

IRDYNA FERYDO



EXPLORER

The Anoikis Field Manual

The Anoikis Field Manual

The Anoikis Field Manual

Volume II: Explorer

Irdyna Ferydo

Unsettling Whisper

A Corporation of Quietly Eerie

Copyright und rechtliche Hinweise

Originaltext © 2026 Irdyna Ferydo. Alle Rechte vorbehalten.

Veröffentlicht von **Unsettling Whisper**, *A Corporation of Quietly Eerie*.

Dies ist eine inoffizielle, von Fans erstellte Publikation für die nicht kommerzielle Verbreitung. Sie steht in keiner Verbindung zu Fenris Creations, wird nicht von Fenris Creations gesponsert und nicht von Fenris Creations unterstützt.

Diese Publikation muss kostenlos verfügbar bleiben. Sie darf ohne vorherige schriftliche Genehmigung der jeweiligen Rechteinhaber weder verkauft noch hinter einer Paywall oder einem Premium-Abonnement angeboten oder zur direkten Umsatzerzeugung durch den Zugang zur Publikation verwendet werden.

EVE Online und alle zugehörigen Namen, Marken, Logos, fiktionalen Schauplätze, Spielelemente und sonstigen geistigen Eigentumsrechte verbleiben bei Fenris Creations oder den jeweiligen Rechteinhabern. Ihre Verwendung in dieser Publikation dient ausschliesslich der Identifikation, Kommentierung und Anleitung im Zusammenhang mit EVE Online.

Die veröffentlichten EVE Online Content Creation Terms schreiben die folgende ältere Formulierung vor. In diesem Zitat ist «CCP Games» der frühere Name von Fenris Creations:

This material is used with limited permission of CCP Games. No official affiliation or endorsement by CCP Games is stated or implied.

EVE Online wird fortlaufend weiterentwickelt. Spielmechaniken, Terminologie, Oberflächenelemente und Zahlenwerte können sich nach der Veröffentlichung ändern. Zeitabhängige Informationen in diesem Buch müssen vor jeder Veröffentlichung anhand aktueller offizieller Quellen überprüft werden.

Kein Teil dieser Publikation ist als offizielle EVE Online-Dokumentation oder als offizielle Aussage von Fenris Creations zu verstehen.

Vollständige und unveränderte Exemplare dieser Publikation dürfen kostenlos zu nicht kommerziellen Zwecken weitergegeben werden, sofern Autor, Herausgeber, rechtliche Hinweise und Credits vollständig erhalten bleiben. Verkauf, Verbreitung hinter einer Paywall, veränderte Weiterverbreitung und Unterlizenzierung des Originaltexts bedürfen der vorherigen schriftlichen Genehmigung von Irdyna Ferydo und, soweit anwendbar, der jeweiligen Rechteinhaber von Drittinhalten.

Texte, Bilder, Schriften und andere Assets Dritter dürfen nur verwendet werden, wenn Herkunft, Urheber und Lizenz in den Credits der Publikation dokumentiert sind. Die jeweiligen Lizenzbedingungen gelten weiterhin.

Manuskriptinformationen

Feld	Wert
Reihe	The Anoikis Field Manual
Band	Volume II: Explorer
Autor	Irdyna Ferydo
Herausgeber	Unsettling Whisper
Originalsprache	Englisch
Manuskriptstatus	Inhaltlich freigegeben
Deutsche Übersetzung	Vollständig und inhaltlich freigegeben
Vertrieb	Nicht kommerziell
Produktionsstatus	Nicht freigegeben

Das englische Manuskript ist das verbindliche Original. Sein Inhalt ist freigegeben und bildet die feste Grundlage für die deutsche Übersetzung. Die Übersetzung erfolgt abschnittsweise und erhält eigene Prüf- und Freigabeentscheide.

Dieses inhaltlich freigegebene Manuskript erlaubt noch kein PDF, Cover, keine Abbildung, Druckdatei oder öffentliche Veröffentlichung. Diese Stufen benötigen eigene spätere Entscheide.

Vorwort

Die erste Nacht in Anoikis ist voller Türen. Jede neue *Connection* wirft die Frage auf, wohin sie führt. Bei jeder Rückkehr stellt sich die Frage, ob die Route noch besteht. Ein ruhiges *Grid* lässt etwas ungeklärt. Die Aufmerksamkeit des Piloten bleibt beim Schiff, weil Bewegung das unmittelbare Problem ist.

Die zweite Nacht ist anders.

Etwas ist zurückgeblieben: ein Schiffsrumpf, ein Container, eine Aufzeichnung oder ein Versprechen an einen anderen Piloten. Vielleicht ist es nur die Erwartung, dass der morgige Tag dort beginnt, wo der heutige endete. Das System ist nicht dauerhaft geworden. Der Pilot hat lediglich begonnen, sich darauf zu verlassen.

Diese Veränderung ist das Thema von *Explorer*.

Pathfinder wurde für den Augenblick zwischen Sehen und Handeln geschrieben. Dieser Band bewahrt diese Disziplin und fügt Zeit hinzu. Eine *Observation* kann nun für jemanden von Bedeutung sein, der sich erst in einigen Stunden einloggt. Eine Route kann jetzt Arbeit statt nur eines Schiffs tragen. Zugleich kann eine *Structure* Tätigkeit ermöglichen und eigene Abhängigkeiten schaffen. Um all dies bleibt derselbe ungewisse Raum bestehen.

Das Leben in Anoikis wird oft anhand von Besitz beschrieben: die *Structure*, die Schiffe, der Treibstoff, die gemeinsame Routenaufzeichnung. Diese Dinge sind wichtig, doch sie erklären nicht das Leben, das zwischen ihnen entsteht. *Home* besteht ebenso aus Aufmerksamkeit, Pflegeverantwortung, Zugriff, Ersatzmöglichkeiten und der Bereitschaft zu erkennen, wann eine Anordnung nicht mehr funktioniert.

Dies ist deshalb kein Buch darüber, die grösstmögliche Präsenz aufzubauen oder den höchsten Ertrag herauszuholen. Es geht darum, Verpflichtungen zu erkennen, bevor Gewohnheit sie verdeckt. Das Buch

fragt, wovon eine Gruppe abhängt und wer den Zustand dieser Abhängigkeit erkennen kann. Danach verfolgt es, was geschieht, wenn die Abhängigkeit versagt, und wie sie verringert werden kann, bevor eine Krise an ihrer Stelle entscheidet.

Die folgenden Seiten werden *Home* nicht dauerhaft machen. Sie sollen seine Verpflichtungen lesbar machen.

Irdyna Ferydo

Dieses Handbuch nutzen

Lies *Explorer* nach *Pathfinder* – oder sobald du Anoikis bereits betreten, eine Route bewahren, eine *Observation* von einer Schlussfolgerung unterscheiden und zurückkehren kannst, ohne die *Chain* als dauerhafte Geografie zu behandeln.

Die vier Parts folgen dem Leben einer Verpflichtung. *Wurzeln schlagen* fragt, was sich ändert, wenn aus einem Besuch Anwesenheit wird. *Home erhalten* macht gemeinsamen Zustand und Infrastruktur sichtbar. *Hier arbeiten* folgt der Exposition, die wiederholte Tätigkeit erzeugt. *Handlungsfähig bleiben* untersucht *Degradation*, Verringerung und ein bewusstes Ende.

Jedes Kapitel lässt sich später einzeln wieder aufschlagen, doch die erste Lektüre sollte der Reihenfolge folgen. Eine Entscheidung über Zugriffsrechte wird verständlicher, nachdem benannt wurde, was gemeinsam getragen wird. Eine Auseinandersetzung mit Ausfällen wird nützlicher, nachdem die Abhängigkeiten, die ausfallen können, sichtbar geworden sind.

Fünf Fragen begleiten jede dauerhafte Entscheidung:

1. Welchem Zweck dient diese Verpflichtung?
2. Wer trägt sie?
3. Wie wird ihr Zustand sichtbar gemacht?
4. Was folgt, wenn sie versagt?
5. Wie soll sie enden?

Diese Fragen ersetzen nicht den in *Pathfinder* entwickelten Arbeitsrhythmus. Sie erweitern ihn über mehr Menschen und mehr Zeit.

Mechaniken, Oberflächenbezeichnungen und Zahlenwerte in EVE Online können sich ändern. Dieses Handbuch erklärt Zusammenhänge, statt einen dauerhaften Klickpfad zu versprechen. Scheinen der *Live*

Client und das Buch einander zu widersprechen, bewahre den Unterschied und überprüfe ihn anhand aktueller offizieller Informationen.

Inhaltsverzeichnis

Vorwort	9
Dieses Handbuch nutzen	11
Einleitung	14
Part I - Wurzeln schlagen	17
Kapitel 1 - Jenseits des Besuchs	19
Kapitel 2 - <i>Home</i> wählen	31
Kapitel 3 - Verpflichtung wiegt	45
Kapitel 4 - Gemeinsamer Boden	59
Part II - <i>Home</i> erhalten	73
Kapitel 5 - <i>Baseline</i> lesen	75
Kapitel 6 - <i>Chain</i> pflegen	89
Kapitel 7 - <i>Access</i> und <i>Ownership</i>	103
Kapitel 8 - Infrastruktur pflegen	117
Part III - Hier arbeiten	131
Kapitel 9 - <i>Work</i> erzeugt <i>Exposure</i>	133
Kapitel 10 - Ressourcenoperationen	145
Kapitel 11 - Wert bewegen	159
Kapitel 12 - Lagerung und <i>Replacement</i>	173
Part IV - Handlungsfähig bleiben	187
Kapitel 13 - Routine im Zeitverlauf	189
Kapitel 14 - <i>Failure</i> vor Krise	203
Kapitel 15 - Bewusst aufbrechen	217
Kapitel 16 - Bereit für <i>Survivor</i>	233
Glossar und Index	245
Weiterführende Literatur	255
Danksagung	259
Über den Verlag	261

Einleitung

Nach der Rückkehr

Ein Pathfinder plant die Rückkehr vor der ersten Durchquerung. Die Route ist wichtig, weil das Schiff den Ort wieder verlassen soll. Ein *Explorer* stellt eine weitere Frage: Was wird noch von diesem Ort abhängen, nachdem der Pilot die Tastatur verlassen hat?

Vielleicht ist es nicht viel. Ein Schiff bleibt in *Home* eingelagert. Ein *Shared Location Folder* enthält *Bookmarks* für die nächste Schicht. Materialien warten auf einen späteren Einsatz. Jemand hat zugesagt, eine *Connection* erneut zu prüfen. Nichts davon ist dramatisch, doch jede dieser Handlungen überträgt einen Teil der heutigen Entscheidung auf morgen.

Diese Übertragung ist der Beginn von Anwesenheit. *Ownership* an einer *Structure* ist dafür nicht erforderlich. Der Gastzugang kann von einer anderen Gruppe abhängen. Eine kleine *Corporation* kann Infrastruktur besitzen und dennoch bei der Pflege des Wissens in ihrem Umfeld von einer einzigen Person abhängen. *Ownership*, *Access*, Verantwortung und *Capability* können sich überschneiden, bezeichnen aber nicht denselben Zustand.

Anwesenheit wird System

Längere Anwesenheit verbindet Dinge, die sich bei einem kurzen Besuch voneinander trennen lassen. Die *Chain* beeinflusst den Transport. Der Transport beeinflusst *Replacement*. *Replacement* bestimmt, wie viel *Work* nach einem Verlust fortgesetzt werden kann. *Access* entscheidet, wer die Werkzeuge und Vorräte erreichen kann, die diese *Work* tragen sollen. Ein veralteter Bericht kann deshalb Folgen haben, die weit über den Piloten hinausreichen, der ihm zuerst vertraute.

Das Ergebnis ist keine Maschine, die sich durch das Erledigen jeder Aufgabe absichern liesse. Stattdessen entsteht eine lebendige Anordnung,

deren Teile unterschiedlich schnell altern. *Connections* verändern sich. *Observations* veralten. Menschen kommen und gehen. *Services* verbrauchen Ressourcen. Der Zweck verschiebt sich. Gute Arbeit beginnt damit, diese Veränderungen sichtbar zu machen, statt sie hinter Routine zu verbergen.

Der Arbeitsrhythmus

Explorer behält den Rhythmus des ersten Bands bei:

Observe → Record → Understand → Evaluate → Decide → Act → Observe again

Der Unterschied liegt in der Reichweite. Eine *Observation* kann nun länger bestehen als der unmittelbare Flug. Aufzeichnungen können jemanden leiten, der die jeweilige *Source* nicht selbst erlebt hat. Entscheidungen können Vermögenswerte binden, die mehreren Personen gehören. Bei der abschliessenden *Observation* lautet die Frage nicht nur, ob eine Handlung funktioniert hat, sondern auch, ob der gemeinsame Zustand dies bestätigt.

Darum begegnet uns der *Steward* im ganzen Buch. Seine Aufgabe besteht nicht darin, jede Arbeit persönlich zu erledigen. Er sorgt dafür, dass für eine Abhängigkeit jemand verantwortlich ist, ihr Zustand abgelesen werden kann und ihr *Handover* bestätigt statt vorausgesetzt wird.

Die Grenze

Dieser Band behandelt längerfristige Anwesenheit, gemeinsame Informationen, *Access* und Infrastruktur. Er verbindet sie mit gewöhnlicher *Work*, Lagerung, *Replacement*, *Degradation* und einem geplanten *Exit*. Der Band schreibt kein universelles Modell für eine *Corporation* vor, erstellt keine Ertragsrangliste der Systeme und bietet kein *Fitting* für jede Tätigkeit.

Combat Doctrine, *Structure Defence* und *Evictions* gehören zu *Survivor*. Strategische Flottenbewegungen und Logistik im Massstab einer *Corpo*-

ration gehören zu *Navigator*. *Explorer* kann zeigen, wo diese Belastungen auf eine Entscheidung treffen, macht sie aber nicht selbst zum Lernstoff.

Diese Grenze ist wichtig. Ein nützliches Feldhandbuch sollte nicht so tun, als gehörte jede Folge zur selben Fähigkeit. Dieses Buch bereitet einen Bewohner darauf vor, zu erkennen, was exponiert ist, bevor aktiver Konflikt zum Thema wird.

Das Gewicht

Das erste Kapitel beginnt in der zweiten Nacht. Das System ist vertraut genug, um zu Annahmen zu verleiten, aber nicht beständig genug, um sie zu rechtfertigen. Etwas wurde zurückgelassen. Eine Route ist gealtert. Die Notiz einer anderen Person wartet darauf, dass man ihr vertraut.

Bevor der *Explorer Home* wählt, Routine entwickelt oder *Work* organisiert, muss er benennen, was bereits zu einer Verpflichtung geworden ist.

Part I - Wurzeln schlagen

Ein Besuch kann enden, wenn das Schiff zurückkehrt. Anwesenheit wirkt in dem fort, was bleibt: Vermögenswerte, Aufzeichnungen, *Access*, *Work* und Erwartungen. Part I folgt dieser Veränderung von der ersten wiederholten Nacht bis zu dem Punkt, an dem mehrere Piloten benennen können, was sie teilen und was keiner von ihnen voraussetzen sollte.

Kapitel 1 - Jenseits des Besuchs

Am folgenden Abend kehrt der Pilot in dasselbe System zurück.

Der Name im *Client* ist vertraut. Auch die *Structure* am Endpunkt des letzten Flugs ist bekannt. Ein Schiff für das *Scanning* wartet dort, wo es zurückgelassen wurde. Die Notizen der vergangenen Nacht beschreiben noch immer eine Route nach *Known Space*. Neben einer der *Connections* hat ein anderer Pilot eine kurze Nachricht ergänzt: ruhig bei der Prüfung.

Nichts an diesem Bild muss falsch sein.

Nichts daran muss noch aktuell sein.

Gestern führte die Route ein Schiff in Richtung *Home*. Heute Abend scheinen dieselben Markierungen einen Ausgangspunkt zu bieten. Dieser Anschein ist die erste Prüfung längerer Anwesenheit. Ein Besucher fragt, ob sich eine Route jetzt nutzen lässt. Ein Bewohner ist versucht zu fragen, weshalb sie nicht noch immer nutzbar sein sollte.

Zwischen diesen Fragen liegt das Gewicht von allem, was zurückgeblieben ist.

Die zweite Nacht

In der ersten Nacht lag die Ungewissheit nahe an der Schiffshülle. Der Pilot durchquerte die Verbindung, beobachtete das neue *Grid*, erfasste das *Wormhole* und baute genug von der *Chain* auf, um eine Rückkehr zu tragen. Jede Entscheidung gehörte zum laufenden Flug. Endete der Einsatz, endeten mit ihm auch die meisten seiner Verpflichtungen.

Die zweite Nacht beginnt mit einem geerbten Zustand.

Das Schiff für das *Scanning* ist ein Vermögenswert, der bereits an diese Seite der Route gebunden ist. Die *Bookmarks* sind Aufzeichnungen für eine Lage, die sich seither weiter verändert hat. Die Notiz über eine ruhige *Connection* ist *Reported Information* von einer anderen Person

und aus einer anderen Zeit. Selbst die Absicht, dasselbe *Home* zu nutzen, ist eine Annahme, die aus dem Vortag übernommen wurde.

Das macht die Aufzeichnungen nicht nutzlos. Es macht sie zum Ausgangspunkt einer neuen *Observation*, nicht zu deren Ersatz.

Der Pilot kann das geerbte Bild anhand dreier Fragen ordnen.

Was besteht noch? Das Schiff ist vorhanden. Die eingelagerten Materialien sind vorhanden. Die *Structure* ist gegenwärtig erreichbar. Diese Bedingungen lassen sich erneut beobachten.

Was ist lediglich aufgezeichnet? Die frühere *Chain*, der vorherige Zustand der *Connection* und der Grund für die Notiz eines anderen Piloten gehören zu ihren *Sources* und *Timestamps*. Sie beschreiben, was bekannt war, nicht was das System der nächsten Schicht schuldet.

Was ist jetzt *Unknown*? Ob die Route noch vollständig ist, was sich jenseits des gegenwärtigen *Grid* bewegt hat und ob sich die Voraussetzungen des gestrigen Plans verändert haben, bleibt offen, bis neue Belege diese Fragen schliessen.

Dieser Unterschied wirkt kleinlich, solange nichts schiefgegangen ist. Gerade weil keine dramatische Veränderung angekündigt wurde, ist er wertvoll. Längere Anwesenheit beginnt selten mit dem formellen Entscheid, eine Abhängigkeit zu akzeptieren. Sie beginnt, wenn die gestrige Antwort die heutige Frage ersetzen darf.

Die zweite Nacht beginnt deshalb nicht damit, jede Handlung der ersten zu wiederholen. Zuerst gilt es zu erkennen, welche Teile des geerbten Zustands für die nächste Entscheidung wichtig sind. Ein ungenutztes *Bookmark* kann ohne unmittelbare Folge altern. Eine *Connection*, welche die einzige verfügbare Route tragen soll, verlangt jetzt Aufmerksamkeit. Ein eingelagertes Schiff, das ohne Auswirkung auf den Plan unerreichbar bleiben kann, stellt eine andere Verpflichtung dar als ein Schiff, das für den Beginn des Plans benötigt wird.

Anwesenheit ergänzt die Beobachtung um Auswahl. Der Bewohner muss nicht nur wissen, was sich verändert hat, sondern auch, welche unveränderte Annahme am schwersten wiegen würde, falls sie falsch wäre.

Veränderte Konsequenzen

Anoikis wird durch Dauer nicht zu einem anderen Ort. Es verändert sich, was der Pilot von diesem Ort tragen lassen will.

Ein Besucher kann für eine kurze, umkehrbare Bewegung ein unvollständiges Bild akzeptieren. Ein Bewohner kann dieselbe *Chain* verwenden, um Materialien zu bewegen, *Work* zu planen, einen anderen Piloten zu unterstützen oder zu entscheiden, was sicher in *Home* bleiben kann. Die *Connection* ist dasselbe Objekt. Die daran gebundene Konsequenz ist gewachsen.

Deshalb kann Vertrautheit täuschen, ohne eine einzige falsche Tatsache zu enthalten. Der Systemname stimmt. Die *Structure* befindet sich an ihrem alten Ort. Die gemeinsame Aufzeichnung kann genau wiedergeben, was die vorherige Schicht sah. Der Fehler entsteht, wenn Vertrautheit als Beleg dafür gilt, dass die Beziehung zwischen diesen Dingen unverändert geblieben ist.

Wormhole Connections sind kurzlebig. Ihre Informationen geben nur grobe Hinweise auf verbleibende Lebensdauer und bereits getragene Masse. Zugleich unterliegen einzelne Transits und der gesamte Durchgang Massengrenzen. Eine bekannte *Chain* ist deshalb eine datierte Darstellung vorübergehender Beziehungen. Sie kann eine Entscheidung nur tragen, wenn ihr Alter und ihre gegenwärtige Nutzung gemeinsam mit den Kosten eines Versagens beurteilt werden.

Längere Anwesenheit schafft mehr Wege, auf denen sich dieses Versagen ausbreiten kann. Verliert ein Scout eine Route, verändert sich ein Flug. Erwarten eingelagertes Material, geplante *Work* und mehrere Piloten dieselbe Route, erreicht derselbe Verlust mehrere Entscheidungen.

Das Problem besteht nicht bloss darin, dass das *Wormhole* zusammenbrechen könnte. Es besteht darin, dass die Gruppe mehrere Bedürfnisse an dieselbe unsichere Bedingung gebunden hat.

Ein *Explorer* lernt, diesen gemeinsamen Zustand zu erkennen, bevor er versucht, ihn zu beseitigen. Ein Teil dieser Abhängigkeit ist Sinn und Zweck des Lebens in *Home*. Schiffe müssen sich irgendwo befinden. Informationen müssen auf irgendeine Weise geteilt werden. *Work* benötigt Werkzeuge, Zeit und *Access*. Das Ziel ist nicht die Unabhängigkeit von jedem beweglichen Teil. Es ist eine ehrliche Darstellung davon, welche beweglichen Teile nun Gewicht tragen.

Das einfachste Anzeichen ist ein Satz, der mit *wir erwarten* beginnt. Die Route soll verfügbar bleiben. Das eingelagerte Schiff soll bereit sein, ein anderer Pilot soll die Karte aktualisieren und *Access* soll bestehen bleiben. Jede Erwartung kann vernünftig sein. Zur Verpflichtung wird sie, sobald die Entscheidungen von morgen von ihr abhängen.

Vermögenswerte wiegen

Vermögenswerte machen Verpflichtungen leicht sichtbar, weil sie auch dann einen Ort einnehmen, wenn ihr Eigentümer abwesend ist. Ihr Preis allein zeigt jedoch nicht ihr Gewicht.

Ein bescheidenes Schiff für das *Scanning* kann unverzichtbar sein, wenn es als einziges eine Route wiederaufbauen kann. Ein kostspieliger Gegenstand kann leicht wiegen, wenn sein Verlust keine unmittelbare Entscheidung verändert und *Replacement* bereits anderswo erreichbar ist. Entscheidend ist nicht nur der Wert eines Vermögenswerts, sondern auch, welche *Capability* mit ihm verschwindet, wie viele andere Pläne von ihm abhängen und wie weit ein Ersatz entfernt ist.

Dieser Unterschied verhindert zwei schwache Gewohnheiten.

Die erste behandelt alles in *Home* als gleichermassen exponiert. So entsteht eine grosse Zahl, ohne sichtbar zu machen, welcher Ausfall die Gruppe tatsächlich unterbrechen würde. Die zweite behandelt vertrau-

ten Lagerraum wie bereits wiedererlangten Wert. Ein Vermögenswert ist nicht sicher, nur weil er sich nicht mehr bewegt. Seine *Exposure* hat sich vom Transit auf Standort, *Access* und Infrastruktur verlagert.

Diese Veränderung ist im *Wormhole Space* besonders wichtig. *Upwell Structures* verwenden dort nicht das gewöhnliche Ergebnis von *Asset Safety*. Würde *Asset Safety* normalerweise ausgelöst, werden betroffene Vermögenswerte stattdessen in den Weltraum ausgestossen und können von anderen geborgen werden. Lagerung kann praktisch, notwendig und sorgfältig verwaltet sein. Sie darf jedoch nicht als Versprechen einer späteren Wiedererlangung verstanden werden.

Die Lehre lautet nicht, *Home* nach jeder Nutzung zu leeren. Das würde aus dem Leben dort wieder eine Folge von Besuchen machen und gewöhnliche *Work* verunmöglichen. Vermögenswerte sollen vielmehr bewusst zurückgelassen werden.

Für jeden wichtigen Vermögenswert sollte der Bewohner benennen können, was er ermöglicht, wer von ihm abhängt und was geschieht, wenn er nicht erreichbar ist. Die Antwort muss keine Inventartabelle sein. Ein Schiff kann die einzige lokale *Scanning Capability* der Gruppe tragen. Ein gemeinsames Lager kann mehreren Eigentümern gehören. Wird ein einziges *Replacement* nach *Home* gebracht, kann die Route hinaus unbemerkt wichtiger geworden sein.

Vermögenswerte erhalten Gewicht, wenn sie eine zukünftige Entscheidung weniger umkehrbar machen. Ein Pilot, der gehen kann, ohne einen anderen Plan zu verändern, bleibt einem Besucher nahe. Ein Pilot, der zuerst Schiffe zurückholen, *Access* klären, Aufzeichnungen bewahren oder eine Abwesenheit erklären muss, hat bereits Wurzeln geschlagen.

Rückkehr verlangt Arbeit

Home hebt die Notwendigkeit nicht auf, *Home* zu finden.

Dieser Satz klingt nur widersprüchlich, solange *Home* als fester Punkt mit einer festen Strasse verstanden wird. Das System besteht fort, doch die *Wormhole Connections*, über die ein Pilot es erreicht, sind vorübergehend. Eine gestern aufgezeichnete Route ist keine Strasse, die nur wieder geöffnet werden muss. Sie ist ein Beleg dafür, dass eine Beziehung bestand, und kann helfen zu bestimmen, was als Nächstes geprüft werden muss.

Für einen Besucher ist das *Scanning* der Rückkehr Teil eines einzelnen Einsatzes. Für einen Bewohner wird die Pflege der *Chain* zu wiederkehrender *Work*. Jemand beobachtet. Jemand zeichnet auf. Ein benannter *Steward* entscheidet, welcher alte Zweig noch eine Bewegung leiten kann und welcher als *Unknown* markiert oder geschlossen werden muss. Spätere Kapitel untersuchen, wie diese *Work* geteilt wird. Die erste Lektion lautet schlicht, dass es sie gibt.

Sie zu ignorieren bewahrt die *Chain* nicht. Es überträgt die Kosten der Entdeckung auf die nächste Person, die sich bewegen muss.

Die Nachricht neben der *Connection* – ruhig bei der Prüfung – muss deshalb an ihre Grenzen gebunden bleiben. Sie kann ein sorgfältiger und genauer Bericht sein. Dem gegenwärtigen Piloten kann sie nicht sagen, was nach der Prüfung geschah, ob die Route die geplante Bewegung noch trägt oder ob ruhig Abwesenheit, Unaufmerksamkeit oder bloss keine beobachtete Veränderung bedeutete.

Die verantwortungsvolle Reaktion ist kein Misstrauen gegenüber dem anderen Piloten. Sie respektiert die Form des Berichts. Bewahre, wer was wann beobachtet hat. Führe dann die neue *Observation* durch, welche die neue Entscheidung verlangt.

Die Arbeit an der Rückkehr macht auch den Zweck sichtbar. Wer eine Route für einen entbehrlichen Scout prüft, benötigt möglicherweise weniger Gewissheit als jemand, der eine *Capability* bewegen will, die sich in *Home* nicht ersetzen lässt. Die *Chain* kennt keinen einzelnen Zustand namens bereit. Sie ist *für etwas* bereit – unter benannten

Ungewissheiten und so lange, wie die stützenden *Observations* nützlich bleiben.

Die zweite Nacht wird handhabbar, sobald der Bewohner nicht mehr fragt, ob die alte Route noch wahr ist. Er fragt, was für den Zweck dieses Abends wahr sein muss. Diese Frage macht aus Wiederholung Feldarbeit.

Verpflichtung benennen

Eine Verpflichtung wird gefährlich, wenn sie zwar spürbar ist, aber nicht beschrieben werden kann. Menschen merken, dass der Aufbruch schwierig wäre, dass immer eine Aufgabe wartet oder dass ein abwesendes Mitglied das ganze Bild zu tragen scheint. Ohne eine klarere Sprache reagiert die Gruppe vielleicht mit mehr Arbeit rund um die Abhängigkeit, statt zu entscheiden, ob diese noch bestehen sollte.

Fünf Fragen machen die Verpflichtung sichtbar.

Welchem Zweck dient sie?

Der Zweck trennt eine nützliche Abhängigkeit von angesammelter Gewohnheit. Ein Schiff für das *Scanning* kann den Wiederaufbau der *Chain* unterstützen. Gemeinsame Aufzeichnungen können ein *Handover* tragen. Eingelagerte Materialien können einer bestimmten Phase von *Work* dienen. *Weil wir hier leben* ist noch kein Zweck. Der Satz erklärt Vertrautheit, nicht Bedarf.

Ein Zweck sollte eng genug gefasst sein, um enden zu können. Unterstützt ein Vermögenswert, ein *Service* oder eine Routine alles, kann niemand erkennen, wann er nicht mehr benötigt wird. Wird die konkrete *Capability* benannt, die er schützt, zeigt sich ausserdem, wann sich mehrere Verpflichtungen unbemerkt um dasselbe Objekt gesammelt haben.

Wer trägt sie?

Der Träger ist nicht immer der Eigentümer. Ein Pilot kann das Schiff besitzen, während ein anderer die Aufzeichnung pflegt, die es nutzbar macht. Eine *Corporation* kann eine *Structure* besitzen, während wenige Menschen die Aufmerksamkeit aufbringen, die nötig ist, um ihren Zustand zu erkennen. Ein Gast kann von *Access* abhängen, ohne ihn zu kontrollieren.

Die Frage lautet nicht, wem bei einem Ausfall die Schuld gegeben werden kann. Sie lautet, wer zugestimmt hat, die Verpflichtung zu beobachten, zu aktualisieren oder zu schliessen. Ist die Antwort *alle*, lautet die praktische Antwort oft *noch niemand*.

Wie wird ihr Zustand sichtbar?

Eine Verpflichtung, die nur in der Erinnerung besteht, lässt sich nicht übergeben. Der sichtbare Zustand kann eine aktuelle Aufzeichnung, eine klare Unterscheidung im Inventar, eine benannte verantwortliche Person oder ein bestätigter Bericht sein. Er sollte ausreichen, damit eine andere Person erkennt, ob die Abhängigkeit intakt, ungewiss oder bereits beeinträchtigt ist.

Sichtbarkeit verlangt nicht, jedes Detail für jedes Mitglied zu veröffentlichen. Sie verlangt, dass die Personen, von denen eine Handlung erwartet wird, den Zustand sehen können, von dem diese Handlung abhängt. *Access* zu Informationen ist Teil der Verpflichtung, kein nachträglicher Gedanke.

Was folgt, wenn sie versagt?

Failure umfasst mehr als Zerstörung. Eine Route kann unbrauchbar werden. *Access* kann sich verändern. Eine Aufzeichnung kann zu alt werden. Die einzige Person, die eine Aufgabe ausführen kann, kann abwesend sein. Der Vermögenswert kann noch bestehen, während die operative *Capability* verschwunden ist.

Beschreibe die erste verlorene *Capability*, bevor du dir das schlimmste mögliche Ereignis vorstellst. Der Verlust dieser *Connection* stoppt den

Transport dieses Abends. Ohne dieses Schiff kann das lokale *Scanning* nicht wie geplant beginnen. Ein veralteter Bericht zwingt den nächsten Piloten, den betroffenen Teil des Bildes neu aufzubauen. Diese Antworten führen zu verhältnismässiger Vorbereitung statt zu unbestimmter Angst.

Wie soll sie enden?

Jede Verpflichtung benötigt ein vorgesehenes Ende, auch wenn kein Datum bekannt ist. Ein Vermögenswert kann entfernt werden, wenn sein Zweck endet. Eine vorübergehende Regelung für *Access* kann nach der unterstützten *Work* überprüft werden. Eine wiederkehrende Prüfung kann enden, sobald die Gruppe nicht mehr von ihrem Ergebnis abhängt.

Eine Endbedingung bewahrt Wahlmöglichkeiten. Ohne sie kommt jede neue Abhängigkeit zu den alten hinzu und die Anwesenheit wächst durch Ansammlung. Die Gruppe kann noch immer gehen, entdeckt die wahre Grösse der Verpflichtung aber erst bei dem Versuch, sie abzubauen.

Die Fragen wirken zusammen. Ein Zweck ohne Träger bleibt eine Absicht. Fehlt einem Träger der sichtbare Zustand, entsteht eine private Abhängigkeit. Sichtbarer Zustand ohne Folge eines *Failure* erzeugt Aufzeichnungen, von denen niemand weiss, wie sie zu verwenden sind. Eine Reaktion ohne Endbedingung kann das unmittelbare Problem lösen und dabei die Verpflichtung vergrössern.

Betrachte das in *Home* wartende Schiff für das *Scanning*. Sein Zweck ist nicht einfach *Scanning*. Heute Abend soll es genug von der *Chain* wiederherstellen, um eine bestimmte Verlegung zu tragen. Ein Pilot übernimmt diese erste Prüfung. Die gemeinsame Aufzeichnung zeigt anschliessend, welche *Connections* frisch *Observed* wurden, welche *Reported* bleiben und welche *Unknown* geworden sind. Kann die Schiffshülle nicht eingesetzt werden, wartet die Verlegung, bis eine andere *Scanning Capability* verfügbar ist. Die Verpflichtung endet, wenn die

Route der Verlegung gedient hat und das Schiff in seinen vorgesehenen Zustand zurückgebracht wurde – oder wenn die Verlegung abgesagt und nicht länger erwartet wird.

Keine dieser Aussagen macht den Flug sicher. Sie machen seine Abhängigkeit klein genug, um sie zu untersuchen.

Dieselbe Methode zeigt auch, wann das sichtbare Objekt nicht die eigentliche Verpflichtung ist. Eine eingelagerte Ersatzhülle kann wie der wichtige Vermögenswert aussehen, während die tatsächliche Abhängigkeit im *Access* zu ihrem Lagerort besteht. Eine sorgfältig gepflegte Karte wirkt vielleicht wie gemeinsames Wissen, obwohl die eigentliche Abhängigkeit ein einzelner Pilot ist, der weiss, welche Aufzeichnungen noch aktuell sind. Hinter einer Route, die wie der Plan aussieht, kann eine Frist stehen, die keine Zeit für ihren Wiederaufbau lässt.

Wird die Verpflichtung auf der falschen Ebene benannt, entsteht unnötige Arbeit. Die Gruppe zählt Objekte, obwohl sie *Capability* untersuchen sollte, oder aktualisiert Aufzeichnungen, obwohl sie den Zweck hinterfragen sollte, der diese immer wieder erzeugt. Auf der richtigen Ebene geschieht das Gegenteil. Sichtbar werden die kleinste Bedingung, die wahr bleiben muss, und die erste verfügbare Handlung, sobald sie nicht mehr gilt.

Verringerung beginnt hier. Eine Frist kann verschoben, eine Verlegung aufgeteilt oder eine weitere *Capability* in Reichweite bereitgestellt werden. Auch ein zweiter Beobachter ist möglich, ebenso der Entscheid, dass der Zweck das Gewicht nicht mehr rechtfertigt. Verringerung wartet nicht bis zum Rückzug. Sie gehört zum gewöhnlichen Bemühen, die Verpflichtung im Verhältnis zu dem zu halten, was die Gruppe noch beobachten und tragen kann.

Die fünf Antworten müssen nicht vollkommen sein, bevor die zweite Nacht weitergeht. Sie müssen ehrlich genug sein, um das Fehlende sichtbar zu machen.

Der Pilot in *Home* kann die geerbte Szene nun anders lesen. Das Schiff für das *Scanning* besitzt einen Zweck und eine Konsequenz, falls es nicht verfügbar ist. Die alte *Chain* hat eine *Source* und ein Alter statt automatischer Autorität. Innerhalb ihres *Timestamp* bleibt die Notiz über die ruhige Lage nützlich. Die heute Abend benötigte Route kann entsprechend dem Gewicht beobachtet werden, das sie tragen soll.

Nichts ist gewiss geworden. Die Abhängigkeit ist sichtbar geworden.

Das ist die erste Handlung eines *Explorer*.



Abbildung AFH-B2-FIG-001 — Eine Verpflichtung wird durch fünf Angaben sichtbar.

Eine Verpflichtung wird sichtbar, wenn Zweck, Träger, Zustand, Folge eines *Failure* und vorgesehenes Ende benannt sind.

Feldprinzip

Eine Verpflichtung beginnt, bevor eine *Structure* verankert wird oder eine Gruppe ein System als dauerhaft bezeichnet. Sie beginnt, sobald eine zukünftige Entscheidung von etwas abhängt, das in *Home* zurück-

gelassen, mit einer anderen Person geteilt oder weiterhin verfügbar erwartet wird.

Benenne Zweck, Träger, sichtbaren Zustand, Folge eines *Failure* und vorgesehenes Ende. Beobachte danach den Teil der Umgebung, den die Entscheidung dieses Abends tatsächlich verlangt.

Ist das Gewicht des Bleibens sichtbar, lautet die nächste Frage nicht mehr, welches System am besten erscheint. Sie lautet, welches *Home* den Zweck tragen kann, ohne die Zielkonflikte zu verbergen. Dort beginnt Kapitel 2.

Kapitel 2 - *Home* wählen

Zwei mögliche Systeme haben die erste Diskussion überstanden.

Auf der gemeinsamen Seite wirken sie nahezu gleich. Beide sind erreichbar. Beide bieten Raum, in dem die Gruppe arbeiten könnte. In jedem System ist eine *Upwell Structure* sichtbar, doch nur eine gewährt derzeit Gastzugang. Kurze Notizen nennen eine *Class*, einen *Effect* und mehrere beim Scouting beobachtete *Connections*.

Die Frage am oberen Rand der Seite lautet, welches System besser sei. Es ist die falsche Frage.

Ein Kandidat unterstützt die geplante *Work*, rückt *Replacement* jedoch weiter weg. Der andere ist heute bequem, weil eine andere *Corporation Access* gewährt. Diese Bequemlichkeit beruht auf einer Entscheidung, welche die Gruppe nicht kontrolliert. Keines der Systeme ist einfach gut oder schlecht. Jedes verlangt, eine andere Art von Abhängigkeit zu akzeptieren.

Die Wahl von *Home* beginnt mit der Frage, was das System tragen soll.

Zweck vor *Class*

Eine *Class* lässt sich leicht in eine Überschrift setzen. Der Zweck ist schwieriger, denn er zwingt die Gruppe, sich selbst zu beschreiben.

Was werden die Menschen hier tatsächlich tun? Wie oft werden sie es tun? Welche *Capability* muss bei ihrer Ankunft verfügbar sein? Was muss in das System oder aus dem System gebracht werden? Wie lange soll die Anordnung bestehen? Diese Fragen sind weniger elegant als eine Rangliste, doch sie zeigen, was eine Rangliste verbergen würde.

Angenommen, die Gruppe erklärt, *Home* solle Exploration, begrenzte Ressourcenarbeit und grundlegendes *Replacement* für wenige Piloten tragen. Dieser Satz ist ein Anfang, kein vollständiger Zweck. Er lässt offen, ob die Gruppe bei verändertem *Access* pausieren kann, ob Mate-

rialien täglich hinausgebracht werden müssen und ob die Abwesenheit einer Person die *Work* stoppen würde.

Ein brauchbarer Zweck besteht aus fünf Teilen.

Erstens benennt er die *Work*. *Im Wormhole Space leben* ist eine Identität. *Eine kleine Basis für Exploration und eine bestimmte Phase von Ressourcenarbeit unterhalten* beschreibt eine Nutzung.

Zweitens benennt er die Menschen. Ein für einen unabhängigen Piloten gewähltes System erzeugt andere Koordinationskosten als eines, das mehrere Schichten tragen soll. Der Unterschied liegt nicht nur in der Grösse. Mehr Menschen bedeuten mehr *Handovers*, mehr Eigentümer und mehr Gelegenheiten, bei denen eine Annahme unbemerkt weitergegeben wird.

Drittens benennt er die erforderlichen *Capabilities*. Dazu gehören *Scanning*, Lagerung und *Replacement* sowie alle *Services*, welche die *Work* tatsächlich benötigt. Sie sollten sichtbar sein, bevor eine *Structure* wegen ihres Angebots bewundert wird. Eine bloss angenehme *Capability* darf während des Vergleichs nicht unbemerkt zur Voraussetzung werden.

Viertens benennt er die Zeit. Ein Versuch kann Abhängigkeiten akzeptieren, die für eine unbegrenzte Anwesenheit ein schwaches Fundament wären. Eine auf Dauer angelegte Anordnung muss Abwesenheit, verzögertes *Replacement* und veränderlichen *Access* überstehen, ohne so zu tun, als würden diese Bedingungen nie eintreten.

Schliesslich benennt er das Ende. Die Gruppe sollte wissen, welches Ergebnis den Versuch erfolgreich machen würde, welcher Befund das System ungeeignet machen würde und wie viel zurückbleiben darf, solange diese Entscheidung offen ist.

Ein brauchbarer Zweck könnte so lauten: *Während einer Versuchsphase trägt Home die Exploration mehrerer Piloten. Es beherbergt ersetzbare Arbeitsvermögenswerte und bietet nur die Services, die nötig sind, um diese wieder einsetzen zu können. Die Work pausiert, wenn die betreffende Chain*

kein Replacement tragen kann. Der Versuch endet, bevor ein einzelner Vermögenswert oder eine Regelung für Access notwendig wird, um alles andere zurückzuholen.

Die Formulierung sagt bewusst nicht voraus, wie oft nützliche *Connections* erscheinen oder wie viel *Work* die Piloten abschliessen werden. Sie benennt, welche *Capability* das System tragen soll und was die Gruppe noch nicht von ihm verlangen wird. Neue Gelegenheiten können beobachtet werden, ohne während desselben Versuchs zu neuen Verpflichtungen aufzusteigen.

Der Zweck sollte auch für jemanden verständlich sein, der an der Wahl des Systems nicht beteiligt war. Beruht die Erklärung auf Begeisterung, Ruf oder einer ungeschriebenen Geschichte, kann die nächste Schicht sie nicht prüfen. Ein kurzer, ausdrücklicher Zweck gibt späteren Belegen etwas, das sie bestätigen oder infrage stellen können.

Der Zweck wählt das System nicht allein. Er verwandelt Systemeigenschaften in beantwortbare Fragen. Eine *Class* wird in Bezug auf eine bestimmte Tätigkeit bedeutsam. Ein *Effect* wird zur Bedingung, die auf tatsächliche Schiffe und *Work* wirkt. *Connections* werden zu Routen, die eine bestimmte Bewegung tragen sollen. Infrastruktur wird zu einer Gruppe von *Capabilities* mit Verantwortlichen und Grenzen.

Ohne Zweck lädt jedes nützliche Merkmal zu einer weiteren Tätigkeit ein. Die Gruppe wählt ein System, weil dort vieles möglich wäre, und entdeckt später, dass sie nicht erklären kann, welche dieser Möglichkeiten die Verpflichtung rechtfertigte. Mit einem Zweck bleibt das Mögliche vorerst nur möglich. Nur die erforderlichen Bedingungen fließen in die Entscheidung ein.

System lesen

Gewöhnliche *Wormhole Systems* werden üblicherweise in die *Classes C1* bis *C6* eingeteilt. Die offiziellen *Static Data* bilden ausserdem die *Wormhole Class* und die *System Effects* ab. Diese stabilen Systemeigenschaften

lassen sich erfassen und prüfen. Sie sind keine vollständige Beschreibung von *Home*.

Eine *Class* ist zuerst eine Kategorie und erst danach eine Interpretation. Sie hilft festzustellen, welches System betrachtet wird und welche weiteren Fragen zum Plan gehören. Für sich allein zeigt sie weder die morgen verfügbare Route noch die Menschen, die das System teilen werden, oder den *Access*, den ein Gast behalten wird.

Dieser Unterschied ist wichtig, weil Spieler mit der Bezeichnung der *Class* oft fremde Schlussfolgerungen übernehmen. Ein Kandidat wird möglicherweise als einfach oder schwierig, reich oder arm, geeignet oder unerwünscht bezeichnet, bevor der Zweck der Gruppe im Satz vor kommt. Manche dieser Urteile mögen für die Person nützlich gewesen sein, die sie gebildet hat. Es bleiben Interpretationen für einen anderen Plan.

Erfasse zuerst die Eigenschaft.

Das System besitzt eine angegebene *Class*. Erfasse die *Source*, anhand derer sie bestimmt wurde. Das System besitzt einen *Effect* oder es wurde kein relevanter *Effect* festgestellt. Erfasse ihn getrennt. *Structures* und andere sichtbare Infrastruktur gehören zu einer weiteren Ebene. Gegenwärtige *Connections* gehören zur *Chain*, nicht in das Feld der *Class*.

Diese Trennung verhindert, dass eine korrekte Beobachtung eine unbelegte Schlussfolgerung trägt. *C3* kann eine geprüfte Eigenschaft sein. Die Aussage *C3 ist für uns richtig* bezeichnet dagegen eine Entscheidung, die den Zweck, die *Capabilities* und die akzeptierten Kosten der Gruppe benötigt. Die erste Aussage kann wahr bleiben, während sich die zweite verändert.

Static Data sind hier wertvoll, weil sie den Unterschied zwischen den Definitionen des Spiels und den betrieblichen Erfahrungen der Bewohner eines Systems bewahren. Sie enthalten Informationen zu *Wormhole Classes* und *Effects*. Drittentwicklern stellen sie jedoch keine systemspezifischen statischen *Wormholes* bereit, die Spieler möglicherweise aus

Erfahrung oder anderen Aufzeichnungen erwarten. Eine Feldnotiz über eine erwartete *Connection* muss deshalb ihre eigene *Source* und Unge-
wissheit bewahren.

Dieses Kapitel benötigt keinen Katalog der *Classes*. Ein solcher Katalog würde zur Wahl nach Bezeichnung verleiten und überall dort veralten, wo er Werte, Ertrag oder eine vorausgesetzte Lebensweise einschmugelt. Der *Explorer* benötigt eine klarere Fähigkeit: Lies jede Systemeigenschaft und frage danach, welche Konsequenz sie für diesen Zweck besitzt.

Kann eine Gruppe diese Frage nicht beantworten, sollte sie die Lücke nicht mit dem Ruf des Systems füllen. Sie kann die Konsequenz prüfen, eine aktuelle *Source* suchen oder die Eigenschaft ungeklärt lassen. *Unknown* ist nützlicher als eine übernommene Rangliste.

Connections lesen

Der erste Kandidat bot während mehrerer Scouting-Besuche eine bequeme Route. Beim zweiten waren wiederholt mehr Schritte nötig, um die von der Gruppe genutzten Orte zu erreichen. Es liegt nahe, aus diesen Beobachtungen dauerhafte Eigenschaften der Systeme zu machen. Sie sind keine dauerhaften Eigenschaften.

Wormholes bilden kurzlebige *Connections*. Ihre Informationen geben grobe Hinweise auf Lebensdauer und Massenzustand, aber keinen Fahrplan und kein Versprechen für das nächste Ziel. Die um einen Kandidaten beobachtete *Chain* beschreibt den *Access* während dieser *Observation*. Auch wiederholte Aufzeichnungen können sie nicht zu Infrastruktur machen.

Wiederholte *Observations* können dennoch nützlich sein. Sie zeigen, was die Gruppe tatsächlich vorgefunden hat, wie oft sie Routen neu aufbauen musste und welche Bewegungen während der Stichprobe schwierig waren. Ihr Wert beruht auf bewahrten Grenzen: Daten, *Sources*, fehlenden Schichten und dem Zweck, nach dem die Routen beurteilt wurden.

Nenne dies ein Profil der beobachteten *Connections*. Das Wort *Profil* bedeutet keine garantierte Gruppe von Ausgängen. Es bezeichnet eine kontrollierte Geschichte dessen, was die Gruppe vorfand und was dieser Befund von ihr verlangte.

Die Aufzeichnung sollte vier Dinge auseinanderhalten.

Die gegenwärtige *Chain* zeigt die *Connections*, die dem letzten Beobachter zur Verfügung standen. Eine erwartete Beziehung bezeichnet etwas, von dem die Bewohner mit angegebener *Source* glauben, dass es wahrscheinlich gefunden wird. Der Routenbedarf beschreibt, was der Zweck wie oft bewegen muss. Die Routentoleranz legt fest, wie die Gruppe handelt, wenn diese Bewegung nicht wie geplant stattfinden kann.

Diese Ebenen beantworten verschiedene Fragen. Ein heute nützlicher Ausgang kann einen Scouting-Flug tragen, ohne zu beweisen, dass das System regelmässigen Transport unterstützt. Eine erwartete *Connection* kann den nächsten *Scan* leiten, ohne Ladung zum Transport freizugeben, bevor die Route beobachtet wurde. Eine schwierige Route kann akzeptabel sein, wenn die *Work* pausieren kann, aber ungeeignet, wenn *Replacement* von einer Frist abhängt.

Hier wird die Wahl von *Home* operativ. Die Gruppe wählt keine dauerhafte Strasse. Sie entscheidet, wie viel wiederkehrende *Work* sie akzeptieren kann, um vorübergehende Strassen zu entdecken, zu beurteilen und zu nutzen.

Der Vergleich der Kandidaten sollte deshalb Aussagen wie *dieses System hat guten Access* vermeiden. Benenne, was beobachtet wurde und was der Zweck verlangt. Vielleicht erforderte ein Kandidat bisher längere Transportwege, während der andere im gleichen Beobachtungszeitraum leichter erreichbar war. Ein Ergebnis kann bevorzugt werden, doch der Vergleich muss die Möglichkeit bewahren, dass die nächste *Chain* diese Erfahrung umkehrt.

Eine Entscheidung unter dieser Ungewissheit ist nicht schwach. Sie ist ehrlich. Schwach wäre es, einen unumkehrbaren Plan auf einer Routenbeschreibung aufzubauen, die nie beanspruchte, dauerhaft zu sein.

Der Beobachtungszeitraum sollte zur unterstützten Entscheidung passen. Ein kurzer Besuch kann belegen, dass während des Besuchs Bewegung möglich war. Er kann nicht den wiederkehrenden Aufwand des Wohnens belegen. Längere Beobachtung kann Unterschiede zeigen, wird aber nie zur Garantie. Die Aufzeichnung gewinnt an Wert, wenn sie ihren Umfang zeigt, und verliert ihn, wenn ruhige Phasen oder fehlende Beobachter stillschweigend als gewöhnliche Bedingungen behandelt werden.

Warte nicht auf eine perfekte Stichprobe. Auch das Warten kostet Zeit und kann die Gruppe dazu verleiten, so zu handeln, als sei der Kandidat bereits gewählt. Sammle genügend Belege für die Grösse der ersten Verpflichtung, halte diese umkehrbar und beobachte weiter, nachdem der Versuch begonnen hat.

Effect lesen

Ein *Effect* verändert die Bedingungen, unter denen Schiffe in einem *Wormhole System* arbeiten. Die offiziellen *Static Data* bezeichnen die *Effect Types* und die Systemdaten, zu denen sie gehören. So werden das Vorhandensein und der Typ eines *Effect* zu prüfbar Eigenschaften. Die operative Konsequenz hängt weiterhin vom Plan ab.

Derselbe *Effect* kann auf verschiedene Schiffe, *Fittings*, Aufgaben und Toleranzen treffen. Ihn als nützlich oder schädlich zu bezeichnen, bevor diese Beziehungen benannt sind, wiederholt den Fehler bei der *Class*. Eine Eigenschaft wird zur Rangfolge, ohne den Plan unter dem Urteil zu zeigen.

Lies den *Effect* in drei Durchgängen.

Stelle zuerst fest, welcher *Effect* gilt. Halte die aktuellen offiziellen Daten oder eine andere kontrollierte *Source* sichtbar. Verlasse dich nicht auf

eine aus einer alten Karte übernommene Bezeichnung, wenn die Eigenschaft geprüft werden kann.

Bestimme danach, welche erforderliche *Capability* er berührt. Die Frage ist nicht, ob der *Effect* beliebt ist. Sie lautet, ob er die Schiffe oder Handlungen verändert, von denen der benannte Zweck abhängt. Benötigt der Zweck eine *Capability* nicht, sollte ihre Wechselwirkung die Wahl nicht bestimmen.

Entscheide schliesslich, wie die Konsequenz geprüft wird. Eine Gruppe muss möglicherweise bestätigen, dass ihre vorgesehenen Schiffe, Reserven und Arbeitsgewohnheiten weiterhin geeignet sind. Die Prüfung gehört zur Gruppe und ihrem gegenwärtigen Plan. Sie darf nicht als zeitlose Regel für alle Leser verkleidet werden.

Für dieses Entscheidungsmodell ist keine numerische Tabelle der *Effects* nötig. Zahlenwerte würden eine eigene Prüfung aktueller Daten verlangen, und eine vollständige Liste würde zur Optimierung verleiten, bevor die *Work* verstanden wurde. Für *Explorer* soll der *Effect* als Betriebsbedingung sichtbar bleiben, nicht zur Identität des Systems werden.

Das Ergebnis lässt sich in einfacher Sprache festhalten. *Effect bestätigt; Wechselwirkung mit der geplanten Scanning Capability nach einer Prüfung akzeptiert. Oder: Effect bestätigt; Konsequenz für die erforderliche Work bleibt Unknown, daher hängt noch keine Verpflichtung davon ab.*

Eine solche Aufzeichnung ist weniger eindrucksvoll als eine farbcodierte Rangliste. Für die nächste Person ist sie nützlicher, weil die Grenze zwischen Eigenschaft, Interpretation und Entscheidung erhalten bleibt.

Bestehende Infrastruktur

Der zweite Kandidat wirkt einfacher, weil bereits eine *Structure* vorhanden ist. Ihre gegenwärtige Nutzung durch die Gruppe ist ein echter Vorteil. Sie entspricht nicht der Kontrolle darüber.

Upwell Structure Profiles legen fest, welche *Services* anderen Spielern angeboten werden. *Access Lists* bestimmen, wer diese *Services* nutzen kann. Eine sichtbare *Structure*, ein angebotener *Service* und der tatsächliche *Access* eines *Character* sind deshalb getrennte Tatsachen. Jede kann wahr sein, während eine andere falsch ist oder sich später verändert.

Eine Gastregelung sollte als Abhängigkeit von einem anderen Eigentümer gelesen werden. Welche *Capabilities* sind jetzt verfügbar? Wer besitzt die Kontrolle über *Profile* und *Access Lists*? Wie wird eine Veränderung bemerkt? Welche *Work* stoppt, wenn *Access* entzogen wird? Welche Vermögenswerte können ohne diesen *Access* zurückgeholt werden? Die Antworten entscheiden, ob die Nutzung als Gast einen Versuch, gewöhnliche *Work* oder nur eine vorübergehende Bequemlichkeit trägt.

Der Eigentümer trägt andere Verpflichtungen. Nur Mitglieder einer spielergeführten *Corporation* mit der passenden Rolle können eine *Upwell Structure* einsetzen. Die gegenwärtigen offiziellen Regeln für die Bereitstellung schliessen zudem *Shattered Wormhole Systems* und Thera aus. Diese Grenzen sind für die Wahl wichtig, weil die Gruppe nicht in jedem Kandidaten den Gastzugang durch eine eigene *Upwell Structure* ersetzen kann.

Das ist kein Grund, *Ownership* automatisch vorzuziehen. *Ownership* schafft Kontrolle über einige Entscheidungen und Verantwortung für viele weitere. Die *Structure*, ihre *Services*, Betriebsmittel, Zustände, Gestaltung des *Access* und spätere Entfernung werden zu Verpflichtungen. Kapitel 8 wird diese Pflege direkt untersuchen. Vorerst genügt es, die Vorstellung zurückzuweisen, *Ownership* verwandle Ungewissheit in Dauerhaftigkeit.

Sowohl Gastnutzung als auch *Ownership* können ein gültiges *Home* tragen. Sie versagen auf unterschiedliche Weise.

Eine *Capability* des Gastes kann verschwinden, weil ein anderer Eigentümer *Access*, *Services* oder die Vereinbarung selbst verändert. Der Gast kann diese Abhängigkeit verringern, indem er die Lagerung begrenzt,

eine Alternative bewahrt und festlegt, wie schnell *Work* beendet werden kann. Der Eigentümer kann *Access* kontrollieren und trotzdem durch vernachlässigte Betriebsmittel, unklare Verantwortung oder den Zustand der Infrastruktur eine *Capability* verlieren. Kontrolle verlagert die Abhängigkeit; sie kann sie nicht entfernen.

Bestehende Infrastruktur muss ausserdem für jeden *Service* einzeln gelesen werden. Die Anwesenheit der *Structure* belegt, dass das Objekt besteht. Sie kann weder beweisen, dass jeder gewünschte *Service* angeboten wird, noch dass der aktuelle *Character* ihn nutzen kann oder dass er für die Dauer des Plans verfügbar bleibt. Bestätige nur die *Capabilities*, die der Zweck verlangt, und binde die *Observation* an eine Zeit.

Die beiden Kandidaten wirken nun weniger ähnlich. Einer verlangt von der Gruppe, einen grösseren Teil der Infrastrukturverpflichtung selbst zu tragen. Der andere ermöglicht einen leichteren Anfang, legt aber eine wichtige Entscheidung in fremde Hände. Der bedeutsame Unterschied liegt nicht zwischen Unabhängigkeit und Abhängigkeit. Er liegt darin, welche Abhängigkeit die Gruppe sehen, begrenzen und beenden kann.

Ein einfacher Beginn darf nicht mit einer kleinen Verpflichtung verwechselt werden. Gastzugang kann die unmittelbare *Work* der Bereitstellung beseitigen und zugleich Vermögenswerte und Routinen hinter einer Berechtigung ansammeln, welche die Gruppe nicht erhalten kann. *Ownership* kann mehr Vorbereitung verlangen und zugleich die kontrollierenden Entscheidungen verdeutlichen. Beide Anordnungen werden schwer, sobald die dahinter angesammelten Vermögenswerte nicht mehr nach dem Zeitplan der Gruppe entfernt werden können.

Deshalb gehört *Access* selbst dann in die Aufzeichnung der Auswahl, wenn er unkompliziert wirkt. Notiere die tatsächlich genutzte *Capability*, die kontrollierende Partei, die jüngste Bestätigung und die verfügbare Handlung, falls sie sich verändert. Die Aufzeichnung macht einen

anderen Eigentümer nicht berechenbar. Sie verhindert, dass Bequemlichkeit mit Kontrolle verwechselt wird.

Zielkonflikt akzeptieren

Nachdem die Eigenschaften getrennt wurden, gewinnt keiner der Kandidaten jede Spalte.

Das erste System passt zur geplanten *Work* und gibt der Gruppe mehr Kontrolle über die erforderliche *Capability*. Seine beobachteten Routen haben *Replacement* langsamer und aufwendiger gemacht. Seine Wahl bedeutet, den Transportaufwand zu akzeptieren und zu begrenzen, was zwischen nützlichen *Connections* unersetzbar werden darf.

Das zweite erlaubt der Gruppe, mit weniger eigener Infrastruktur zu beginnen. Der gegenwärtige Gastzugang trägt die erforderlichen *Services*. Seine Wahl bedeutet, zu akzeptieren, dass eine andere *Corporation* eine *Capability* kontrolliert, von der die *Work* abhängt. Die gewählte Anordnung muss der Gruppe erlauben, zu pausieren, Vermögenswerte zurückzuholen oder zu gehen, wenn sich dieser *Access* verändert.

Keine Rechnung macht diese Kosten gleich. Die Entscheidung hängt davon ab, welchen *Failure* der Zweck toleriert und welche Last die Gruppe tatsächlich tragen kann. Eine Gruppe mit wenig Zeit für Infrastruktur kann begrenzte Gastnutzung vernünftigerweise vorziehen. Eine Gruppe, deren *Work* keinen ungewissen *Access* toleriert, kann die Transportlast bevorzugen. Kehre den Zweck um und dieselben Systeme können ihre Reihenfolge tauschen.

Behandle die Wahl als begrenzte Hypothese.

Benenne den Zweck: welche *Work*, für wen und für wie lange. Erfasse die geprüften Eigenschaften: *Class*, *Effect*, beobachtete *Connections* und erforderliche Infrastruktur. Benenne den akzeptierten Zielkonflikt in einem Satz. Lege danach fest, welche Belege eine Überprüfung auslösen.

Die Entscheidungsaufzeichnung sollte auch verworfene Interpretationen bewahren. *Die Class wurde erfasst, aber nicht als Ertragsrangliste verwendet. Der gegenwärtige Gastzugang wurde bestätigt, aber nicht als dauerhaft behandelt. Beobachtete Connections flossen in den Versuch ein, wurden jedoch nicht zu garantierten Routen erhoben.* Diese Grenzen schützen davor, dass die Entscheidung beim Nacherzählen gewisser wird.

Eine Überprüfung kann durch wiederholtes Versagen notwendiger Bewegungen, veränderten *Access* oder einen nicht verfügbaren *Service* ausgelöst werden. Eine Wechselwirkung des *Effect*, welche die erforderliche *Work* ungeeignet macht, oder eine von der Gruppe nicht tragbare Arbeitslast kann die Überprüfung ebenfalls auslösen. Die Bedingung sollte beobachtbar sein. *Wir mögen das System nicht mehr lässt sich schwer in eine Handlung übersetzen. Die Route für Replacement hat wiederholt die in unserem Versuch erlaubte Verzögerung überschritten zeigt, welche Annahme versagte.*

Begrenze schliesslich die erste Verpflichtung. Bewege nur die Vermögenswerte, die zur Prüfung des Zwecks nötig sind. Baue nicht mehrere Tätigkeiten um den Kandidaten auf, bevor die erste ihre Tragfähigkeit gezeigt hat. Lege einen Termin oder ein Ereignis für die Überprüfung fest und bewahre einen Weg, den Versuch zu beenden, ohne dass das System ausgerechnet beim Aufbruch bequem werden muss.

Dieses Vorgehen garantiert nicht die Wahl des besseren Kandidaten. Es macht den Grund für die Wahl sichtbar. Ändern sich die Bedingungen, kann die Gruppe eine aufgezeichnete Hypothese überarbeiten, statt eine Identität zu verteidigen.

Home ist nicht das System, das nichts verlangt. Ein solches System gibt es nicht. *Home* ist ein System, dessen notwendige Zielkonflikte genau genug beobachtet wurden, um sie zu akzeptieren, und dessen verbleibende *Unknowns* klein genug gehalten wurden, um einen Irrtum zu überstehen.

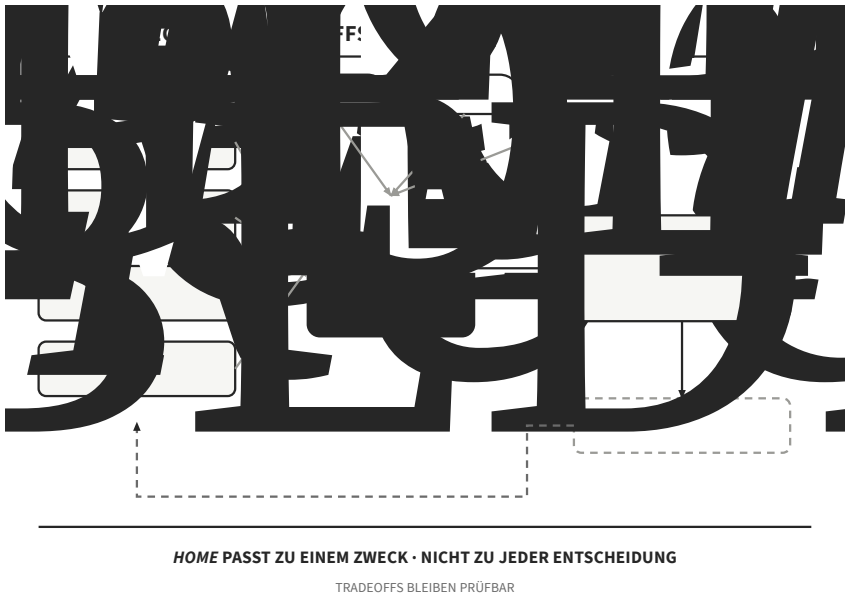


Abbildung AFH-B2-FIG-002 — Home Fit ist eine begrenzte Hypothese. Home passt durch ausdrücklich benannte Tradeoffs zu einem Zweck, nicht durch eine Eigenschaft, die jede Entscheidung günstig macht.

Feldprinzip

Wähle den Zweck vor der Eigenschaft.

Erfasse *Class*, *Effect*, *Connections* und Infrastruktur als getrennte Ebenen. Lass jede von ihnen erst in Beziehung zu *Work*, Menschen, Zeit und *Exit* der Gruppe günstig oder ungünstig werden. Behandle gegenwärtigen *Access* als gegenwärtig, wiederholte Routen als beobachtete Geschichte und *Ownership* als Verantwortung statt als Sicherheit.

Der Entscheid zu bleiben sollte eine begrenzte Hypothese mit Bedingungen für die Überprüfung sein, keine Erklärung, das ideale System sei gefunden worden.

Nach der Wahl von *Home* stellt sich die Frage, wie viel an diese Hypothese gebunden werden darf. Kapitel 3 wendet sich von der Auswahl

dem Gewicht von Vermögenswerten, *Replacement* und konzentriertem Verlust zu.

Kapitel 3 - Verpflichtung wiegt

Das Schiff ist nicht teuer genug, um jemanden zu beunruhigen.

Es ist eine gewöhnliche Transporthülle, bereits in *Home* und bereit für die nächste Bewegung. Ihr Marktwert könnte ersetzt werden, ohne die Gruppe zu gefährden. In der Vermögensliste erscheint sie deshalb als kleine Zeile zwischen grösseren.

Dann fragt ein Pilot, was geschieht, wenn sie nicht verfügbar ist.

Das Schiff ist die einzige gegenwärtig vorbereitete Hülle, die Treibstoff und Arbeitsmaterial durch die erwarteten Routen der Gruppe bewegen kann. Es gehört ausserdem zum Plan, Vermögenswerte zu entfernen, falls der Versuch in *Home* endet. Eine andere Hülle könnte diese Rolle übernehmen, doch dieses *Replacement* befindet sich ausserhalb der gegenwärtigen *Chain*. Es hereinzubringen würde eine Route, Zeit und einen Piloten verlangen, der die Bewegung abschliessen kann.

Der Preis des Schiffs hat sich nicht verändert.

Sein Gewicht schon.

Was zurückbleibt

Ein Inventar beantwortet die Frage «Was ist hier?». Eine Verpflichtung fragt, wofür die aufgeführten Objekte verantwortlich gemacht wurden.

Der Unterschied beginnt bei *Ownership*. Eine Hülle in persönlichem Besitz, ein Stapel im Hangar einer *Corporation* und ein über einen anderen Piloten verfügbarer Gegenstand können alle dieselbe *Work* zu tragen scheinen. Sie bieten nicht dieselbe Kontrolle. Wer einen Gegenstand sehen kann, kann ihn möglicherweise nicht entfernen. Wer ihn entfernen kann, trägt vielleicht nicht die Entscheidung, für die er zurückgelegt wurde.

Corporation Roles machen diese Unterschiede konkret. Das Recht, einen Hangar zu sehen, seinen Inhalt zu entnehmen oder Container zu

entfernen, besteht aus getrennten Berechtigungen. *Management Roles* und *Industry Roles* gewähren weitere Fähigkeiten. Eine gemeinsame Bezeichnung am Hangar macht nicht jeden Gegenstand für jedes Mitglied gleichermassen verfügbar.

Kapitel 7 untersucht *Access* im Detail. Vorerst ist entscheidend, dass eine Vermögensaufzeichnung sowohl den Eigentümer als auch die nutzbare *Capability* bewahrt. *Vermögenswert der Corporation* ist zu ungenau, wenn ein Pilot den Gegenstand sehen, ein anderer ihn entnehmen und ein dritter ihn für gemeinsame *Work* reserviert glauben kann.

Als Nächstes folgt der Zweck. Was ermöglicht das Objekt der Gruppe? Die Transporthülle ermöglicht Bewegung. Eine Hülle für das *Scanning* ermöglicht den Wiederaufbau der *Chain*. Treibstoff verhindert, dass ein benötigter *Service* nur noch als Symbol in einem Plan besteht. Ein *Blueprint* kann zukünftige *Work* darstellen, die zusätzlich von Materialien, *Access* und einem verfügbaren *Service* abhängt.

Diese Sicht erfasst unfertige *Work* ebenso wie eingelagerte Objekte. Für eine Aufgabe zurückgelegte Materialien und ein beladenes Schiff, das auf einen Piloten wartet, binden künftige Entscheidungen. Auch ein Vermögenswert, der einer späteren Schicht versprochen wurde, bindet künftige Entscheidungen. Sie alle können noch am selben Ort erscheinen, besitzen aber inzwischen eine Zuordnung.

Kapitel 1 benannte Zweck, Träger, sichtbaren Zustand, *Failure* und vorgesehenes Ende jeder Verpflichtung. Eine Vermögensaufzeichnung schärft diese Methode mit fünf praktischen Fragen:

Wem gehört der Vermögenswert? Welche *Capability* nutzt ihn? Wo befindet sich sein *Replacement*? Was fällt gleichzeitig aus? Wie kann seine *Exposure* verringert werden?

Diese Felder sind nicht für jede Munition oder jedes preiswerte Werkzeug bestimmt. Sie gehören zu Vermögenswerten, deren Fehlen den Plan verändern würde. Das Ziel ist kein vollkommenes Inventar.

Gesucht werden die wenigen Zeilen, die mehr *Work* tragen, als ihr Preis vermuten lässt.

Die Transporthülle wird durch diese Prüfung sichtbar. Sie gehört einem Piloten, unterstützt mehrere gemeinsame Bewegungen und besitzt in *Home* keinen unmittelbar verfügbaren Ersatz. Ihr Verlust wäre nicht der grösste finanzielle Schaden. Er würde die meisten Versprechen zugleich brechen.

Dieser Unterschied verändert das Gespräch, bevor etwas bewegt wird. Der Eigentümer kann bereit sein, die Hülle zu riskieren, während die Gruppe die daran gebundene *Work* nicht verlieren will. Keines der Urteile hebt das andere auf. Das eine beschreibt persönliche Toleranz, das andere eine gemeinsame Abhängigkeit. Beide müssen sichtbar sein, bevor Zustimmung zu einem mit Zustimmung zu beidem verwechselt wird.

Die Aufzeichnung muss diesen Unterschied nicht mit einem Preis versehen. Sie muss zeigen, wo sich die Versprechen trennen. Der Eigentümer kann den Verlust der Hülle akzeptieren. Die Treibstoffbewegung kann anders zugewiesen werden. Der Rückzugsplan kann eine andere Methode benennen. Sind diese Entscheide getrennt, wird das Objekt wieder zu einem gewöhnlichen Vermögenswert statt zur stillen Antwort auf drei verschiedene Probleme.

Eine Verpflichtung sollte deshalb in dem Moment überprüft werden, in dem ein Vermögenswert einen neuen Zweck erhält, nicht nur bei seinem ersten Eintreffen in *Home*. Eine als Ersatz eingetroffene Hülle kann nach einer Änderung am *Fitting*, einer aufgebrauchten *Reserve* oder dem Weggang eines Piloten unverzichtbar werden. Inventaränderungen sind sichtbar. Veränderungen der Verantwortung oft nicht.

Entfernung des *Replacement*

Über *Replacement* wird oft gesprochen, als handle es sich um einen Kauf. In Anoikis ist das Objekt nur der Anfang.

Ein *Replacement* muss bestehen, dem vorgesehenen Eigentümer zur Verfügung stehen und den Ort erreichen, an dem seine *Capability* benötigt wird. Jeder Schritt kann Entfernung hinzufügen. Der Markt kann im *Known Space* nahe liegen, während die gegenwärtige Route ungeeignet ist. Eine Hülle kann bereits anderswo bestehen und trotzdem einen Piloten, *Access* und Zeit verlangen, über welche die Gruppe beim Eintritt eines *Failure* nicht verfügt.

Entfernung wird deshalb ebenso in Entscheidungen wie in Systemen gemessen.

Beginne bei der Verfügbarkeit. Besteht ein *Replacement* tatsächlich oder nimmt der Plan nur an, eines beschaffen zu können? Kontrolliert es dieselbe Person, die es einsetzen soll? Verlangen *Fitting*, Beladung oder Übertragung eine weitere *Capability*, die zusammen mit dem ursprünglichen Vermögenswert ausfallen könnte?

Lies danach die Route. *Wormhole Connections* sind kurzlebig und begrenzen sowohl einzelne Transits als auch die gesamte Masse. Die gegenwärtige *Chain* kann jetzt eine Bewegung für *Replacement* tragen, ohne zu versprechen, dass dieselbe Bewegung weiterhin möglich bleibt. Ausserhalb von *Home* wird ein Vermögenswert erst zur nutzbaren Rückfallebene, wenn die für seinen Transport benötigte Route für die tatsächliche Bewegung beobachtet wurde.

Als Nächstes folgt die Zeit. Wie lange kann die Gruppe ohne die *Capability* arbeiten? Ein Weg für *Replacement*, der mehrere Sitzungen beansprucht, kann für optionale *Work* ausreichen. Derselbe Weg ist schwach, wenn *Scanning*, *Access* zu eingelagerten Vermögenswerten oder ein geplanter *Exit* unmittelbar vom fehlenden Objekt abhängen.

Benenne schliesslich die Menschen. *Replacement* kann jemanden mit dem richtigen *Access* zu Vermögenswerten, einen Piloten für den Transport und eine Person mit der Befugnis zur Änderung des Plans verlangen. Gehören alle drei Rollen einem abwesenden Mitglied, ist das Objekt nicht die einzige konzentrierte Abhängigkeit.

Das ist die nicht finanzielle Seite von *Replacement*. Eine Gruppe kann mehrere Hüllen besitzen und dennoch nur über eine nutzbare *Capability* verfügen. Die Begrenzung kann in einer Rolle, einem *Timestamp*, einem *Bookmark* oder einer Entscheidung liegen. Sie kann auch eine Person sein, die weiss, weshalb ein Gegenstand an seinem Ort liegt.

Die Lösung besteht nicht darin, alles zu verdoppeln. Verdoppelung ohne Trennung kann einen grösseren Stapel schaffen, der demselben *Failure* ausgesetzt ist. *Replacement* erhält Bedeutung, wenn es die Bedingung übersteht, welche das Original voraussichtlich beseitigen wird.

Diese Bedingung sollte klar benannt werden. Geht es beim erwarteten *Failure* um den Verlust einer Hülle im Flug, kann eine weitere Hülle in *Home* genügen. Betrifft er den verlorenen *Access* zu *Home*, ist dieselbe *Reserve* überhaupt keine *Reserve*. Die erwartete Abwesenheit des Eigentümers verändert, was als Rückfallebene zählt: Ein weiterer persönlich kontrollierter Vermögenswert wiederholt die Abhängigkeit, auch wenn er anderswo liegt.

Die Gruppe kann dies prüfen, ohne ein Unglück nachzustellen. Wähle eine erforderliche *Capability* und nimm an, ihre primäre *Source* sei nicht verfügbar. Beginne mit den Aufzeichnungen, Menschen und Routen, die verbleiben würden. Wer bemerkt die Abwesenheit? Wer darf die Rückfallebene aktivieren? Was muss bewegt werden, bevor die *Capability* wieder funktioniert? Die Übung endet, wenn die Gruppe handeln kann, nicht wenn jemand auf einen weiteren Gegenstand in der Vermögensliste zeigt.

Die Antwort kann zeigen, dass kein weiterer Kauf nötig ist. Vielleicht muss ein *Bookmark* geteilt, eine bestehende Hülle neu zugewiesen oder ein zweiter Pilot in den Kontext versetzt werden, das Vorhandene zu nutzen. Die Entfernung des *Replacement* wird durch das Entfernen von Entscheidungen und Abhängigkeiten ebenso verkürzt wie durch das Annähern eines Objekts.

Bei der Transporthülle kann ein zweites, identisches Schiff daneben gegen einen mechanischen Verlust schützen. Dieses Duplikat bietet keinen Schutz vor verlorenem *Access* zur *Structure*, einer Route, die den Transport nicht tragen kann, oder einem abwesenden Piloten, dem beide Hüllen gehören. Die nützliche Frage lautet nicht, ob ein weiteres Objekt besteht. Sie lautet, ob die *Capability* aus dem Zustand der Rückfallebene wiederhergestellt werden kann, mit dem die Gruppe rechnet.

Konzentration verändert Verlust

Konzentration ermöglicht die *Work in Home*. Schiffe, Materialien, Aufzeichnungen und *Services* werden zusammengebracht, weil Nähe den Aufwand verringert. Dieselbe Anordnung ermöglicht zugleich, dass ein Ereignis mehrere Verpflichtungen erreicht.

Finanzielle Konzentration ist die sichtbarste Form. Viele wertvolle Vermögenswerte an einem Ort können gemeinsam exponiert sein. Der Marktwert ist jedoch nur ein Massstab. Auch *Capability* kann konzentriert sein, wenn die Objekte preiswert sind.

Die Transporthülle trägt Bewegung, Treibstofflieferung und einen Teil des Plans für den *Exit*. Drei Zwecke haben sich um einen Vermögenswert gesammelt. Ein zweites Schiff würde nur helfen, wenn es unter einem anderen *Failure* verfügbar bliebe. Andernfalls hat die Gruppe Wert hinzugefügt, ohne *Capability* zu trennen.

Der Standort ist eine weitere Form. Persönliche Vermögenswerte und jene einer *Corporation* können verschiedene Eigentümer haben und trotzdem dieselbe *Structure*, denselben Weg für *Access* und dieselbe Route hinaus teilen. Sie getrennt zu zählen ändert nichts daran, dass ein verlorener Standort oder eine veränderte Bedingung für *Access* sie gemeinsam betreffen kann.

Auch Zeit kann *Exposure* konzentrieren. Mehrere unabhängige Bewegungen können für sich vernünftig sein. Hängen sie alle von derselben

alternden *Connection* ab, ist ihr Risiko nicht länger unabhängig. Die Route soll mehrere Verpflichtungen tragen, bevor sie sich verändert.

Menschen schaffen Konzentration, wenn eine Person mehrere erforderliche Fähigkeiten trägt. Das Mitglied, dem die Transporthülle gehört, kann zugleich die einzige Person sein, welche die betreffenden Vermögenswerte der *Corporation* entfernen darf und weiss, welche Materialien reserviert wurden. Nichts muss zerstört werden, damit diese *Capability* verschwindet. Abwesenheit genügt.

Informationen können auf dieselbe Weise versagen. Eine Liste kann jedes Objekt erfassen und zugleich Zweck und Eigentümer in der Erinnerung einer Person belassen. Das Inventar bleibt bestehen, doch die Gruppe weiss nicht mehr, welcher Vermögenswert verbraucht, bewegt oder aufgegeben werden darf.

Lies Konzentration, indem du fragst, was gemeinsam ausfallen würde. Beginne nicht mit einer hypothetischen Katastrophe. Beginne mit einer nicht verfügbaren Hülle, einer veränderten Bedingung für *Access*, einer unbrauchbaren Route oder einer abwesenden Person. Verfolge den Plan, während sich die ersten Folgen entfalten.

Folge diesen Konsequenzen nur so weit, wie die Gruppe reagieren müsste. Vielleicht pausiert der Transport, was den Treibstoff verzögert und eine geplante Phase von *Work* in eine Bewegung zur Rückgewinnung verwandelt. Vielleicht bleibt *Access* bestehen, doch der einzige Pilot, der die Reservierungsliste versteht, ist abwesend. Niemand kann dann erkennen, welches Material bedenkenlos verbraucht werden darf. Das erste Ereignis ist klein; die angesammelte Verpflichtung gibt ihm Reichweite.

Getrennte Aufzeichnungen können diese Reichweite verbergen. Eine persönliche Vermögensliste, ein Inventar der *Corporation* und ein Bewegungsplan können jeweils vernünftig aussehen. Ihre gemeinsame Abhängigkeit erscheint erst, wenn sie zusammen gelesen werden. Dieselbe Hülle kann eine kleine persönliche *Exposure*, eine erforderliche

Capability der *Corporation* und die unausgesprochene Verbindung in einem Aufbruchsplan sein. Keine einzelne Aufzeichnung ist falsch. Die Schlussfolgerung aus jeder allein bleibt unvollständig.

Enden mehrere Verpflichtungen am selben Punkt, trägt dieser mehr Gewicht, als sein isolierter Wert vermuten lässt.

Verringerung kann verschiedene Formen annehmen. Trenne eine Rückfallebene vom primären Standort. Gib einer weiteren Person den *Access* und den Kontext, die für ihre Nutzung nötig sind. Lege nicht mehrere dringende Bewegungen auf dieselbe *Connection*. Entferne alte Vermögenswerte, bevor neue aus einem Versuch eine Ansammlung machen.

Keine dieser Handlungen beseitigt Verlust. Sie verringern die Zahl der *Capabilities*, die gleichzeitig wiederhergestellt werden müssen.



GEWICHT STEIGT, WENN KÜNFTIGE MÖGLICHKEITEN EINE ABHÄNGIGKEIT TEILEN

Abbildung AFH-B2-FIG-003 — Verpflichtungsgewicht durch Abhängigkeitskonzentration. Verpflichtung gewinnt Gewicht, wenn mehrere künftige *Capabilities* vom selben Vermögenswert, *Access Path*, Weg, Zeitpunkt oder Menschen abhängen.

Andere Rückgewinnung

Der Ausdruck *Asset Safety* kann das falsche Bild erzeugen, wenn er aus anderen Gebieten in den *Wormhole Space* übertragen wird.

Upwell Structures im *Wormhole Space* verwenden nicht das gewöhnliche System von *Asset Safety*. Unter Umständen, die anderswo zu *Asset Safety* führen, werden betroffene Vermögenswerte stattdessen in den Weltraum ausgestossen und stehen danach allen Spielern zur Bergung offen. Dies gilt für persönlichen Besitz und betroffene Bestände einer *Corporation*.

Die Konsequenz ist einfach und schwerwiegend: Lagerung in einer *Wormhole Structure* darf nicht als verzögerter Dienst zur Rückgewinnung behandelt werden.

Das bedeutet nicht, dass jeder eingelagerte Vermögenswert verloren geht. Der Plan kann das gewöhnliche Ergebnis von *Asset Safety* jedoch nicht als Rückfallebene beanspruchen. Er muss berücksichtigen, was gemeinsam verloren gehen kann, was bewusst entfernt werden kann und welche *Capability* ausserhalb des exponierten Standorts bestehen muss.

Ownership bleibt wichtig. Persönliche Vermögenswerte und Bestände einer *Corporation* haben vor einem Verlust unterschiedliche Entscheidungsträger und Wege für *Access*. Ein Hangar der *Corporation* kann einen gemeinsamen Zweck bewahren und zugleich begrenzen, wer handeln darf. Eine persönliche Hülle kann von ihrem Eigentümer rasch kontrolliert werden, bleibt aber für alle anderen unerreichbar. Das Ergebnis im *Wormhole Space* beseitigt diese Unterschiede nicht; es macht sie wichtiger, bevor Rückgewinnung unmöglich wird.

Das Wort *Rückgewinnung* sollte deshalb vorsichtig verwendet werden. Es kann bedeuten, die *Capability* von anderswo zu ersetzen, exponierte Vermögenswerte bei bestehendem *Access* zu entfernen, Aufzeichnungen neu aufzubauen oder zu akzeptieren, dass eine geplante *Reserve*

verloren ist. Es darf nicht stillschweigend bedeuten, dass dieselben Gegenstände später an einen bequemen Ort geliefert werden.

Diese Bedeutungen verlangen verschiedene Vorbereitungen. *Replacement* benötigt eine überlebende *Source* und eine Route. Bewusstes Entfernen benötigt einen Auslöser, der früh genug zum Handeln führt. Der Wiederaufbau von Aufzeichnungen benötigt Kopien, die nicht denselben verlorenen Weg für *Access* teilen. Akzeptanz verlangt, dass Eigentümer verstehen, was verschwinden kann, bevor ihre Vermögenswerte gebunden werden. Alle vier Ergebnisse *Rückgewinnung* zu nennen verbirgt die Entscheidung bis zu dem Moment, in dem die Gruppe den geringsten Spielraum dafür besitzt.

Die nützliche Vorbereitung erfolgt deshalb, solange *Home* noch gewöhnlich wirkt. Benenne, welche Reaktion zu jeder wichtigen Verpflichtung gehört. Gib für *Replacement* die *Source* an. Bewusstes Entfernen benötigt eine Bedingung, die es auslöst; Akzeptanz verlangt eine ausdrückliche *Exposure*. Eine unbestimmte Hoffnung auf spätere Rückgewinnung ist keine vierte Möglichkeit.

Dieser Unterschied verändert die Erfassung von Wert. Ein teures Objekt mit getrenntem *Replacement* kann verkraftbar sein. Kontrolliert ein preiswertes Objekt den einzigen Weg zum *Replacement*, kann es entscheidend sein. Ein grosses gemeinsames Lager bleibt möglicherweise handhabbar, wenn jeder Eigentümer die *Exposure* versteht und *Work* ohne dieses Lager weitergehen kann. Zu schwer kann dagegen ein kleines Lager sein, dessen Verlust alle übrigen Vermögenswerte hinter derselben fehlenden *Capability* festsitzen lässt.

Asset Safety ist nicht der einzige Grund, Konzentration zu begrenzen. *Connections*, *Access*, Menschen und Zeit machen dies bereits nötig. Das besondere Ergebnis des *Wormhole Space* beseitigt eine bequeme Annahme und macht den verbleibenden Plan sichtbar.

Exposure begrenzen

Es gibt keinen universellen Betrag in *ISK*, den ein Bewohner sicher in *Home* zurücklassen darf. Dieselbe Zahl kann für eine Gruppe unbedeutend, für eine andere unersetzbar und für beide irreführend sein.

Eine Grenze für *Exposure* sollte zuerst *Capability* und erst danach den Preis beschreiben.

Frage, was die Gruppe nach dem akzeptierten Verlust weiterhin tun können muss. Kann sie die *Chain* durch *Scanning* neu aufbauen? Jeder Eigentümer sollte eine brauchbare Rückfallebene erreichen können. *Work* sollte pausieren können, ohne einen weiteren Notfall zu verursachen. Ebenso muss klar sein, ob die Gruppe das Verbleibende entfernen kann oder ob der Plan für den *Exit* bereits mit dem ersten Vermögenswert versagt hat.

Die Antworten bestimmen einen Zustand für die Rückfallebene. Er kann aus verringerter *Work* bestehen, während *Scanning* und Aufbruch weiterhin verfügbar sind. Ein anderer gültiger Zustand ist eine vollständige Pause, während *Replacement* durch eine neu beobachtete Route hereingebracht wird. Stattdessen kann die Gruppe entscheiden, den Versuch zu beenden, bevor ein weiterer Vermögenswert nach *Home* gelangt.

Eine brauchbare Grenze lässt sich ohne Preis ausdrücken: *Kein einzelner Verlust darf zugleich unsere für die Work benötigte Capability und die Mittel für ihr Replacement beseitigen.* Eine andere Gruppe kann wählen: *Hinter dem Gastzugang bleiben nur Vermögenswerte, deren Eigentümer den vollständigen Verlust akzeptieren können.* Die Formulierung unterscheidet sich, weil sich die Verpflichtungen unterscheiden.

Wende die Grenze auf gleichzeitige *Exposure* an. Eine Bewegung kann akzeptabel sein, während drei ähnliche Bewegungen von derselben Route abhängen würden. Eine *Reserve* kann vernünftig sein, bis primäre Quelle und Rückfallebene gemeinsam gelagert werden und der Unter-

schied verschwindet. Ein abwesender Spezialist kann handhabbar sein, bis jede Handlung zur Rückgewinnung auf diese Person wartet.

Lege danach Auslöser für die Verringerung fest. Füge keine Vermögenswerte mehr hinzu, wenn die Rückfallebene unklar wird. Bewege oder verbrauche alte Bestände, bevor ein zweiter Zweck dieselbe *Reserve* beansprucht. Trenne persönlichen Besitz von der Verantwortung der *Corporation*. Bestätige, wer gemeinsame Bestände sehen, entnehmen und bewegen kann, bevor die *Capability* benötigt wird.

Die Gruppe sollte ausserdem benennen, wer die Grenze auslösen darf. Wenn alle übermässige *Exposure* erkennen können, aber niemand die Bewegung anhalten darf, ist die Grenze nur ein Kommentar. Befugnis und Verantwortung müssen zusammentreffen, bevor der nächste Vermögenswert die Verbindung durchquert.

Grenzen benötigen auch einen Zeitpunkt für die Überprüfung. Der Versuch kann mit wenigen Vermögenswerten in *Home* und einer klaren Rückfallebene beginnen und danach mit jeder vernünftigen Bewegung Materialien, spezialisierte Hüllen und unfertige Aufgaben ansammeln. Nichts überschreitet eine dramatische Schwelle, doch die bewahrte *Capability* wird allmählich von allem hinzugefügten abhängig. Eine Überprüfung nach einer bedeutenden Bewegung, einer Veränderung des *Access* oder einem neuen gemeinsamen Zweck ist nützlicher, als auf ein Kalenderdatum zu warten.

Die Überprüfung kann kurz sein. Lies die gegenwärtigen Verpflichtungen laut, entferne die primäre *Capability* aus dem Bild und beschreibe den verbleibenden Zustand. Kann die Gruppe weiterhin pausieren, durch *Scanning* eine Route aufbauen, entscheiden und das wiederherstellen, was sie bewahren wollte, beschreibt die Grenze noch die Wirklichkeit. Hängt die Erklärung von der Rückkehr des fehlenden Vermögenswerts, einer unbestätigten Route oder der Reaktion eines abwesenden Eigentümers ab, hat die Gruppe sie bereits überschritten.

Übermäßige *Exposure* verlangt nicht immer eine Evakuierung. Vielleicht muss eine Aufgabe abgeschlossen werden, bevor eine andere beginnt. Eine geliehene *Capability* kann zurückgegeben, eine *Reserve* verringert oder eine Aufzeichnung über den *Access* hinaus verschoben werden, den sie beschreibt. Die kleinste Handlung, welche die Rückfallebene wiederherstellt, ist oft besser als ein weiteres Objekt zur Lösung einer Verpflichtung, die durch zu viele Objekte entstanden ist.

Kehre zur Transporthülle zurück. Ihr Marktwert bleibt gewöhnlich. Unter der Grenze für *Exposure* kann sie jedoch nicht zugleich die einzige vorbereitete *Capability* für Bewegung, der einzige Treibstofftransporter und die vorausgesetzte Methode des Rückzugs sein. Ein Zweck kann getrennt, eine Rückfallebene anderswo vorbereitet oder die Zahl der Vermögenswerte verringert werden, die von der Rückkehr der Hülle abhängen.

Die richtige Handlung wird nicht von der Hülle bestimmt. Sie wird von dem Zustand der Rückfallebene bestimmt, den die Gruppe bewahren will.

Feldprinzip

Wert wird zu Gewicht, wenn mehrere künftige Entscheidungen von demselben Vermögenswert, demselben Weg für *Access*, derselben Route, derselben Zeit oder derselben Person abhängen.

Erfasse *Ownership*, Zweck, Entfernung des *Replacement*, gleichzeitigen *Failure* und die verfügbare Verringerung. Zähle *Replacement* nur, wenn die Gruppe die *Capability* aus dem nach einem Verlust erwarteten Zustand wiederherstellen kann. Betrachte das gewöhnliche Ergebnis von *Asset Safety* bei einer *Upwell Structure* im *Wormhole Space* nicht als solchen Zustand.

Setze Grenzen für *Exposure* um das, was möglich bleiben muss, nicht um einen universellen Preis. Verringere die Verpflichtung, bevor der erste Verlust sowohl die *Work* als auch den Rückweg daraus beseitigt.

Sobald mehrere Menschen dieselben Vermögenswerte und Aufzeichnungen nutzen, ist das nächste Problem nicht mehr nur ihr Gewicht. Die Frage lautet, was tatsächlich geteilt wird, wer es pflegt und welche Annahmen *Access* nicht rechtfertigt. Kapitel 4 beginnt auf diesem gemeinsamen Boden.

Kapitel 4 - Gemeinsamer Boden

Die *Chain* ist aktuell.

Zwei Piloten sagen es innerhalb derselben Minute.

Der erste meint, dass jede zu Beginn der Arbeitsphase bekannte *Connection* mit der für ihn sichtbaren Aufzeichnung abgeglichen wurde. Der zweite meint, dass er nach seiner Rückkehr nach *Home* eine neue *Observation* hinzugefügt hat. Ihre Aussagen klingen gleich, weil beide weder *Source* noch Zeit nennen.

Ein dritter Pilot hört aus derselben *Structure* zu. Er ist Gast, kann andocken, die ihm überlassenen gemeinsamen Routennotizen lesen und sich auf die geplante *Work* vorbereiten. Als der Zeitpunkt kommt, steht ihm der benötigte *Service* der *Structure* jedoch nicht zur Verfügung. Die Erlaubnis zum Andocken und der *Access* zum *Service* wurden wie eine einzige Berechtigung behandelt. Das sind sie nicht.

Alle drei Piloten befinden sich im selben *Home*. Jeder handelt aus einer anderen Version davon.

Nichts an dieser Szene setzt Täuschung oder Nachlässigkeit voraus. Ein Pilot beobachtete, ein anderer meldete und ein Gast vertraute der Einladung. Der *Failure* liegt zwischen diesen Handlungen. Niemand bestimmte, welcher Zustand geteilt wurde, wer ihn pflegte oder woran eine andere Person erkennen würde, dass die Verantwortung weitergegeben worden war.

Ein Ort wird gemeinsam, wenn sein veränderlicher Zustand die zufällig anwesenden Menschen überdauern kann. Bis dahin treffen im selben System nur mehrere private Wirklichkeiten aufeinander.

Ein *Home*, viele

Das Zusammenleben macht es leicht, Gleichheit vorauszusetzen.

Dieselbe *Structure* erscheint in jeder *Overview*. Derselbe Systemname steht an jedem Ort. Piloten sprechen über *die Chain*, *den Treibstoff*, *den Hangar* und *den Plan*, als hätte der bestimmte Artikel *Ownership*, *Access* und *Bedeutung* bereits geklärt.

Der gemeinsame Ort ist real, aber er bildet die dünnste Ebene eines gemeinsam getragenen *Home*. Er sagt der Gruppe, wo Menschen und Vermögenswerte zusammentreffen. Ob sie dieselben Informationen sehen, dieselben *Capabilities* nutzen oder von derselben Person die nächste Handlung erwarten, kann er nicht zeigen.

Gemeinsame Information ist eine andere Ebene. Zwei Piloten können beide Routenaufzeichnungen besitzen, während eine davon eine spätere *Observation* enthält. Beide können einen Vermögenswert sehen, während nur einer weiss, dass er reserviert wurde. Eine Notiz kann für jedes Mitglied sichtbar sein und trotzdem verschieden verstanden werden, weil der Autor nie angegeben hat, ob es sich um eine *Observation*, eine Schlussfolgerung oder einen ungeklärten Bericht handelt.

Gemeinsame Verantwortung bildet eine weitere Ebene. Ein Pilot, der Informationen ergänzt, hat nicht zwingend ihrer Pflege zugestimmt. Wer einen Vermögenswert nutzen kann, trägt damit noch nicht dessen Zweck. *Access* zu einem *Service* sagt nichts darüber aus, wer bemerkt, dass dieser die von ihm erwartete *Work* nicht mehr trägt.

Diese Ebenen können sich trennen, ohne einen offensichtlichen Alarm auszulösen. Die *Structure* akzeptiert weiterhin die Piloten, die andocken dürfen. Der gemeinsame Ordner enthält weiterhin *Locations*. Im Hangar der *Corporation* liegen weiterhin Vermögenswerte. *Home* wirkt vollständig, weil seine Objekte sichtbar bleiben, während die Vereinbarung um sie herum bereits auseinandergeht.

Der Gast aus der Eingangsszene macht diese Trennung leichter erkennbar. Eigentümer ergänzen fehlenden Kontext oft aus Gewohnheit. Sie erinnern sich, welche Notizen alt sind, welche Vermögenswerte persönlich sind und wen sie wegen *Access* fragen müssen. Ein Gast besitzt

keinen solchen Hintergrund. Seine Sicht legt jede Anweisung offen, die nur in der Erinnerung besteht.

Gäste sind kein Ausnahmefall. Jedes Mitglied, das nach einer Abwesenheit zurückkehrt, an einer späteren Arbeitsphase teilnimmt oder eine unvertraute Aufgabe übernimmt, begegnet *Home* auf dieselbe Weise. Muss es dessen Bedeutung aus den gerade anwesenden Personen rekonstruieren, teilt die Gruppe Anwesenheit statt Zustand.

Der erste nützliche Unterschied ist deshalb einfach. Frage getrennt, wo die Gruppe zusammentrifft, was sie gegenwärtig glaubt und wer dafür verantwortlich ist, diesen Glauben nutzbar zu halten. Übereinstimmung in einer Antwort darf nicht für die anderen beansprucht werden.

Verschiedene Antworten müssen nicht in eine falsche Übereinstimmung gezwungen werden. Zwei Berichte können nebeneinander bestehen, wenn Zeit, *Source* oder Geltungsbereich voneinander abweichen. Ein Gast kann an der gegenwärtigen *Work* teilnehmen, ohne als Eigentümer behandelt zu werden. Eine Person kann als *Steward* eine Aufzeichnung pflegen, während eine andere die davon betroffenen Vermögenswerte besitzt. Der gemeinsame Zustand wird stärker, wenn diese Unterschiede bewahrt statt geglättet werden.

Die Gefahr beginnt, wenn ein breites Wort den Unterschied verbirgt. *Unser* kann sich auf Standort, *Access*, Eigentum, Verantwortung oder Zweck beziehen. Vor einer darauf gestützten Handlung sollte die Gruppe benennen können, welche Bedeutung gilt. Das verlangt nicht in jedem Gespräch formelle Sprache. Präzision wird wichtig, wenn eine Entscheidung aus der Wirklichkeit einer Person in die einer anderen übergeht.

***Baseline* definieren**

Ein gemeinsamer Zustand benötigt einen Ausgangspunkt, von dem aus Veränderung erkannt werden kann. Dieser Punkt ist eine *Baseline*.

Kapitel 5 untersucht, was rund um *Home* beobachtet werden kann und wie Abweichungen gelesen werden. Hier besitzt die *Baseline* einen engeren Zweck: Sie erfasst, was die Gruppe als ihre gegenwärtige Arbeitsbedingung vereinbart hat.

Diese Bedingung beginnt mit dem Zweck. Ein *Home*, das für eine kurze Phase des *Scanning* vorbereitet wurde, benötigt nicht dieselben *Capabilities* wie ein *Home*, das mehrere laufende Aufgaben, Vermögensbewegungen und Gäste trägt. Ohne benannten Zweck kann ein Pilot die Lage für normal halten, weil *Scanning* noch möglich ist. Ein anderer kann sie für unzureichend halten, weil sein geplanter *Service* nicht verfügbar ist. Beide können ihre eigene *Work* zutreffend beschreiben.

Die *Baseline* trennt danach *Observed* von dem, was abgeleitet, *Reported* oder *Unknown* gelassen wurde. Eine von einem benannten Piloten zu einer bekannten Zeit geprüfte *Connection* ist eine *Observation*. Die Annahme, dass eine Route für eine spätere Bewegung geeignet bleibt, ist eine Interpretation. Ohne direkte Bestätigung weitergegebene Information ist ein Bericht. Was nicht untersucht wurde, bleibt *Unknown*, auch wenn Gewohnheit eine wahrscheinliche Antwort liefert.

Diese Unterschiede verhindern, dass Gewissheit weiter reicht als die Belege. Sie erlauben einem anderen Piloten ausserdem, nur den veränderten Teil zu aktualisieren. Eine neue *Observation* muss die frühere Aufzeichnung nicht löschen. Sie muss kenntlich machen, was sie ersetzt und welche Schlussfolgerungen neu betrachtet werden müssen.

Zeit gehört in die *Baseline*, nicht neben sie. *Aktuell* bedeutet nichts, solange die Gruppe nicht weiss, seit wann, laut wem und für welche Entscheidung etwas aktuell ist. Verschiedene Teile von *Home* altern unterschiedlich schnell. Die *Baseline* macht sie nicht dauerhaft. Sie macht ihr Alter weit genug sichtbar, um zu entscheiden, ob sie die nächste Handlung noch tragen.

Auch *Access* gehört zum vereinbarten Zustand. Die Gruppe benötigt in keinem Statusbericht einen vollständigen Katalog aller Berechtigungen.

Sie muss aber wissen, ob die erwarteten Personen den Ort, die Informationen, Vermögenswerte und *Services* erreichen können, die ihre *Work* erfordert. Dass ein Gast andocken kann, beantwortet nur eine dieser Fragen.

Eine brauchbare *Baseline* ist kurz genug, um vor einer Handlung gelesen zu werden, und genau genug, um infrage gestellt werden zu können. Sie beschreibt, was *Home* tragen soll, welche wichtigen Bedingungen bestätigt wurden, was *Unknown* bleibt und wann die Vereinbarung erneut betrachtet werden muss. Sie beansprucht keine Sicherheit, sondern beschreibt den Ausgangspunkt, von dem aus die Gruppe Veränderung bemerken will.

Ohne diesen Ausgangspunkt wird jede Aktualisierung zu einer isolierten Tatsache. Mit ihm kann eine *Observation* zu einem bedeutsamen Unterschied werden.

Steward bestimmen

Eine Vereinbarung zerfällt, wenn alle sie aktualisieren können und niemand für ihren Zustand verantwortlich ist.

Die Antwort ist Pflegeverantwortung. Ein *Steward* pflegt einen begrenzten Teil des gemeinsamen Zustands, bis er geschlossen oder übergeben wird. Die Rolle macht diese Person weder zum Eigentümer aller beteiligten Objekte noch zum *Commander* der Gruppe oder zum einzigen Piloten, der beobachten darf. Sie benennt, wer bemerken muss, ob die gemeinsame Darstellung nutzbar bleibt.

Für die *Chain* kann Pflegeverantwortung bedeuten, neue *Observations* mit der gegenwärtigen Aufzeichnung abzugleichen, Ungewissheit zu markieren und sicherzustellen, dass der nächste Pilot weiss, was noch bestätigt werden muss. Bei einer gemeinsamen Verpflichtung um einen Vermögenswert kann sie bedeuten, den benannten Zweck zu bewahren und zu erkennen, wann die versprochene *Capability* nicht mehr verfüg-

bar ist. *Work* von Gästen kann die Bestätigung verlangen, dass der vom vereinbarten Zweck benötigte *Access* tatsächlich besteht.

Der *Steward* muss nicht jede Handlung ausführen. Andere Piloten können durch *Scanning* Routen aufbauen, Vermögenswerte bewegen, *Access* prüfen oder Veränderungen melden. Die Pflegeverantwortung umfasst die offene Schleife – die Lücke zwischen dem Erkennen eines Ereignisses und dem Nachführen des gemeinsamen Zustands auf das, worauf sich die Gruppe nun verlassen kann.

So wird ein häufiges Verschwinden von Verantwortung verhindert. Ein Pilot beobachtet eine Veränderung und nimmt an, der Verwalter der Aufzeichnung werde sie sehen. Dieser sieht eine Nachricht und nimmt an, der Beobachter habe die *Source* aktualisiert. Alle haben etwas beigetragen, doch niemand weiss, ob die *Work* abgeschlossen ist.

Wird ein *Steward* benannt, lässt sich die letzte Frage beantworten: Wer bestätigt, dass die Veränderung aufgenommen wurde?

Kleine Gruppen benötigen kein grosses Organigramm. Eine Person kann während einer ruhigen Phase mehrere Zustände pflegen. Der Unterschied bleibt wichtig, weil dieselbe Person die *Chain* übergeben kann, ohne Entscheidungen über Vermögenswerte zu übergeben. Sie kann *Home* verlassen und zugleich Eigentümerin persönlichen Besitzes bleiben. Werden die Aufgaben getrennt benannt, wird eine Übertragung nicht mit allen anderen verwechselt.

Auch Pflegeverantwortung benötigt eine Grenze. Sie beginnt, wenn eine Person den Zustand annimmt, und endet, wenn der Zweck abgeschlossen, die Verantwortung zurückgegeben oder das *Handover* von einer anderen Person bestätigt wird. Eine unbestätigte Nachricht verschiebt diese Grenze nicht. Ebenso macht Abwesenheit nicht stillschweigend den aktivsten verbliebenen Piloten zum neuen *Steward*.

Die Rolle sollte eng genug sein, um abgeschlossen werden zu können. *Kümmere dich um Home* ist kein brauchbarer Auftrag, weil er zu viele verschiedene Bedingungen verbirgt. *Pflege während dieser Arbeitsphase*

den vereinbarten Zustand der Chain kann beobachtet, übergeben und geschlossen werden. Auch der Auftrag *Bestätige die für die heutige Aufgabe benötigten Capabilities des Gastes* lässt sich so begrenzen.

Die Zuweisung muss ausserdem für die davon abhängigen Menschen sichtbar sein. Ist ein *Steward* nur seinem Auftraggeber bekannt, kann er weder Berichte erhalten noch infrage gestellt werden, wenn die Aufzeichnung einer *Observation* widerspricht. Die gemeinsame Darstellung sollte zeigen, wer den Zustand gegenwärtig pflegt, wie diese Person erreicht werden kann und was geschieht, wenn sie nicht weitermachen kann.

Diese Rückfallebene muss kein weiteres dauerhaftes Amt sein. Sie kann aus einem benannten Stellvertreter, einer Regel zum Pausieren der *Work* oder der Rückgabe der Verantwortung an die Gruppe bestehen. Entscheidend ist, dass Abwesenheit zu einer bekannten Verringerung statt zu einer unsichtbaren Lücke führt. Gemeinsame Verantwortung entsteht nicht dadurch, dass eine Person unverzichtbar gemacht wird.

Ein *Steward* garantiert keine korrekten Informationen. Der Wert der Rolle liegt darin, dass Ungewissheit einen Ort erhält. Widersprüche können als *Unknown* bewahrt, veraltete Berichte infrage gestellt und fehlende Bestätigung kann daran gehindert werden, unbemerkt zum Wissen der Gruppe zu werden.

Access ist bedingt

Access fühlt sich wie eine Tür an: offen oder geschlossen. EVE vergibt ihn in kleineren Teilen.

Shared Location Folders unterscheiden zwischen *View*, *Use*, *Manage* und *Admin*. Diese Stufen ermöglichen verschiedene Beziehungen zu derselben Sammlung von *Locations*. Eine Route sehen zu können umfasst weder die Pflege des Ordners noch die Änderung, wer ihn erreichen kann.

Der *Access* zu einer *Upwell Structure* trennt ein weiteres Fragenpaar. Ein *Structure Profile* legt fest, welche *Services* anderen Spielern angeboten werden. Zugewiesene *Access Lists* bestimmen, welche Spieler diese *Services* nutzen können. *Services* können eigene Zuweisungen besitzen. Die Erlaubnis zum Andocken verspricht deshalb nicht die Erlaubnis, jeden von der *Work* benötigten *Service* zu verwenden.

Corporation Roles folgen demselben breiteren Muster. Sie gewähren bestimmte Fähigkeiten statt eines allgemeinen Zustands der Verantwortung. Den Inhalt eines Hangars der *Corporation* sehen zu können unterscheidet sich davon, Gegenstände daraus entfernen zu dürfen. Keine der beiden Fähigkeiten erklärt, wer den Zweck der Gegenstände trägt.

Kapitel 7 untersucht dieses Modell für *Access* im Detail. Hier geht es noch nicht um seine Konfiguration. Entscheidend ist, dass jeder gemeinsame Plan die vorausgesetzte *Capability* benennt. *Hat Access* ist zu ungenau, um eine Entscheidung zu tragen.

Dem Gast aus der Eingangsszene wurde *Home* nicht verweigert. Er erhielt eine reale Form von *Access*, während ihm eine andere fehlte. Der Fehler bestand darin, die erste als Beleg für die zweite zu behandeln. Hätte der Plan den benötigten *Service* und die für seine Bestätigung verantwortliche Person benannt, hätte der Unterschied vor Beginn der *Work* entdeckt werden können.

Auch *Access* verändert sich. Ein Ordner kann für eine Gruppe verfügbar sein und für eine andere nicht. *Profile* und zugewiesene *Access Lists* können verändert werden. Die Fähigkeiten innerhalb einer *Corporation* können sich zwischen Mitgliedern unterscheiden. Der operative Zustand ist deshalb nicht «*Wir haben ihnen einmal Access gegeben.*» Er lautet: *Die erforderlichen Personen können die erforderliche Capability für die aktuelle Entscheidung nutzen.*

Diese Aussage überträgt noch immer kein *Ownership*. Eine *Location* kann einem Gast zur Verfügung stehen, ohne die umgebenden Informa-

tionen zu übertragen. Ein Mitglied kann einen Gegenstand entnehmen dürfen, ohne seinen Zweck neu ausrichten zu dürfen. Ein Verwalter kann *Access* ändern, ohne für die davon abhängige *Work* verantwortlich zu werden.

Die Trennung dieser Bedeutungen schützt beide Seiten. Eigentümer verwechseln Sichtbarkeit nicht mit Zustimmung. Gäste verwechseln eine Einladung nicht mit einem Versprechen auf Dauer. *Stewards* nehmen nicht an, dass eine technische Befugnis jemandem automatisch die Pflicht zur Pflege des Zustands zuweist.

Lies *Access* vor dem Beginn gemeinsamer *Work* als Gruppe von Bedingungen. Bestimme, ob die Person den Ort erreichen und die Informationen sehen kann. Stelle getrennt fest, ob sie den benötigten Vermögenswert oder *Service* verwenden kann. Zusätzlich müssen ein Weg zum Melden von Veränderungen und die Befugnis zu einer Reaktion bestehen. Jede Frage kann eine andere Antwort besitzen.

Das Ziel ist nicht möglichst breiter *Access*. Breiterer *Access* kann *Capability* verteilen und zugleich unklare Annahmen verbreiten. Die gewährte *Capability* soll zur *Work* passen. Halte danach fest, wer die Abhängigkeit pflegt und wo ihre Grenzen liegen.

***Handover* ohne Annahme**

Der erste Pilot beobachtet. Der zweite meldet. Die nächste Person muss bestätigen.

Das sind drei verschiedene Ereignisse.

Eine *Observation* verändert, was eine Person weiss. Ein Bericht macht dieses Wissen anderen zugänglich. Eine Bestätigung stellt fest, dass der vorgesehene Empfänger die Verantwortung erhalten, den Zustand verstanden hat und daraus handeln kann.

Eine gemeinsame Notiz kann den Meldeschritt abschliessen, ohne das *Handover* abzuschliessen. Ebenso kann eine Gesprächsnachricht, eine veränderte Bezeichnung oder ein *Bookmark* in einem gemeinsamen

Ordner nur die Meldung abschliessen. Die Information kann sichtbar sein, ohne dass jemand die damit verbundene offene *Work* angenommen hat.

Kehre zu den beiden Aussagen zurück, die *Chain* sei aktuell. Der Widerspruch wird nicht aufgelöst, indem automatisch die spätere Nachricht gewählt wird. Die Gruppe muss wissen, was jeder Pilot wann beobachtete und welchen Teil des früheren Zustands der spätere Bericht verändert. Alles nicht Abgeglichenes bleibt *Unknown*.

Der bisherige *Steward* kann danach eine begrenzte Darstellung übergeben: den Zweck des gegenwärtigen Zustands, seine letzten bestätigten *Observations*, bereits gemeldete Veränderungen, ungelöste Unterschiede und die nächste erforderliche Entscheidung. Der neue *Steward* bestätigt nicht nur den Eingang der Worte, sondern auch, dass er die *Work* übernehmen kann.

Diese Bestätigung kann eine fehlende *Capability* aufdecken. Der Empfänger hat möglicherweise keinen *Access* zu den gemeinsamen *Locations*, den betroffenen Vermögenswerten oder dem benötigten *Service*. Vielleicht steht sein Aufbruch bevor oder ihm fehlt der Kontext, um einen Bericht von einer *Observation* zu unterscheiden. Die Entdeckung dieser Lücke ist eine erfolgreiche Prüfung des *Handover*. Sie verhindert, dass der Titel *Steward* weitergegeben wird, während die Fähigkeit zur Pflege zurückbleibt.

Kann niemand übernehmen, sollte sich der Zustand sichtbar verschlechtern. Die *Work* kann pausieren, die *Baseline* kann enger werden und unbestätigte Aussagen können zu *Unknown* zurückkehren. Nicht geschehen darf, dass die Verantwortung verschwindet, während die Aufzeichnung weiterhin massgeblich wirkt.

Die Bestätigung muss genau genug sein, um die Schleife zu schliessen. *Gesehen* kann bestätigen, dass eine Nachricht einen Bildschirm erreicht hat. Es bestätigt nicht zwingend Verantwortung. Eine nützliche Antwort benennt, was angenommen wurde und was nicht. Der Zustand der

Chain kann übernommen werden, während die Frage zum Gastzugang bei der gegenwärtig verantwortlichen Person bleibt. Geplante *Work* kann warten, bis beides geklärt ist.

Die Bestätigung sollte ausserdem die erste vom Empfänger erwartete Handlung zeigen. Vielleicht muss er eine alte *Observation* überprüfen, den Gast über die Pause der *Work* informieren oder den gegenwärtigen Zustand lediglich bis zur Ankunft eines weiteren Piloten bewahren. Ein *Handover*, das eine grosse Darstellung ohne nächste Entscheidung überträgt, kann Informationen weitergeben und dabei die Priorität zurücklassen.

Der Empfänger kann nur einen Teil der Darstellung annehmen. Das ist besser als eine breite Bestätigung, der eine private Annahme folgt. Der Rest verbleibt bei seinem bisherigen *Steward*, geht an die Gruppe zurück oder wird zu einer ausdrücklichen Lücke. Teilweise Bestätigung hält die Grenze der Verantwortung ehrlich.

Diese Disziplin ist über verschiedene Arbeitsphasen hinweg besonders wertvoll. Die Person, die eine *Observation* gemacht hat, kann bei Eintritt ihrer Konsequenz abwesend sein. Ein bestätigtes *Handover* lässt *Home* weiterbestehen, ohne so zu tun, als teilten alle Erinnerung, Zeitplan oder *Access*.

Werkzeuge können die Darstellung anzeigen, speichern und verteilen. Sie können nicht entscheiden, dass ein Mensch eine Verpflichtung verstanden hat. Unabhängig vom gewählten Medium ist das *Handover* erst abgeschlossen, wenn der Empfänger sowohl Zustand als auch Geltungsbereich bestätigt.

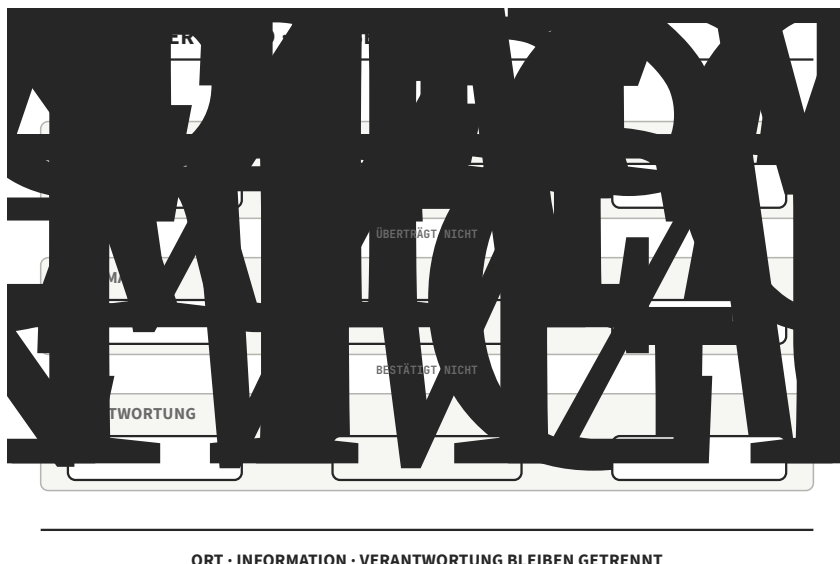


Abbildung AFH-B2-FIG-004 — Gemeinsamer Ort, Information und Verantwortung bleiben getrennt. Gemeinsamer Ort, geteilte Information und bestätigte Verantwortung sind verschiedene Beziehungen.

Feldprinzip

Ein gemeinsam getragenes *Home* wird nicht durch die Zahl der Piloten bestimmt, die es betreten können. Entscheidend ist, welche veränderlichen Bedingungen sie verstehen, pflegen und übergeben können, ohne Übereinstimmung zu unterstellen, wo sie fehlt.

Trenne Ort, Information und Verantwortung. Lege eine an Zweck, *Source*, Zeit und bekannte Ungewissheit gebundene *Baseline* fest. Gib jedem entscheidenden Zustand einen *Steward*, dessen *Work* einen Anfang, eine Grenze und ein bestätigtes Ende besitzt. Behandle *Access* als bedingte *Capability*, niemals als stillen Beleg für *Ownership*, Zustimmung oder Dauerhaftigkeit.

Eine *Observation* gehört zuerst dem Beobachter. Ein Bericht macht sie verfügbar. Nur eine Bestätigung überträgt die Pflicht, sie weiterzutragen.

Die drei Piloten können ihr *Home* nun abgleichen. Bis eine neue *Observation* den Unterschied auflöst, bewahren die beiden Routendarstellungen ihre unterschiedlichen Zeiten. Die dem Gast erteilte Andockerlaubnis bleibt bestehen, ohne mit *Access* zu einem *Service* verwechselt zu werden. Ein benannter *Steward* trägt jede offene Frage, bis sie beantwortet oder übergeben ist.

Der gemeinsame Boden hat *Home* nicht gewiss gemacht. Er hat Uneinigkeit sichtbar gemacht, bevor eine Handlung von ihr abhängt.

Diese Sichtbarkeit erzeugt die nächste Aufgabe. Eine Gruppe kann Veränderung nur erkennen, wenn sie weiss, wie ein gewöhnliches *Home* aussehen sollte. Part II beginnt damit, diese *Baseline* lesen zu lernen.

Part II - *Home* erhalten

Home wird selten in einem einzigen Augenblick unbrauchbar. Häufiger verschiebt sich sein gewöhnlicher Zustand. *Observations* altern, *Connections* verändern sich und *Access* wird enger. *Services* tragen die um sie herum aufgebaute *Work* nicht mehr. Part II macht aus dem gemeinsamen Boden von Part I eine Praxis der Pflege. Er folgt den Signalen, die Veränderungen sichtbar machen, und der *Chain*, die Bewegung trägt. Danach wendet er sich den Berechtigungen zu, die *Capability* formen, sowie der Infrastruktur, deren anhaltender Nutzen von Aufmerksamkeit abhängt.

Kapitel 5 - *Baseline* lesen

Zu Beginn der Arbeitsphase wirkt *Home* unverändert.

Dieselbe *Structure* steht auf demselben *Grid*. Die erwarteten Schiffe liegen weiterhin in ihren Hangars, und die gemeinsame Aufzeichnung zeigt noch immer eine Route, welche die frühere Schicht benutzt hat. Der erste Blick liefert keinen Grund zur Beunruhigung.

Dann fallen drei kleine Unterschiede auf.

Bei der Prüfung durch die neue Schicht zeigt der *Structure Browser* einen *Service* nicht mehr an, den die geplante *Work* benötigt. Eine erneute Prüfung im *Probe Scanner* zeigt eine zuvor erfasste *Cosmic Signature* nicht. Der betreffende Zweig der *Chain* wurde zuletzt von einem anderen Piloten während einer früheren Arbeitsphase bestätigt.

Diese *Observations* beweisen keine Gefahr. Vielleicht ist der *Service* für diesen *Character* nicht verfügbar, statt für alle zu fehlen. Die *Signature* kann sich verändert haben, ohne den Zweck dieses Abends zu berühren. Die übernommene Aufzeichnung der *Chain* kann weiterhin eine nutzbare Route beschreiben.

Zusammen brechen die Unterschiede jedoch eine wichtige Annahme: Der von der früheren Schicht übernommene Zustand kann nicht mehr ohne Prüfung verwendet werden.

Kapitel 4 legte eine *Baseline* als gemeinsame Vereinbarung fest. Part II beginnt damit, diese Vereinbarung beobachtbar zu machen. Vertrautheit ist nicht die Frage. Die Bedingungen, welche die nächste Entscheidung tragen, müssen weiterhin dem datierten Zustand entsprechen, auf den sich die Gruppe geeinigt hat.

Gewöhnliches als Evidenz

Gewöhnliche Bedingungen werden leicht übersehen, weil sie keine Reaktion verlangen. Die vertraute *Structure* ist vorhanden. Die erwartete

te Routenaufzeichnung besteht. Ein funktionierender *Service* erscheint dort, wo ihn die Gruppe erwartet. Die nächste Aufgabe besitzt einen Verantwortlichen.

Jede dieser stillen Tatsachen wird erst nützlich, wenn sie als Teil eines Zwecks festgehalten wird.

Eine *Baseline* ist diese Aufzeichnung. Sie sagt, welche Unterstützung von *Home* zu einem bestimmten Zeitpunkt, über bestimmte *Sources* und für eine benannte Art von *Work* beobachtet wurde. Die Aufzeichnung behauptet nicht, *Home* sei sicher, dauerhaft oder vollständig.

Diese Unterscheidung ist wichtig, weil Normalität kein allgemeingültiger Zustand ist. Für einen kurzen Flug zum *Scanning* benötigt ein Pilot vielleicht eine aktuelle Sicht auf das lokale System und die ersten relevanten *Connections*. Eine Gruppe, die unersetzliche *Capability* bewegt, kann eine tiefere Routendarstellung, bestätigten *Access* und eine bekannte Rückfallebene verlangen. Für den ersten Zweck kann *Home* gewöhnlich und für den zweiten noch nicht bereit sein.

Solche Unterschiede benötigen einen Bezugspunkt, bevor sie Bedeutung erhalten. Die frühere Schicht meldete nicht bloss, *alles sei in Ordnung*. Sie hinterliess eine begrenzte Darstellung. Ein benannter Pilot hatte die für die Rückkehr verwendete Route beobachtet. Der erforderliche *Service* war für die *Characters* sichtbar, welche diese *Work* ausführten. Entscheidende *Scanning Capability* war vorhanden, und keine offene Aufgabe hing von einem abwesenden Verantwortlichen ab. Diese Darstellung kann zutreffen und dennoch alt sein.

Alter macht die frühere Arbeit nicht nachlässig. Es macht die Darstellung historisch. Gegenwärtige *Observations* lassen sich mit ihr vergleichen, statt die Erinnerung darüber entscheiden zu lassen, ob sich ein Unterschied wichtig anfühlt.

Evidenz über das Gewöhnliche muss bescheiden bleiben. Das *Sensor Overlay* kann einen Überblick über die mit seiner gewählten Anzeige

sichtbaren Orte der Exploration liefern. Der *Probe Scanner* kann *Cosmic Anomalies* und *Cosmic Signatures* anzeigen; *Anomalies* benötigen keine *Probes*, um gefunden zu werden, *Cosmic Signatures* dagegen schon. Diese Oberflächen legen verschiedene beobachtbare Ebenen offen. Keine von ihnen zeigt jedes Ereignis im System, und keine macht aus dem Fehlen in einer Ansicht einen Beweis für Sicherheit.

Dieselbe Zurückhaltung gilt für Infrastruktur. Der *Structure Browser* zeigt *Structures*, auf die ein *Character* zugreifen kann, sowie die dort angebotenen *Services*. *Characters* mit der entsprechenden Befugnis innerhalb der *Corporation* können zusätzliche Informationen über ihre eigenen *Structures* sehen, darunter den verbleibenden *Fuel* und den *Vulnerability State*. Die Ansicht gehört dem beobachtenden *Character*. Ein Gast und ein Eigentümer können deshalb im selben Moment unterschiedliche, korrekte *Observations* besitzen.

Eine *Baseline* bewahrt diese Grenze. Sie hält fest: *für diesen Beobachter zu diesem Zeitpunkt sichtbar – nicht: für alle wahr, bis etwas anderes bekannt wird.*

Werden gewöhnliche Bedingungen so beschrieben, lässt sich Veränderung erkennen, ohne sie zu dramatisieren. Die *Baseline* sagt den nächsten *Failure* nicht voraus. Damit kann sich ein neuer Zustand nicht hinter einem alten Gefühl der Vertrautheit verbergen.

Ebenen von Home

Home besitzt nicht nur einen Zustand.

Sein System, seine *Chain*, seine Infrastruktur, seine Vermögenswerte und seine *Work* können jeweils eine andere Antwort tragen. Sie in der Aussage *Home ist in Ordnung* zusammenzufassen, entfernt genau die Unterschiede, die benötigt werden, sobald sich ein Teil verändert.

Beginne mit der Ebene des Systems. Zu einem festgehaltenen Zeitpunkt beobachtet ein benannter Pilot die für die vorgesehene *Work* relevanten *Cosmic Signatures* und *Cosmic Anomalies*. Die *Source* umfasst die Oberflä-

che und die aktive Ansicht, die für die *Observation* verwendet wurden. Das Ergebnis ist kein vollständiges Inventar des Raums. Es bildet eine datierte Darstellung dessen, was der Pilot sehen konnte und was ausserhalb des gewählten Umfangs blieb.

Die fehlende *Signature* gehört auf diese Ebene. Eine neue Ansicht im *Probe Scanner* weicht von der früheren Aufzeichnung ab. Diese Tatsache erklärt noch nicht, ob die frühere *Signature* abgelaufen, aus der Ansicht gefiltert oder anderweitig falsch erkannt worden ist. Die *Baseline* hat ihre Aufgabe erfüllt, sobald sie die Abweichung sichtbar macht und die Ursache zu *Unknown* zurückführt.

Die Ebene der *Chain* erfasst bekannte *Connections* und das Alter der *Observations*, auf denen sie beruhen. Frische darf nicht von der Systemebene übertragen werden. Ein neuer *Scan* von *Home* bestätigt nicht automatisch jeden entfernten Zweig erneut. Ebenso kann eine alte Route auf einer *Shared Map* verbleiben, ohne für eine neue Bewegung geeignet zu bleiben.

Kapitel 6 entwickelt diese Ebene vollständig. Für die *Baseline* ist das Minimum klar: Welcher Zweig ist für den gegenwärtigen Zweck wichtig, wer hat ihn zuletzt beobachtet, wann geschah das und was wurde seither nicht geprüft?

Die Infrastruktur bildet eine weitere Ebene. Trenne die *Structure* von ihren *Services*, dem *Access* und ihrem sichtbaren Zustand. Die Hülle kann stehen bleiben, während ein *Service* nicht verfügbar ist. Das Angebot eines *Service* kann bestehen, während dem eintreffenden *Character* der benötigte *Access* fehlt. Informationen über *Fuel* und den *Vulnerability State*, die ein befugter Eigentümer sieht, können in der Ansicht eines Gastes fehlen.

Für die Eröffnungsszene lässt sich deshalb eine genaue *Observation* festhalten: *Der benötigte Service war für diesen Character zu diesem Zeitpunkt nicht sichtbar.* Die Ursache bleibt offen. Diese Aussage ist nützlicher als

die *Structure* ist defekt, weil sie zeigt, was als Nächstes geprüft werden muss, ohne mehr zu behaupten, als die Ansicht belegt.

Vermögenswerte bilden eine vierte Ebene. Eine vollständige Preisliste ist nicht nötig. Die *Baseline* benötigt die kleine Gruppe von *Capabilities*, deren Fehlen die vereinbarte *Work* verändert. Eine Schiffshülle für das *Scanning* kann vorhanden, für den vorgesehenen Piloten aber nicht verfügbar sein. *Fuel* kann bestehen, jedoch über einen Zugriffspfad kontrolliert werden, den die für den *Service* verantwortliche Person nicht nutzen kann. Ein *Replacement* kann ausserhalb von *Home* erfasst sein, während seine Route unbestätigt bleibt.

Halte *Capability* statt beruhigender Menge fest. Rückfallebene für das *Scanning* vorhanden und beim letzten Check für den benannten Piloten zugänglich trägt eine Entscheidung. In *Home* liegen mehrere Schiffe tut das nicht.

Die letzte Ebene ist *Work*. Aufgaben, Bewegungen und Prüfungen können offen bleiben, nachdem die Person, die sie begonnen hat, gegangen ist. Ihr Zustand in der *Baseline* sollte den gegenwärtigen Zweck, den *Steward* und die nächste Entscheidung benennen. Eine nicht abgeschlossene Aufgabe ohne Verantwortlichen kann jede andere Ebene gesünder wirken lassen, als sie ist, weil noch niemand die *Capability* prüft, welche die Aufgabe benötigen wird.

Diese fünf Ebenen wirken zusammen, sollten aber erst aufeinander-treffen, nachdem jede ehrlich gelesen worden ist. Ein fehlender *Service* beeinflusst die *Work*, wenn sie ihn benötigt. Eine alternde *Chain* erhält eine andere Bedeutung, wenn ein *Replacement* oder ein Aufbruch von ihr abhängt. Eine neue *Signature* wird wichtig, sobald sie die Routen-frage verändert, welche die Gruppe beantworten muss.

Die *Baseline* ist kein einzelnes grünes Zeichen neben *Home*. Sie bildet eine kurze Sammlung datierter Aussagen, deren Beziehungen untersucht werden können, wenn sich der Plan verändert.

Jede *Observation* altert

Eine *Observation* ohne Zeit leiht sich bereits Gewissheit von der Zukunft. *Service verfügbar* kann wahr gewesen sein, als der frühere Pilot hinsah. *Route nutzbar* kann eine einzelne abgeschlossene Bewegung beschreiben. *Schiff für das Scanning bereit* kann den Zustand wiedergeben, den sein Eigentümer vorbereitet hatte. Werden *Source* und *Alter* entfernt, wird jede Aussage zu einem Versprechen, das niemand gegeben hat.

Für jede tragende *Observation* gelten vier Grenzen: Was wurde gesehen, wer sah es, wann wurde es gesehen und für welchen Zweck galt es als ausreichend?

Die *Source* ist mehr als der Name eines *Characters*. Sie umfasst den Weg, über den der Zustand sichtbar wurde. Ein direkter *Scan*, eine Ansicht im *Structure Browser* und die Nachricht eines anderen Piloten sind unterschiedliche *Sources*. Die Nachricht kann zutreffen, bleibt aber *Reported*, bis die gegenwärtige Entscheidung die von ihr benötigte Bestätigung erhält.

Der *Timestamp* zeigt der nächsten Person, wann die Zuverlässigkeit zu verfallen begann. Er liefert kein allgemeingültiges Ablaufdatum. Eine lokale *Observation* kann für eine Entscheidung nützlich bleiben, während ein entfernter Zweig vor einer gewichtigen Bewegung eine weitere Prüfung benötigt. Wie schnell die Information altert, ergibt sich aus dem vorübergehenden System und aus der Konsequenz eines Irrtums.

Der Zweck bestimmt diese Konsequenz. Eine für einen Scout beobachtete Route wurde nicht zwingend für einen Transport geprüft. Ein für seinen Eigentümer bestätigter *Service* wurde nicht zwingend für einen Gast bestätigt. Ein für die Lagerung gezählter Vermögenswert wurde nicht zwingend für den sofortigen Einsatz vorbereitet.

Shared Location Folders verstärken dieselbe Lektion. Sie verwenden verschiedene Zugriffsstufen, können von einem *Character* als online oder offline verbunden werden und *Locations* mit Ablaufdatum enthalten.

Fehlt eine *Location* in der aktiven Ansicht eines Piloten, kann das am Zustand des Ordners oder am *Access* liegen statt an einem veränderten *Grid*. Die *Baseline* sollte genügend Kontext zur *Source* erfassen, um diese Möglichkeit offenzuhalten.

Alter lässt sich zeigen, ohne die Aufzeichnung in eine Wand aus Uhren zu verwandeln. Die Gruppe benötigt *Timestamps* nur dort, wo Zeit die Deutung verändert. Die letzte Bestätigung der Route, welche die Bewegung dieses Abends trägt, ist wichtig. Das genaue Alter eines ungenutzten Zweigs vielleicht nicht.

Diese Auswahl verhindert zwei Fehler. Wird kein Alter festgehalten, wirken übernommene Aussagen gegenwärtig. Wird jede denkbare Zeit erfasst, gehen die Aussagen unter, welche eine Entscheidung tragen.

Die nützliche Frage lautet nicht: *Wie alt sind alle unsere Informationen?* Frage stattdessen: *Welche alte Observation würde die nächste Handlung verändern, wenn sie nicht mehr zuträfe?*

Veränderungen brauchen Bedeutung

Ein Unterschied ist Evidenz. Er ist noch keine Deutung.

Die verschwundene *Signature*, der nicht verfügbare *Service* und der alternde Zweig der *Chain* sind drei Abweichungen von der übernommenen *Baseline*. Keine von ihnen sollte unmittelbar zu einer *Threat*, einem *Failure* oder einem Grund erklärt werden, *Home* aufzugeben.

Frage zuerst, ob die Veränderung den genannten Zweck berührt. Eine fehlende *Signature* ohne Bezug zur *Work* dieses Abends kann lediglich die Darstellung des Systems verengen. Ein nicht verfügbarer *Service* ist sofort bedeutsam, wenn eine Aufgabe gleich von ihm abhängen wird. Eine alte Aufzeichnung der *Chain* wird dringend, sobald die Route das einzige *Replacement* tragen soll.

Bestimme danach die kleinste Aussage, die sich verändert hat. Der *Character* sah den *Service* nicht. Das gegenwärtige Ergebnis im *Probe Scanner* wich vom früheren Ergebnis ab. Der betreffende Zweig besitzt

keine Bestätigung aus der gegenwärtigen Arbeitsphase. Diese Aussagen bewahren die *Observation* vor der Erklärung.

Vergleiche als Nächstes die *Sources*. Stammt die frühere *Observation* des *Service* von einem Eigentümer und die neue von einem Gast, kann *Access* den Unterschied erklären. Aufzeichnungen der *Signatures*, die unter verschiedenen sichtbaren Auswahlen entstanden, sind noch nicht vergleichbar. Ein der früheren Schicht nur als *Reported* übermittelter Zweig der *Chain* kann älter sein als die Nachricht, die ihn übermittelte. Was nach diesem Vergleich verbleibt, kann ohne Drama eingeordnet werden.

Manche Abweichung ist zu erwarten und unterbricht die *Work* nicht. Manche Veränderung beeinträchtigt eine *Capability* und verlangt eine neue Entscheidung. Ein Widerspruch zwischen *Sources* verlangt Bestätigung. Fehlt Kontext, kehrt der Zustand zu *Unknown* zurück.

Unknown ist kein Vorwurf. Es ist korrekt, sobald die *Baseline* die beabsichtigte Schlussfolgerung nicht mehr tragen kann. Die betroffene *Work* kann pausieren, während nicht betroffene *Work* fortgesetzt wird.

Die Unterschiede aus der Eröffnungsszene führen nun zu verhältnismässigen Reaktionen. Der benötigte *Service* wird über die Ansicht des *Characters* geprüft, der ihn verwenden soll, und wenn nötig über einen befugten Eigentümer. Die fehlende *Signature* wird unter einer bekannten Ansicht erneut gescannt, bevor eine Schlussfolgerung über die Route verändert wird. Der alternde Zweig der *Chain* wird nicht für die geplante Bewegung freigegeben, bis die betreffenden *Connections* für diesen Zweck beobachtet worden sind.

Es wurde kein Notfall ausgerufen. Die *Baseline* hat Zeit für eine Untersuchung geschaffen, bevor sich mehrere Annahmen zu einer verbinden.

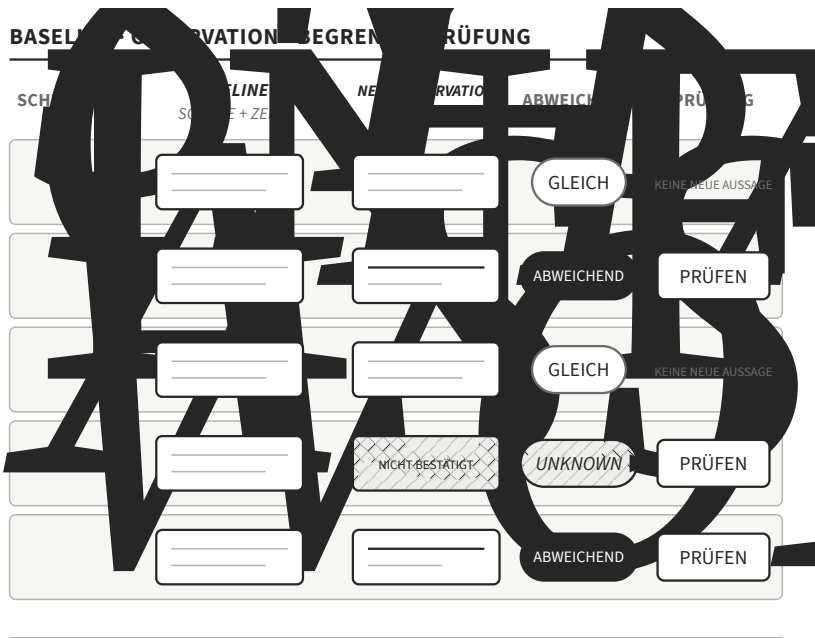


Abbildung AFH-B2-FIG-005 — Eine *Baseline* macht Abweichung sichtbar. Eine *Baseline* macht Abweichung sichtbar, bevor von einem Hinweis verlangt wird, seine Ursache zu benennen.

Wesentliches beobachten

Eine nützliche Überwachung ist kleiner als *Home*.

Der Versuch, jedes Objekt, jeden Vermögenswert, jede Berechtigung und jede Tätigkeit zu überwachen, würde aus gewöhnlicher Anwesenheit eine dauernde Berichterstattung machen. Das Ergebnis wäre detailliert und unlesbar. Wichtige Veränderungen könnten sich in genau jenem Aufwand verbergen, der sie sichtbar machen soll.

Wähle die wenigen Signale, welche den gegenwärtigen Zweck tragen.

Für das System kann dies die Darstellung der *Signatures* sein, die benötigt wird, um die betreffende Route wiederaufzubauen oder die geplante

Work zu beginnen. Ihre Aufzeichnung nennt den beobachtenden Piloten, die Zeit und die Ansicht. Nicht jede unbeteiligte *Anomaly* muss wie ein dauerhafter Alarm behandelt werden.

Beobachte für die *Chain* die letzte bestätigte *Observation* des Zweigs, der Bewegung, *Replacement* oder Rückkehr trägt. Das Signal ist nicht die Zahl der *Connections* auf der Karte. Es fragt, ob die benötigte Beziehung in der Tiefe, welche die nächste Entscheidung verlangt, über aktuelle Evidenz verfügt.

Beobachte bei der Infrastruktur den bestimmten *Service* und den Zugriffspfad, von denen die *Work* abhängt. Ein allgemeiner Zustand der *Structure* kann eine Prüfung durch den *Character*, der handeln soll, nicht ersetzen. Eigentümer können *Observations* zu *Fuel* und zum *Vulnerability State* ergänzen, wenn diese Bedingungen den Zweck tragen; Gäste sollten nicht vorgeben, Informationen zu sehen, die nur Eigentümern offenstehen.

Beobachte bei den Vermögenswerten eine oder zwei *Capabilities*, deren Fehlen den Plan verändern würde. Das Signal kann eine zugängliche *Scanning Capability* oder die für den Rückzug benötigte Rückfallebene sein. Wert und Menge bleiben gegenüber der Nutzung zweitrangig.

Halte bei der *Work* den *Steward* und die nächste Entscheidung sichtbar. Eine offene Aufgabe ohne bestätigten Verantwortlichen ist selbst eine Abweichung, auch wenn jede physische Eingabe weiterhin vorhanden ist.

Jedes Signal benötigt einen Auslöser. Dieser ist nicht immer ein fester Zeitplan. Er kann mit einer Arbeitsphase, einer Bewegung wichtiger *Capability*, einem *Handover*, einer Änderung des *Access* oder in dem Moment beginnen, in dem eine neue Aufgabe von der Bedingung abhängt. Prüfe erneut, sobald sich die Konsequenz veralteter Information verändert.

Der Beginn einer Arbeitsphase bietet ein brauchbares Beispiel. Wer übernimmt, wiederholt nicht jede *Observation*, die jemals mit *Home*

verbunden war. Die Person liest den von der früheren Schicht hinterlassenen Zweck und bestimmt die Bedingungen, die dieser Zweck weitergibt. Heute Abend sind das der für die Bewegung vorgesehene Zweig, der nach dieser Bewegung erforderliche *Service* und die *Capability*, die bei einem nötigen Wiederaufbau der Route gebraucht wird.

Der erste Durchgang bestimmt, was sofort fortgesetzt werden kann. Unbeteiligte Vermögenswerte müssen nicht erneut gezählt werden. Ein entfernter Zweig ausserhalb der vorgesehenen Route kann ausserhalb der Überwachung bleiben. *Work*, die weder vom fehlenden *Service* noch von der alternden *Connection* abhängt, kann unter ihrer eigenen *Baseline* fortgesetzt werden.

Der zweite Durchgang behandelt die Abweichung. Der Pilot löscht die frühere *Observation* des *Service* nicht, nur weil die aktuelle Ansicht davon abweicht. Beide *Observations* bleiben innerhalb ihrer *Sources* wahr: Der frühere *Character* sah den *Service* damals; der eintreffende *Character* sieht ihn jetzt nicht. Die nächste Bestätigung wird über den Zugriffspfad zugewiesen, den die geplante *Work* tatsächlich verwenden wird.

Der Zweig der *Chain* erhält dieselbe Behandlung. Sein Eintrag auf der *Shared Map* bleibt als Geschichte erhalten, doch seine alte Bestätigung wird nicht als aktuelles Routenwissen dargestellt. Ein der Bewegung zugewiesener Pilot beobachtet die betreffende *Connection* erneut. Bis dahin wartet nur die betroffene Bewegung. Die Karte hat nicht versagt; ihre Ungewissheit wurde sichtbar, bevor sie mit einem Versprechen verwechselt werden konnte.

Beim *Handover* wird die Überwachung zu einer kurzen Darstellung veränderter Bedingungen und offener Entscheidungen. Die nächste Schicht benötigt keinen Leistungsbericht. Sie benötigt die noch offene Frage zur *Signature* und die Grenzen der Bestätigung des *Service*. Zudem muss sie wissen, dass der Routenzweig beobachtet wurde, nachdem die Bewegung verschoben worden war. Jede Aussage behält ihre *Source* und Zeit.

Dieser Rhythmus ist wichtiger als eine dauerhafte Checkliste. Lies den übernommenen Zweck. Bestätige die Bedingungen, welche die nächste Konsequenz tragen. Bewahre Abweichungen, bis ihre *Sources* verglichen werden können. Übergib die verbleibenden *Unknowns* mit Verantwortlichen. Die Einzelheiten verändern sich mit der *Work*, während die Disziplin erkennbar bleibt.

Die Überwachung benötigt ausserdem eine Reaktionsgrenze. Weicht ein Signal von der *Baseline* ab, halte die *Observation* fest, begrenze den betroffenen Zweck und weise die nächste Bestätigung zu. Mache nicht aus jeder Abweichung einen allgemeinen Stopp. Lass die *Work* nicht bloss deshalb weiterlaufen, weil kein einzelner Unterschied dramatisch wirkt.

Diese kleine Auswahl kann sich zusammen mit *Home* verändern. Ein während einer Phase des *Scanning* unbedeutender *Service* kann die Überwachung verlassen. Eine Transportroute kann hinzukommen, sobald *Replacement* beginnt. Die *Baseline* bleibt nützlich, weil sie der Abhängigkeit statt der Gewohnheit folgt.

Nun kann der Pilot aus der Eröffnungsszene *Home* beschreiben, ohne es sicher oder unsicher zu nennen. Die Darstellung des Systems enthält eine ungeklärte Abweichung bei einer *Signature*. Die für die Bewegung benötigte Route ist alt und wartet auf eine neue *Observation*. Ein erforderlicher *Service* ist für den vorgesehenen *Character* nicht bestätigt. Vermögenswerte und unbeteiligte *Work* bleiben innerhalb ihrer genannten Bedingungen.

Das genügt, um zu entscheiden, was pausiert und was weitergeht.

Feldprinzip

Eine *Baseline* ist ein datierter Vergleich, kein Versprechen, dass gewöhnliche Bedingungen bestehen bleiben.

Trenne System, *Chain*, Infrastruktur, Vermögenswerte und *Work*. Gib jeder tragenden *Observation* eine *Source*, einen *Timestamp* und einen

Zweck. Behandle eine Abweichung als Evidenz, bevor du ihre Ursache benennst. Beobachte nur die Bedingungen, deren Veränderung die nächste Entscheidung beeinflussen würde.

Home wird lesbar, wenn eine neue *Observation* drei Fragen beantworten kann: Was wurde erwartet, was ist anders und welche *Work* verlangt nun eine Prüfung?

Die Antwort muss nicht jede Ursache klären. Sie muss die Grenze der Entscheidung offenlegen. Ein *Character*, der einen benötigten *Service* nicht sieht, hat eine Frage zum *Access* festgestellt, keinen *Failure* der gesamten *Structure*. Ein verändertes Ergebnis beim *Scanning* hat eine Abweichung im System festgestellt, keine feindliche Absicht. Eine alternde Routenaufzeichnung hat den Bedarf nach Bestätigung festgestellt, nicht den gegenwärtigen Zustand jeder *Connection*, die sie einst beschrieb. Genauigkeit verhindert, dass sich ein kleines *Unknown* zu einem allgemeinen Urteil über *Home* ausbreitet.

Diese Genauigkeit schützt auch den Aufwand. Piloten prüfen die Bedingung erneut, von der die nächste Entscheidung abhängt. Unbeteiligte *Observations* müssen nicht wiederholt werden, nur damit die *Baseline* vollständig wirkt.

Darin liegt ihr praktischer Wert. Die *Baseline* gibt gewöhnlicher Anwesenheit ein Gedächtnis, das infrage gestellt werden kann. Neue Evidenz kann es verändern, ohne umzuschreiben, was frühere Beobachter tatsächlich sahen. Und *Work* kann begrenzt werden, ohne aus jeder Ungewissheit einen Alarm zu machen.

Die *Chain* bleibt die älteste ungeklärte Bedingung. Ihre sichtbaren Zweige tragen noch immer frühere *Observations*, Berichte und Annahmen aus einer anderen Arbeitsphase. Eine *Baseline* kann dieses Alter offenlegen. Die Route ehrlich halten kann sie allein nicht.

Kapitel 6 folgt der *Chain* durch die Zeit.

Kapitel 6 - *Chain* pflegen

Die *Shared Map* wirkt vollständig, als die frühe Schicht *Home* verlässt.

Ein Zweig führt zu einer Route, die für Nachschub verwendet wurde. Ein anderer endet in einem System, das auf dem Rückweg gescannt worden ist. Bezeichnungen der *Connections*, *Locations* und kurze Notizen zeigen, wohin die Piloten reisten und was sie beobachteten. Die Karte ist nicht elegant, doch sie trug wirkliche Bewegung und brachte alle zurück.

Stunden später bereitet ein Pilot eine wichtige *Capability* für die Bewegung entlang des Nachschubzweigs vor. Auf dem Bildschirm stehen dieselben Farben. Dieselben Linien verbinden dieselben Systeme. Die Zeichnung verkündet nicht, dass Zeit vergangen ist.

Eine Notiz enthält den Unterschied. Die erste *Connection* wurde zu Beginn der früheren Schicht beobachtet, und seit der letzten Nutzung der Route hat niemand die entfernte Seite bestätigt. Weiter draussen im Zweig stammt ein Eintrag aus einem mündlichen Bericht statt aus direkter *Observation*. Die geplante Bewegung erbt nicht die Gewissheit, welche die Rückkehr getragen hat.

Die Karte ist nicht auf einmal falsch geworden. Sie ist zu einer Mischung aus Geschichte, aktueller Evidenz und unbeantworteten Fragen geworden.

Kapitel 5 trennte die *Baseline* in datierte Ebenen. Die *Chain* ist jene Ebene, die am ehesten wie ein beständiges Objekt aussieht, während sie sich ausserhalb der Aufzeichnung verändert. Sie zu pflegen bedeutet nicht, jeden Zweig eingezeichnet zu lassen. Es bedeutet, die Bedeutung jedes nützlichen Zweigs so ehrlich zu halten, dass sie die von ihr verlangte Entscheidung tragen kann.

Karten veralten

Eine *Shared Map* ist eine Publikation über die *Chain*, nicht die *Chain* selbst.

Dieser Unterschied kann während gewöhnlicher Reisen verschwinden. Ein Pilot beobachtet eine *Connection*, fügt beide Seiten der Aufzeichnung hinzu und durchquert sie. Wenig später nutzt ein weiterer Pilot denselben Zweig. Weil Publikation und Umgebung weiterhin übereinstimmen, beginnt sich die Route wie ein in *Home* eingebauter Korridor anzufühlen.

Ihre scheinbare Beständigkeit stammt aus erfolgreicher Nutzung, nicht aus Dauerhaftigkeit.

Jede Linie auf der Karte beruht auf einer *Observation*. Ein Pilot sah eine Seite einer *Connection* zu einer bestimmten Zeit und in einem bestimmten Zustand. Durch eine Reise oder ein zuverlässiges *Handover* verband er sie mit der anderen Seite. Die Karte verdichtet diese Ereignisse zu einer Form, die leichter zu verwenden ist. Diese Verdichtung ist nötig, kann aber das Alter und die Grenzen der darunterliegenden Evidenz verdecken.

Eine gute Aufzeichnung der *Chain* trägt deshalb mehr als ihre Topologie. Bei einem wichtigen Zweig bewahrt sie die *Connection*, die beobachtete Richtung, die *Source*, den *Timestamp* und jeden sichtbaren Zustand, der ihre Deutung verändert. Sie sagt auch, weshalb die *Observation* als ausreichend galt. Eine für einen Scout bestätigte Linie wird nicht stillschweigend zu einer Transportroute erhoben.

Der Zweig aus der Eröffnungsszene enthält alle benötigten Systeme. Seine Schwäche liegt in den Aussagen, die sie verbinden. Die erste *Connection* besitzt eine alte direkte *Observation*. Der nächste Abschnitt beruht auf *Reported* Information. Während der gegenwärtigen Schicht wurde die entfernte Seite nicht gesehen. Diese Verbindungen in einer Farbe zu zeichnen, liesse drei verschiedene Zustände der Zuverlässigkeit gleich aussehen.

Frische gehört zu jeder einzelnen Aussage, nicht zur Karte als Ganzem. Die Seite von *Home* kann ein Pilot aktualisieren, ohne zu bestätigen, was dahinter liegt. Eine erfolgreiche Reise kann die geflogene Richtung bestätigen und den Zustand der Rückkehr unbeobachtet lassen. Die neue Darstellung der *Signatures* in einem System aktualisiert keinen zwei Sprünge entfernten Zweig.

So wird ein vertrauter Pflegefehler vermieden: Tätigkeit auf der Karte als Beleg dafür zu behandeln, dass die Karte aktuell sei. Eine neu gefundene *Connection* beweist nichts über eine ältere Route. Das Entfernen einer toten Notiz bestätigt die benachbarten Linien nicht. Eine geschäftige Publikation kann dennoch einen veralteten Entscheidungsweg tragen.

Lies die Route von der vorgesehenen Bewegung rückwärts. Welche *Connection* würde den Zweck stoppen, falls die Aufzeichnung falsch wäre? Welche *Observation* trägt diese *Connection*? Wann und von welcher Seite wurde sie gemacht? Was geschah seither, das die Aufzeichnung nicht zeigt?

Die Antworten können zeigen, dass nur ein Teil Aufmerksamkeit benötigt. Die gesamte *Chain* vor einer kurzen lokalen Aufgabe neu aufzubauen, ist unnötig, wenn der betreffende Zweig über ausreichende Evidenz verfügt. Umgekehrt gleicht ein schön ausgearbeiteter entfernter Zweig den unbestätigten ersten Schritt aus *Home* nicht aus.

Pflege beginnt deshalb mit Ehrlichkeit über die Form der Karte. Linien sind Aussagen. Bezeichnungen sind Deutungen. Freier Raum kann unerforscht, unbedeutend, abgelaufen oder bloss ausgelassen bedeuten. Eine Route, die noch auf der Seite erscheint, muss ihre Nutzung erneut verdienen, sobald Alter und Konsequenz die alte Evidenz unzureichend machen.

Zeit verändert Routen

Zeit wirkt nicht gleich auf jede Aussage über eine Route.

Wormhole Connections sind vorübergehend. Ein *Timestamp* allein kann jedoch nicht den genauen Augenblick ankündigen, in dem eine bestimmte *Connection* unbrauchbar wird. Der beobachtbare Zustand und die Geschichte der *Connection* sind wichtig. Ebenso wichtig ist der Zweck, für den die Aufzeichnung gelesen wird.

Die Oberfläche liefert begrenzte Information. Eine *Connection* kann dem beobachtenden Piloten eine beschreibende Information zur Lebensdauer zeigen. Die gegenwärtige Terminologie unterscheidet zudem *Reliable Lifetime* von *Expired*: Sobald der durch *Reliable Lifetime* dargestellte garantierte Zeitraum beendet ist, befindet sich die *Connection* im Zustand *Expired*. Dieser Name bedeutet nicht, dass das *Wormhole* bereits verschwunden ist. Er bedeutet, dass die frühere Garantie die Entscheidung nicht mehr trägt und der Kollaps innerhalb der verbleibenden Schwankung eintreten kann.

Genau diese Unterscheidung kann eine *Shared Map* eibebnen. Ein Pilot erinnert sich vielleicht nur daran, dass eine Nachricht vor der Zeit warnte. Für einen anderen bezeichnet *Expired* ein abgeschlossenes Ereignis. Ein dritter sieht die unveränderte Linie und nimmt an, der Zustand habe sich nicht verändert. Die Publikation benötigt die beobachtete Formulierung, die Zeit und die *Source* statt einer selbstsicheren Übersetzung in eine genaue Zukunft.

Ein sichtbarer massenbezogener Zustand schafft eine ähnliche Grenze. Der bei der *Observation* dargestellte Zustand kann von einem Piloten gemeldet werden. Dieser Bericht trägt eine Entscheidung nur innerhalb der Grenzen dessen, was gesehen wurde und welche Bewegungen die *Connection* seither durchquert haben können. Die Karte kann nicht jede Bewegung ableiten, die ausserhalb des Wissens der Gruppe stattgefunden hat.

Zeit verändert auch das Umfeld einer Route. In einem entfernten System können *Signatures* hinzukommen oder verschwinden. Eine nützliche Fortsetzung kann nicht mehr bestehen. Die Bestätigung der

ersten *Connection* bestätigt den dahinterliegenden Zweig nicht. Selbst wenn der physische Weg bestehen bleibt, kann sich seine operative Bedeutung verändern, sobald *Access*, verfügbare Schiffe oder die geplante Ladung wechseln.

Löse diese Ungewissheiten nicht mit einer einzigen allgemeingültigen Regel zur Aktualität. Ein festes Alter kann den Anschein von Disziplin erwecken und dabei die Konsequenz übergehen. Es kann eine unnötige erneute Prüfung eines unbedeutenden Zweigs erzwingen und dennoch eine alte *Observation* für eine Bewegung freigeben, die keinen Zweifel trägt.

Verwende Alter stattdessen als Frage. Sind genug Zeit und unbeobachtete Tätigkeit vergangen, um zu verändern, was diese Aussage tragen kann? Wenn die Antwort wichtig ist, beschaffe eine neue *Observation* in der von der Bewegung verlangten Tiefe. Ist keine Bestätigung verfügbar, lass den Zweig als Geschichte sichtbar und setze die Entscheidung auf *Unknown* zurück.

In der Eröffnungsszene muss der Pilot keinen Kollaps vorhersagen. Die entscheidende *Observation* lautet, dass der Bericht der frühen Schicht die geplante Bewegung nicht mehr rechtfertigt. Die Zeit hat die Zuverlässigkeit der Route verändert, bevor sie eine Veränderung der Route selbst bewiesen hat.

Zweck bestimmt Detail

Nicht jede Reise benötigt dieselbe *Chain*.

Ein Scout, der *Home* verlässt, um das nächste System zu lesen, muss die verwendete *Connection* erkennen und die Rückkehr wiederfinden. Die Publikation benötigt ausserdem genug Kontext, damit andere sie nicht mit einem anderen Zweig verwechseln. Weitere Einzelheiten können folgen, während die Erkundung fortschreitet.

Eine Arbeitsgruppe auf dem Weg zu einem bestimmten Ort benötigt eine breitere Darstellung. Die relevante Fortsetzung, der Rückweg und

Veränderungen, welche die *Work* unterbrechen könnten, müssen sichtbar bleiben, während die Piloten beschäftigt sind. *Locations* müssen den *Characters* zur Verfügung stehen, die handeln sollen, und nicht nur der Person, die sie erstellt hat.

Eine Bewegung wichtiger *Capability* verlangt mehr von derselben Karte. Die Route muss bis zu jener Tiefe bestätigt werden, von der die Bewegung abhängt. *Access* am Ziel ist wichtig. Die Publikation muss zeigen, wo die Bewegung pausieren kann, wer die *Chain* aktualisieren kann und welche Entscheidung den Versuch beendet, wenn eine entscheidende Aussage zu *Unknown* wird.

Dies sind Unterschiede der Konsequenz, nicht des Status. Der Transportpilot hat keinen Anspruch auf eine eindrucksvollere Publikation als der Scout. Seine Bewegung legt bloss mehr Abhängigkeiten offen, falls die Aufzeichnung falsch ist.

Der Zweck sollte deshalb mit der Bestätigung verbunden werden. Eine Notiz wie *Zweig geprüft* sagt zu wenig. Sie lässt offen, ob der Pilot beide Seiten erkannt, nur das nächste System untersucht oder die Route bis zum vorgesehenen Ende verfolgt hat. Für die Aufgabe, aus der sie entstand, kann die Notiz genügen und für eine andere Wiederverwendung dennoch unsicher sein.

Eine nützliche Bestätigung beantwortet vier Fragen. Welcher Teil des Zweigs wurde beobachtet? Aus welcher Richtung? Zu welcher Zeit? Für welche nächste Handlung wurde das Ergebnis angenommen? Die Antwort kann kurz bleiben, weil die *Shared Map* die Topologie bereits trägt. Ihre Aufgabe ist es, die von der Linie verdeckte Grenze offenzulegen.

Details jenseits dieser Grenze sind nicht automatisch wertvoll. Jedes auf einer Route angetroffene Objekt aufzuzeichnen, kann die Publikation langsamer lesbar und bei einem *Handover* schwerer handhabbar machen. Nur Objekte, welche *Connections* erkennen lassen, *Access* unterstützen, Bewegung verändern oder die Rückkehr bewahren, gehören

in die unmittelbare Aufzeichnung. Andere *Observations* können in Arbeitsnotizen bleiben, bis ein Zweck sie benötigt.

Dasselbe Prinzip bestimmt die Tiefe. Ein Zweig kann weiter reichen als die gegenwärtige Entscheidung. Wer ihn pflegt, darf nicht nahelegen, das ungesehene Ende sei frisch, nur weil der erste Teil bestätigt wurde. Markiere den Punkt, an dem die direkte *Observation* endet. Was dahinter liegt, kann *Reported*, historisch oder *Unknown* bleiben.

Betrachte den Unterschied zwischen einer Rückkehr und einer Lieferung. Ein allein zurückkehrender Pilot benötigt vielleicht die nächste bestätigte *Connection* und genug Kontext, um *Home* zu erkennen. Trägt diese *Connection* den Versuch nicht mehr, kann der Pilot anhalten, scannen und eine neue Entscheidung bilden, bevor weitere *Capability* eingesetzt wird.

Eine Lieferung sammelt bereits hinter dem ersten Sprung mehrere Bedingungen. Das Transportschiff, der *Access* am Ziel, die Personen, die auf das Material warten, und die nachfolgende *Work* können alle von der Route abhängen. Die Bestätigung muss jenen Punkt erreichen, an dem diese Abhängigkeiten angenommen oder gelöst werden können. Eine nur bis zum ersten entfernten System geprüfte Route lässt den Rest der Lieferentscheidung offen.

Die Karte kann dies ausdrücken, ohne eine zweite Karte hinzuzufügen. Gib der Bewegung ein begrenztes Ziel und markiere den letzten direkt bestätigten Abschnitt. Halte die Rückfallebene fest, die vor der Verpflichtung verfügbar bleibt. Kann die benötigte Tiefe nicht beobachtet werden, verkleinere oder verschiebe die Bewegung, statt die Teilroute als vollständig zu beschreiben.

Das begrenzt zugleich unnötige Details. Der Pilot, der eine Lieferroute bestätigt, muss nicht jeden unterwegs gefundenen Zweig lösen. Solche Zweige können ausserhalb des Zwecks bleiben, sofern sie nicht die Rückkehr verändern, die vereinbarte Rückfallebene anbieten oder auf

der verwendeten Route unmittelbare Mehrdeutigkeit schaffen. Die Tiefe folgt der Abhängigkeit, die Breite der Bedeutung.

Nach Abschluss der Lieferung wird ihre tiefere Bestätigung nicht zum dauerhaften Standard für jede spätere *Work*. Sie wird zur datierten Aufzeichnung dessen, was diese Bewegung trug. Der nächste Zweck darf die Evidenz erst wiederverwenden, nachdem er geprüft hat, ob Zeit, Zustand und Konsequenz dies weiterhin erlauben.

Das ist besonders wichtig, wenn mehrere Piloten beitragen. Einer scannt das erste System. Ein weiterer folgt der Fortsetzung. Ein dritter erhält das Ergebnis, ohne die Route zu bereisen. Gemeinsam können sie eine belastbare Route erzeugen – aber nur, wenn die Publikation bewahrt, welche Teile jede *Source* tatsächlich beobachtet hat.

Für die Bewegung aus der Eröffnungsszene wählt die Gruppe eine kleinere Aussage als *die Nachschubroute ist gut*. Der erste Pilot bestätigt die *Connection* von *Home* und ihre entfernte Seite. Ein zweiter folgt dem Zweig bis zu dem von der Bewegung benötigten Punkt. Der gemeldete Abschnitt bleibt *Reported*, bis einer von ihnen ihn direkt sieht. Erst dann kann die Route für diesen Zweck angenommen werden.

Der Zweck hat die *Chain* nicht dauerhaft gemacht. Er hat die benötigte Ungewissheit sichtbar gemacht, bevor Piloten und Vermögenswerte an sie gebunden werden.

Handover bewahrt Zweifel

Ein *Handover* wird oft danach beurteilt, wie schnell sich der nächste Pilot bewegen kann.

Geschwindigkeit ist wichtig, doch ein glatter Satz kann jene Vorsicht entfernen, welche die frühere *Observation* nützlich machte. Die Aussage «Die Route war in Ordnung, als ich hindurchkam» enthält eine abgeschlossene Reise, eine nicht genannte Zeit, eine Richtung und eine undefinierte Vorstellung von *in Ordnung*. Wird sie als gegenwärtige Tatsache in die *Shared Map* übernommen, wird sie gewisser als ihre *Source*.

Bewahre die Ungewissheit des Berichts.

Hat ein Pilot eine *Connection* direkt beobachtet, halte sie mit der betreffenden Zeit und Richtung als *Observed* fest. Wiederholt ein anderer Pilot diese Information, ohne sie zu prüfen, besitzt der Empfänger *Reported Information*. Wiederholung verwandelt einen Bericht nicht in eine Bestätigung. Eine angesehene *Source* kann das Vertrauen in das Gesagte erhöhen, aber keine *Observation* hinzufügen, die nicht stattgefunden hat.

Diese Unterscheidung schützt sowohl den ursprünglichen Piloten als auch die nächste Entscheidung. Der Sprecher kann genau beschrieben haben, was er sah. Der Fehler entsteht, wenn die Publikation den zeitlichen Hinweis fallen lässt und nur die Aussage «Die Route ist in Ordnung» behält.

Ein nützliches *Handover* bewahrt drei Arten der Unvollständigkeit.

Erstens bewahrt es Lücken. Hat niemand das entfernte Ende untersucht, bleibt dieses Ende *Unknown*. Die Lücke sollte nicht mit einer wahrscheinlichen Fortsetzung gefüllt werden, bloss damit der Zweig abgeschlossen wirkt.

Zweitens bewahrt es Widerspruch. Melden zwei Piloten unterschiedliche sichtbare Zustände, behalten beide *Observations* ihre *Sources* und Zeiten, bis eine neue Prüfung sie erklären oder ersetzen kann. Den unbequemen Eintrag zu löschen, kann die Karte aufräumen und zugleich Evidenz einer Veränderung vernichten.

Drittens bewahrt es die Absicht. Der nächste Pilot muss wissen, weshalb ein offener Zweig wichtig ist. *Vor der Nachschubbewegung bestätigen* lenkt die Aufmerksamkeit. *Muss gescannt werden* kann alles zwischen Neugier und unmittelbarer Abhängigkeit bedeuten.

Ownership vervollständigt das *Handover*. Ein *Unknown* ohne *Steward* wird zum gemeinsamen Hintergrund: Alle können es sehen, und niemand weiss, ob er handeln soll. Ein benannter Pilot oder eine benannte

Rolle übernimmt die nächste *Observation*, aktualisiert die betreffende Aufzeichnung und meldet, ob die Route ihren Zweck tragen kann.

Der Verantwortliche garantiert keine Antwort. Er garantiert, dass die Frage zwischen den Schichten nicht verschwindet. Kann die *Connection* nicht gefunden werden, fehlt *Access* oder hat sich der Zweig verändert, veröffentlicht er diese Grenze, statt die alte Route aus der Erinnerung zu verlängern.

Gute Sprache beim *Handover* bleibt ruhig. Sie zeigt Ungewissheit, ohne den nächsten Piloten zu beunruhigen oder die Tatsachen aus Zeitnot abzuschwächen. Das Ergebnis bietet Evidenz, Alter, Zweck und die nächste ungeklärte Entscheidung.

In der Eröffnungsszene bleibt der frühere mündliche Bericht mit seiner *Source* verbunden. Der Pilot, der die Bewegung vorbereitet, kann erkennen, wo die direkte *Observation* endet. Der früheren Schicht zu misstrauen, ist nicht nötig. Für jenen Abschnitt, dessen alter Bericht nun eine grössere Konsequenz tragen soll, bleibt Bestätigung nötig.

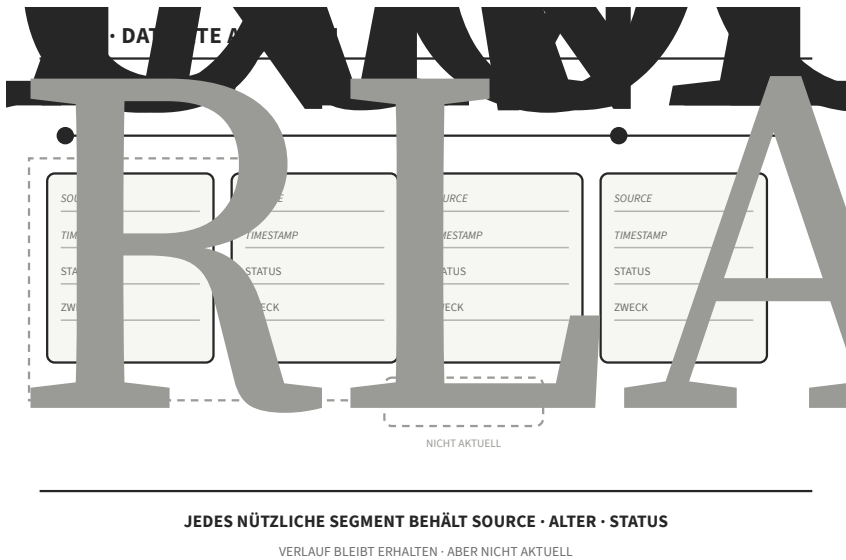


Abbildung AFH-B2-FIG-006 — Eine Chain besteht aus datierten Aussagen. Eine Chain bleibt ehrlich, wenn jedes nützliche Segment *Source*, Alter und Evidenzstatus behält.

Tote Zweige schliessen

Alte Zweige erzeugen Druck, selbst wenn sie keinem Zweck mehr dienen.

Sie verlängern die Karte, ziehen neue Notizen an und bieten eiligen Piloten vertraute Formen. Eine Linie, die Geschichte hätte werden sollen, kann mit einer Route verwechselt werden, nur weil sie sichtbarer bleibt als eine ungeklärte Alternative.

Einen Zweig zu schliessen bedeutet, seine Geltung für gegenwärtige Entscheidungen zu beenden. Der genaue Zeitpunkt, an dem jede *Connection* verschwand, muss dafür nicht immer bewiesen werden.

Direkte Evidenz kann einen festen Abschluss rechtfertigen. Ein neuer *Scan* stützt die aufgezeichnete *Connection* vielleicht nicht mehr. Beim erwarteten *Grid* kann ein Pilot das Objekt unter einer bekannten Ansicht nicht vorfinden. Die Aufzeichnung kann festhalten, was wann

beobachtet wurde, und den Zweig danach aus der gegenwärtigen *Chain* entfernen.

Manchmal ist die Evidenz schwächer. Der Zweig wurde nicht geprüft, sein Zweck ist beendet und seine *Observations* sind für eine Wiederverwendung zu alt. Kennzeichne ihn in diesem Fall als historisch oder für gegenwärtige *Work* geschlossen, ohne einen beobachteten Kollaps zu behaupten. Die richtige Aussage lautet *für gegenwärtige Entscheidungen nicht gepflegt, nicht nachweislich verschwunden*.

Dieser Unterschied ist später wichtig. Erscheint ein ähnliches System oder eine ähnliche *Signature*, kann die historische Aufzeichnung erklären helfen, was die Piloten einst sahen. Sie darf nicht mit ihrer früheren Zuverlässigkeit unbemerkt in die gegenwärtige *Chain* zurückkehren. Eine neue *Connection* benötigt eine neue *Observation*, selbst wenn die Zeichnung vertraut wirkt.

Bewahre nicht jeden toten Zweig als Denkmal des Aufwands. Behalte genug Geschichte, um *Handovers*, ungeklärte Bewegung und wichtige Veränderungen zu verstehen. Entferne oder archiviere den Rest gemäss der Publikationspraxis der Gruppe. Die *Shared Map* sollte den aktuellen Entscheidungsweg leichter erkennbar machen als die ungenutzten Möglichkeiten von gestern.

Der Abschluss beendet auch *Ownership*. Ein für einen Zweig verantwortlicher *Steward* sollte wissen, wann dessen Zweck abgeschlossen, übertragen oder aufgegeben ist. Sonst wird aus Gewohnheit weiter gepflegt und die Gruppe verwendet Aufmerksamkeit für eine Route, die keine *Work* mehr trägt.

Der Nachschubzweig aus der Eröffnungsszene erhält ein klares Ergebnis. Stützt die neue *Observation* die Bewegung, erhalten seine relevanten Abschnitte neue *Sources*, Zeiten und einen Zweck. Scheitert die Bestätigung des Zweigs, stoppt die Bewegung und die alte Route kehrt in die Geschichte zurück. Bei einer teilweisen Prüfung bleibt der beobachtete Teil sichtbar, während der Rest zu *Unknown* zurückkehrt.

Keines dieser Ergebnisse setzt Wissen voraus, das nicht vorhanden ist. Der Zweig wird gepflegt, indem seine veröffentlichte Geltung mit der jetzt verfügbaren Evidenz in Einklang gebracht wird.

Feldprinzip

Die *Chain* besteht aus datierten Aussagen über vorübergehende Beziehungen.

Pflege die Aussage, nicht bloss die Linie. Bewahre Richtung, *Source*, *Timestamp*, sichtbaren Zustand und Zweck überall dort, wo sie eine Entscheidung tragen. Lass einen Bericht *Reported* bleiben, bis eine direkte *Observation* bestätigt, was die nächste Handlung benötigt. Lass einen ungenutzten Zweig Geschichte werden, bevor Vertrautheit ihn wieder in eine Route verwandelt.

Eine *Shared Map* erfüllt ihren Zweck, wenn Piloten auf einen Blick drei Dinge unterscheiden können: Was wurde beobachtet, was wurde übernommen und was muss vor einer Bewegung geprüft werden? Sie versagt, wenn eine alte Form auf dem Bildschirm, auf dem sie eingezeichnet bleibt, gegenwärtige Gewissheit ausleiht.

Nützliche Pflege kann deshalb klein sein. Eine rechtzeitige *Observation* des tragenden Abschnitts kann die Entscheidung besser stützen als ein breiter *Scan*, der das Alter dieses Abschnitts ungeklärt lässt.

Die veraltete Route vom Anfang des Kapitels kann ohne Vorhersage behandelt werden. Ihre alte Evidenz bleibt erhalten. Ihr entscheidender Abschnitt wird zur Bestätigung zugewiesen. Die geplante Bewegung wartet nur dort, wo Konsequenz auf *Unknown* trifft.

Die Pflege der Route beantwortet, wohin sich die Gruppe bewegen kann. Dieses Wissen beantwortet nicht, wer die *Structure* am Ende benutzen darf, wer die benötigten *Locations* sehen kann oder wer die Befugnis besitzt, einen dieser Zustände zu verändern.

Kapitel 7 trennt *Access* von *Ownership*.

Kapitel 7 - Access und Ownership

Die Route aus Kapitel 6 wird bestätigt, bevor die Bewegung beginnt.

Am Ziel dockt der erste *Character* in der erwarteten *Structure* an. Er kann seine Schiffe und sein persönliches Inventar sehen, doch der für den nächsten Schritt benötigte *Service* scheint ihm nicht zur Verfügung zu stehen. Ein zweiter *Character* gehört zur besitzenden *Corporation* und trägt die Rolle *Factory Manager*. Er kann die betreffende Aufgabe der *Corporation* einsehen, doch das für ihre Fortsetzung benötigte Material lässt sich nicht aus der Hangardivision entnehmen.

In keiner der beiden Szenen muss etwas defekt sein.

Andocken versprach keinen *Access* zu jedem *Service*. Eine *Corporation Role* versprach keinen *Access* zu jedem Gegenstand. Aus dem *Ownership* der *Structure* wurde nicht jedes Mitglied zu ihrem Administrator, und aus der Sichtbarkeit einer Aufgabe wurde ihr Material nicht verwendbar.

Die Route beantwortete, wohin die Piloten reisen konnten. Sie konnte nicht beantworten, was ihre *Characters* nach der Ankunft tun konnten.

Diese *Capability* muss als Pfad gelesen werden. *Ownership*, Sichtbarkeit, Nutzung und Administration treffen auf unterschiedliche Systeme, bevor eine Handlung möglich wird. Verstehen lässt sich der *Failure* nur, wenn die Gruppe diese Fragen lange genug auseinanderhält, um die fehlende Bedingung zu finden.

Vier getrennte Fragen

Beginne mit *Ownership*: Wem gehört die *Structure*, der Vermögenswert oder die Information?

Ownership bezeichnet die Person oder *Corporation*, deren Systeme das betreffende Objekt enthalten. Es kann darauf hinweisen, wer die Befug-

nis letztlich zuweisen kann. Für sich allein beweist *Ownership* nicht, dass der *Character* auf dem *Grid* diese Befugnis jetzt besitzt.

Die *Structure* aus der Eröffnungsszene gehört einer *Corporation*. Diese Tatsache sagt nichts darüber, welches Mitglied ihr *Profile* bearbeiten darf, wer einen bestimmten *Service* nutzen kann oder wer Material aus einem Hangar der *Corporation* entnehmen darf. In einer *Structure*, die ihm nicht gehört, kann ein Gast nützliche *Work* ausführen. Ein Mitglied des Eigentümers kann dieselbe *Work* möglicherweise nicht abschließen.

Die zweite Frage betrifft die Sichtbarkeit: Was kann dieser *Character* sehen?

Eine sichtbare *Structure*, ein sichtbarer *Service*, eine sichtbare Aufgabe oder Inventardivision liefert dem *Character* Information. Diese kann genügen, um einen Zustand zu melden. Eine solche Ansicht gewährt nicht immer das Recht, das Sichtbare zu verwenden oder zu verändern. Umgekehrt kann ein *Character* eine enge Fähigkeit besitzen, ohne eine bequeme Sicht auf alle umliegenden Informationen zu erhalten.

Die dritte Frage betrifft die Nutzung: Kann der *Character* den *Service* erhalten oder das für die Handlung benötigte Material verwenden?

Die Nutzung kann von einem *Structure Profile*, einer oder mehreren *Access Lists*, den Berechtigungen für Hangars der *Corporation* und dem Zustand des *Service* abhängen. Diese Bedingungen verschmelzen nicht, nur weil sie in derselben *Structure* zusammentreffen. Eine erfüllte Bedingung kann den *Character* an der nächsten noch immer stoppen.

Die vierte Frage betrifft die Administration: Wer kann den bestimmten Zustand verändern?

Ein Administrator kann ein *Profile* bearbeiten oder eine Aufgabe der *Corporation* verwalten, ohne der Pilot zu sein, der die *Work* ausführen soll. Der arbeitende Pilot kann den *Service* nutzen, ohne die Befugnis zu besitzen, dessen Zielgruppe zu verändern. Administration ermöglicht

es, andere Nutzer zu beeinflussen, und darf nicht daraus abgeleitet werden, dass ein *Character* eine einzelne Aufgabe abgeschlossen hat.

Material verdient einen eigenen Platz innerhalb dieses Modells. Sichtbar und nutzbar kann ein *Service* sein, während seine Eingabe in einer Division liegt, die der *Character* weder einsehen noch daraus entnehmen kann. Ebenso begründet *Access* zum Material keinen *Access* zu jenem *Service*, der es verarbeiten soll.

Erst wenn diese Ebenen bekannt sind, sollte die Gruppe nach *Capability* fragen. Die eigentliche Frage ist nicht, ob eine einzelne Berechtigung besteht. *Capability* hängt vom vollständigen Pfad ab. Der vorgesehene *Character* muss am vorgesehenen Ort und zur vorgesehenen Zeit vom Erkennen der Aufgabe bis zu ihrem Abschluss mit dem benötigten Schiff, Material und *Service* gelangen.

Der erste *Character* aus der Eröffnungsszene hat damit das Andocken und einen Teil seiner persönlichen Sichtbarkeit bestätigt. Der zweite hat die Befugnis zur Verwaltung der Aufgabe bestätigt. Keines der Ergebnisse beantwortet die nächste Handlung. Die fehlenden Bedingungen unterscheiden sich, obwohl beide *Characters* dasselbe Ergebnis beschreiben: *Ich kann nicht fortfahren.*

Profiles bieten Services

Ein *Structure Profile* legt fest, welche *Services* anderen Spielern zur Verfügung stehen. Die innerhalb dieses *Profile* zugewiesenen *Access Lists* bestimmen, wer auf jeden *Service* zugreifen kann.

Diese Trennung ist der erste Schlüssel zum Problem des angedockten Gastes. Andocken ist ein über das *Profile* geregelter *Service*. Der nach dem Andocken benötigte *Service* kann eine andere Zuweisung besitzen. Wer die erste Zielgruppe passiert, gehört dadurch nicht zu jeder weiteren.

Einem einzelnen *Service* innerhalb eines *Profile* können eine oder mehrere *Access Lists* unabhängig zugewiesen werden. Dasselbe *Profile* kann

für mehrere *Structures* verwendet werden; eine Änderung daran betrifft alle *Structures*, die es verwenden. Ein *Profile* ist deshalb nicht bloss eine Bezeichnung an einer Hülle. Es wirkt als gemeinsame Festlegung des *Service Access*, deren Umfang vor einer Änderung verstanden werden muss.

Das *Profile* macht einen *Service* nicht physisch funktionsfähig. Es legt die Verfügbarkeit für Nutzer fest. Kapitel 8 trennt dieses Angebot vom *Service Module*, vom *Fuel* und von dem Zustand, die für die Bereitstellung der *Capability* nötig sind. Hier ist entscheidend, dass eine Konfiguration des *Access* erklären kann, weshalb zwei *Characters* bei derselben *Structure* unterschiedliche Ergebnisse erhalten.

Das gegenwärtige *Profile* kann vom *CEO*, den *Directors* und Mitgliedern mit der Rolle *Station Manager* in der besitzenden *Corporation* verändert werden. Diese Befugnis gehört zur Administration. Ihr Umfang ist grösser als die Fähigkeit des Gastes, anzudocken oder einen einzelnen *Service* zu nutzen. Der arbeitende Pilot sollte sie nicht erhalten, nur um einen ungeklärten Zugriffspfad zu lösen.

Der Administrator muss auch den richtigen Umfang erkennen. Die Bearbeitung eines *Profile*, das mehrere *Structures* verwenden, kann den *Access* über die *Structure* aus der Eröffnungsszene hinaus verändern. Ein örtliches Problem sollte keine gemeinsame Änderung auslösen, bevor der Eigentümer weiss, welches *Profile*, welcher *Service* und welche *Access Lists* das Ergebnis erzeugen.

Deshalb ist ein genauer Bericht wertvoll. Die Aussage «Der *Service* fehlt» vermischt mehrere Möglichkeiten. «Dieser *Character* kann andocken, aber zu diesem Zeitpunkt nicht auf den benötigten *Service* in dieser *Structure* zugreifen» benennt den geprüften Pfad. Ein Eigentümer kann die Zuweisung des *Profile*, die *Access Lists* des *Service* und den Betriebszustand des *Service* vergleichen, ohne eine einzelne *Observation* zu einem allgemeinen *Failure* zu erklären.

Profiles legen damit eine Beziehung offen, keine Garantie. Sie verbinden die angebotenen *Services* einer *Structure* mit Zielgruppen. Der Nutzer muss von diesen Zielgruppen weiterhin ausgewählt sein, alle getrennten Materialberechtigungen besitzen und eintreffen, während die Infrastruktur die *Work* tatsächlich ausführen kann.

Access Lists wählen

Access Lists sind Listen von Personen, die zusammen mit *Profiles* den Access zu *Services* in einer *Upwell Structure* kontrollieren.

Eine *Access List* kann einzelne *Characters*, *Corporations*, *Alliances* oder *Everyone* enthalten. Der betreffende Eintrag kann einem einzelnen *Character* den Access erlauben oder verweigern. Das Ergebnis folgt nicht der breitesten Art von Gruppe auf der Seite. Innerhalb einer Liste bestimmt die feinste zutreffende Einstellung: *Alliance* ist genauer als *Everyone*, *Corporation* genauer als *Alliance* und *Character* genauer als *Corporation*.

Angenommen, die *Corporation* des Gastes darf den benötigten *Service* nutzen, doch derselbe Gast besitzt in derselben *Access List* einen ausdrücklich blockierenden Eintrag für seinen *Character*. Die genauere Einstellung des *Character* bestimmt das Ergebnis dieser Liste. Der *Character* kann weiterhin der *Corporation* angehören und über eine andere Zuweisung andocken dürfen. Das Ergebnis beim *Service* ist kein Widerspruch.

Mehrere *Access Lists* fügen eine weitere Grenze hinzu. Sind demselben *Service* mehrere Listen zugewiesen, werden die zutreffenden Ergebnisse jeder Liste kombiniert. Eine Erlaubnis aus einer zugewiesenen Liste kann damit den Access gewähren, obwohl eine andere zugewiesene Liste den *Character* blockiert. Nur die Liste mit dem blockierenden Eintrag zu lesen, würde die endgültige Zielgruppe des *Service* nicht erklären.

Diese Verbindung muss verstanden sein, bevor jemand versucht, den Access zu reparieren. Das Hinzufügen eines *Character* zu einer weiteren Liste kann eine Erlaubnis erzeugen und den ursprünglichen Wider-

spruch bestehen lassen. Das Entfernen einer *Corporation* aus einer Liste kann wirkungslos bleiben, weil eine zweite Liste sie weiterhin zulässt. Der geprüfte *Service* und alle ihm zugewiesenen Listen bilden den betreffenden Pfad.

Auch die Administration einer *Access List* ist vom *Ownership* der *Structure* getrennt. Nur *Admins* oder *Managers* einer *Access List* können die von ihnen verwalteten Listen sehen und verändern. Ein *Station Manager*, der ein *Structure Profile* ändern kann, beweist damit keine administrative Kontrolle über jede zur Zuweisung verfügbare *Access List*. Für jede Ebene muss bestimmt werden, wer sie verändern kann, statt nach einer einzigen Rolle zu suchen, die ausreichend hochrangig klingt.

Das Modell rechtfertigt keinen breiten *Access*. Es ist auch keine Sicherheitsdoktrin zur Suche nach illoyalen Mitgliedern. Die Regel dient der Deutung: Beginne beim vorgesehenen *Character* und *Service*. Folge dem *Profile* zu jeder zugewiesenen *Access List*. Wende die spezifischen Einträge und ihre Kombination an. Halte an, sobald das beobachtbare Ergebnis erklärt ist, oder führe den ungeklärten Zustand zu *Unknown* zurück.

Der gescheiterte Pfad zum *Service* kann untersucht werden, ohne das *Ownership* des Gastes zu verändern. Zuerst prüft der Eigentümer den Pfad des *Service* im *Profile*. Anschliessend testet der Gast das Ergebnis mit demselben *Character*. *Access* ist erst bestätigt, wenn der vorgesehene Pfad funktioniert – nicht, wenn ein Administrator einen plausiblen Namen auf einer Liste sieht.

Roles ermöglichen Handeln

Corporation Roles gewähren bestimmte Fähigkeiten. Sie bilden keine allgemeine Vertrauensstufe, und ihre Namen können nicht für den Handlungspfad stehen.

Der zweite *Character* trägt *Factory Manager*. Diese Rolle erlaubt die Verwaltung von Aufgaben der *Corporation*, darunter Aufgaben, die von

anderen Mitgliedern eingerichtet wurden. Sie erlaubt zudem unabhängig vom Zugriff auf den Hangar der *Corporation* die Auflistung und Nutzung von Blueprints der *Corporation* in der betreffenden *Industry View*.

Dieselbe offizielle Beschreibung bewahrt die fehlende Grenze: Eine Aufgabe, die Eingabematerial benötigt, verlangt weiterhin den richtigen *Take Access* für den betreffenden Hangar oder Container. Befugnis über die Aufgabe und Befugnis über das Material sind nicht dasselbe.

Die Hangardivisionen der *Corporation* unterscheiden mehrere Zugriffsfähigkeiten. *Query* erlaubt dem *Character*, den Inhalt des betreffenden Hangars zu sehen, aber keine Entnahme. *Take* erlaubt die Entnahme von Gegenständen und das Öffnen von Containern in diesem Hangar, nicht jedoch die Entnahme des Containers selbst. *Container Take* regelt die Entnahme von Containern. Auch der Standortkontext der Rolle muss dem Ort entsprechen, an dem das Material liegt.

Diese Unterschiede erklären mehrere Ergebnisse, die sonst willkürlich wirken. Ein *Character* kann die Eingabe sehen und sie nicht entnehmen. Ein anderer kann *Take* ohne *Query* besitzen und keine gewöhnliche Inventaraufstellung erhalten, während eine Aufgabe in *Industry* das benötigte Material aus dem erlaubten Hangar dennoch prüfen und entnehmen kann. Beim dritten *Character* lassen sich Gegenstände aus einem Container entnehmen, nicht aber der Container selbst.

Das Kapitel benötigt nicht jede *Corporation Role*, um diesen Punkt zu zeigen. Es benötigt nur die von der Szene berührten Fähigkeiten. *Factory Manager* beantwortet, wer die Aufgabe verwalten darf. *Query*, *Take* und – wenn die Aufgabe den Container selbst bewegt – *Container Take* beantworten, wie das Material gesehen und gehandhabt werden kann. Das *Profile* und die *Access Lists* beantworten, ob der *Service* der *Structure* dem *Character* zur Verfügung steht.

Station Manager gehört zu einem anderen Pfad. Die Rolle bietet Verwaltungsoptionen für *Upwell Structures*, welche der *Corporation* gehören,

darunter *Profiles*. Sie kann die für die Aufgabe benötigten Materialberechtigungen nicht ersetzen. Sie dem arbeitenden Mitglied zu geben, würde administrative Befugnis hinzufügen, ohne zwingend die fehlende Eingabe zugänglich zu machen.

Verfolge die Aufgabe aus der Eröffnungsszene vom Ergebnis rückwärts. Der *Character* muss die Aufgabe der *Corporation* verwalten, auf den richtigen *Service* zugreifen, das benötigte Material erkennen und dessen Verwendung durch die Aufgabe erlauben können. Die vorhandene Rolle bestätigt nur den ersten Teil. Der *Failure* sieht nicht länger nach einer unzuverlässigen Oberfläche oder einer defekten *Structure* aus. Der sichtbare Bruch liegt zwischen zwei berechtigten Ebenen der Zugriffsrechte.

Gastabhängigkeit unterscheidet sich

Gäste können echte *Capability* besitzen, ohne deren Grundlage zu besitzen.

Der erste *Character* dockt an, lagert persönliche Schiffe und kann vielleicht einige *Services* nutzen. Diese Handlungen sind weder eingebildet noch zweitrangig. Sie tragen wirkliche *Work*. Ihre Abhängigkeit unterscheidet sich, weil ein anderer Eigentümer das *Profile*, die *Access Lists*, die *Service Modules* und den Zustand der *Structure* kontrolliert, auf denen die *Work* beruht.

Damit ein Schiff an einer *Upwell Structure* ein *Tether* erhält, muss ausserdem die Erlaubnis zum Andocken bestehen. Diese Beziehung macht aus der Erlaubnis kein Sicherheitsversprechen. Ein aktives *Tether* besitzt bestimmte Wirkungen, während verschiedene Handlungen und Zustände es verhindern oder unterbrechen können. Der Pilot muss den aktiven Zustand beobachten, statt ihn aus der früher verfügbaren Andockerlaubnis abzuleiten.

Dasselbe Prinzip gilt innerhalb der *Structure*. Ein Gast, der gestern einen *Service* nutzte, besitzt Evidenz über den gestrigen *Access* und Betriebszustand. Das *Profile* oder eine zugewiesene *Access List* kann der Eigentümer

verändern. Auch der *Service* kann in einen anderen Zustand wechseln. Ein Plan des Gastes sollte auf einer aktuellen Prüfung beruhen, wenn die Konsequenz dies verlangt.

Auch Eigentümer tragen Abhängigkeiten. Eine *Corporation* kann die Hülle besitzen, während die einzige Person, die eine benötigte *Access List* verändern kann, abwesend ist. Material kann der *Corporation* gehören, während dem vorgesehenen Betreiber die passenden Hangarrechte fehlen. *Ownership* bündelt die letztliche Kontrolle, garantiert aber nicht, dass die benötigten Personen und Berechtigungen zur benötigten Zeit zusammentreffen.

Dadurch unterscheidet sich die Rückfallebene von Eigentümern und Gästen. Ein Eigentümer kann einen *Steward* einsetzen, der den Pfad von *Access* und Infrastruktur innerhalb der gewählten Befugnisse der *Corporation* pflegt. Ein Gast kann diese Veränderung nicht voraussetzen. Er kann die angebotene *Capability* bestätigen, Alternativen bewahren und die *Work* begrenzen, die von der Konfiguration eines anderen abhängig ist.

Nichts davon macht Gastzugang grundsätzlich ungeeignet. Die Abhängigkeit ist widerrufbar und wird von anderen verwaltet; damit wird sie Teil der *Baseline*. Kann die *Work* bei einer Änderung des *Access* pausieren oder umziehen, ist die Abhängigkeit vielleicht akzeptabel. Würde ein nicht verfügbarer *Service* unersetzliche *Capability* oder offene Verpflichtungen festsetzen, benötigt der Plan eine andere Grenze, bevor die *Work* beginnt.

Der angedockte Gast aus der Eröffnungsszene sollte nicht nach dem *Ownership* der *Structure* fragen. Die nötige Antwort lautet, ob der benötigte *Service* seinem *Character* absichtlich zur Verfügung steht, wer ein unbeabsichtigtes Ergebnis korrigieren kann und was mit der geplanten *Work* geschieht, falls der *Access* fehlt.

Den Pfad prüfen

Namen von Berechtigungen sind Evidenz über eine Konfiguration. Eine abgeschlossene Handlung ist Evidenz über *Capability*.

Prüfe bei einer entscheidenden Aufgabe den Pfad mit jenem *Character*, der sie ausführen soll. Beginne beim Ergebnis und gehe rückwärts. Bestätige, dass der *Character* die Aufgabe erkennen, den *Service* erreichen und das benötigte Material unter den betreffenden Rechten der *Corporation* verwenden kann. Stelle danach fest, ob das richtige Schiff oder eine andere *Capability* vorhanden ist. Ein gesonderter Verantwortlicher muss die Befugnis besitzen, eine gescheiterte Ebene zu beheben, ohne unbeteiligten *Access* zu verändern.

Ort und Zeit gehören zur Prüfung. Eine mit dem falschen Hangarkontext verbundene Berechtigung kann das hier liegende Material nicht tragen. Das Ergebnis eines *Service* aus einer anderen *Structure* kann das zugewiesene *Profile* dieser *Structure* nicht bestätigen. Die erfolgreiche Handlung von gestern bleibt nützliche Geschichte, darf aber eine aktuelle Prüfung nicht unbemerkt ersetzen, wenn die *Work* nun eine grössere Konsequenz trägt. Diese Grenzen verhindern, dass ein vertrauter Name eines *Character* mehrere verschiedene Pfade wie eine einzige Berechtigung wirken lässt.

Die Prüfung sollte verhältnismässig sein. Sie muss kein wertvolles Material verbrauchen oder eine unumkehrbare Aufgabe beginnen, nur um eine Annahme zu beweisen. Der *Character* kann die betreffenden Ansichten untersuchen, das Erscheinen des *Service* bestätigen, den vorgesehenen Materialpfad prüfen und vor der Verpflichtung anhalten. Kann die Handlung nicht sicher getestet werden, halte fest, welche Teile direkt beobachtet wurden und welche abgeleitet bleiben.

Verwende den *Character* und nicht die Vorschau eines Administrators als *Source* für die *Capability* des Nutzers. Ein Eigentümer sieht vielleicht, dass ein *Profile* eine *Access List* enthält. Erst das Ergebnis des vorgese-

henen *Character* bestätigt, wie der verbundene Pfad gegenwärtig für diesen *Character* aufgelöst wird.

Folge dem Pfad des Gastes aus der Eröffnungsszene. Das beabsichtigte Ergebnis ist nicht bloss der Eintritt in die *Structure*. Der *Character* muss andocken, den benötigten *Service* finden und den Punkt erreichen, an dem die geplante *Work* beginnen kann. Das Andocken gelingt; dieser Teil des Pfads erhält damit eine aktuelle *Observation*. Der fehlende *Service* stoppt die Prüfung, bevor Material eingesetzt wird.

Ein Eigentümer liest die Konfiguration danach vom selben Ergebnis rückwärts. Welches *Profile* ist dieser *Structure* zugewiesen? Welche *Access Lists* sind diesem *Service* zugewiesen? Wo wird der Gast innerhalb jeder Liste eingeordnet, und wie werden die Ergebnisse der Listen verbunden? Die Untersuchung bewahrt ausserdem eine getrennte Frage für Kapitel 8: Ist der *Service* selbst betriebsfähig?

Angenommen, der Eigentümer findet die *Corporation* des Gastes in einer erlaubenden Liste. Das ist noch kein Grund, die Frage zu schliessen. Ein genauerer Eintrag für den *Character* kann innerhalb derselben Liste bestimmen. Eine andere Liste kann ein eigenes Ergebnis beitragen. Der Eigentümer hat die betreffende Konfiguration erkannt; der Gast liefert weiterhin die endgültige Evidenz über den *Access*.

Nach einer bewussten Korrektur wiederholt der Gast nur den gescheiterten Teil. Ein neuer Versuch kann zeigen, dass der *Service* nun für den vorgesehenen *Character* verfügbar ist. Dieses Ergebnis beweist weder denselben *Access* für jedes Mitglied der *Corporation* des Gastes noch die Verfügbarkeit aller *Services*. Es kann auch nicht versprechen, dass der *Access* unverändert bleibt. Die Bestätigung ist an diesen *Character* und Zweck gebunden.

Der Pfad des Mitglieds der *Corporation* beginnt anderswo. Das Mitglied kann die Aufgabe sehen und verwalten; *Factory Manager* ist für diesen Teil der Aufgabe damit bereits belegt. Der *Failure* tritt dort ein, wo die

Work auf ihre Eingabe trifft. Das Mitglied bestätigt die betreffende Hangardivision und was die Oberfläche dort sehen oder entnehmen lässt.

Ist die Eingabe unsichtbar, untersucht die Gruppe *Query* im richtigen Standortkontext. Ist sie sichtbar, kann aber nicht wie benötigt verwendet werden, wird *Take* relevant. Soll der Container selbst bewegt werden, stellt sich die engere Frage nach *Container Take*. Jede Prüfung folgt dem Objekt, das die Aufgabe tatsächlich benötigt. Keine beginnt damit, in der Hoffnung auf Erfolg ein Bündel von Rollen zu vergeben.

Diese Methode schützt spätere *Handovers*. Die Aufzeichnung kann sagen, dass der Pfad eines Gastes zum *Service* bestätigt und der Materialpfad eines Mitglieds der *Corporation* korrigiert wurde. Die nächste Schicht erbt zwei begrenzte *Observations* statt der irreführenden Schlussfolgerung, *der Access sei repariert*.

Das Gegenteil gilt ebenso. Eine erfolgreiche Prüfung durch den Nutzer bestätigt nicht die Fähigkeit des Eigentümers, den Pfad später zu verändern. Die Administration sollte durch die dafür verantwortliche Person geprüft werden. Die beiden *Sources* beantworten unterschiedliche Fragen.

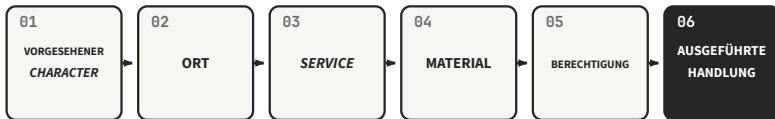
Halte die Prüfung mit ihrer Zeit und ihrem Zweck fest. *Access geprüft* altert schlecht, weil die Aussage *Character*, *Service*, *Material* und *Handlung* auslöst. Die Feststellung «Der benannte *Character* konnte vor dieser Aufgabe den benötigten *Service* und den Materialpfad sehen» kann die genannte *Work* tragen, ohne dauerhaften *Access* zu behaupten.

Zwei begrenzte Ergebnisse bleiben. Die Prüfung folgt dem Pfad des Gastes über das *Profile* und die zugewiesenen *Access Lists*, bis der vorgesehene *Character* den *Service* nutzen kann oder die *Work* verschoben wird. Der Pfad des Mitglieds zur Aufgabe wird über *Factory Manager* und den betreffenden *Access* zum Hangar geprüft, bis die Eingabe genutzt werden kann oder ein anderer befugter Betreiber die Aufgabe annimmt. Niemand musste aufgrund einer Vermutung weitreichende Befugnisse vergeben. Der Name einer Rolle wurde nicht als Beleg für Bereitschaft

behandelt. Jede fehlende *Capability* wurde auf jener Ebene gefunden, die sie tatsächlich bereitstellen konnte.

CAPABILITY · DEN GANZEN PFAD PRÜFEN

GEPRÜFTER PFAD · ABGESCHLOSSEN



ANGENOMMENER PFAD · ENDET HIER



EIN FUNKTIONIERENDER SCHRITT VERVOLLSTÄNDIGT DEN PFAD NICHT

DURCH DEN CHARACTER PRÜFEN, DER HANDELN MUSS

Abbildung AFH-B2-FIG-007 — *Capability* benötigt einen vollständigen geprüften Pfad.

Capability besteht erst, wenn der vorgesehene *Character* den ganzen Pfad vom Ort bis zur Handlung abschliessen kann.

Feldprinzip

Access ist nicht *Ownership*. Etwas zu sehen, gewährt nicht die Erlaubnis, es zu verwenden. Eine Rolle ist nicht die abgeschlossene Handlung, und Administration ist nicht Bereitschaft.

Lies *Capability* als Pfad durch den vorgesehenen *Character*, den Ort, den *Service*, das Material und die Befugnis. Lass *Profiles* die angebotenen *Services* beschreiben, *Access Lists* ihre Zielgruppen auswählen und *Corporation Roles* nur jene Fähigkeiten bereitstellen, die sie tatsächlich benennen. Bestätige das Ergebnis durch den *Character*, der handeln muss.

Die Route aus Kapitel 6 brachte beide Piloten zur richtigen *Structure*. Kapitel 7 erklärt, weshalb sie nach ihrer Ankunft dennoch nicht arbeiten konnten. Sobald die Pfade der Berechtigungen korrigiert sind, bleibt eine weitere Abhängigkeit ausserhalb jeder *Access List* und jeder Rolle.

Eine vollständige Berechtigung kann noch immer zu einem *Service* ohne *Fuel*, in einem Offlinezustand oder ohne Unterstützung für die umgebende *Work* führen. Rechte ermöglichen den Versuch; sie halten die Infrastruktur nicht funktionsfähig.

Kapitel 8 folgt dem *Service* selbst.

Kapitel 8 - Infrastruktur pflegen

Die Berechtigungspfade aus Kapitel 7 werden korrigiert, bevor die *Work* weitergeht.

Der vorgesehene *Character* kann andocken. Die *Structure* zeigt den benötigten *Service*, der Materialpfad ist sichtbar und das Mitglied der *Corporation* besitzt die für die Verwaltung der Aufgabe benötigten Fähigkeiten. Jede Frage zum *Access* liefert nun die erwartete Antwort.

Die Aufgabe kann trotzdem nicht beginnen. Der *Service* ist offline.

Die *Structure* steht auf demselben *Grid* wie zuvor. Piloten können andocken, ihre Hangars öffnen und den vertrauten Innenraum sehen. Die Hülle hat den Anschein der Beständigkeit bewahrt, während eine der Funktionen eingestellt wurde, um die herum die Gruppe ihre *Work* organisiert hat.

Der *Failure* wirkt nur plötzlich, weil die Abhängigkeit verborgen war. Der *Service* benötigte ein *Service Module*, das online bleiben musste. Das Modul benötigte *Fuel*. Der *Fuel* benötigte eine Route, eine *Reserve*, *Access* zur *Fuel Bay* und eine Person, die erkennen sollte, wann die *Reserve* die vorgesehene *Work* nicht mehr tragen konnte. Erst bei der Aufgabe am Ende dieser Kette liess sich die fehlende Pflege nicht mehr übersehen.

Infrastruktur ist nicht das im Raum stehende Objekt. Sie besteht aus der gepflegten Beziehung zwischen diesem Objekt, seinen Funktionen und den Menschen, die von ihnen abhängen.

Services brauchen Mittel

Service Modules verleihen *Upwell Structures* ihre Funktionalität.

Die *Structure* selbst verbraucht nicht allein durch ihre Existenz *Fuel*. Installierte *Service Modules* verbrauchen dagegen *Fuel*, wenn sie online sind. Verschiedene *Services* können unterschiedliche Anforderungen besitzen, die beim jeweiligen Modul sichtbar sind. Dieses Kapitel benö-

tigt keine allgemeingültige Menge, weil die nützliche Frage vor der Berechnung kommt: Welche Funktion ist online, was hält sie online und welche *Work* scheitert, wenn sie stoppt?

Diese Unterscheidung verändert die Eröffnungsszene. Die *Structure* ist nicht verschwunden. Ihre Funktionen zum Andocken und Lagern geben der Gruppe weiterhin einen Ort, an dem sie ankommen kann. Der für die Aufgabe benötigte *Service* ist nicht verfügbar, weil seine eigene Betriebskette unterbrochen ist.

Beginne mit der Funktion statt mit der Hülle. Benenne den *Service*, den die Gruppe tatsächlich nutzt. Bestimme das *Service Module*, das ihn bereitstellt, und ob dieses Modul online ist. Halte den Zustand des *Fuel* über einen *Character* fest, der ihn sehen und pflegen kann. Folge der Abhängigkeit dann nach aussen bis zur *Work*.

Die Kette kann an mehreren Stellen brechen, ohne dieselbe *Observation* zu erzeugen. Das Modul kann installiert, aber offline sein. *Fuel* kann in der *Structure* liegen, während der verantwortliche *Character* ihn nicht dorthin bringen kann, wo das aktive Modul ihn verwendet. Zur Verwaltung der *Structure* kann ein *Character* befugt, aber abwesend sein, wenn sich die Route für den Transport des zusätzlichen *Fuel* schliesst. Der Nutzer kann *Access* zum *Service* besitzen, während der Eigentümer niemanden mit der Pflege des Moduls beauftragt hat.

Die Aussage «Wir besitzen eine *Structure*» löst keine dieser Bedingungen. *Ownership* sagt, wo die letzte Kontrolle liegt, ohne *Fuel* zu bewegen, einen *Service* online zu halten oder den verantwortlichen *Character* verfügbar zu machen.

Das Gegenteil ist ebenso wichtig. Auch wenn ein *Service* hauptsächlich *Work* von Gästen trägt, bleibt seine Pflege eine Verpflichtung des Eigentümers. Die Nachfrage der Gäste erzeugt weder automatisch *Fuel* noch einen Verantwortlichen oder eine Entscheidung über die Wiederherstellung. Bietet eine Gruppe eine *Capability* an, muss jemand wissen,

ob dieses Versprechen beabsichtigt ist und wie lange die gegenwärtigen Mittel es tragen können.

Verfolge *Fuel* als *Capability* statt als Inventar. Eine *Reserve* ausserhalb von *Home* muss weiterhin eine nutzbare *Chain* durchqueren. Eine *Reserve* innerhalb der *Structure* muss einem *Character* zur Verfügung stehen, der sie in die *Fuel Bay* legen und den betreffenden *Service* verwalten kann. *Fuel* in der richtigen *Bay* muss noch immer für die erwarteten aktiven *Service Modules* ausreichen. Jeder Schritt kann wahr sein, während der nächste *Unknown* bleibt.

Eine beruhigende Gesamtsumme ohne verbundenen Pfad bringt der Gruppe deshalb wenig. «*Fuel* ist vorhanden» kann Material auf dem Weg, in einem unzugänglichen Lager oder eine für einen anderen Zweck bestimmte *Reserve* beschreiben. Die Aussage «Der befugte Verantwortliche beobachtete vor dieser *Work* den *Fuel*, der diesen aktiven *Services* zur Verfügung steht» gibt der nächsten Entscheidung eine nutzbare Grenze.

Der Verbrauch verändert sich zudem mit den ausgewählten *Services*. Nimmt ein Eigentümer ein neues Modul online, verändert er die Bedingungen jeder bestehenden Schätzung. Nimmt ein Verantwortlicher eines offline, kann er die verbleibenden Mittel länger reichen lassen und zugleich eine Funktion entfernen, von der noch jemand abhängt. Eine nützliche Aufzeichnung benötigt sowohl die *Observation* der Mittel als auch die Gruppe von *Services*, die sie tragen sollten.

Damit wird ein bewusster Entscheid für offline zu einem Teil der Pflege und nicht zu ihrem Gegenteil. Manchmal kann die Gruppe einen *Service* nicht erhalten oder will seine *Work* nicht länger unterstützen. Ihn aus der operativen *Baseline* zu entfernen, ist dann ehrlicher, als Nutzer um eine *Capability* planen zu lassen, die ohne Verantwortlichen verschwinden wird. Der Entscheid muss diese Nutzer vor ihrer nächsten Verpflichtung erreichen.

Pflege verlangt nicht, dass jeder installierte *Service* online bleibt. Eine ungenutzte Funktion kann Aufmerksamkeit und Ressourcen verbrauchen, ohne dem gegenwärtigen Zweck zu dienen. Zur Pflege gehört der Entscheid, welche *Services Home* tragen soll, und das Entfernen aufgebener Erwartungen aus der *Baseline*, bevor Nutzer darauf *Work* aufbauen.

Der *Service*, der in der Eröffnungsszene offline ist, legt damit zwei Fragen offen. Welches Mittel fehlt jetzt? Weshalb bemerkte niemand, dass die Abhängigkeit ihre Grenze erreichte? Die Wiederherstellung des Moduls kann die erste beantworten. Nur eine zugewiesene Pflegepraxis kann die zweite beantworten.

Bedeutung von Power

Eine *Upwell Structure* befindet sich in *Full Power*, wenn mindestens ein *Service Module* online ist. Ist kein *Service Module* installiert oder online, befindet sich die *Structure* in *Low Power*.

Diese Begriffe beschreiben einen wirklichen Zustand, beantworten aber nicht, ob ein bestimmter *Service* verfügbar ist. Eine *Structure* mit einem aktiven *Service Module* kann sich in *Full Power* befinden, während das für die *Work* der Gruppe benötigte *Service Module* offline bleibt. *Full Power* als *alle Funktionen bereit* zu lesen, würde den Fehler wiederholen, der Andocken als Beweis für *Service Access* behandelte.

Low Power liefert eine andere *Observation*. Gegenwärtig ist kein *Service Module* online. Diese Tatsache reicht weiter als eine einzelne fehlende Funktion für eine Aufgabe, erklärt aber noch immer nicht die Ursache. Vielleicht fehlt *Fuel*. Vielleicht wurden *Service Modules* bewusst offline genommen. Ein anderer Zustand der *Structure* kann beteiligt sein. Der sichtbare Zustand begrenzt die Untersuchung, ohne eine vollständige Geschichte zu liefern.

Der *Power State* wirkt auch über gewöhnliche *Work* hinaus. Das offizielle System unterscheidet diese Zustände bei Verwundbarkeit und Betrieb

einer *Structure*. Verteidigung, Verstärkung und Reaktion im Kampf liegen ausserhalb dieses Kapitels. Die operative Lektion lautet lediglich, dass *Full Power* und *Low Power* keine schmückenden Bezeichnungen sind und ein Wechsel zwischen ihnen in die gemeinsame *Baseline* gehört.

Ein längeres Ausbleiben des Verbrauchs von *Fuel* durch *Service Modules* kann eine *Structure* von *Low Power* in *Abandoned* versetzen. In diesem Zustand stehen bestimmte Fähigkeiten der *Structure* nicht mehr zur Verfügung. Für eine Pflegepraxis wird die genaue Übergangszeit nicht benötigt. Auf eine *Observation* von *Abandoned* zu warten, würde den früheren Pflegefehler zu einem deutlich breiteren Zustand heranwachsen lassen.

Die drei Bezeichnungen sollten getrennt bleiben. *Full Power* beweist mindestens ein aktives *Service Module*, nicht jede vorgesehene Funktion. *Low Power* beweist, dass kein *Service Module* online ist, nicht den Grund dafür. *Abandoned* ist ein eigener Zustand, der nach anhaltend fehlendem Verbrauch von *Fuel* durch *Service Modules* erreicht wird und mit weiteren verlorenen Fähigkeiten verbunden ist.

Mit diesem Vokabular kann die Gruppe den *Failure* aus der Eröffnungsszene genau melden. Bleibt eine *Structure* in *Full Power*, kann der benötigte *Service* als bestimmtes Modul oder als bestimmte Abhängigkeit untersucht werden. Eine *Observation* von *Low Power* weitet die Untersuchung auf jeden vorgesehenen *Service* aus. Eine *Observation* von *Abandoned* zeigt, dass die Planung gewöhnlicher *Work* bereits Annahmen über die ursprüngliche Aufgabe hinaus verloren hat.

Power besitzt Bedeutung, wenn der Zustand den Umfang der nächsten Frage verändert. Irreführend wird er, wenn eine Bezeichnung als allgemeines Urteil über die Gesundheit von *Home* verwendet wird.

Zustand sichtbar machen

Pflege benötigt eine *Source*, welche den gepflegten Zustand sehen kann.

Der *Structure Browser* zeigt zugängliche *Structures* und die angebotenen *Services*. Diese Ansicht hilft einem Nutzer festzustellen, ob seinem *Character* die benötigte *Capability* angeboten wird. Die Bedingungen des Eigentümers hinter diesem Angebot bleiben ausserhalb dieser Ansicht. Mitglieder der besitzenden *Corporation* können mit der entsprechenden Befugnis die Ansicht *My Structures* sehen. Diese Information umfasst den verbleibenden *Fuel* und den gegenwärtigen *Vulnerability State* ihrer *Structures* sowie Informationen zu *Profile* und Zuweisungen. Die Ansicht des Eigentümers kann damit *Observations* tragen, die einem Gast nicht zur Verfügung stehen.

Die beiden Perspektiven sollten zusammentreffen, ohne vermischt zu werden. Gäste können melden, dass der *Service* dem vorgesehenen *Character* zu einer bestimmten Zeit nicht zur Verfügung steht. Ein befugter Eigentümer kann melden, dass das Modul offline ist, welcher Zustand des *Fuel* sichtbar ist und welcher Zustand der *Structure* gilt. Zusammen erklären die *Observations* mehr als eine von ihnen allein.

Mache aus der breiteren Ansicht des Eigentümers kein dauerhaftes Wissen. *Fuel* wird verbraucht, solange aktive *Service Modules* laufen. Ein zu Beginn einer Arbeitsphase gesehener Wert altert, während diese *Services* arbeiten und Menschen die *Structure* verändern. Der *Timestamp* gehört auch dann zur *Observation*, wenn die Oberfläche eine genaue Zahl anzeigt.

Der Zustand des *Service* verdient ebenfalls eine eigene Zeile. Ein aktives Modul, der *Power State* der *Structure* und die Fähigkeit des Nutzers, auf den *Service* zuzugreifen, hängen zusammen, bleiben aber getrennt. Einer kann sich verändern, ohne bei allen anderen dasselbe Ergebnis auszulösen. Nur *Structure grün* festzuhalten, entfernt jene Ebene, welche die nächste Person prüfen muss.

Sichtbarkeit muss zu einer Entscheidungsgrenze führen. Eine *Observation* des *Fuel* ist wichtig, wenn sie zeigt, ob der vorgesehene *Service* durch die *Work* hindurch und bis zum Handeln der nächsten verantwortlichen

Person online bleiben kann. Nicht jeder wechselnde Wert muss für jeden Nutzer veröffentlicht werden. Sichtbar sein muss der Zustand für die Person, welche die Reaktion trägt.

Ist die Ansicht nicht verfügbar, wird der Zustand zu *Unknown*, statt vorausgesetzt zu werden. Die Abwesenheit des Eigentümers weist vielleicht nicht auf ein Problem mit dem *Fuel* hin, kann der Gruppe aber die Fähigkeit nehmen, den Zustand zu bestätigen oder zu korrigieren. Das ist ein eigener *Failure* der *Capability*.

Die Piloten können nun trennen, was jeder weiss. Der arbeitende *Character* beobachtet den *Service* offline. Der befugte Eigentümer bestätigt den Zustand des Moduls und des *Fuel*. Ihre gemeinsame Darstellung verbindet diese *Observations* mit der pausierten Aufgabe, ohne zu behaupten, jede Funktion der *Structure* sei ausgefallen.

Failure wirkt weiter

Ein *Failure* eines *Service* endet selten beim *Service*.

Manufacturing und *Research* können in *Upwell Structures* über die entsprechenden *Service Modules* bereitgestellt werden. Geht das betreffende *Service Module* offline, können laufende Aufgaben dieser Art pausieren, bis der *Service* zurückkehrt. Wird das Modul entfernt oder die *Structure* zerstört, können die betreffenden Aufgaben abgebrochen werden.

Diese mechanischen Ergebnisse wirken über die abhängige *Work* auf den Betrieb. Eine pausierte Aufgabe verhindert, dass ihre erwartete Ausgabe erscheint. Eine spätere Aufgabe kann auf diese Ausgabe warten. Ein Pilot und ein Schiff können bereits für ihren Transport eingeteilt sein. Die *Chain* kann für eine Lieferung bestätigt worden sein, deren Material nicht bereit sein wird.

Verfolge den *Failure* aus der Eröffnungsszene nach aussen. Der *Service* ist offline, deshalb kann die Aufgabe nicht beginnen. Das Material bleibt im Lager, statt in die Aufgabe einzugehen. Die geplante Abschlusszeit

verliert ihre Grundlage. Ein von einem anderen Piloten erwarteter Ersatzgegenstand wird nicht verfügbar sein. Die für seinen Transport vorbereitete Route besitzt keinen gegenwärtigen Zweck mehr.

Kein einzelner Schritt ist eine Krise. Die *Exposure* entsteht, wenn jeder abhängige Plan die alte Annahme weiterverwendet. Ein Transportpilot, der auf einer Route auf Material wartet, das nicht hergestellt werden kann, verwendet Aufmerksamkeit für einen toten Zweig. Setzt die Gruppe ein weiteres Schiff ein, weil sie mit der Ausgabe rechnet, kann sie denselben Mangel vertiefen.

Die Reaktion sollte sich in der entgegengesetzten Richtung durch die Kette bewegen. Pausiere die Entscheidung über die Aufgabe. Informiere den Verantwortlichen der abhängigen *Work*. Löse Bewegungen, die keine Ladung oder keinen Zweck mehr besitzen. Entscheide, ob die Wiederherstellung dieses *Service* der richtige Einsatz von *Fuel* und Befugnis ist oder ob die *Work* zu einer bestehenden Alternative wechseln soll.

Access kann den *Failure* schwerer lesbar machen. Der *Service* wirkt vielleicht für einen *Character* wegen des *Profile* nicht verfügbar und bleibt für einen anderen online. Umgekehrt kann ein vollumfänglicher *Access* des Nutzers ein Modul, das offline ist, nicht funktionsfähig machen. Beide Pfade benötigen eine Bestätigung, bevor die Ursache zugewiesen wird.

Ein Abbruch besitzt eine stärkere Grenze als eine Pause. Eine pausierte Aufgabe kann fortgesetzt werden, wenn ihr *Service* zurückkehrt. Die Entfernung des betreffenden Moduls oder die Zerstörung der *Structure* kann die *Work* stattdessen abbrechen. Die Pflegeaufzeichnung muss eine behebbare Unterbrechung von einer Entscheidung unterscheiden, welche die Aufgabe beendet oder ersetzt.

Andere *Services* erzeugen andere äussere Konsequenzen. Ein Katalog liegt ausserhalb dieses Kapitels. Die Methode bleibt dieselbe: Benenne die Funktion und bestimme die *Work*, die sie voraussetzt. Beschreibe

den ersten sichtbaren *Failure*. Entscheide danach, was stoppen muss, bevor sich die fehlende Ausgabe auf einen weiteren Plan ausbreitet.

Zeit verändert die Grösse der Konsequenz. Ein vor Beginn einer neuen Aufgabe offline entdeckter *Service* kostet vielleicht nur eine verschobene Entscheidung. Derselbe *Failure*, der erst gefunden wird, nachdem mehrere Aufgaben und Bewegungen um ihre erwarteten Ausgaben herum organisiert wurden, kann Piloten, Material und Routen festhalten. Frühe Sichtbarkeit verkleinert nicht den mechanischen Zustand; sie bindet weniger Verpflichtungen daran.

Deshalb gehört der Verantwortliche der *Work* in die Reaktion. Wer die *Structure* pflegt, kann melden, ob das Modul zurückkehren kann. Diese Person kann nicht allein entscheiden, ob der alte Zeitplan, das Material und die Bewegung noch sinnvoll sind. Die Wiederherstellung eines *Service* nach einer langen Unterbrechung stellt die Annahmen vor der Pause nicht wieder her.

Baue diese Annahmen der Reihe nach neu auf. Bestätige den *Service*. Beurteile die Aufgabe und ihre Eingaben erneut. Gib eine neue Erwartung über den Abschluss an die abhängige *Work* weiter. Verpflichte erst danach die Bewegung erneut. Während der fehlenden Verfügbarkeit des *Service* kann sich die *Chain* verändert haben, und der einst für den Transport eingeteilte Pilot kann inzwischen einen anderen Zweck tragen.

Ist die Wiederherstellung ungewiss, muss die Rückfallebene mit derselben Disziplin verglichen werden. Eine andere *Structure* bietet dem vorgesehenen *Character* vielleicht den *Service*, besitzt aber keine bestätigte Route. Die Bewegung des Materials kann mehr *Exposure* erzeugen als das Warten. Die richtige Entscheidung ergibt sich aus der gesamten Abhängigkeit, nicht aus dem Wunsch, den ursprünglichen Zeitplan unversehrt wirken zu lassen.

Infrastruktur wird verständlich, wenn ihre Konsequenzen vor einem *Failure* verfolgt werden – nicht erst, nachdem jeder abhängige Pilot bereits gehandelt hat.



EINE BEDINGUNG IST NICHT DER GANZE PFAD

EIN BRUCH BETRIFFT BEGRENZTE WORK • KEINEN ERFUNDENEN TOTALAUSFALL

Abbildung AFH-B2-FIG-008 — Service hängt von mehreren gepflegten Bedingungen
ab. *Infrastructure* trägt *Work* durch mehrere gepflegte Bedingungen, von denen keine allein für den ganzen Pfad steht.

Abhängigkeiten zuweisen

Kritische Infrastruktur benötigt einen Verantwortlichen, bevor sie eine Reparatur benötigt.

Beginne beim *Service*, der gegenwärtige *Work* trägt. Halte seinen vorgesehenen Zweck, das bereitstellende *Service Module* und den Verantwortlichen fest, der dessen Betriebszustand und Versorgung mit *Fuel* sehen kann. Benenne die Person, die handeln soll, sobald der beobachtete Zustand diesen Zweck nicht mehr trägt.

Beobachter und Verantwortlicher können unterschiedliche *Characters* sein. Ein Nutzer kann bemerken, dass ein *Service* verschwunden ist. Den Zustand des *Fuel* und des Moduls kann ein befugter Eigentümer prüfen. Der zusätzliche Betriebsstoff kann wiederum von einem Piloten durch die *Chain* getragen werden. Die Pflegepraxis gelingt, wenn diese Beiträge bei einer benannten Entscheidung zusammentreffen – nicht, wenn jede Person jede Befugnis erhält.

Gib der Abhängigkeit einen Auslöser. Eine Prüfung kann an den Beginn einer Arbeitsphase, vor eine lange Aufgabe, vor den Aufbruch des einzigen Verantwortlichen oder an eine Veränderung der Route gehören. Die nützliche Häufigkeit ergibt sich aus dem Verbrauch von *Fuel*, dem *Access* zu *Replacement* und der Konsequenz einer Unterbrechung. Kein allgemeingültiger Zeitplan kann diese Bedingungen ersetzen.

Füge die abhängige *Work* hinzu. Ein *Service* ohne benannten Nutzer ist vielleicht nicht entscheidend, selbst wenn sein Betrieb teuer ist. Eine bescheidene Funktion kann tragend sein, wenn mehrere offene Verpflichtungen sie voraussetzen. Der Zweck bestimmt, was die Gruppe pflegt.

Füge die Folge eines *Failure* und die erste Reaktion hinzu. Welche *Work* pausiert, wenn das Modul offline geht? Wer muss informiert werden? Welche Bewegung kann gelöst werden? Welcher Zustand verlangt die Ausweitung einer Frage zum *Service* auf eine Prüfung der gesamten *Structure*? Diese Entscheidungen sollten bestehen, bevor die Oberfläche den *Failure* zeigt.

Benenne zuletzt den Pfad für *Replacement*. Die Alternative kann ein anderer verfügbarer *Service*, eine verschobene Aufgabe, eine kleinere Verpflichtung oder eine Route zur Wiederherstellung der Betriebsmittel sein. Eine Rückfallebene ist nur glaubwürdig, wenn ihr *Access*, ihre *Chain* und ihre Materialbedingungen mit derselben Sorgfalt gelesen wurden wie beim primären Pfad.

Diese kurze Darstellung der Abhängigkeit ist kein dauerhaftes Inventar jedes Moduls. Sie folgt dem aktiven Zweck. Wird eine Aufgabe geschlossen, kann ihr *Service* die entscheidende Gruppe verlassen. Beginnt neue *Work*, kann ein zuvor nebensächlicher *Service* mit einem *Steward*, einer *Source* für die *Observation* und einer Reaktionsgrenze hinzukommen.

Stell dir die Aufzeichnung zu Beginn der nächsten Arbeitsphase vor. Sie nennt den für die offene Aufgabe benötigten *Service* und den befugten *Character*, der sein Modul zuletzt online beobachtet hat. Derselbe Eintrag bezeichnet die für den *Fuel* verantwortliche Person, die Zeit der *Observation* des *Fuel* und den Punkt, vor dem eine weitere Prüfung nötig ist. Er nennt den Verantwortlichen der Aufgabe, die auf ihre Ausgabe wartende Bewegung und die Rückfallebene, falls der *Service* die *Work* nicht tragen kann.

Der nächste Pilot kann diese Aufzeichnung lesen, ohne jedes Verwaltungsrecht zu benötigen. Bleibt der *Service* sichtbar und ist die nächste Prüfung noch nicht fällig, weiss der Pilot, welche datierte Evidenz den Plan trägt. Fehlt der *Service*, erkennt er die zu pausierende *Work* und die zu kontaktierende Person. Ist der Verantwortliche nicht verfügbar, tritt eine vorbereitete Rückfallebene an die Stelle einer improvisierten Suche nach Befugnis.

Ein *Handover* muss Grenzen bewahren. Ein Verantwortlicher, der vor seinem Aufbruch *Fuel* beobachtete, bestätigt nicht, was nach einer unbestimmten Zeit des Verbrauchs verbleiben wird. Er benennt die aktiven *Services*, die von der *Observation* zu tragende *Work* und den Zeitpunkt, an dem die Verantwortung zur nächsten Person übergeht. Der neue Verantwortliche bestätigt die Annahme, wie ein *Steward* der *Chain* einen ungeklärten Zweig annimmt.

Nimmt niemand an, sollte die Infrastruktur im Plan sichtbar degradieren. Verschiebe die neue Aufgabe, verkürze den unterstützten Zeitraum oder entferne den *Service* aus den versprochenen *Capabilities*. Unverän-

dert fortzufahren, würde die Abhängigkeit dem Zufall zuweisen und die Nutzer glauben lassen, sie besitze weiterhin einen Verantwortlichen.

Verantwortung umfasst auch ausreichende Befugnis, ohne sie anzuhäufen. Die verantwortliche Person benötigt den für *Observation* und Handlung nötigen *Access*, aber nicht jede Fähigkeit der *Structure* oder *Corporation*. Ein getrennter Administrator kann die breitere Kontrolle behalten, solange der Reaktionspfad ausdrücklich benannt und innerhalb der vom *Service* tolerierten Zeit verfügbar ist.

Diese Anordnung übersteht Abwesenheit besser als ein vertrauter Name in einer Notiz. Sie sagt, wer beobachtet, wer handelt, wer über die *Work* entscheidet und wer das nächste *Handover* annimmt. Mehrere Positionen kann derselbe *Character* einnehmen, doch die Funktionen bleiben klar genug getrennt, um beim Aufbruch dieses *Character* übertragen zu werden.

Das Modul, das offline ist, kann wiederhergestellt werden, ohne den *Failure* zu verbergen, der seine Abhängigkeiten offengelegt hat. Der Eigentümer bestätigt den Zustand von *Fuel* und Modul. Der Verantwortliche der *Work* entscheidet, ob die Aufgabe weiterhin wichtig ist. Eine verantwortliche Person nimmt die nächste Prüfung an, und die Gruppe hält fest, was geschehen soll, wenn der *Service* während der *Work* nicht online gehalten werden kann.

Eine Reparatur stellt Funktionalität wieder her. Zuweisung macht ihre Rückkehr tragfähig.

Feldprinzip

Infrastruktur ist eine fortlaufende Verpflichtung, keine einmal platzierte Hülle.

Trenne *Structure*, *Service Module*, *Fuel*, *Access* des Nutzers und abhängige *Work*. Lies *Full Power*, *Low Power* und *Abandoned* als begrenzte Zustände statt als allgemeine Urteile. Gib jedem entscheidenden *Service* einen

sichtbaren Zustand, eine verantwortliche Person, einen Auslöser zur Prüfung und eine Alternative für den Fall, dass dieser Zustand versagt.

Die *Structure* aus der Eröffnungsszene bewegte sich nie. Verändert hat sich die gepflegte Beziehung zwischen ihrem *Service* und der darauf aufgebauten *Work*. Indem die Gruppe dieser Beziehung rückwärts folgt, kann sie *Capability* wiederherstellen, ohne Berechtigung, *Power State* oder *Ownership* mit der vollständigen Antwort zu verwechseln.

Part II begann damit, gewöhnliches *Home* beobachtbar zu machen. Er hielt die *Chain* ehrlich, trennte Befugnis von *Capability* und legte die Pflege unter einer vertrauten *Structure* offen. *Home* kann nun *Work* tragen, ohne vorzugeben, diese Unterstützung geschehe von selbst.

Funktion allein ist kein Zweck. Die nächste Frage lautet, was die Gruppe mit dieser *Capability* tun will und welche neue *Exposure* beginnt, sobald Aufmerksamkeit, Schiffe, Routen und Material an die Aufgabe gebunden werden.

Part III beginnt mit *Work*.

Part III - Hier arbeiten

Home wird exponiert, sobald *Capability* eingesetzt wird. Ein Schiff verlässt das Lager, Aufmerksamkeit verengt sich, Material sammelt sich und die gepflegte *Chain* beginnt neben Bewegung auch Rückkehr zu tragen. Part III folgt der *Work* von ihrer ersten Verpflichtung über Ressourcenoperationen, Transport, Lagerung und *Replacement*. Ranglisten des Ertrags bleiben ausserhalb dieses Blickfelds. Jede Aufgabe muss offenlegen, was sie bindet, welche Bedingungen beobachtbar bleiben müssen und wie *Work* enden kann, bevor ihr Zweck die Fähigkeit zur Wiederherstellung überholt.

Kapitel 9 - *Work* erzeugt *Exposure*

Das *Grid* war lange genug ruhig geblieben, um leer zu wirken.

Es war nicht leer. Ein Scanner blieb nahe am Rand des Systems, ein Pilot beobachtete die offenen *Connections* und einige Schiffe warteten in der *Structure*. Nichts verlangte jedoch sofortige Aufmerksamkeit. Dieses fehlende Verlangen liess den Ort still erscheinen.

Dann dockte das erste Arbeitsschiff ab.

Sein Pilot überquerte das *Grid*, brachte die Aufgabe in Reichweite und begann den ersten Zyklus. Ein zweiter Pilot folgte. Das gemeinsame Gespräch veränderte sich. Einen Augenblick zuvor war die *Chain* das Thema gewesen; nun wandte sich die Aufmerksamkeit der *Work* zu: Position, Fortschritt, verfügbarer Raum und nächste Handlung. Die Route hatte sich nicht verändert. Keine neue *Signature* war erschienen. Dennoch besass eine *Observation*, die zehn Minuten zuvor gewöhnlich gewesen wäre, jetzt mehr Gewicht. Zwei Schiffe und ihre Piloten konnten sich langsamer von dem Zweck lösen, der sie band.

Work hatte das Feld verändert, bevor sie etwas hervorbrachte.

Das ist die erste Tatsache des Arbeitens von *Home* aus. Eine Aufgabe ist nicht bloss eine Handlung im Raum. Jede Aufgabe schafft eine vorübergehende Anordnung aus Schiffen, Aufmerksamkeit, Zeit, Routen, Material und Menschen. Diese Anordnung kann vernünftig, nötig und ertragreich sein. Sie bleibt *Exposure*. Ein *Explorer* beginnt die *Work*, indem er diese *Exposure* erkennt, während sie gewählt wird, statt sie erst zu entdecken, wenn sich die *Work* nur noch schwer verlassen lässt.

***Work* hinterlässt Spuren**

Jede Aufgabe bringt etwas dorthin, wo es zuvor nicht benötigt wurde.

Das Schiff ist der sichtbarste Teil. Es verlässt das Lager, betritt ein *Grid* und nimmt eine Arbeitsposition ein. Doch das Schiff bildet nur

einen Rand der Spur. Jemand muss es fliegen. Eine Route muss es hinaus- und zurücktragen. Kapazität muss aufnehmen, was die Aufgabe hervorbringt. Andere Schiffe müssen es vielleicht scannen, bewegen oder ersetzen. Die *Structure* muss vielleicht Lagerung oder einen *Service* tragen. Eine Aufzeichnung muss der nächsten Person möglicherweise sagen, was sich verändert hat.

Die Spur reicht damit über den Ort hinaus, an dem die *Modules* ihre Zyklen durchlaufen. Sie führt rückwärts in die Vorbereitung und vorwärts in Rückkehr, Lagerung und *Replacement*. Seitlich reicht sie in *Capabilities*, die vorübergehend nicht für andere *Work* verfügbar sind.

Betrachte das erste Schiff auf dem ruhigen *Grid*. Vor dem Abdocken war die Hülle eine Möglichkeit. Sie konnte eine andere Notwendigkeit beantworten, als *Replacement* warten oder nicht exponiert bleiben. Mit Beginn der *Work* erhielt dieselbe Hülle eine Position, einen Zweck und einen Pfad, der rückgängig gemacht werden musste. Das an Bord gesammelte Material erzeugte ein Ziel. Die *Observations* des Piloten konkurrierten mit dem übrigen System. Keine dieser Veränderungen war ein Notfall. Zusammen bildeten sie eine operative Spur.

Diese Spur lässt sich durch sechs Fragen beschreiben.

Welches Schiff oder welche andere *Capability* wird belegt? Welche *Observations* werden leichter übersehen, während die Aufmerksamkeit auf der Aufgabe liegt? Wie lange kann die *Work* weitergehen, ohne eine geordnete Rückkehr zweifelhaft zu machen? Welche *Connections* müssen das Schiff und sein Ergebnis tragen? Wo sammelt sich Wert? Wer kann die Operation beobachten, fortsetzen, stoppen und wiederherstellen?

Dies sind keine Bewertungskategorien. Sie legen verschiedene Formen der Verpflichtung offen. Eine Aufgabe nahe bei *Home* benötigt vielleicht nur wenig Routentiefe, konzentriert aber Material auf einem *Grid*. Eine entfernte Aufgabe kann wenig Ladung hervorbringen und zugleich einen Scanner belegen sowie mehrere Aussagen über die *Chain* altern lassen. *Manufacturing* kann Schiffe sicher in einer *Structure* halten und

dabei Materialien, einen *Blueprint*, einen passenden *Service* und einen Plan für die Ausgabe miteinander verbinden. Transport bringt vielleicht überhaupt nichts hervor, platziert jedoch für *Replacement* entscheidende Gegenstände hinter einer veränderlichen Route.

Die Spur sollte vor Beginn der *Work* benannt werden. Sobald die Tätigkeit läuft, beschreiben Menschen natürlicherweise, was noch gewonnen werden kann. Eine erklärte Spur hält sichtbar, was noch gelöst werden muss.

Der Zweck bestimmt, welche Teile am wichtigsten sind. Eine Aufgabe, die einen entscheidenden Gegenstand ersetzen soll, erhält ihren Wert aus der wiederhergestellten *Capability* statt aus dem Marktpreis. Bei einer Aufgabe, die den Inhalt einer *Site* verstehen soll, kann die Information wichtig sein, selbst wenn wenig Ladung zurückkehrt. Beim Räumen eines Lagers stehen Bewegung und Ziel im Zentrum. Die Spur verändert sich mit dem Zweck, auch wenn dasselbe Schiff und *Grid* verwendet werden.

Work kann deshalb nicht nur anhand ihres Ertrags beurteilt werden. Ertrag beschreibt ein Ergebnis. Die Spur beschreibt den Zustand, der beim Streben danach entsteht.



WORK BINDET, BEVOR ERFOLG BEKANNT IST

JEDE BINDUNG BENÖTIGT EINEN EIGENEN RELEASE PATH

Abbildung AFH-B2-FIG-009 — Work erzeugt sechs gleichzeitige Bindungen. *Work* erzeugt einen Fussabdruck, indem sie Schiffe, Aufmerksamkeit, Zeit, Wege, Inventar und Menschen bindet, bevor Erfolg bekannt ist.

Aufmerksamkeit wandert

Aufmerksamkeit ist ein Arbeitsmittel, obwohl sie nie in einem Hangar erscheint.

Beim Annähern an eine Aufgabe sieht ein Pilot Distanz, Position und den Zustand der *Modules*. Dazu kommen Zyklus und Ladung. Jeder Punkt kann Aufmerksamkeit verdienen. Je flüssiger die Aufgabe läuft, desto leichter wird ihr Rhythmus zum ganzen Blickfeld. Fortschritt liefert eine stetige Folge kleiner Entscheidungen, und diese Entscheidungen wirken unmittelbar. Eine Veränderung anderswo im System kann im Vergleich still bleiben.

Das Problem ist nicht, dass arbeitende Piloten nachlässig werden. Die *Work* selbst verlangt, dass sie manches auf Kosten von anderem bemerken.

Vor dem ersten Zyklus auf dem *Grid* der Eröffnungsszene wäre eine neue *Signature* eine Frage der Kartierung gewesen. Wer hat sie beobachtet, wie alt ist die *Observation* und verändert sie die bekannte *Chain*? Nachdem zwei Schiffe der *Work* verpflichtet sind, wird dieselbe *Signature* zusätzlich zu einer operativen Frage. Kann die Gruppe die Aufgabe erhalten, während sie die *Signature* identifiziert? Soll der nächste Zyklus beginnen? Welcher Pilot bleibt für eine *Observation* frei? Die Evidenz ist nicht feindlicher geworden. Die umgebende Verpflichtung hat ihre Konsequenz verändert.

Diese Unterscheidung verhindert zwei häufige Fehler. Der erste erklärt jede Veränderung zur Gefahr, nur weil *Work* läuft. Jede neue *Signature* bleibt *Unknown*, bis sie beobachtet wurde. Eine Erklärung zu erfinden, verbessert die Entscheidung nicht. Der andere Fehler behandelt die Veränderung als unbedeutend, weil sie keine Gefahr beweist. Ein *Unknown* kann wichtig sein, ohne als Feind gedeutet zu werden. Bis jemand das fehlende Wissen wiederherstellt, kann es genügen, keine weitere Verpflichtung einzugehen.

Aufmerksamkeit während der *Work* benötigt deshalb eine Grenze. *Ownership* der Aufgabe kann bei einer Person liegen, die *Observation* des *Grid* bei einer anderen und jene der *Chain* bei einer dritten. In einer kleinen Gruppe kann ein Pilot mehrere Funktionen tragen; sie müssen dennoch benannt werden. Sonst können alle mit demselben sichtbaren Fortschritt beschäftigt sein, während jeder annimmt, jemand anderes bewahre das weitere Bild.

Das *Handover* zwischen diesen Funktionen ist wichtig. «Die *Chain* beobachten» ist zu ungenau, wenn niemand weiss, welche *Sources* beobachtet werden oder wann die letzte *Observation* stattfand. Alle Beteiligten müssen auch wissen, welche Veränderung die *Work* unterbrechen soll.

Eine nützliche Bedingung lässt sich einfach formulieren: Der gegenwärtige Rückkehrzweig bleibt innerhalb des angenommenen Alters seiner *Observation*, und eine benannte Person trägt ungelöste *Signatures*. Ein verpasstes Update stoppt jede weitere Verpflichtung, bis die *Observation* wiederhergestellt ist.

Solche Bedingungen versprechen keine Sicherheit. Sie machen die Kosten verengter Aufmerksamkeit sichtbar. Zudem erlauben sie, eine Operation ehrlich zu verkleinern. Ist nur ein Pilot verfügbar, kann die Gruppe eine Aufgabe mit kleinerer Aufmerksamkeitsspur wählen, näher bei *Home* bleiben oder nicht beginnen. Die Antwort ist nicht immer eine weitere Rolle. Manchmal muss vom Feld weniger verlangt werden.

Eine *Observation* während der *Work* sollte kurz genug bleiben, um der Handlung zu dienen. Ein langer Bericht, den beim Arbeiten niemand lesen kann, ist keine Aufmerksamkeit. Die Arbeitsaufzeichnung benötigt die Veränderung, *Source* und Zeit, den betroffenen Zustand sowie die Person, welche die nächste Prüfung trägt. Die Deutung kann später vertieft werden. Unmittelbar soll die Aufgabe mit der umgebenden Welt verbunden bleiben.

Schiffe werden gebunden

Ein Schiff im Lager ist auf eine Weise verfügbar, die einem Schiff bei der *Work* fehlt.

Dieser Unterschied wird leicht unterschätzt, weil die Hülle weiterhin demselben Piloten oder derselben *Corporation* gehört. Sie kann unbeschädigt und nahe bei *Home* bleiben. Sobald sie jedoch eine Arbeitsposition eingenommen hat, verlangt ihre Freigabe Handlungen: den gegenwärtigen Zyklus abschliessen oder unterbrechen, das *Grid* verlassen, die Route durchqueren, entladen und für einen anderen Zweck bereit werden. Ladung, Drohnen, Werkzeuge oder die Geometrie der Aufgabe

können weitere Schritte hinzufügen. Aus Verfügbarkeit als Tatsache des Inventars ist ein Wiederherstellungspfad geworden.

Die Bindung besitzt Abstufungen. Ein ausgerichtetes Schiff mit wenig Ladung kann rasch freigegeben werden. Ein Schiff tief in einer *Chain*, mit gesammeltem Material und abhängig von einem anderen Piloten für ein vollständiges Bild der Rückkehr, ist stärker gebunden. Die wichtige Frage lautet nicht, ob das Schiff sicher oder unsicher ist. Sie lautet, wie viele Bedingungen für seine geordnete Freigabe wahr bleiben müssen.

Replacement verändert diese Lesart. Ist die arbeitende Hülle das einzige lokale Schiff für eine entscheidende Funktion, entfernt ihre Bindung diese Funktion zugleich aus der *Reserve*. Eine kleine Unterbrechung kann dann über die Aufgabe hinausreichen. Andere wertvolle Schiffe mögen bestehen, doch Wert ersetzt die fehlende *Capability* nicht in dem Augenblick, in dem sie benötigt wird.

Benenne vor der *Work* die gebundene *Capability* und die verbleibende Rückfallebene. Frage, ob ein anderes Schiff das Ergebnis bergen und ein anderer Pilot die *Observation* fortsetzen kann. Entscheide, ob die Aufgabe stoppen kann, ohne ihre Ausgabe unbrauchbar zu machen. Benötigt die Freigabe des Schiffs einen *Service*, ein Lagerrecht oder eine Person, die vielleicht nicht verfügbar bleibt?

Die Antwort muss nicht immer eine Verdoppelung sein. Das einzige geeignete Schiff bewusst einzusetzen, kann vernünftig sein. Der Unterschied liegt in der benannten Konsequenz. Welche *Work* wird als Nächstes unmöglich, wenn das Schiff verloren geht, verspätet zurückkehrt oder jenseits einer kollabierten *Connection* bleibt? Diese Konsequenz kann eine kleinere Ladung, einen kürzeren Zeitraum oder einen anderen Zeitpunkt rechtfertigen. Ebenso kann sie das Fortfahren rechtfertigen, weil der Zweck wichtig genug ist. *Explorer* entfernt die Verpflichtung nicht; er macht sie bewusst.

Menschen werden zusammen mit ihren Schiffen gebunden. Piloten, die eine vollständige Rückkehr versprechen, sind in der nächsten Schicht vielleicht nicht verfügbar. Ein Scanner, der mehrere arbeitende Piloten unterstützt, kann zum einzigen Pfad werden, von dem ihre Wiederherstellung abhängt. Ein Transportschiff, das auf die Ausgabe wartet, kann Zeit belegen, die eine andere Operation erwartet. Deshalb benötigt eine neben der Aufgabe festgehaltene Rolle sowohl einen Verantwortlichen als auch eine vorgesehene Freigabe. «Ich erledige das» beginnt Verantwortung. «Bis zu dieser Bedingung bin ich verantwortlich, danach übernimmt diese Person oder Rückfallebene» macht die Verantwortung übertragbar.

Die beiden Schiffe auf dem *Grid* der Eröffnungsszene sind somit nicht bloss zwei Hüllen, die eine Ausgabe erzeugen. Sie bilden einen vorübergehenden Entzug von *Capability* aus der Gruppe. Dieser Entzug kann vollkommen angemessen sein. Er gehört dennoch in die Entscheidung.

Routen tragen Konsequenz

Work gibt der *Chain* eine zukünftige Last.

Auf dem Hinweg kann eine *Connection* ein leeres Schiff und eine Absicht tragen. Auf der Rückkehr muss derselbe Pfad vielleicht das Schiff, sein gesammeltes Material und den Wert des *Replacement* für alles an Bord tragen. Auch andere Piloten können die *Connection* während der *Work* benutzt haben. Zeit ist vergangen. Die Route, welche die Aufgabe erreichbar machte, wird durch den Entscheid zum Beginn nicht eingefroren.

Jede *Wormhole Connection* ist kurzlebig und sowohl durch die Masse einer einzelnen Durchquerung als auch durch die Gesamtmasse begrenzt. Ihre Information liefert grobe Signale über Lebensdauer und Masse, keine Reservierung für eine spätere Rückkehr. Der Arbeitsplan muss die Route deshalb als veränderliche Abhängigkeit behandeln.

«Wir kamen hier hindurch» ist Evidenz für vergangenen *Access*, kein Beweis für spätere Wiederherstellung.

Die Rückkehr beginnt im Plan für den Hinweg. Die Gruppe bestimmt, welcher Zweig die Operation trägt, welche *Observation* ihn stützt, wer die Aktualisierungen trägt und welche Veränderung verhindert, dass mehr Wert von *Home* weggebracht wird. Zudem wird benannt, wo die *Work* endet, falls die bevorzugte Route den ursprünglichen Plan nicht tragen kann. Eine Rückfallebene kann bedeuten, Material vorübergehend zwischenzulagern, früher zurückzukehren, die Aufgabe aufzuteilen oder nicht zu beginnen. Sie garantiert keinen *Exit*. Stattdessen verhindert sie, dass die erste Routenänderung zum ersten Augenblick wird, in dem jemand eine Alternative erwägt.

Die Konsequenz einer Route wird nicht nur in Sprüngen gemessen. Ein kurzer Zweig, den viele Schiffe benutzen, kann eine grössere gemeinsame Konsequenz für die Masse erhalten als ein längerer, der einmal benutzt wird. Geordnet kann ein entferntes *Grid* bleiben, wenn seine Aufzeichnung der Rückkehr aktiv gepflegt wird. Ein nahe gelegenes *Grid* lässt sich vielleicht schwer freigeben, weil alle annahmen, die *Connection* werde bestehen bleiben. Tiefe ist sichtbar. Abhängigkeit ist es häufig nicht.

Die *Work* verändert auch die Bedeutung des Alters der *Chain*. Eine alternde *Observation* auf einem unbenutzten Zweig kann zu einer späteren Prüfung einladen. Dasselbe Alter auf der einzigen geplanten Rückkehr beladener Schiffe kann den nächsten Zyklus jetzt stoppen. Kein allgemeingültiges Alter macht aus einer guten eine schlechte Route. Der Zweck bestimmt, wann die Ungewissheit für die nächste Verpflichtung zu gross geworden ist.

Diese Überlegung schützt vor dem verführerischen Satz «noch eine Ladung». Diese Ladung durchquert keinen abstrakten Tunnel. Sie tritt nach allen früheren Durchquerungen und vor einer ungewissen Zukunft in eine gemeinsam genutzte *Connection* ein. Ihr Wert kann

die Konsequenz einer Schliessung erhöhen. Die Entscheidung gehört zum jetzt sichtbaren Zustand, nicht zur Zahl der bereits erfolgreich abgeschlossenen Rückkehrbewegungen.

Ein Routenplan endet erst, wenn die Operation zurückgekehrt ist, ihr Ergebnis das vorgesehene Lager erreicht hat und die Aufzeichnung der *Chain* zeigt, was die Bewegung verändert hat. Bis dahin ist die *Chain* Teil des Arbeitsorts.

Rechtzeitig stoppen

Eine Operation, die beginnen kann, aber ihr Ende nicht beschreiben kann, ist nicht vollständig vorbereitet.

Stoppen wird häufig als Reaktion auf einen offensichtlichen *Failure* vorgestellt. Ein *Module* fällt aus, eine Route schliesst sich oder eine *Threat* erscheint, und die Gruppe reagiert. Diese Momente sind wichtig, aber späte Grenzen. Eine nützliche Stoppbedingung ist meist stiller. Sie greift, bevor die Operation die letzte einfache Wahl verbraucht.

Die Bedingung kann Aufmerksamkeit betreffen: Kein benannter Beobachter bleibt frei, um ein neues *Unknown* zu übernehmen. Eine Schiffsgrenze ist erreicht, wenn die für Rückkehr oder *Replacement* benötigte *Reserve Capability* nicht mehr verfügbar ist. Zeit wird zur Einschränkung, wenn die *Observation* der Route über das von dieser Aufgabe angenommene Alter hinausgeht. Lagerung kann die Grenze bilden, sobald die nächste Ladung eine nicht verwaltete Konzentration erzeugen würde. Eine Grenze der Menschen erscheint, wenn der für die Wiederherstellung verantwortliche Pilot gehen muss, bevor ein weiterer vollständiger Zyklus zurückgebracht werden kann.

Jede Bedingung sollte beobachtbar und mit einer Handlung verbunden sein. «Stoppe, wenn es riskant wird» lässt beide Begriffe in jenem Moment undefiniert, in dem sie am ehesten diskutiert werden. «Beginne keinen weiteren Zyklus, solange der Rückkehrzweig ein *Unknown* ohne Verantwortlichen enthält» benennt die Evidenz und die unmittelbare

Reaktion. Die Operation kann pausieren, ohne zu behaupten, was das *Unknown* bedeutet.

Stoppen bedeutet nicht immer, die Aufgabe aufzugeben. Es kann bedeuten, den gegenwärtigen umkehrbaren Schritt abzuschliessen, ein Schiff zurückzubringen, die Zahl aktiver Piloten zu verringern oder eine *Observation* wiederherzustellen, bevor die *Work* fortgesetzt wird. Die Handlung sollte zu der veränderten *Exposure* passen. Ist die Lagerung die Einschränkung, löst mehr Material am selben nicht verwalteten Ort das Problem nicht. Mehr *Ownership* beim *Scanning* kann eine Einschränkung des Routenwissens beantworten. Sobald die einzige Hülle für *Replacement* in die *Work* eingetreten ist, kann die Verkleinerung der Operation die *Reserve* wiederherstellen.

Der Stopp benötigt ebenfalls einen Verantwortlichen. Eine Bedingung, die alle beobachten und niemand trägt, kann zum Gesprächsthema werden, während der nächste Zyklus bereits beginnt. Die verantwortliche Person benötigt kein Kommando über alle Beteiligten. Innerhalb der Operation braucht sie eine klare Befugnis, das Erreichen der vorher vereinbarten Grenze auszusprechen. Eine erneute Beurteilung kann später mit einem neuen Plan und neuen Bedingungen erfolgen. Das unterscheidet sich davon, dass die Dynamik den ursprünglichen Entscheidungsprozess umschreibt.

Die schwierigste Grenze ist oft die Genügsamkeit. Fortschritt ist sichtbar, zukünftige Ungewissheit nicht. Entscheide, wie ein abgeschlossener Zweck aussieht, bevor die Menge zum einzigen verbleibenden Mass wird. Genug Material für das benannte *Replacement*, genug Information zur Beantwortung der Frage, genug Bewegung zur Wiederherstellung der *Reserve*: Diese Enden geben *Capability* frei. «So viel wie möglich» besitzt kein natürliches Ende, bis das Feld eines erzwingt.

Auf dem *Grid* der Eröffnungsszene muss die neue *Signature* keinen Angriff beweisen. Sie erfüllt die Bedingung, nach der keine weitere Verpflichtung beginnt, solange ein wichtiges *Unknown* keinen Verant-

wortlichen besitzt. Ein Schiff beendet die bereits laufende umkehrbare Handlung. Das zweite löst sich von der *Work* und stellt die *Observation* wieder her. Die Gruppe ist nicht geflohen. Sie hat die Fähigkeit bewahrt, das Weitere zu wählen.

Feldprinzip

Work beginnt, indem sie *Exposure* verändert.

Benenne vor einer Messung des möglichen Ergebnisses, was die Aufgabe bindet. Berücksichtige Schiff, Aufmerksamkeit, Zeit, Route, Inventar und Menschen. Beschreibe, wie jede Verpflichtung gelöst werden kann und welche *Observation* den nächsten unumkehrbaren Schritt verhindert. Gib dem Stopp einen Verantwortlichen, bevor Fortschritt das Stoppen unbequem macht.

Das ruhige *Grid* wurde nicht gefährlich, nur weil Schiffe zu arbeiten begannen. Es erhielt Konsequenz. Eine *Signature*, eine alternde Route und eine belegte Hülle bedeuteten nun etwas anderes, weil die Gruppe einen Zweck um sie gelegt hatte. Ein *Explorer* liest diesen Unterschied, ohne eine *Threat* zu erfinden und ohne die *Exposure* zu übergehen.

Diese Lesart bleibt nützlich, wenn die Aufgabe gelingt. Eine abgeschlossene Ladung kann weiterhin ermüdete Aufmerksamkeit, eine alternde Route, konzentriertes Inventar und eine Hülle hinterlassen, die noch nicht in die *Reserve* zurückgekehrt ist. Erfolg gehört zum Ergebnis; Abschluss gehört zur gesamten Spur. Beides zu verwechseln, verleitet die nächste Operation dazu, mit übernommenen, aber unbenannten Verpflichtungen zu beginnen.

Kapitel 9 hat die Spur beschrieben, die viele Arten von *Work* teilen. Das nächste Kapitel folgt einer Operation vom Zweck bis zum Bericht. Es zeigt, wie Exploration und Ressourcenarbeit dieselbe Überlegung verwenden können, ohne so zu tun, als seien sie dieselbe Aufgabe.

- Ressourcenoperationen

Im *Probe Scanner* erschien eine *Site*, die ausreichend aufgelöst war, um sie zu betreten.

Diese Tatsache übte stillen Druck aus. Die *Work* des Auffindens war abgeschlossen, das Schiff war bereit und die Route bekannt. Die nächste Handlung schien durch das Ergebnis selbst ausgewählt: zur *Site* warpen, eintreten, beginnen.

Der Pilot wartete.

Nichts im Ergebnis der Prüfung erklärte, weshalb die *Site* jetzt genutzt werden sollte. Es benannte die wiederherzustellende *Capability* nicht. Auch die zu beantwortende Frage und das zu lagernde Material blieben offen. Es sagte nichts über den Rückkehrzweig, seinen Beobachter oder jene *Observation*, welche den Versuch beenden sollte. Das Ergebnis hatte eine Gelegenheit offengelegt. Eine Operation hatte es nicht geschaffen.

Diese Pause trennt Finden von Wählen. Exploration, Ernte und andere Ressourcenarbeit unterscheiden sich in Schiffen, Werkzeugen, sichtbaren Veränderungen und spezialisiertem Wissen. Sie sollten nicht in eine allgemeingültige Checkliste eingeebnet werden. Eine operative Bewegung können sie dennoch teilen. Zweck und Ort kommen zuerst. Vorbereitung führt zu *Observation*, Handlung und erneuter Beurteilung. Auf den Stopp folgen Rückkehr, Lagerung und Bericht.

Die Bewegung dauert länger als die Tätigkeit auf dem *Grid*, weil die *Exposure* länger dauert. Eine *Site* ist nicht abgeschlossen, sobald ein *Module* stoppt. Die Aufgabe bleibt offen, solange das Ergebnis unterwegs, das Ziel der Lagerung ungeklärt oder in der gemeinsamen Aufzeichnung noch der Zustand vor der Operation beschrieben ist.

Zweck wählen

Der Zweck sollte in einen Satz passen.

«Wir gehen hinaus, um zu sehen, was da ist» klingt nach Exploration, lässt das Ende aber offen. Ein *Grid* zu sehen, eine *Signature* aufzulösen oder einen ganzen Zweig zu untersuchen könnte den Satz erfüllen. «Wir gehen hinaus, um Ressourcen zu sammeln» ist ebenso dehnbar. Jeder weitere Zyklus kann vorgeben, derselben Anweisung zu dienen.

Ein brauchbarer Zweck benennt die Veränderung, welche die Gruppe herbeiführen will. Vielleicht sucht er genug Material für eine bestimmte *Reserve*. Möglich sind auch eine Antwort darauf, ob eine *Site* eine spätere Entscheidung tragen kann, oder eine Probe zur Prüfung einer Abhängigkeit. Der Satz muss keine Menge enthalten, wenn keine nützliche Menge bekannt ist. Er muss den Abschluss erkennbar machen.

Der Zweck kommt vor der Auswahl, weil Auswahl Dynamik erzeugt. Eine sichtbare *Site*, eine vertraute Tätigkeit oder ein wartendes Schiff kann verfügbare *Work* notwendig wirken lassen. Die *Work* beginnt dann beim Objekt, und die Gruppe erfindet den Grund danach. Diese Umkehrung verdeckt, ob das Ergebnis verwendet werden kann. Material ohne Ziel, Information ohne Frage und Ausgabe ohne Verantwortlichen können die *Exposure* erhöhen und wie Fortschritt wirken.

Der Zweck bestimmt zudem die annehmbare Spur. Muss eine entscheidende *Reserve* bald wiederhergestellt werden, kann eine bescheidene Operation mit kurzer Rückkehr besser dienen als eine ergiebigere Möglichkeit hinter mehreren alternden *Connections*. Soll eine unbekannte *Site* verstanden werden, kann die Operation die *Observation* und eine saubere Freigabe höher gewichten als Ladung. Bei einem Übungszweck kann eine lesbare Abfolge wichtiger sein als Geschwindigkeit. Dieselbe Gelegenheit kann für einen Zweck geeignet und für einen anderen ungeeignet sein, ohne ihre Eigenschaften zu verändern.

Bevor der Pilot die *Site* betritt, folgen aus dem Satz vier Konsequenzen. Welches Ergebnis ist genug? Wohin wird dieses Ergebnis gebracht?

Welche *Capability* darf dafür gebunden werden? Welche Evidenz würde dazu führen, dass dieser Versuch dem Zweck nicht mehr dient?

Die letzte Frage schützt die erste. Angenommen, die Operation soll eine *Reserve* für *Replacement* wiederherstellen. Eine veränderte Route kann die nächste Ladung eher jenseits von *Home* festsetzen, als sie zur *Reserve* zu bringen. Das Fortfahren kann noch immer die Gesamtmenge erhöhen und sich zugleich vom benannten Ergebnis entfernen. Der Stopp verwirft den Wert nicht. Er bleibt dem Zweck treu.

Verschiedene Menschen können verschiedene Zwecke vorschlagen. Sie sollten den Unterschied auflösen, bevor Schiffe gebunden werden. Ein Pilot versteht die *Site* vielleicht als Lerngelegenheit, ein anderer erwartet einen Produktionsinput und ein dritter nimmt an, die Aufgabe werde fortgesetzt, solange die *Site* besteht. Bleiben diese Erwartungen unausgesprochen, lesen alle Beteiligten denselben Fortschritt und denselben Stopp anders.

Der Satz macht die Operation besprechbar: «Kehre mit dem für das benannte *Replacement* benötigten Material zurück. Nutze den gegenwärtigen Zweig nur, solange die Bedingung seiner *Observation* erfüllt bleibt.» Er verspricht weder Ertrag noch eine allgemeingültige Methode. Er sagt jedoch, weshalb sich das Feld gleich verändern wird.

Site lesen

Der Ort beginnt bei der Information, die EVE tatsächlich liefert.

Das *Sensor Overlay* zeigt Explorationsinhalte um das Schiff. Der *Probe Scanner* trennt *Cosmic Anomalies*, für deren Ortung keine *Probes* nötig sind, von *Cosmic Signatures*, die mit *Core Scanner Probes* aufgelöst werden müssen. Diese Unterscheidung beschreibt, wie eine Gelegenheit erreichbar wird. Ob die Gruppe bereit ist, sie zu nutzen, beantwortet sie nicht.

Das Ergebnis der Prüfung sollte deshalb auf zwei Ebenen gelesen werden. Die erste ist das beobachtete Objekt: seine *Source*, sein gegen-

wärtiger Zustand im *Scan*, sein Ort und die Zeit, zu der diese Tatsachen festgestellt wurden. Die zweite ist die umgebende Operation: die Route zum *Grid*, die beabsichtigte *Work*, die dafür benötigten Werkzeuge und das erwartete Ergebnis, das wieder entfernt werden soll.

Nur die erste Ebene ist für den Piloten aus der Eröffnungsszene vollständig. Die *Signature* ist aufgelöst. Die *Site* kann betreten werden. Bevor der Pilot aus diesem Ergebnis *Work* macht, vergleicht er es mit dem Zweck und der gegenwärtigen *Baseline*. Ist dies das Objekt, das die Gruppe finden wollte? Haben sich System oder *Chain* seit der Vorbereitung der Route verändert? Hängt bereits eine andere Operation von demselben Beobachter, Schiff, Lager oder derselben *Connection* ab?

Eine *Site* zu lesen, verlangt nicht, alle ihre Ereignisse als bekannt darzustellen. Spezialisierte Tätigkeiten können Auslöser, Gefahren oder Mechaniken enthalten, die dieses Feldhandbuch nicht lehrt. Hängt der Erfolg von solchem Wissen ab, muss die Gruppe es aus einer aktuellen spezialisierten *Source* beziehen, bevor sie die Aufgabe als vorbereitet behandelt. Unwissenheit lässt sich nicht durch die allgemeine Anweisung reparieren, vorsichtig zu sein. Sie sollte als *Unknown* mit einem Verantwortlichen oder als Grund benannt werden, nicht zu beginnen.

Die *Site* besitzt zudem einen sichtbaren Zustand, der sich durch Nutzung verändern kann. Objekte können verbraucht, Container geöffnet, Schiffe bewegt und Kapazitäten gefüllt werden. Entscheide, welche Veränderungen für die nächste Handlung wichtig sind. Eine Ernteaufgabe kann auf verbleibendes nutzbares Material und freien Frachtraum achten. Eine Explorationsaufgabe kann darauf achten, welche *Observations* die Frage bereits beantwortet haben. Bei einer Untersuchung ist die Entfernung von Material vielleicht weniger wichtig als die Möglichkeit, den aufgezeichneten Zustand weiterhin diesem Besuch zuzuordnen.

Hier darf aus einem gemeinsamen Denkmodell keine gemeinsame Checkliste werden. «Verbleibender Frachtraum» kann für eine Aufgabe zentral und für eine andere bedeutungslos sein. «Die Frage ist beant-

wortet» kann einen Explorationsbesuch beenden, obwohl das *Grid* weiterhin Tätigkeit anbietet. Die gemeinsame Form ist keine Liste von Tatsachen über die *Site*. Sie verbindet Zweck, sichtbaren Zustand und nächste Verpflichtung.

Halte nur fest, was die Verwendung überdauert. Ein Name ohne *Timestamp* kann dem nächsten Piloten nicht sagen, wann die *Site* bekannt war. Ein Ergebnis ohne *Source* lässt sich nicht von einer Erwartung unterscheiden. Die Aussage, die *Site* sei «klar», kann verbergen, welche Frage tatsächlich beantwortet wurde. Genaue, begrenzte *Observations* verkleinern spätere *Work*. Vages Vertrauen lässt sie die Entdeckung wiederholen.

Sobald die *Site* lesbar ist, kann der Pilot entscheiden, ob sie der richtige Ort für diese Operation ist. Verfügbarkeit ist schliesslich zu Bedeutung geworden, doch die *Work* hat noch nicht begonnen.

Exit vorbereiten

Der *Exit* sollte vorbereitet werden, solange das Verlassen noch einfach ist.

Das bedeutet mehr als eine *Location* zu speichern. Zur Vorbereitung gehören ein gegenwärtiger Rückweg, ein Verantwortlicher für dessen *Observation*, ein Ziel für das Ergebnis und ein Freigabepfad für jedes gebundene Schiff. Der Plan bestimmt, welche Handlung sauber gestoppt werden kann und welcher Schritt eine Abhängigkeit erzeugt, die abgeschlossen werden muss. Er benötigt ausserdem einen Punkt, nach dem keine neue *Work* beginnt, selbst wenn die gegenwärtige Handlung noch beendet werden kann.

Der Rückweg ist eine datierte Aussage über die *Chain*. Seine *Wormhole Connections* sind kurzlebig und unterliegen sowohl Grenzen für einzelne Durchquerungen als auch einer Gesamtmassengrenze. Ihre groben Signale helfen, den gegenwärtigen Zustand zu beschreiben, reservieren die *Connection* aber nicht. Vorbereitung verbindet deshalb Routennut-

zung mit Routenbeobachtung. Für den betreffenden Zweig wird ein Pilot benannt, der ihn aktuell hält. Die Operation legt fest, welche fehlende oder veränderte Evidenz sie pausiert.

Danach kommt das Schiff. Die arbeitende Hülle muss sich unter den von der Operation angenommenen Bedingungen vom *Grid* lösen können. Diese Frage hängt von der Tätigkeit ab und lässt sich in diesem Kapitel nicht durch ein allgemeingültiges *Fit* beantworten. Sie umfasst die für die Aufgabe benötigten Werkzeuge, aber auch Kapazität, den Weg zur Rückkehr des Ergebnisses und die Auswirkung der gebundenen Hülle auf die lokale *Reserve*. Tätigkeitsbezogene Details gehören in eine aktuelle *Source* und eine ausdrückliche Vorbereitungsprüfung, sobald die Operation sie verlangt.

Das Ergebnis benötigt ein Ziel, bevor es besteht. «*Home*» ist noch kein Lagerplan. Welche persönliche *Location* oder welche *Location* der *Corporation* soll es aufnehmen? Wer kann es einlagern, wer muss es verwenden, und kann der Empfänger es danach sehen und entnehmen? Handelt es sich beim Zweck um Information, welche gemeinsame Aufzeichnung nimmt die begrenzte *Observation* auf? Handelt es sich um Material, erzeugt das Ziel eine nicht verwaltete Konzentration oder vermischt es für den Zweck wichtige Bestände mit entbehrlicher Ausgabe? Menschen vervollständigen die Vorbereitung. Eine Person entscheidet, ob der Zweck weiterhin erfüllt wird. Der Beobachter trägt Veränderungen ausserhalb der unmittelbaren *Work*. Für die Rückkehr hält ein Verantwortlicher Pfad und Ziel lesbar. Eine kleine Operation kann alle drei Funktionen einer Person zuweisen; der Plan sollte dann klein genug werden, damit diese Verbindung glaubwürdig ist. Ein Titel erzeugt keine Aufmerksamkeit.

Die Vorbereitung benennt auch das Teilergebnis. Bleibt etwas zurück, das nach der ersten Handlung die Rückkehr wert ist, eine bewahrenswerte *Observation* oder ein zu meldender Zustand? Eine Aufgabe, die erst nach vielen unumkehrbaren Schritten nützlich wird, kann stärkere

Bedingungen benötigen als eine, die ein sauberes Teilergebnis erzeugt. Diesen Unterschied zu erkennen, verhindert, dass bereits investierte *Work* zum Argument wird, spätere Evidenz zu ignorieren.

Für den Piloten aus der Eröffnungsszene besteht jetzt mehr als ein mit *Warp* erreichbares Ergebnis. Die *Site* dient einer benannten Notwendigkeit. Eine aktuelle *Observation* und ein Verantwortlicher tragen den Rückkehrzweig. Das Ergebnis besitzt ein Ziel. Die Stoppbedingungen sind für alle Beteiligten sichtbar. Der Eintritt in das *Grid* ist nicht länger die nächste vom Scanner nahegelegte Handlung. Er ist eine bewusste Verpflichtung der Operation.

Unter Veränderung arbeiten

Die *Work* beginnt, und das Feld bleibt nicht still.

Der Pilot betritt die *Site*, erreicht die vorgesehene Position und beginnt die erste umkehrbare Handlung. Die Aufgabe erzeugt einen eigenen Strom von Information: gegenwärtiger Fortschritt, Raum an Bord, Schiffszustand und verbleibende *Work*. Gleichzeitig verändern sich System und *Chain* weiter. Die Operation muss beide Ströme beobachten, ohne einen von ihnen für das ganze Bild zu halten.

Eine neue Beurteilung sollte an Entscheidungspunkten stattfinden statt als dauerndes Gefühl. Vergleiche den sichtbaren Zustand mit den vorbereiteten Bedingungen, bevor ein weiterer Zyklus, ein weiterer Container, eine weitere Durchquerung oder ein weiteres Schiff gebunden wird. Hat sich der Zweck verändert? Ist die *Observation* der Rückkehr noch annehmbar? Erhöht der nächste Schritt die Konzentration über den Lagerplan hinaus? Besitzt die Operation weiterhin die benannten Menschen und die benannte *Reserve Capability*?

Der Abstand hängt von der *Work* ab. Dieses Buch legt keinen allgemeingültigen Zeitplan fest. Stattdessen wählt die Operation Momente, in denen eine Verpflichtung noch zurückgehalten werden kann. Eine Prü-

fung nach einem unumkehrbaren Schritt kann den Verlust festhalten, aber nicht die Wahl wiederherstellen, die zuvor bestand.

Veränderung muss durch Evidenz begrenzt bleiben. Behandle eine neue *Signature* als neue *Signature*, nicht als Beweis für einen Akteur. Ein fehlender Beobachter ist eine Lücke im *Ownership* der *Observation*, kein Beweis dafür, dass die Route versagt hat. Ein verändertes Signal einer *Connection* ist das gegenwärtige Signal, keine genaue Vorhersage ihres Kollapses. Jede Tatsache kann eine Stoppbedingung erfüllen, ohne zu einer Geschichte ausgeschmückt zu werden.

Diese disziplinierte Lesart hält zwei entgegengesetzte Impulse im Gleichgewicht. Die Dynamik sagt, die *Work* sei bisher erfolgreich gewesen und solle fortfahren. Alarm sagt, jede Veränderung mache die ganze Operation ungültig. Evidenz stellt eine engere Frage: Überschreitet diese Veränderung eine mit dem Zweck verbundene Grenze? Falls nicht, halte sie fest und fahre innerhalb des Plans fort. Tut sie es, führe die vorbereitete Handlung aus, bevor eine grössere Erklärung diskutiert wird.

Auch Fortschritt kann die Grenze verändern. Ein Frachtraum, der beim Eintritt nahezu leer war, trägt nun ein Ergebnis. Weitere Schiffe können hinzugekommen sein. Dieselbe *Connection* hat zusätzliche Durchquerungen getragen. Die verbleibende Ausbeute der *Site* kann im Verhältnis zum Aufwand für eine weitere Ladung gesunken sein. Selbst bei unverändertem äusserem Feld ist die eigene Spur der Operation gewachsen.

Erfolgreiche Zyklen beweisen deshalb nicht, dass der nächste klug ist. Sie sind Evidenz dafür, dass frühere Verpflichtungen unter früheren Bedingungen ein Ergebnis hervorgebracht haben. Das Fortfahren hängt vom gegenwärtigen Vergleich ab.

Innerhalb dieser Struktur kann die *Work* ruhig bleiben. Piloten müssen nicht jeden kleinen Zustand erzählen. An den gewählten Entscheidungspunkten benötigt der Verantwortliche der Aufgabe klare Veränderungen, der Beobachter das *Ownership* relevanter *Unknowns*, und alle

müssen wissen, wann die Operation von *Work* zu Freigabe übergeht. Das Modell ist gerade deshalb wertvoll, weil es Diskussionen bei verengter Aufmerksamkeit verringert.

Vor Gier lagern

Material an Bord eines zurückkehrenden Schiffs ist noch keine wiederhergestellte *Capability*.

Es ist bewegliche Konzentration. Bis es den vorgesehenen Ort erreicht und von der vorgesehenen Person verwendet werden kann, bleibt es Teil der *Exposure* der Operation. Der Satz «Wir haben die Ladung» kann deshalb zu früh kommen. Die Gruppe hat etwas erhalten, aber nicht zwingend den Zweck abgeschlossen.

Das vor dem Aufbruch vorbereitete Ziel begrenzt nun die Versuchung, weiterzuarbeiten. Kann das Ziel eine nützliche Ladung sauber aufnehmen, verdoppelt ein zweites volles Schiff ohne geklärten Bestimmungsort den Erfolg nicht. Es fügt eine Lagerfrage hinzu, während Route und Aufmerksamkeit gebunden bleiben. Lagere ein nutzbares Ergebnis, bevor die Spur nur deshalb vergrößert wird, weil die *Site* weitere Tätigkeit erlaubt.

Lagerung ist kein neutrales letztes Feld. Persönliches *Ownership* und jenes der *Corporation* erzeugen unterschiedliche Zugriffspfade. Ein *Character*, der etwas einlagern kann, kann sich von der Person unterscheiden, die es später entnehmen, installieren oder verteilen darf. Ein Container kann Ordnung schaffen, während seine Berechtigungen den Inhalt weniger nutzbar machen. Eine *Structure* kann Material enthalten, während der für seine Verarbeitung vorgesehene *Service* nicht verfügbar ist. Kapitel 12 untersucht diese Abhängigkeiten genau. In diesem Stadium muss die Operation nur beweisen, dass ihr Ergebnis den gewählten Pfad erreicht, statt «irgendwo in *Home*» zu verschwinden.

Auch Information benötigt Lagerung. Eine *Observation* der *Site* sollte mit ihrer *Source* und Zeit in die gemeinsame Aufzeichnung gelangen.

Veränderungen der *Chain*, die Nutzung einer *Connection* oder die Auflösung eines *Unknown* müssen die betreffende Aufzeichnung erreichen. Sie darf den früheren Zustand nicht länger als gegenwärtig darstellen. Bleibt die Route nur in persönlicher Erinnerung, schliesst das keine gemeinsame Operation ab.

Das Wort Gier ist hier auch dann nützlich, wenn kein Wert in ISK diskutiert wird. Gier ist der Wunsch, die sichtbare Gelegenheit offenzuhalten, nachdem der Zweck bereits erfüllt ist. Sie kann mehr Material, vollständigeres Wissen oder eine effizientere Nutzung der Reise anstreben. Der Impuls ist verständlich. Die Vorbereitung war aufwendig, die *Work* läuft und die Freigabe fühlt sich an, als werde die verbleibende Möglichkeit aufgegeben.

Doch ein Zweck, der nie wichtiger als die Gelegenheit werden kann, ist kein Zweck. Sobald das benannte Ergebnis eingelagert ist und die nächste Ladung ein neues Ziel, eine neue *Exposure* oder eine neue Zeitbedingung erzeugen würde, hat die gegenwärtige Operation ihr Ende erreicht. Eine weitere Operation kann sofort vorgeschlagen werden. Sie sollte mit einem eigenen Satz, gegenwärtigen *Observations* und Stoppbedingungen beginnen, statt die Erlaubnis vom vorherigen Erfolg zu erben.

Das erste nützliche Ergebnis kehrt zurück. Es erreicht den vorgesehenen Ort, und die verantwortliche Person bestätigt, dass es dem Zweck dienen kann. Diese kleine Handlung verändert die Entscheidung. Das Fortfahren wird nicht mehr benötigt, um bereits gesammelten Wert zu retten. Der nächste Schritt kann gewählt werden, ohne dass der Frachtraum zugunsten der *Site* argumentiert.

Nach Evidenz stoppen

Die Stoppbedingung gehört zum Plan; der Stopppentscheid gehört zum gegenwärtigen Feld.

Während der Operation wird sich irgendwann eine der vorbereiteten *Observations* verändern. Der Zweck kann erfüllt sein. Die Route kann über die angenommene Bedingung hinaus altern. Ein neues *Unknown* kann ohne Verantwortlichen bleiben. Das Lager kann seine verwaltete Grenze erreichen. Ein Pilot oder ein Schiff der *Reserve* kann nicht mehr verfügbar sein. Jede Tatsache kann unspektakulär bleiben. Sie muss wahr, zuordenbar und mit der vereinbarten Handlung verbunden sein. Evidenz besitzt eine *Source* und eine Zeit. «Die *Chain* fühlt sich alt an» kann nicht bestimmen, welcher Zweig die Rückkehr beeinflusst. «Jemand sagte, es habe eine Veränderung gegeben» zeigt nicht, ob der Bericht zum gegenwärtigen *Grid* gehört. «Wir sollten wahrscheinlich bald stoppen» benennt die Grenze nicht. Die Operation benötigt die kleinste Aussage, welche die Handlung tragen kann: Was wurde beobachtet, woher stammt es, wann wurde es beobachtet, welche Bedingung erfüllt es und wer trägt die Reaktion?

Nach dieser Aussage zu handeln, ist keine Panik. Die Handlung schliesst eine Entscheidung ab, die bei breiterer Aufmerksamkeit getroffen wurde. Sobald die Bedingung erfüllt ist, kündigt der Verantwortliche der Aufgabe an, dass keine neue Verpflichtung beginnt. Piloten beenden nur die im Plan bestimmten umkehrbaren Schritte, geben ihre Schiffe frei und folgen dem Rückweg, solange seine gegenwärtigen *Observations* ihn tragen. Ist die Route selbst die veränderte Bedingung, verwenden sie die vorbereitete Rückfallebene, statt eine Garantie zu improvisieren. Auf die Rückkehr folgt Lagerung. Auf die Lagerung folgt der Bericht. Dieser muss den Versuch weder feiern noch verurteilen. Er hält fest, ob der Zweck erfüllt wurde, was zurückkehrte und wo das Ergebnis jetzt liegt. Zudem nennt er veränderte *Observations*, verbleibende *Unknowns* und die Person, welche die nächste Prüfung trägt. Eine unvollständige Operation kann weiterhin einen wertvollen Bericht erzeugen, wenn er die nächste Entscheidung verkleinert.

Der Bericht muss die *Site* von der Operation unterscheiden. Eine *Site* kann verfügbar bleiben, nachdem die Gruppe stoppt. Das macht den Stopp nicht verfrüht. Die Operation endete, weil ihr eigener Zweck oder ihre eigene Grenze erreicht war. Umgekehrt schliesst das Verschwinden einer *Site* die *Work* nicht automatisch ab; Schiffe, Material und Aufzeichnungen müssen vielleicht noch zurückkehren.

Löse nach dem Abschluss die vorübergehenden Rollen. Ein beendeter Zweck benötigt keinen Verantwortlichen der Aufgabe mehr. Der Beobachter der Route schliesst entweder das betreffende *Handover* oder führt sie zur gewöhnlichen Pflege der *Chain* zurück. Schiffe werden entladen und verfügbar oder als pflege- beziehungsweise ersatzbedürftig benannt. So wird jene *Capability* wiederhergestellt, die eine einfache Nachricht mit dem Inhalt «erledigt» uneindeutig liesse.

Am Ende der Operation kann der Pilot auf eine vollständige Bewegung zurückblicken. Der Scanner fand eine *Site*. Der Zweck wählte sie aus. Vorbereitung machte den Eintritt vernünftig. *Observation* formte jede nächste Verpflichtung. Evidenz beendete die *Work*. Rückkehr, Lagerung und Bericht gaben frei, was die Operation gebunden hatte.

Die sichtbare Tätigkeit belegte nur die Mitte.



WORK IST NICHT DAS ENDE DER OPERATION

RÜCKKEHR · LAGERUNG · BERICHT TREFFEN WIEDER AUF DEN ZWECK

Abbildung AFH-B2-FIG-010 — Eine Resource Operation schliesst nach Rückkehr, Lagerung und Bericht. Eine Resource Operation endet erst, wenn Work, Rückkehr, Lagerung und Bericht wieder mit dem ursprünglichen Zweck verbunden sind.

Feldprinzip

Eine Ressourcenoperation beginnt vor der ersten Handlung auf ihrem Grid und endet, nachdem das letzte Ergebnis es verlassen hat.

Beginne mit Zweck und Ort. Fahre mit Vorbereitung, *Observation*, *Work* und erneuter Beurteilung fort. Stoppe danach, kehre zurück, lagere und berichte. Bewahre diese Reihenfolge, während jede Tätigkeit ihre eigenen Werkzeuge, ihr spezialisiertes Wissen, ihre sichtbaren Veränderungen und ihre umkehrbaren Schritte bestimmt. Verwende niemals ein allgemeines Modell, um ein tätigkeitsspezifisches *Unknown* zu verbergen.

Die Site aus der Eröffnungsszene sagte dem Piloten nie, was zu tun sei. Sie lieferte eine beobachtete Möglichkeit. Erst als die Gruppe diese Möglichkeit mit einer Notwendigkeit, einer Route, einem Ziel und einem Stopp verband, wurde die Operation kontrolliert. Ihr Erfolg bestand

nicht in der Zahl abgeschlossener Handlungen. Er bestand darin, ein nutzbares Ergebnis zu schaffen, ohne die umgebenden Verpflichtungen aus den Augen zu verlieren.

Dieses Ergebnis liegt nun in *Home*. Bevor es etwas wiederherstellen kann, muss Wert jedoch oft mehr von der *Chain* durchqueren, als ein Arbeitsschiff auf einmal tragen kann. Kapitel 11 folgt dieser Bewegung und fragt, wie viel Konsequenz gemeinsam reisen soll.

Kapitel 11 - Wert bewegen

Die Ladung war vor der Route bereit.

Container waren in der *Structure* angeordnet und ein Pilot hatte sich für ihren Transport gemeldet. Im Inventar fehlte nichts. Die Ladung passte in das verfügbare Schiff, und die erste *Connection* erschien noch immer in der *Chain*.

Einen Augenblick lang wirkte die Aufgabe wie eine einfache Umwandlung: Gegenstände in einem Hangar würden zu Gegenständen in einem anderen.

Dann wurde die Routenaufzeichnung geöffnet. Eine *Connection* hatte seit ihrer letzten *Observation* der Masse mehrere Arbeitsschiffe getragen. Eine weitere lag nicht mehr innerhalb des Zeitraums, den die Gruppe als verlässlich offen behandeln wollte. Laut Karte blieb das Ziel erreichbar, doch diese Aussage verband *Observations* unterschiedlichen Alters und Zwecks. Die Ladung war bereit. Ihre Bedingungen zur Wiederherstellung waren es nicht.

Bewegung exponiert Wert anders als *Work*. Die *Site* bindet keine Aufmerksamkeit mehr, doch das Ergebnis ist nun an Bord eines Schiffs konzentriert und von jeder *Connection* zwischen seinem gegenwärtigen und vorgesehenen Ort abhängig. Der Pilot muss nach mehr fragen, als ob das Schiff hindurchpasst. Bewegung verlangt deshalb einen Entscheid darüber, welche Konsequenz gemeinsam hindurchgehen soll, was die Durchquerungen für andere verbrauchen und wann eine Ankunft abgeschlossen ist.

Wert verändert Bewegung

Ladung erhält ihre Bedeutung durch das, was auf sie wartet.

Ein preiswerter Gegenstand kann das einzige lokale *Replacement* für eine *Capability* sein, welche die Gruppe verwenden will. Ein kostspieliger

Gegenstand kann einen Überschuss ohne unmittelbaren Zweck bilden. Material mag am Markt gewöhnlich und durch die gegenwärtige *Chain* dennoch schwer ersetzbar sein. Ein *Blueprint*, Werkzeug oder Bauteil kann *Work* ermöglichen, die sein Volumen weit übersteigt. Information besitzt überhaupt kein Frachtvolumen und kann dennoch bestimmen, wo die Bewegung enden soll.

Der Marktwert ist wichtig, wenn ein Verlust getragen werden muss, aber er ist nicht die ganze Konsequenz. Beschreibe die Ladung vor dem Beladen durch vier Fragen. Welche *Capability* stellt sie wieder her oder ermöglicht sie? Wie bald wird diese *Capability* benötigt? Welches weitere Exemplar oder welcher Ersatz bleibt auf jeder Seite der Route? Welche spätere Bewegung wird nötig, falls diese Ladung nicht ankommt?

Diese Fragen verändern die Form der Ladung. Angenommen, der *Structure* in *Home* fehlt ein Gegenstand für *Replacement*, während ausserhalb zwei Exemplare liegen. Beide gemeinsam zu bewegen, kann effizient wirken. Es setzt zugleich die gesamte Menge für *Replacement* in einem Schiff und einem Routenversuch aus. Wird zuerst ein Exemplar bewegt, entsteht wieder eine *Reserve* in *Home*; das zweite kann unter einer späteren *Observation* folgen. Die transportierte Gesamtmenge bleibt gleich, aber die grösste auf einmal exponierte Konsequenz wird kleiner.

Auch das Gegenteil kann gelten. Das Aufteilen einer Gruppe, die nur vollständig nützlich ist, kann mehrere Durchquerungen erzeugen, ohne ein nutzbares Teilergebnis zu schaffen. Die erste Ankunft würde Lager belegen, aber keine Funktion wiederherstellen. Jede spätere Bewegung trüge weiterhin die Konsequenz, die Gruppe zu vervollständigen. Ladung aufzuteilen ist deshalb nicht automatisch umsichtig. Aufgeteilt wird die operative Konsequenz, nicht bloss das Volumen.

Benenne diese Konsequenz im Zweck der Bewegung. «Nachschub bewegen» liefert keine Grundlage für die Reihenfolge. «Ein nutzbares *Replacement* in *Home* wiederherstellen und ein weiteres ausserhalb der Route bewahren» bestimmt eine Reihenfolge und einen Stopp. «Die

vollständige Eingabegruppe für die freigegebene Aufgabe bewegen» erklärt, weshalb eine Teilladung keine eigene Durchquerung rechtfertigen kann.

Ownership fügt eine weitere Bedeutung hinzu. Persönliches Material, das an einen Ort der *Corporation* bewegt wird, kann über einen anderen Zugriffspfad verfügbar werden. Material der *Corporation* in einem persönlichen Hangar kann für die erwarteten Nutzer unerreichbar werden. Das Schiff kann genau am angewiesenen Ort ankommen und den Zweck dennoch verfehlen, weil das Ziel als Ort statt als nutzbarer *Access* beschrieben wurde.

Wert verändert Bewegung, indem er verändert, was wiederherstellbar bleiben muss. Ein Gegenstand mit kleinem Volumen, der zukünftige *Work* ermöglicht, kann Vorrang verdienen. Ein teurer Gegenstand ohne gegenwärtigen Zweck kann eine Verzögerung rechtfertigen. Hier gibt es keine allgemeingültige Grenze für Ladung, weil sich Gruppen, Routen und der Zustand von *Replacement* unterscheiden. Die Methode fragt nach der grössten Konsequenz, welche die Gruppe hinter dasselbe ungelöste Ereignis legen will.

Dieser Vergleich gewinnt durch ein Manifest, das nach Nutzung geordnet ist. Trenne Gegenstände zur Wiederherstellung einer benannten *Capability* von gewöhnlichem Nachschub und von Gegenständen ohne gegenwärtiges Ziel. Diese Kategorien schaffen keine dauerhafte Rangfolge. Sie zeigen, weshalb jeder Gegenstand heute auf dieser Route reist. Ein Objekt kann von optional zu entscheidend wechseln, sobald das letzte lokale *Replacement* verbraucht wird, oder von dringend zu gewöhnlich, wenn eine andere Lieferung zuerst ankommt.

Das Manifest sollte auch Abhängigkeiten zwischen Einträgen benennen. Ein Werkzeug, das ohne seine passende Eingabe nutzlos ist, darf nicht als vollständiges Teilergebnis gelten. Eine Hülle für *Replacement* ohne die für ihre Funktion benötigten Gegenstände kann weniger *Capability* wiederherstellen, als der Name im Inventar vermuten lässt. Ein

kleines fehlendes Bauteil kann umgekehrt das Einzige sein, das jetzt bewegt werden muss, wenn alles andere bereits in *Home* wartet. Diese Beziehungen zu lesen, verhindert, dass Volumen und Preis die wirkliche Einheit des Werts verbergen.

Masse wird geteilt

Eine *Wormhole Connection* besitzt mehr als eine Massengrenze.

Es gibt eine maximale Masse für eine einzelne Durchquerung und eine grössere Grenze für die gesamte Lebensdauer der *Connection*. Diese Grenzen sind verschieden. Dass ein Schiff einmal hindurchpasst, bedeutet nicht, dass die *Connection* eine unbegrenzte Folge vergleichbarer Durchquerungen tragen kann. Die Durchquerung früherer Schiffe gehört zum Zustand, den spätere Piloten übernehmen.

Die offiziellen Informationen liefern grobe Massensignale statt eines genauen gemeinsamen Kontos. Die Operation darf aus diesen Signalen keine Genauigkeit machen, die sie nicht enthalten. Eine *Observation* der Masse hält fest, was sichtbar war, wann es sichtbar war und welche bekannten Bewegungen folgten. Sie kann nicht beweisen, dass keine unbeobachtete Durchquerung stattfand.

Masse wird dadurch zur gemeinsamen Konsequenz. Das Transportschiff aus der Eröffnungsszene ist nicht der einzige Nutzer der *Connection*. Scanner müssen vielleicht zurückkehren, Arbeitsschiffe können weiterhin ausserhalb liegen und eine spätere Operation kann vom selben Zweig abhängen. Jede Durchquerung verbraucht einen Teil eines veränderlichen Pfads, dessen verbleibende Kapazität nicht dem Piloten gehört, der ihn als Nächstes benutzt.

Eine Bewegungsaufzeichnung kann diesen gemeinsamen Kontext bewahren, ohne einen genauen Rest vorzugeben. Sie verbindet das zuletzt beobachtete Massensignal mit den darauffolgenden bekannten Durchquerungen und nennt die noch erwarteten Rückkehrbewegungen. Kann ein Pilot nicht jede Bewegung erklären, sagt die Aufzeichnung

das. Dieses *Unknown* kann verändern, welche Ladung die Gruppe als Nächstes bindet, obwohl es nicht beweisen kann, wie viel Kapazität verbleibt.

Gemeinsam bedeutet nicht, dass Bewegung auf allgemeine Zustimmung warten muss. Stattdessen müssen relevante Verpflichtungen zusammen sichtbar sein. Dazu gehören die Schiffe jenseits der *Connection* und die Bewegung, welche eine entscheidende *Capability* wiederherstellt. Welche bekannten Durchquerungen erfolgten seit dem letzten Massensignal? Wer übernimmt die aktualisierte *Observation* nach dieser Durchquerung?

Der Unterschied zwischen Einzel- und Gesamtmasse prägt auch den Rückkehrplan. Ein Schiff, das hinauskommt, benötigt eine tragbare Durchquerung zurück, falls die Operation seine Rückkehr erwartet. Auf die einseitige Bewegung von Ladung kann später ein leeres oder anderes Schiff folgen. Nur die beladene Durchquerung zu zählen, verdeckt die Hälfte der vorgesehenen Bewegung. Der Plan sollte die vollständige Folge beschreiben, welche die *Connection* tragen soll, ohne einen genauen, nicht beobachteten Rest zu behaupten.

Dieser Abschnitt lehrt keinen absichtlichen Kollaps einer *Connection*. Diese Nutzung gehört zu einem anderen operativen Zweck und einer späteren Stufe der Doktrin. Hier wird Masse als Bedingung gewöhnlicher Bewegung gelesen. Verändert sich das Signal oder macht bekannter Verkehr die ursprüngliche Folge zweifelhaft, bindet die Gruppe keinen weiteren Wert und stellt die *Observation* wieder her. Aus unvollständiger Evidenz wird keine Berechnung eines Kollapses improvisiert.

Die Konsequenz der Masse kann auf verschiedene Weise verkleinert werden, ohne eine allgemeingültige Regel für Schiff oder Ladung festzulegen. Weniger Durchquerungen können mehr gemeinsame Kapazität bewahren. Eine kleinere erste Konsequenz kann eine entscheidende *Reserve* wiederherstellen, bevor optionales Material bewegt wird. Das Warten auf eine neue Route kann den Druck von einer *Connection*

nehmen, die weitere Verpflichtungen trägt. Die Bewegung abzulehnen, kann richtig sein, wenn ihr Zweck schwächer ist als die Wiederherstellung, die sie stören könnte.

Die erste Frage an der *Connection* lautet deshalb nicht bloss: «Kann dieses Schiff springen?» Sie lautet: «Welche geplanten Bewegungen und verbleibenden Verpflichtungen verändert dieser Sprung?»

Zeit wird geteilt

Connections altern, während Ladung wartet und während sie bewegt wird.

Die gegenwärtige Information zum *Wormhole* verwendet *Reliable Lifetime* für den frühesten Zeitpunkt, an dem die *Connection* kollabieren kann. Sobald dieser garantierte Zeitraum beendet ist, kann *Expired* darauf hinweisen, dass ein Kollaps jederzeit eintreten kann. Keiner der Begriffe verspricht, dass die Route für den ganzen Plan bestehen bleibt. Die nützliche Tatsache ist begrenzt: Die *Connection* besitzt ein gegenwärtiges Signal der Lebensdauer, und die Bewegung verbraucht Zeit innerhalb dieses Signals.

Zeit beginnt vor dem Abdocken. Das Beladen, die Zuweisung eines Piloten, die Bestätigung des *Access* und die Prüfung der *Chain* geschehen, während *Connections* altern. Eine Route, die den Vorschlag getragen hat, trägt vielleicht nicht mehr die ganze Bewegung, wenn das Schiff bereit ist. Die *Observation*, welche die Aufgabe freigab, muss deshalb mit dem Augenblick verglichen werden, in dem die Verpflichtung tatsächlich beginnt.

Zeit wird geteilt, weil auch andere Pläne vom selben Zeitraum abhängen. Ein Scanner will vielleicht später zurückkehren. Eine Arbeitsgruppe bereitet ihre letzte Ladung vor. Ein Pilot ausserhalb von *Home* weiss möglicherweise noch nicht, dass die Bewegung verändert, wie die Gruppe den Zweig liest. Eine alternde *Connection* ist nicht der Timer

eines Piloten. Der Zeitraum ist eine gemeinsame Bedingung mehrerer möglicher Wiederherstellungen.

Eine von der Gruppe erfundene genaue Frist macht die zugrunde liegende Lebensdauer nicht genau. Interne Grenzen bleiben nützlich, wenn sie ehrlich beschrieben werden. Neue Bewegungen von Wert kann die Gruppe nach einer bestimmten Bedingung der *Observation* verweigern oder vor der nächsten Ladung eine neue Routenprüfung verlangen. Eine solche Regel bestimmt die eigene Verpflichtung. Die *Connection* kann sie nicht vorhersagen.

Die vollständige Bewegung benötigt Zeit auf beiden Seiten jeder Durchquerung. Ein Pilot muss sich nähern, hindurchgehen, ausrichten, fortfahren und die Folge vielleicht wiederholen. Am Ziel muss *Access* funktionieren und die Ladung angenommen werden. Eine erforderliche Rückkehr des Schiffs bedeutet, dass die Routenabhängigkeit nicht beim Entladen endet. Nur den Augenblick der Durchquerung zu planen, verdeckt die Zeit, in der Wert zwischen nutzbaren Zuständen verbleibt.

Die Tiefe der *Chain* vervielfacht diese Vergleiche. Jede *Connection* besitzt ihre eigene *Source*, *Observation* und ihr eigenes Alter. Eine als «gegenwärtig» beschriebene Route kann weiterhin einen Zweig enthalten, dessen Evidenz die Konsequenz der Ladung nicht mehr trägt. Für den Zweck zählt die schwächste Aussage mehr als die durchschnittliche Frische der Karte.

Das Transportschiff aus der Eröffnungsszene trifft genau auf dieses Problem. Jede *Connection* ist eingezeichnet, doch eine *Observation* der Lebensdauer hat die Arbeitsgrenze der Gruppe überschritten. Die Route wird nicht für geschlossen erklärt, und es wird keine Gefahr erfunden. Der Bewegung fehlt lediglich die Evidenz, um diesen Wert jenseits des Zweigs zu binden. Die Ladung bleibt eingelagert, während ein benannter Pilot die *Observation* wiederherstellt oder eine andere Route bestimmt.

Warten kann ein aktiver Transportentscheid sein. Es schützt Wert, vermeidet weitere Konsequenz für die Masse und erlaubt, die Route anhand gegenwärtiger Evidenz auszuwählen. Bewegung gelingt nicht bloss deshalb, weil sie rasch beginnt.

Exposure aufteilen

Sobald die Route lesbar ist, entscheide, was gemeinsam exponiert werden soll.

Aufgeteilt werden können Ladung, Zeit, Schiffe, Ziele oder Verantwortung. Eine Aufteilung sollte nicht bloss den Anschein von Vorsicht erzeugen. Jeder Teil benötigt einen nützlichen Zweck, eine vollständige Bewegung und eine eigene Stoppbedingung.

Beginne mit *Replacement*. Ist nach Verlust oder Verzögerung einer Ladung kein Exemplar einer benötigten *Capability* mehr auf einer Seite verfügbar, ist die Konzentration hoch. Bewege genug, um eine nutzbare *Reserve* wiederherzustellen, ohne automatisch die ganze *Reserve* an Bord zu nehmen. Bildet die Ladung eine unteilbare Gruppe, erkenne diese Gruppe als Einheit der Konsequenz und prüfe, ob die gegenwärtige Route sie überhaupt tragen soll.

Betrachte danach die Reihenfolge. Die erste Ladung kann enthalten, was eine spätere Verzögerung tragbar macht: ein nutzbares *Replacement*, ein benötigtes Werkzeug oder jenes Teil, das eine blockierte Funktion löst. Optionale Menge kann unter einem neuen Vergleich folgen. Durch diese Reihenfolge wird die frühe Ankunft zur verringerten *Exposure* statt bloss zu einem kleineren Teil des Manifests.

Zeit lässt sich aufteilen, indem eine Bewegung geschlossen wird, bevor die nächste freigegeben wird. Der Pilot kehrt zurück oder die empfangende Seite bestätigt die Lagerung. Die Aufzeichnung der *Chain* nimmt die bekannte Durchquerung auf; verbleibende Ladung wird anhand der neuen Signale für Masse und Lebensdauer neu beurteilt. Eine erfolgreiche erste Durchquerung erteilt der zweiten keine Erlaubnis.

Sie verändert die Evidenz, auf deren Grundlage die zweite entschieden werden muss.

Ziele können Konzentration verringern, wenn sie einer wirklichen Nutzung entsprechen. Jedes *Replacement* in einer *Structure* zu bewahren, kann das Zählen vereinfachen und zugleich ein einzelnes Ergebnis für den Vermögenswert folgenreicher machen. Gegenstände ohne *Ownership* oder Inventaraufzeichnung zu verstreuen, erzeugt einen anderen *Failure*. Verteilung ist nützlich, wenn jeder Ort einen benannten Zweck, einen Zugriffspfad und eine für seinen Zustand verantwortliche Person besitzt.

Verantwortung sollte nicht in Fragmente aufgeteilt werden, die niemand schliessen kann. Eine Person trägt den Zweck der Bewegung. Eine andere kann die *Observation* der Route tragen, und ein Empfänger kann Ankunft und *Access* bestätigen. Ihre *Handovers* müssen ausdrücklich sein. Der Pilot darf nicht erst nach der Durchquerung entdecken, dass der Empfänger annahm, jemand anders könne Ladung der *Corporation* annehmen.

Failure sollte nur aufgeteilt werden, wenn das Ergebnis wirklich unabhängig ist. Zwei Schiffe, die nahezu gleichzeitig derselben alten Route folgen, können die Ladung aufteilen, ohne die Konsequenz der Route zu teilen. Ein Versand zu Zielen, die vom selben nicht verfügbaren *Access* abhängen, kann derselbe *Failure* verdoppelt werden. Dagegen schafft ein nutzbares, bestätigtes *Replacement* vor dem zweiten Aufbruch eine wirkliche Grenze: Eine spätere Verzögerung entfernt die bereits wiederhergestellte *Capability* nicht mehr.

Die Gruppe legt ausserdem in ihrer eigenen Sprache eine maximale gleichzeitige Konsequenz fest. Die einzigen zwei Exemplare einer *Capability* können getrennt reisen. Eine weitere Ladung kann warten, bis die erste nutzbar wird. Optionales Material kann eingelagert bleiben, während Menschen jenseits desselben alternden Zweigs sind. Dies

sind örtliche Entscheide, keine allgemeingültigen Zahlen. Ihre Stärke stammt aus der Konsequenz, die sie benennen.

Das Aufteilen der *Exposure* wirkt oft langsamer, weil es Augenblicke einfügt, in denen die Gruppe stoppen kann. Diese Augenblicke sind der Zweck. Effizienz, die jeden sauberen Stopp entfernt, überträgt den Entscheid an den ersten *Failure*.



Abbildung AFH-B2-FIG-011 — Geteilte Bewegung kann *Capability* bewahren. Geteilte Bewegung kann *Capability* bewahren, wenn eine teilweise Ankunft nützt und kein einzelnes Ereignis die gesamte Folge trägt.

Ankunft ist unvollständig

Das Transportschiff dockt an. Die Ladung bleibt an Bord.

Auf der Karte ist die Bewegung beendet. Operativ hat der Wert einen neuen Container, aber noch keinen zwingend nutzbaren Zustand erreicht. Auf die Ankunft müssen Empfang, Platzierung, Bestätigung des *Access* und eine Aufzeichnung dessen folgen, was die Route getragen hat.

Der Empfang beantwortet, ob das vorgesehene Ziel wie geplant besteht. Kann der Pilot die Ladung dort ablegen? Ist der Kontext des empfangenden *Character* oder der *Corporation* korrekt? Besitzt die für die nächste Nutzung verantwortliche Person den benötigten *Access*? Eine Bewegung in den falschen Pfad des *Ownership* kann zugänglichen Wert in ein Berechtigungsproblem verwandeln, ohne seinen physischen Ort zu verändern.

Die Platzierung verbindet Ladung mit Zweck. Ein *Replacement* gehört dorthin, wo es ersetzen kann. Eine Eingabe für *Manufacturing* gehört zum freigegebenen Pfad der Aufgabe, nicht in einen undifferenzierten Haufen, den später jemand identifizieren soll. Material für einen zukünftigen *Exit* sollte nicht mit Bestand vermischt werden, der in *Home* bleiben soll. Je klarer das Ziel seine Funktion zeigt, desto weniger Wissen muss unter Druck wiederaufgebaut werden.

Die Inventaraufzeichnung sollte sich nun in denselben Begriffen wie das Manifest verändern. Sie nennt, was ankam, den Kontext des *Ownership*, den Ort, die vorgesehene Nutzung und die für die nächste Handlung verantwortliche Person. Hat die Bewegung einen anderen Ort unter seine angenommene *Reserve* gebracht, bleibt diese Konsequenz ebenfalls sichtbar. Transport hat beide Seiten verändert. Nur das Ziel aufzuzeichnen, liesse die *Source* fähiger erscheinen, als sie geworden ist.

Die Bestätigung schliesst das *Handover* des Empfängers. «Angedockt» meldet das Schiff. «Am benannten Ort empfangen, für den benannten Nutzer sichtbar» meldet das Ergebnis. Scheitert der *Access*, wird die Ladung nicht stillschweigend als abgeschlossen erklärt. Der Empfänger hält das *Unknown* fest, benennt die Person, die es auflösen kann, und verhindert die Wiederholung desselben Pfads durch die nächste Ladung, bis das Ziel nutzbar ist.

Auch die Aufzeichnung der *Chain* verändert sich. Bekannte Durchquerungen gehören zum Massensignal, das ihnen vorausging. Die Zeit der

Bewegung ist für spätere Entscheide zur Lebensdauer wichtig. Kehrt der Pilot über einen anderen Zweig zurück, muss die tatsächlich verwendete Route auf der *Shared Map* erscheinen. Eine frühere Aussage, die zu alt geworden ist, darf nicht als gegenwärtiger Plan dargestellt bleiben.

Gib zuletzt Schiff und Menschen frei. Soll das Transportschiff zurückkehren, bleibt die Bewegung offen, bis sein Wiederherstellungspfad geschlossen oder übergeben ist. Bleibt es am Ziel, ist seine *Capability* mit der Ladung umgezogen; das Inventar von *Home* darf es nicht weiter als lokale *Reserve* zählen. Der Beobachter der Route bestätigt, wer nun den betreffenden Zweig trägt. Der Empfänger bestätigt, was noch bewegt werden muss.

Erst dann kann die Gruppe das Ergebnis mit dem Zweck vergleichen. Sie hat die benannte *Capability* wiederhergestellt, ein *Replacement* auf der anderen Seite bewahrt und die Nutzung der *Connection* aufgezeichnet. Optionale Ladung bleibt an ihrem Ausgangsort. Dieser Rest ist kein *Failure*. Er wird zu einer späteren Entscheidung, an der weniger Konsequenz hängt.

Die Ladung aus der Eröffnungsszene war vor der Route bereit. Weil die Gruppe Bereitschaft nicht mit Erlaubnis verwechselte, liess sie gegenwärtige Evidenz über die Bewegung entscheiden. Indem sie Andocken nicht mit Abschluss verwechselte, machte sie aus transportierten Gegenständen verfügbare *Capability*.

Feldprinzip

Bewege Konsequenz, nicht bloss Ladung.

Beschreibe Wert durch die wiederhergestellte *Capability*, den Zustand seines *Replacement* und den am Ziel benötigten *Access*. Halte die Masse einer einzelnen Durchquerung vom gemeinsamen Zustand der Gesamtmasse einer *Connection* getrennt. Handle *Reliable Lifetime* als begrenztes Signal, nicht als genaue Reservierung für die Rückkehr.

Begrenze, was hinter demselben Ereignis verloren gehen oder verspätet werden kann. Teile *Exposure*, wenn eine Teilladung nützlich wird. Lass eine unteilbare Gruppe nur dann zusammen, wenn die gegenwärtige Route die gesamte Konsequenz rechtfertigt. Beurteile jede neue Ladung erneut, nachdem die vorausgehende Bewegung Masse, Zeit und Verteilung verändert hat.

Die Ladung aus der Eröffnungsszene verlor beim Warten nicht an Wert. Das Warten schützte den Zweck, bis die Route ihn tragen konnte. Nach der Bewegung machten Empfang und *Access* den Wert für *Home* wirklich.

Bewusste Verzögerung bewahrte die Wahl, welche verfrühte Effizienz verbraucht hätte.

Wert benötigt nun einen Ort zum Bleiben und einen Weg, zu *Replacement* zu werden. Kapitel 12 folgt ihm in Lagerung, *Ownership* und *Industry*. Dort kann ein Gegenstand innerhalb der *Structure* und für die davon abhängige *Work* dennoch nicht verfügbar sein.

Kapitel 12 - Lagerung und *Replacement*

Das *Replacement* war sichtbar, aber nicht nutzbar.

Sein Name erschien genau dort im Inventar der *Structure*, wo ihn der Bewegungsbericht verortet hatte. Der darauf angewiesene Pilot konnte die *Structure* erreichen und die betreffende Division der *Corporation* öffnen. Der Gegenstand lag zwischen den übrigen Beständen und schien zu beweisen, dass *Home* vorbereitet war.

Als der Pilot ihn entnehmen wollte, verschwand diese Vorbereitung. Der *Character* konnte den Inhalt sehen, aber nicht entfernen, was die nächste Aufgabe benötigte. Die Person, die alles angeordnet hatte, war offline. Ein anderer Pilot trug breite Verantwortung für die *Structure*, aber nicht den bestimmten *Access* innerhalb der *Corporation*, der den Gegenstand freigeben konnte. Nichts war verloren gegangen. Nichts hatte sich bewegt. Dennoch fehlte die *Capability* in dem Augenblick, in dem sie gezählt wurde.

Lagerung kann ein Objekt bewahren und zugleich die Bedingungen seiner Nutzung verbergen. Ein Hangar, ein Container oder eine *Structure* beantwortet, wo etwas liegt. Der Ort allein beantwortet nicht, wem es gehört, wer es bewegen kann, welcher *Service* es umwandeln kann oder was geschieht, wenn sich der umgebende Zustand der *Structure* verschlechtert. *Replacement* besteht nur, wenn diese Bedingungen von der Notwendigkeit bis zur Handlung verfolgt werden können.

Lagerung ist Risiko

Ansammlung gibt *Home* Ausdauer und einem *Failure* Konzentration.

Derselbe Ort, der Materialien leicht auffindbar macht, kann mehrere *Capabilities* hinter einem Zugriffspfad und einem möglichen Ergeb-

nis der *Structure* platzieren. Derselbe Container, der entscheidenden Bestand von gewöhnlicher Ausgabe trennt, kann von einer Person abhängen, die niemand benannt hat. Dieselbe Gesamtsumme im Inventar, welche die Gruppe beruhigt, kann nutzbare Gegenstände, reservierte Eingaben, persönliches Eigentum und bereits einer anderen Aufgabe versprochenes Material vermischen.

Das Risiko der Lagerung ist damit kein Argument gegen das Lagern von Wert. Der Begriff benennt die operative Form des gewählten Orts. Beschreibe diese Form durch Zweck, *Ownership*, *Access*, Konzentration und Wiederherstellung.

Der Zweck erklärt, weshalb der Bestand bleibt. Eine *Reserve* für *Replacement* besitzt eine andere Freigaberegeln als Material, das auf Bewegung wartet. Eingaben für *Industry* dürfen nicht als freier Bestand gezählt werden, wenn eine freigegebene Aufgabe bereits von ihnen abhängt. Eine Ausgabe ohne vorgesehenen Nutzer kann Wert besitzen, ist aber noch keine *Capability* für *Replacement*. Klare Zwecke verhindern, dass derselbe Gegenstand zweimal versprochen wird.

Ownership bestimmt, wessen Vermögenspfad das Objekt enthält. Vermögenswerte im persönlichen Besitz und solche der *Corporation* sind nicht austauschbar, nur weil sie sich in derselben *Structure* befinden. Der Hangar eines *Character* kann für die Gruppe unzugänglich bleiben. Eine *Division* der *Corporation* kann für einen *Character* ohne die bestimmte Rolle oder Berechtigung unbrauchbar bleiben. Das Inventar sollte den Kontext der Bestände nennen, statt nur eine gemeinsame Summe zu melden.

Access fragt nach möglichen Handlungen. Einen Gegenstand zu sehen, beweist nicht die Fähigkeit, ihn zu entnehmen. Einlagern beweist keine spätere Entnahme. Eine *Structure* zu verwalten, ist nicht dieselbe Berechtigung wie das Bewegen von Eigentum einer *Corporation*. Die genauen Rechte sollten mit einem unwichtigen Gegenstand geprüft

werden, bevor entscheidende Bestände von ihnen abhängen, wie Kapitel 7 festlegte.

Konzentration fragt, was unter einem *Failure* aus dem Plan verschwindet. Die Grenze der Wiederherstellung im *Wormhole Space* aus Kapitel 3 macht diese Frage zu einer Lagerfrage. Inventar, das jetzt in einer *Structure* sichtbar ist, besitzt keine garantierte spätere Wiederherstellung. Das Feldhandbuch macht aus dieser Grenze kein Kampfszenario. Es verwirft damit die Annahme, Lagerung in einer *Structure* erzeuge automatisch eine spätere Wiederherstellung.

Wiederherstellung fragt, wie die Gruppe diese Konzentration vor einer Krise erkennt und verringert. Einige Gegenstände für *Replacement* können in verschiedenen Kontexten des *Ownership* oder an verschiedenen Orten bleiben. Entscheidendes Wissen über Bestände muss die Person überdauern, die sie angeordnet hat. Material ohne gegenwärtigen Zweck kann den Ort verlassen, statt die Ansammlung zu vertiefen. Die richtige Verteilung hängt von der Gruppe ab, doch jeder Ort benötigt einen Verantwortlichen und ein vorgesehenes Ende.

Ein Inventar wird ehrlicher, wenn es Zustände trennt, statt alle Namen zu einer Summe zu addieren. Eine nutzbare *Reserve* ist vorhanden, zugänglich und noch nicht gebunden. Reservierter Bestand gehört zu einer benannten Aufgabe oder Bewegung. Eingehender Bestand hängt noch von einer Route oder einem Empfänger ab. Nicht bestätigter Bestand wurde gemeldet, aber nicht durch seinen Zugriffspfad geprüft. Überschuss besitzt keinen gegenwärtigen operativen Zweck. Diese Bezeichnungen lassen sich ohne Preise oder allgemeingültige Mengen pflegen. Sie zeigen, welcher Teil der Ansammlung jetzt eine Notwendigkeit beantworten kann.

Das Alter gehört neben den Zustand. Eine Aufzeichnung aus dem vergangenen Monat kann korrekt beschreiben, was eingelagert wurde, und dennoch verfehlen, was verbleibt. Der Verantwortliche bestimmt die nächste Prüfung anhand der Konsequenz eines Irrtums. Entschei-

dendes *Replacement* benötigt einen kürzeren Weg zur Bestätigung als optionaler Überschuss. Trägt niemand diese Bestätigung, stufe die Aussage zurück, statt altes Vertrauen weiterzuführen.

Das sichtbare *Replacement* aus der Eröffnungsszene versagte am *Access*, nicht an der Menge. Einen Gegenstand zu zählen, war korrekt und dennoch irreführend. Ein nützliches Inventar muss die Bedingungen zählen, welche den Gegenstand freigebbar machen.

Ownership formt Access

Die Lagerung einer *Corporation* verbindet *Ownership* von Vermögenswerten mit der Gestaltung von Rollen.

Das offizielle Rollenmodell unterscheidet Fähigkeiten, statt eine allgemeingültige Vertrauensstufe zu vergeben. *Query* erlaubt einem *Character*, die betreffenden Inhalte eines Hangars der *Corporation* zu sehen. *Take* erlaubt dort die Entnahme von Gegenständen. *Container Take* betrifft die Entnahme der Container selbst. *Industry Roles* und Fähigkeiten zur Verwaltung von *Structures* beantworten andere Fragen. Diese Berechtigungen dürfen nicht als Synonyme behandelt werden.

Dieser Abschnitt bildet die Rollenliste nicht ab. Er soll zeigen, weshalb der Pfad zu einem *Replacement* brechen kann, obwohl jede Person befugt wirkt. Sichtbarkeit besass der Pilot aus der Eröffnungsszene. Eine andere Person konnte Teile der *Structure* verwalten. Keine der Tatsachen gewährte die fehlende Handlung am eingelagerten Gegenstand.

Modelliere den Pfad aus der Sicht des Nutzers. Die Person muss die *Structure* erreichen und den richtigen Kontext des *Ownership* öffnen. Danach muss sie den Bestand erkennen, ihn entnehmen oder verwenden und jedes daraus entstehende Objekt für die nächste Aufgabe zugänglich platzieren. Jedes Verb kann von einer anderen Berechtigung abhängen. Prüfe den vollständigen Pfad, statt zu Beginn eine eindrucksvolle Rolle zu bestätigen.

Das Prinzip der geringsten Rechte bleibt wichtig. Jeder pflegenden Person jede *Corporation Role* zu geben, würde den Zugriffspfad leichter erinnerbar machen und zugleich die Konsequenzen eines Irrtums oder Aufbruchs stark vergrössern. Vergib nur die von der Verantwortung benötigten Fähigkeiten. Halte fest, wer sie besitzt, und bewahre einen ausdrücklichen Eskalationspfad für Handlungen, die getrennt bleiben. Trennung benötigt auch Verfügbarkeit. Kann nur ein *Character* entscheidenden Bestand freigeben, erbt die *Reserve* den Zeitplan dieser Person. Ein zweiter Pfad muss nicht jede Befugnis verdoppeln. Er kann aus einer geprüften Stellvertretung für die betreffende Division, einer kleineren zugänglichen *Reserve* oder einem Verfahren bestehen, das ein *Replacement* bewegt, bevor der einzige Berechtigte nicht mehr verfügbar ist. Die Gestaltung sollte dazu passen, wie lange die *Capability* warten kann.

Container verdienen dieselbe Prüfung der Verben. Sie können Bestand ordnen, einen Zweck kennzeichnen und versehentliche Vermischung verringern. Sie können auch ein weiteres Objekt hinzufügen, dessen *Access* oder Entnahme anders funktioniert als bei den darin enthaltenen Gegenständen. Leite das Verhalten eines Containers nicht aus der Fähigkeit ab, seinen Namen zu sehen. Prüfe die Handlung, welche die Operation tatsächlich benötigen wird.

Persönlicher Bestand besitzt einen einfacheren Berechtigungspfad, aber einen engeren sozialen Pfad. Die direkte Nutzung bleibt seinem Eigentümer vorbehalten. Ist der Eigentümer abwesend, kann die Gruppe keinen *Access* annehmen. Einen persönlichen Gegenstand «unser *Replacement*» zu nennen, schafft eine gemeinsame Erwartung ohne gemeinsame Kontrolle. Er kann dem Plan weiterhin dienen, wenn Eigentümer und Verfügbarkeit ehrlich benannt werden.

Access sollte geprüft werden, bevor ein Träger von Rollen aufbricht, seine Verantwortung verändert oder ein entscheidendes Verb von einer Person abhängig lässt. Die Prüfung beginnt bei den eingelagerten Zwecken statt beim vollständigen Katalog der Rollen. Bestimme *Reserves*, die

nur dieser *Character* freigeben kann, und Aufgaben, die nur diese installierende Person fortsetzen kann. Finde Container, die sichtbar, aber unbeweglich würden. Die Antworten führen zu einer engen Übertragung oder einer verringerten Abhängigkeit, solange der gegenwärtige Berechtigte das Ergebnis noch prüfen kann.

Eine Übertragung ist nicht mit der Zuweisung einer Rolle abgeschlossen. Der empfangende *Character* führt denselben Pfad mit einem unwichtigen Testobjekt aus und bestätigt das Ergebnis. Danach kann der frühere Berechtigte aus der Abhängigkeit entfernt werden, ohne die undokumentierte Rückfallebene zu bleiben, welche später alle anzusprechen erwarten.

Sobald der fehlende Pfad für *Take* aus der Eröffnungsszene erkannt ist, besitzt die Gruppe zwei Aufgaben. Sie gibt das unmittelbare *Replacement* durch eine befugte Person frei. Zudem verändert sie die Gestaltung der Lagerung, damit die nächste Notwendigkeit nicht vom Wiederaufbau derselben Berechtigung abhängt. Den Gegenstand wiederherzustellen, löst das heutige Problem. Seinen Pfad sichtbar zu machen, stellt die Vorbereitung wieder her.

Industry braucht Kontinuität

Replacement kann als Material statt als Gegenstand beginnen.

Manufacturing wandelt Materialien und Komponenten mithilfe eines *Blueprint* in einer geeigneten Einrichtung in eine Ausgabe um. Die Oberfläche *Industry* trennt *Blueprint*, Eingabematerialien, gewählte Einrichtung, Orte für Eingabe und Ausgabe, Dauer und Kosten. Optimierung bleibt ausserhalb dieses Kapitels. Sein Thema ist die dadurch offengelegte Kette von Abhängigkeiten.

Ein Plan für *Manufacturing* benötigt mehr als eine ausreichende Gesamtmenge an Material. Die Eingaben müssen am Ort und im Kontext des *Ownership* liegen, welche die Aufgabe verwenden kann. Der *Blueprint* muss dem *Character* zur Verfügung stehen, der die Aufgabe

einrichtet. Die gewählte Einrichtung muss den richtigen *Service* anbieten. Der Ort der Ausgabe muss für die Person zugänglich sein, welche das Ergebnis empfängt und verteilt. Die Zeit muss zu dem Augenblick passen, in dem *Replacement* benötigt wird.

Jede dieser Bedingungen kann einzeln wahr sein, während die ganze Kette versagt. Materialien können in einem persönlichen Hangar liegen, obwohl die freigegebenen Eingaben als Bestand der *Corporation* gezählt wurden. Ein *Blueprint* kann sichtbar, aber für die installierende Person nicht verfügbar sein. Ein *Service* kann in der *Structure* eingebaut, aber offline sein. Eine Ausgabe kann an einem Ort abgeschlossen werden, auf den der vorgesehene Nutzer nicht zugreifen kann. Eine Summe im Inventar kann diese Brüche der Kontinuität nicht offenlegen.

Schreibe die Aufgabe als Pfad. Beginne bei Notwendigkeit, *Blueprint* und Eingabegruppe. Benenne danach *Ownership*, Ort der Eingabe, installierende Person und Einrichtung. Schliesse mit *Service*, Ort der Ausgabe, Empfänger und vorgesehener Nutzung. Jeder Schritt besitzt gegenwärtige Evidenz und einen Verantwortlichen. Der Pfad ist kein dauerhaftes Rezept; er ist der operative Zustand dieses *Replacement*.

Eine Rückfallebene sollte die Abhängigkeit verändern, statt dieselbe Annahme zu wiederholen. Kann der lokale *Service* die Aufgabe nicht tragen, ist die Bewegung derselben Eingaben zu einem anderen nicht unterstützten Ort keine Alternative. Stattdessen kann die Gruppe einen fertigen Gegenstand sichern, eine Einrichtung mit geprüftem *Access* wählen oder eine Zeit ohne diese *Capability* annehmen. Die richtige Wahl hängt von der gegenwärtigen *Chain* und vom Zweck ab. Die Rückfallebene muss einen anderen funktionierenden Pfad benennen.

Der Pfad sollte auch jenen Entscheid offenlegen, der sich nicht sauber rückgängig machen lässt. Materialien können flexibel bleiben, solange sie in zugänglicher Lagerung liegen, und nach der Einrichtung der Aufgabe gebunden sein. Eine Bewegung kann eine Eingabe für eine Einrichtung verfügbar machen und sie zugleich aus einer anderen

Reserve entfernen. Die Freigabe unmittelbar vor dieser Veränderung zu platzieren, bewahrt die Möglichkeit, auf neue Evidenz zu reagieren, ohne den Plan zwischen Zuständen festzusetzen.

Hier wird die Unterscheidung zwischen reserviertem und verfügbarem Material wichtig. Sobald Eingaben einer freigegebenen Aufgabe zugeordnet sind, verspricht ihre Zählung als allgemeine *Reserve* sie zweimal. Kennzeichne die Reservierung vor der Einrichtung und nicht erst, nachdem ein anderer Pilot denselben Bestand nutzen will. Wird die Aufgabe abgebrochen, muss die Aufzeichnung festhalten, welche nutzbaren Materialien verbleiben, statt anzunehmen, der ursprüngliche Plan habe sich einfach umgekehrt.

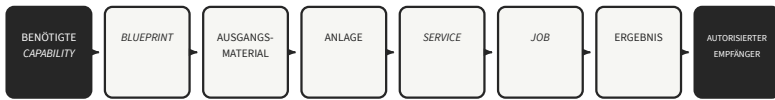
Industry trägt auch Informationsabhängigkeiten. Der Bedarf an einer Ausgabe kann bekannt sein, ohne dass die spezialisierten gegenwärtigen Einzelheiten für ihre effiziente oder korrekte Herstellung bekannt sind. Dieses Buch füllt diese Lücke nicht mit alten Zahlen. Aktuelle, tätigkeitsspezifische Information beschafft der Verantwortliche der Aufgabe dort, wo sie wichtig ist. Aufgezeichnet wird nur der daraus entstandene Entscheid, den die gemeinsame Operation benötigt.

Kontinuität umfasst die Menschen um den Timer. Eine Aufgabe einzurichten und zu gehen, weist ihren Empfang niemandem zu. Jemand muss wissen, wann geprüft werden soll, welcher Zustand die Erwartung unterbrechen kann und wohin die Ausgabe danach gelangt. Wird die installierende Person nicht verfügbar sein, geschieht das *Handover*, solange Aufgabe und Zweck noch lesbar sind.

Das *Replacement* aus der Eröffnungsszene war bereits hergestellt, doch sein *Failure* beim *Access* legt dieselbe Lektion offen. Produktion ist keine vollständige *Capability*. Die Kette endet erst, wenn der vorgesehene Nutzer die Ausgabe erhalten kann und die Aufzeichnung der *Reserve* sie erkennt.

REPLACEMENT · VOLLSTÄNDIGE KETTE

BEDARF BIS EMPFANG



GEPRÜFTER ACCESS



EIN VORHANDENER GEGENSTAND IST NOCH KEINE NUTZBARE CAPABILITY

JEDER KNOTEN BLEIBT EINE EIGENE BEDINGUNG

Abbildung AFH-B2-FIG-012 — Replacement ist eine vollständige Capability-Kette. Replacement wird erst zur Capability, wenn Bedarf, Ausgangsmaterial, Service, Ergebnis und autorisierter Empfänger verbunden bleiben.

Services können pausieren

Industry hängt von gepflegter Infrastruktur ab.

Dass ein geeigneter *Service* einer *Structure* heute besteht, macht ihn nicht zu einer dauerhaften Eigenschaft von *Home*. *Service Modules* verbrauchen *Fuel*. Ein betreffender *Service* kann offline genommen werden und dadurch verbundene Aufgaben in *Industry* pausieren. Das Entfernen des *Service* oder der Verlust der *Structure* kann *Work* abbrechen. Die genauen Konsequenzen einer bestimmten Aufgabe sollten vor der Verpflichtung geprüft werden; die allgemeine Abhängigkeit genügt, um den Plan zu formen.

Der Unterschied zwischen Pause und Abbruch ist wichtig. Eine Pause verzögert die erwartete Ausgabe und kann eine Frist für *Replacement* brechen, selbst wenn die Aufgabe später fortgesetzt wird. Ein Abbruch kann die Gruppe ohne das geplante Ergebnis zurücklassen, während sich nicht jede Konsequenz der Eingaben rückgängig machen lässt.

Keines der Ergebnisse darf auf «die *Structure* besteht noch» reduziert werden.

Verbinde die Aufgabe vor ihrer Einrichtung mit der Infrastrukturaufzeichnung aus Kapitel 8. Welchen *Service* benötigt sie? Wer pflegt den Pfad des *Fuel*? Was wurde zuletzt über den Zustand des *Service* beobachtet, und wann muss diese *Observation* erneuert werden? Wer kann die Aufgabe verschieben, falls der Zeitraum der Unterstützung zweifelhaft ist? Welche Alternative stellt die *Capability* wieder her, wenn der *Service* die *Work* nicht tragen kann?

Der Zeitraum der Unterstützung sollte mehr als die angezeigte Dauer der Aufgabe umfassen. Eingaben müssen vorbereitet, die Aufgabe muss eingerichtet, die Ausgabe ausgeliefert werden und jemand muss bis zum Empfang verantwortlich bleiben. Ein *Service*, der nur bis zu einem optimistischen Abschlusszeitpunkt bestehen soll, deckt den vollständigen Pfad des *Replacement* nicht ab.

Auch das Vertrauen in einen *Service* besitzt eine *Source*. Das eingebaute Modul, sein gegenwärtiger Onlinezustand und der Plan seines Verantwortlichen sind verschiedene *Observations*. Das sichtbare Modul beweist nicht, dass es online ist. Den *Service* online zu sehen, beweist nicht, wie lange *Fuel* und *Ownership* ihn tragen werden. Die Absicht des Verantwortlichen ersetzt den sichtbaren Zustand nicht. Die Aufzeichnung der Aufgabe hält diese Aussagen getrennt, damit keine beruhigende Tatsache für den ganzen Zeitraum der Unterstützung steht.

Wird die Unterstützung vor der Einrichtung ungewiss, hält ein Aufschub die Eingaben für einen anderen Pfad verfügbar. Tritt die Ungewissheit während einer Aufgabe auf, hält ihr Verantwortlicher den beobachteten Zustand der Aufgabe und seine Auswirkung auf die erwartete Ausgabe fest. Dasselbe Wort – nicht verfügbar – darf nicht verbergen, ob die *Work* wartet, abgebrochen oder für den Beobachter lediglich unzugänglich ist.

Versprechen an andere *Work* müssen abgeglichen werden. Ein Verantwortlicher will einen *Service* vielleicht nach einer gegenwärtigen Aufgabe offline nehmen, während ein anderer Pilot eine neue Aufgabe vorbereitet, die seine anhaltende Verfügbarkeit voraussetzt. Beide Pläne können getrennt vernünftig sein. In der gemeinsamen Aufzeichnung muss ihre Überschneidung sichtbar werden, bevor die Einrichtung daraus einen Streit mit bereits gebundenem Material macht.

Verändert sich der Zustand des *Service*, folgt die Reaktion der Evidenz. Halte den beobachteten Zustand fest, bestimme die betroffenen Aufgaben und stoppe neue Verpflichtungen. Behaupte nicht, jede Aufgabe habe versagt, wenn das sichtbare Ergebnis eine Pause ist. Versprich nicht, die *Work* werde fortfahren, nur weil das Modul eingebaut bleibt. Die Verantwortlichen von Aufgabe und Infrastruktur bestimmen anhand des wirklichen Zustands von *Structure*, *Service* und *Fuel*, was getragen werden kann.

Doktrin für Verteidigung und Reparatur bleibt ausserhalb dieses Kapitels. Der Text erklärt nicht über das Beobachtbare hinaus, weshalb ein *Service* nicht verfügbar wurde. Die operative Frage ist enger: Welche versprochene *Capability* besitzt keinen durchgehenden Pfad mehr, und wie kann die *Exposure* verringert werden, bevor weitere *Work* von ihr abhängt?

Das sichtbare *Replacement* vom Anfang vermied diese Abhängigkeit vom *Service*, weil seine Aufgabe bereits abgeschlossen war. Zukünftiges *Replacement* wird sie vielleicht besitzen. Vorbereitung sollte vollständige, zugängliche *Capability* anders zählen als Material in einer laufenden oder nicht unterstützten Aufgabe.

Vor Leere ersetzen

Der letzte nutzbare Gegenstand ist keine *Reserve*. Er wirkt als Countdown ohne zugehörige Zeit.

Beginnt *Replacement* erst nach Verbrauch des letzten Exemplars, werden alle Abhängigkeiten des Pfads gleichzeitig dringend. Materialien, *Blueprint*, *Access*, *Service*, Route und Menschen müssen ohne Verzögerung verfügbar bleiben. Eine kleine Pause wird dann zur Lücke in der *Capability*.

Setze den Auslöser vor die Abwesenheit. Er sollte auf Zeit und Ungewissheit des Pfads zum *Replacement* statt auf einer allgemeingültigen Bestandszahl beruhen. Ein lokal hergestellter Gegenstand kann weiterhin von eingeführten Eingaben abhängen. Ein grosser sichtbarer Bestand kann für seine Nutzer unzugänglich sein. Eine kleine *Reserve* kann genügen, wenn Ersatzmöglichkeiten und ein kurzer geprüfter Pfad bestehen. Die Menge allein kann die Grenze nicht setzen.

Beginne mit Verbrauch oder Verlust, den die Gruppe beobachten kann. Welche Ereignisse verringern die nutzbare *Reserve*? Wer aktualisiert die Zahl? Wird ein Gegenstand nicht verfügbar, wenn er anderswo eingebaut, für eine Aufgabe reserviert oder jenseits der gegenwärtigen *Chain* bewegt wird? Eine Zahl, die sich erst nach physischer Zerstörung verändert, übersieht mehrere Wege, auf denen *Capability Home* verlässt.

Beschreibe danach die Vorlaufzeit des *Replacement* als Kette von Bedingungen statt als selbstsichere Dauer. Die Erkennung beginnt den Prozess. Ein Verantwortlicher gibt die Notwendigkeit frei. Eingaben werden freigegeben oder bewegt. Die Aufgabe wird eingerichtet oder ein bestehendes *Replacement* transportiert. Die Ausgabe erreicht ein nutzbares Lager. Jede Stufe kann auf Menschen, *Access*, *Service* oder *Connections* warten. Der Auslöser muss Raum für jene ungewissen Stufen lassen, welche die Gruppe tatsächlich besitzt.

Zwei Horizonte machen diese Ungewissheit sichtbar. Der Nutzungshorizont fragt, wie lange die gegenwärtige *Reserve* den angenommenen Zweck tragen kann. Der Horizont des *Replacement* fragt, wie lange der langsamste glaubwürdige Pfad zu einem weiteren nutzbaren Gegenstand dauern kann. *Replacement* beginnt, bevor sich die Horizonte

treffen. Keiner benötigt falsche Genauigkeit. Eine Bandbreite, eine begrenzende Abhängigkeit und die nächste Prüfung sind ehrlicher als ein selbstsicheres Datum aus optimalen Schritten.

Bereits laufende *Work* gehört weder zur *Reserve* noch zum abgeschlossenen *Replacement*. Zähle sie als offene Abhängigkeit mit ihrem *Service*, ihrem Verantwortlichen und dem erwarteten Empfang. So beruhigt eine laufende Aufgabe nicht jene Nutzer, die jetzt einen Gegenstand benötigen. Zugleich erkennt die Gruppe, dass *Replacement* begonnen hat.

Die Prioritäten des *Replacement* folgen der Konsequenz. Stelle jene *Capability* zuerst wieder her, deren Fehlen *Observation*, Rückkehr, *Access* oder einen weiteren Pfad des *Replacement* blockiert. Daraus folgt keine allgemeingültige Rangfolge von Gegenständen. Frage, welche fehlende Funktion die weiteren Versprechen der Gruppe falsch machen würde.

Die *Reserve* muss so geprüft werden, wie ein Nutzer sie antrifft. Kann ein befugter *Character* den Gegenstand finden und entnehmen? Liegt die Ausgabe im erwarteten Kontext des *Ownership*? Trennt das Inventar zugänglichen Bestand von reservierten Eingaben und persönlichem Eigentum? Kann die nächste Schicht Verantwortlichen und Auslöser erkennen, ohne die Person zu kontaktieren, die das System entworfen hat?

Wenn der verspätete Gegenstand *Home* erreicht, bringt ihn die Gruppe nicht auf ein unverändertes Regal zurück. Ein nutzbares *Replacement* wird in einem geprüften Zugriffspfad platziert. Die verbleibenden Bestände werden nach Zweck getrennt. Das Inventar benennt *Ownership* und den nächsten Auslöser. Eine zweite befugte Person bestätigt die Handlung, solange die erste noch verfügbar ist.

Das Ergebnis ist keine unendliche Bereitschaft. Es schafft einen ehrlichen Zeitraum, in dem eine *Capability* getragen wird, sowie eine sichtbare Bedingung, die *Replacement* beginnt, bevor dieser Zeitraum endet.

Feldprinzip

Lagerung ist eine Abhängigkeit, keine Pause in der Geschichte des Werts.

Benenne für jeden entscheidenden Gegenstand Zweck, *Ownership* und Ort. Ergänze Zugriffspfad, Konzentration, Nutzer und Auslöser des *Replacement*. Halte persönliche Vermögenswerte von jenen der *Corporation* getrennt. Behandle Sichtbarkeit, Einlagern, Entnahme, Installation und Verteilung als getrennte Handlungen, bis der vollständige Pfad des Nutzers geprüft worden ist.

Folge bei hergestelltem *Replacement* der Kette von der Notwendigkeit über *Blueprint*, Eingaben, Einrichtung, *Service* und Ausgabe bis zu einem befugten Empfänger. Eine Aufgabe kann pausieren oder keine *Capability* liefern, obwohl die *Structure* sichtbar bleibt. Beginne *Replacement*, solange Zeit und Alternativen noch der Gruppe gehören.

Der Gegenstand aus der Eröffnungsszene war nicht verschwunden. Verschwunden waren die umgebenden Annahmen. Durch ihre Verfolgung stellte die Gruppe mehr als den *Access* zu einem Objekt wieder her. Sie machte den nächsten *Failure* vor der nächsten Notwendigkeit sichtbar.

Part III begann mit einem Schiff, das an *Work* gebunden wurde. Er endet mit dem eingelagerten und nutzbaren Ergebnis, das mit *Replacement* verbunden ist. Diese Kontinuität erhält sich nicht von selbst über Tage und Menschen. Part IV beginnt, sobald morgen eine andere Person handeln muss als jene, welche die heutige Anordnung verstand.

Part IV - Handlungsfähig bleiben

Eine *Capability* wird fragil, wenn sie nur in der Erinnerung der Person fortbesteht, die sie zuletzt genutzt hat. Routen altern, Berechtigungen ändern sich, *Services* verlieren ihre Unterstützung, und Bestände der *Reserve* wechseln zwischen nutzbaren und versprochenen Zuständen. Währenddessen sieht *Home* weiterhin vertraut aus. Part IV verfolgt diese Veränderungen über die Zeit. Er verwandelt die heute funktionierende Anordnung in ein *Handover*, erkennt *Degradation*, bevor sie zur Krise wird, und beendet Verpflichtungen, solange noch eine Wahl bleibt. *Home* wird nicht dauerhaft gemacht. Stattdessen bleibt der Unterschied zwischen verfügbar, scheiternd und beendet für jene Menschen sichtbar, die als Nächstes handeln müssen.

Kapitel 13 - Routine im Zeitverlauf

Ein vollständiger Bericht und ein unvollständiges *Home* warteten auf den nächsten Piloten.

Dem Bericht zufolge war der Rückkehrzweig aktuell, der entscheidende *Service* online und das *Replacement* begonnen. Jede Aussage war zum Zeitpunkt ihrer Niederschrift wahr gewesen. Einige Stunden später konnte der Pilot nicht erkennen, welche Wahrheit hätte fortbestehen sollen oder welche Person noch für die nächste Prüfung verantwortlich war. Auch liess «begonnen» offen, ob Materialien reserviert, ein Auftrag installiert oder eine zugängliche Ausgabe zur Abholung bereit war.

Nichts im Bericht war falsch. Seine Gewissheit hatte ihre Evidenz überdauert.

Zeit verändert operatives Wissen, selbst wenn niemand das zugrunde liegende Feld bewusst verändert. Eine *Connection* altert. Eine *Shared Location* kann ablaufen. Die Verfügbarkeit eines Piloten endet. Ein *Service* verbraucht Unterstützung. Bestand wechselt von nutzbar zu reserviert, und ein Auftrag wechselt von beabsichtigt zu aktiv oder unterbrochen. Wer gestern beobachtet hat, kann Aktualität nicht durch selbstbewusstes Schreiben übertragen.

Routine ist die Praxis, begrenztes Wissen durch diese Veränderungen zu tragen. Sie bewahrt weder jedes Detail noch einen Anwesenheitsnachweis. Der nächsten Person gibt sie genügend *Source*, Zeit, Bedeutung, *Unknowns* und Verantwortung. Damit erkennt diese, was noch ausgeführt werden kann und was zuerst erneut beobachtet werden muss.

Wissen altert

Eine *Observation* bleibt eine historische Tatsache, nachdem sie gegenwärtiges Handeln nicht mehr stützt.

Diese Unterscheidung ist wichtig. Hat ein Pilot einen *Service* zu einer festgehaltenen Zeit online gesehen, muss dieses Ereignis nicht gelöscht werden, wenn der *Service* später unsicher wird. Es bleibt nützliche Evidenz dafür, was den früheren Auftrag gestützt hat. Verändert hat sich die Aussage, die sich heute daraus ableiten lässt.

Operatives Alter ist keine universelle Zeitspanne. Dieselbe *Observation* kann für einen Zweck weiterhin genügen und für einen anderen veraltet sein. Eine Routenaufzeichnung, die erklärt, wo ein Pilot gestern unterwegs war, kann für wertvolle Fracht heute zu alt sein. Ein vor einer Arbeitsperiode bestätigtes Inventar kann noch eine nicht dringende Zählung leiten, ohne zu beweisen, dass das letzte entscheidende *Replacement* weiterhin zugänglich ist.

Das Alter sollte deshalb an die Bedeutung gebunden werden. Halte fest, was beobachtet wurde, zusammen mit *Source* und *Timestamp*, und nenne danach die gegenwärtige Entscheidung, welche die *Observation* unterstützen sollte. Ergänze die nächste Prüfung oder die Bedingung, welche diese Unterstützung beendet. Ohne diese Verbindung schmückt ein *Timestamp* die Aussage, steuert aber nicht ihre Verwendung.

Zeit wird durch *Shared Location Folders* noch auf andere Weise sichtbar. *Locations* können ein Ablaufdatum besitzen, und ein Ordner kann für einen *Character* online oder offline sein. Eine gespeicherte *Location*, die einst die *Chain* stützte, kann aus der Nutzung verschwinden oder nicht mehr wie erwartet geteilt sein. Ein Name in einem alten Bericht beweist nicht, dass der nächste Pilot ihn noch sehen oder verwenden kann.

Auch Beobachtungen des *Access* altern. Ein geprüfter Pfad der *Corporation* beweist, dass ein *Character* eine Handlung mit den Rollen und im Kontext des Tests ausführen konnte. *Corporation Roles*, *Access Lists* und Verantwortlichkeiten können sich danach ändern. Für entscheidende

Handlungen braucht es klare Auslöser für eine erneute Prüfung. Die Übertragung einer Rolle, ein veränderter Eigentümer, ein neuer Lagerort oder der Weggang des einzigen geprüften Nutzers können einen solchen Auslöser bilden.

Mit dem Alter verändert sich auch die Stärke einer Schlussfolgerung. Eine direkte *Observation* kann genau den Zustand an ihrem *Timestamp* belegen. Eine spätere Verwendung fügt die Annahme hinzu, dass die entscheidende Bedingung fortbestand. Die Aufzeichnung darf diesen Schritt nicht verbergen, indem sie den ursprünglichen Satz ohne seine Zeitangabe kopiert. Hängt eine Handlung vom Fortbestehen ab, prüft die nächste Kontrolle den Zustand erneut, statt mit dem Alter des ersten Berichts zu argumentieren.

Alte Evidenz kann weiterhin eine Abfolge erklären. Sie kann zeigen, wann eine *Reserve* zuletzt bestätigt wurde, wann ein *Service* einen Auftrag trug oder wann ein Routenzweig in den Plan aufgenommen wurde. Trenne diese Geschichte vom aktuellen *Handover*. Andernfalls kann ein Leser die ausführlichste Darstellung mit der aktuellsten verwechseln.

Der eingangs vorgelegte Bericht lässt sich berichtigen, ohne so zu tun, als verfalle jede Aussage im selben Augenblick. Der Rückkehrzweig erhält eine nächste Prüfung, die an die geplante Bewegung gebunden ist. Die *Observation* des *Service* erhält einen Unterstützungszeitraum, der mit dem offenen Auftrag verbunden ist. Das *Replacement* erhält einen Zustand, der genau genug erkennen lässt, ob es nutzbar, in Arbeit oder nur beabsichtigt ist.

Das Wissen wurde nicht dauerhaft gemacht. Sein Altern wurde sichtbar gemacht.

Jede Schicht übernimmt

Eine Schicht beginnt mit Verpflichtungen von Menschen, die vielleicht nicht mehr anwesend sind.

Schiffe können ausserhalb von *Home* verbleiben. Ein Auftrag kann in der *Structure* weiterlaufen. Eine Route kann eine erwartete Rückkehr tragen. Ein versprochenes *Replacement* kann für jemanden reserviert sein, der es noch nicht übernommen hat. Ein *Unknown* kann bereits einen Verantwortlichen haben, dessen verfügbare Zeit endete, bevor der nächste Beobachter eintraf.

Der ankommende Pilot sollte all dies nicht aus einem Gesprächsverlauf rekonstruieren müssen. Ebenso wenig sollte die ausgehende Person ein vollständiges Tagebuch verfassen. Ein *Handover* wählt jene offenen Bedingungen aus, die verändern können, wozu die nächste Person in der Lage ist.

Jeder Eintrag braucht sieben Teile. Beginne mit *Observation*, *Source* und *Timestamp*. Ergänze danach operative Bedeutung, das verbleibende *Unknown*, den Verantwortlichen und die nächste Prüfung.

Die *Observation* nennt die begrenzte Tatsache. Die *Source* zeigt dem Leser, ob sie aus einem direkten Zustand des Clients, einer gemeinsamen Aufzeichnung, von einem anderen Piloten oder aus einer externen Referenz stammt. Der *Timestamp* verankert sie in der Zeit. Die Bedeutung verbindet die Tatsache mit einem gegenwärtigen Versprechen. Das *Unknown* verhindert, dass sich eine Annahme im Satz verbirgt. Der Verantwortliche bezeichnet, wer handeln oder die Beobachtung wiederherstellen wird. Die nächste Prüfung legt fest, wann die Aussage erneuert, eingeschränkt oder geschlossen werden muss.

Diese Teile können knapp bleiben. «Rückkehrzweig beobachtet; direkte Informationen zum *Wormhole* und *Shared Map*; 21:10; stützt die Rückkehr eines Piloten; späterer Verkehr unbekannt; Mara verantwortlich; vor Frachtbewegung erneut prüfen» ermöglicht mehr Handlung als «Route gut». Der Satz garantiert weder die *Connection* noch eine genaue Lebensdauer. Er lässt die nächste Person die gestützte Rückkehr von einer neuen Bewegung unterscheiden.

Ein *Handover* braucht auch Annahme. Ein Name neben einem *Unknown* beweist nicht, dass die Person es gesehen hat, Zeit dafür besitzt oder über den erforderlichen *Access* verfügt. Die ankommende verantwortliche Person bestätigt die begrenzte Verantwortung. Nimmt sie niemand an, reduziert sich die Operation gemäss Plan: Die Bewegung wird verschoben, der optionale Auftrag geschlossen, die unterstützten *Services* werden verringert oder der Zweig als ohne Verantwortlichen markiert.

Das *Handover* ist keine Anwesenheitskontrolle. Weder eine Erklärung für die Abwesenheit eines Piloten noch eine Messung seines Beitrags ist erforderlich. Es hält nur fest, ob eine operative Funktion gegenwärtig eine verantwortliche Person besitzt. Die Menschen behalten ihre Privatsphäre, während die Gruppe ein ehrliches Bild ihrer *Capabilities* behält.

Eine übernommene Verpflichtung kann auch abgelehnt werden. Die ankommende Schicht ist nicht verpflichtet, jede Aufgabe fortzuführen, nur weil eine frühere Person sie begonnen hat. Lehnt sie ab, sollte das die vorbereitete Verringerung auslösen und nicht zu stiller Vernachlässigung führen. Wer für einen Auftrag verantwortlich ist, kann eine Entscheidung zurückgeben; ein Routenbeobachter kann festhalten, dass der Zweig nicht länger gepflegt wird; ein Empfänger kann Fracht ablehnen, bevor sie bewegt wird. Ehrliche Grenzen bewahren mehr *Capability* als ein unbestätigtes Versprechen.

Aus dem Bericht der Eröffnungsszene wird ein Gespräch mit einem klaren Ende. Die ausgehende Person legt die offenen Verpflichtungen vor. Ankommende Verantwortliche nehmen an, was sie tragen können. Alles, was ohne Verantwortlichen bleibt, ändert seinen Zustand im gemeinsamen Plan, bevor neue *Work* beginnt.

Das *Handover* sollte eine aktive Verpflichtung von nützlichem Kontext unterscheiden. Der Ort gewöhnlicher Überschüsse kann dem ankommenden Piloten helfen, ohne eine Handlung zu verlangen. Ein Pilot jenseits eines alternden Zweigs erzeugt eine aktive Abhängigkeit seiner Rückkehr. Ein *Service*, der keine offene *Work* trägt, kann routinemässi-

ge Pflege, aber keine Ausnahme erfordern. Werden diese Kategorien getrennt, findet der Leser, was angenommen werden muss, bevor optionaler Hintergrund die verfügbare Aufmerksamkeit verbraucht.

Das *Handover* endet mit einer kurzen Rückbestätigung. Jede ankommende verantwortliche Person nennt die angenommene Handlung, Grenze und nächste Prüfung. Unterschiede werden berichtet, solange die *Source* noch anwesend ist. Schweigen lässt den Eintrag unbestätigt und damit der Verringerung unterworfen; es wird nicht als Zustimmung gedeutet.

Berichte brauchen Verantwortliche

Ein Bericht ohne verantwortliche Person beschreibt ein Problem und weist es dem Leser zu.

Für eine öffentliche Notiz mag das genügen, nicht aber für eine operative Ausnahme. Eine Prüfung des *Service*, die vor der Fortsetzung eines offenen Auftrags notwendig ist, braucht eine benannte verantwortliche Person. Ein gescheiterter Zugriffstest verlangt jemanden, der ihn berichtet oder das abhängige Versprechen reduziert. Blockiert ein *Unknown* der Route eine Bewegung, umfasst die Verantwortung entweder die nächste *Observation* oder die Entscheidung zu warten.

Die Verantwortung sollte an die nächste Handlung geknüpft sein, nicht nur an das Thema. «Verantwortlicher für *Industry*» kann zu breit sein. Jemand kann den installierten Auftrag verantworten, eine andere den ihn tragenden *Service* und eine dritte die Entgegennahme der Ausgabe. Die Benennung der Handlung verhindert, dass mehrere fähige Menschen jeweils annehmen, die anderen würden sie abschliessen.

Auch die Verantwortung braucht eine Grenze. Welche Evidenz schliesst die Handlung ab? Wann kehrt die Verantwortung in die gewöhnliche Routine zurück? Wer wird informiert, wenn sich die Bedingung nicht lösen lässt? Eine unbegrenzte Verantwortung wird zu einer dauerhaften

Hintergrundannahme, und die Aufzeichnung wirkt weiterhin zugewiesen, nachdem die Person aufgehört hat, danach zu handeln.

Verantwortung muss ausführbar bleiben. Ein *Character* kann keine Handlung an einem *Service* verantworten, die er weder beobachten noch ausführen kann. Kein Pilot kann einen Lagerpfad der *Corporation* bestätigen, auf den er keinen Zugriff hat. Nach seinem Weggang kann ein Routenbeobachter einen Zweig nicht pflegen, wenn kein bestätigter Nachfolger vorhanden ist. Stimme Verantwortung auf gegenwärtige Befugnis, *Access* und Verfügbarkeit ab, nicht auf Ansehen.

Eine zentrale Führung ist nicht erforderlich. Die Verantwortung kann auf jene Menschen verteilt werden, die der jeweiligen Abhängigkeit am nächsten stehen. Die gemeinsame Anforderung ist der Abschluss. Jede verantwortliche Person hält die neue *Observation* fest, reduziert das Versprechen oder übergibt die Handlung einem bestätigten Nachfolger. Ein Kanal, Dokument oder Kartenwerkzeug kann den Eintrag tragen, doch kein bestimmtes Werkzeug schafft die Verantwortung von selbst.

Mehrere Handlungen kann eine Person nur so lange verantworten, wie deren Grenzen glaubwürdig bleiben. Ist derselbe Pilot für den Rückkehrzweig, eine Prüfung des *Service* und die Entgegennahme ein-treffender Fracht verantwortlich, ordnet der Bericht diese Handlungen nach ihrer zeitlichen Bedeutung. Ihre Priorität folgt Konsequenz und Zeitpunkt, nicht der zuletzt eingetroffenen Nachricht. Handlungen, die sich innerhalb dieser Reihenfolge nicht tragen lassen, werden neu zugewiesen oder reduziert, bevor der Verantwortliche zu einem verborgenen einzelnen Verzögerungspunkt wird.

Geteilte Verantwortung sollte vorsichtig eingesetzt werden. Zwei Menschen können bei einer Handlung zusammenarbeiten, doch eine begrenzte nächste Prüfung braucht weiterhin eine klar benannte berichtende Person. «Wir beide beobachten» kann zu zwei *Observations* führen oder zu keiner, während beide auf die Aktualisierung durch den anderen warten.

Berichte sollten eine Eskalation verhältnismässig machen. Eine kleine Inventardifferenz kann auf die benannte verantwortliche Person warnten. Eine fehlende *Observation* der Rückkehr für einen Piloten, der sich bereits ausserhalb befindet, sollte für jene Menschen sichtbar sein, deren Handlung jetzt davon abhängt. Der Bericht vermeidet eine Alarmkennzeichnung und nennt die Konsequenz: Welche Bewegung, welcher *Service*, welche *Reserve* oder welcher Zugriffspfad bleibt ohne Unterstützung, bis die Prüfung geschlossen ist?

Kann die verantwortliche Person die Handlung nicht abschliessen, ändert sich der Bericht. Sein Zustand kann nicht grün bleiben, nur weil weiterhin ein Name danebensteht. Die Person hält den ungelösten Zustand fest, übergibt ihn, wenn jemand ihn annimmt, oder löst die geplante *Degradation* aus. So wird das Scheitern einer Lösung zu einem ehrlichen operativen Ergebnis und nicht zu einer privaten Last.

Ausnahmen brauchen Abschluss

Routine ist am einfachsten, wenn nichts Ungewöhnliches geschieht. Nützlich wird sie, wenn eine Ausnahme nicht verschwinden will.

Eine Ausnahme ist ein begrenzter Unterschied zwischen dem erwarteten Zustand und der gegenwärtigen Evidenz. Ein als online erwarteter *Service* wird offline beobachtet. Eine in einem *Shared Folder* erwartete *Location* fehlt für den vorgesehenen Nutzer. Eine Inventarzählung stimmt nicht mit dem nutzbaren Bestand überein. Für einen Rückkehrzweig fehlt eine *Observation* innerhalb der Bedingung, welche die offene Bewegung erfordert.

Die erste Reaktion besteht nicht darin, die Ursache zu erklären. Halte den Unterschied, die *Source* und die Zeit fest. Verbinde ihn danach mit den betroffenen Verpflichtungen und weise die nächste Handlung zu. Die Ursache mag später wichtig werden; die *Capability* verändert sich jetzt.

Ein Abschluss kann verschiedene Formen annehmen. Der erwartete Zustand kann wiederhergestellt und geprüft werden. Der Plan kann den neuen Zustand annehmen und sein Versprechen reduzieren. Die Verpflichtung kann enden und damit die Abhängigkeit entfernen. Eine neue verantwortliche Person kann die Handlung mit einer klaren nächsten Prüfung übernehmen. Jeder Abschluss verändert entweder Evidenz, Verantwortung oder Umfang.

«Notiert» ist kein Abschluss. Ebenso wenig ein Gespräch, das endet, ohne die gemeinsame Aufzeichnung zu aktualisieren. Kehrt der *Service* online zurück, braucht die Aufzeichnung die neue *Observation*, und die gestützte *Work* braucht eine neue Entscheidung. Wird ein fehlender Gegenstand in persönlichem Lager gefunden, benötigt das Inventar den berechtigten Pfad des *Ownership*. Andernfalls übernimmt die nächste Schicht sowohl das reparierte Feld als auch den alten *Failure*.

Ausnahmen brauchen ebenfalls ein Alter. Eine vorübergehende Umgehung kann unbemerkt zur Normalität werden, ohne eine der für den Routinebetrieb vorgesehenen Kontrollen zu besitzen. Gibt ein *Character* wiederholt Bestand für andere frei, weil deren *Access* fehlerhaft ist, kann die Gruppe fähig wirken, während sie von der Anwesenheit dieses *Characters* abhängt. Die nächste Prüfung sollte fragen, ob die Umgehung endet, zu einer ausdrücklichen Gestaltung wird oder eine Verringerung der *Capability* erzwingt.

Bewahre nicht jede historische Ausnahme im aktiven *Handover* auf. Ist sie geschlossen, behalte nur, was den gegenwärtigen Zustand erklärt oder den nächsten Auslöser verbessert. Eine endlose Liste lehrt die Leser, die ganze Liste zu ignorieren. Aktive Einträge sollten Handlungen darstellen, die noch jemand ausführen kann.

Der Abschluss sollte eine kleine Spur der Entscheidung hinterlassen. Halte die abschliessende *Observation*, die ausgeführte Handlung und jeden neuen Auslöser fest, der aus der Ausnahme entstand. Diese Spur ist kein Vorfallarchiv. Dadurch öffnet ein späterer Leser dieselbe Frage

nicht wieder, nur weil der Unterschied einfach aus der aktiven Liste verschwunden ist. Ein einzelner Satz kann zeigen, dass der Zustand wiederhergestellt, der Zugriffspfad geprüft und die Routine so angepasst wurde, dass sie nach künftigen Rollenübertragungen kontrolliert wird.

Wiederholte Ausnahmen legen die Gestaltung offen. Scheitern derselbe *Service*, dasselbe *Handover* der Route oder dieselbe Lagerberechtigung immer auf die gleiche Weise, kann der Abschluss jedes einzelnen Auftretens das Muster bewahren. Sie sollte eine begrenzte Gestaltungsfrage aufwerfen: Soll die unterstützte Verpflichtung kleiner werden, soll der Auslöser früher eintreten oder soll die Abhängigkeit einen weiteren geprüften Pfad erhalten? Die Antwort verändert die Routine; die Ausnahme schliesst danach gegenüber dieser neuen Routine.

Zu Beginn verbargen sich drei Ausnahmen in selbstbewussten Sätzen. Sobald Alter und Zustand ergänzt sind, kann jede anders geschlossen werden. Für die nächste Rückkehr wird eine *Observation* der Route erneuert. Ein Verantwortlicher für die Wartung nimmt die Unterstützung des *Service* bis zur Entgegennahme des Auftrags an. Das *Replacement* bleibt in Arbeit und wird nicht länger als nutzbare *Reserve* gezählt. Der Bericht wirkt weniger beruhigend und wird nützlicher.



EINE AUSNAHME SCHLIESST ERST MIT EINEM BESTÄTIGTEN NÄCHSTEN ZUSTAND

EIN BERICHT ALLEIN ÜBERTRÄGT KEINE VERANTWORTUNG

Abbildung AFH-B2-FIG-013 — Eine Ausnahme bleibt bis zum bestätigten nächsten Zustand offen. Eine Ausnahme bleibt offen, bis Verantwortung bestätigt und ihr nächster Zustand geprüft, angenommen, übergeben oder verringert ist.

Routine muss schrumpfen

Eine Routine, die sich nicht aufrechterhalten lässt, sollte weniger von *Home* verlangen.

Gruppen behandeln ein verpasstes *Handover* oft als Aufforderung zu mehr Disziplin. Manchmal ist das richtig. Manchmal verlangt die Routine mehr *Observations*, Verantwortliche und Aktualisierungen, als die verfügbaren Menschen tragen können. Unter diesen Bedingungen das vollständige Versprechen zu wiederholen bewahrt keine *Capability*. Es verbirgt ihren Niedergang.

Kontrollierte *Degradation* reduziert Verpflichtungen in einer bekannten Reihenfolge. Optionale Bewegung wartet, wenn die *Chain* keine verantwortliche Person besitzt. Neue *Industry Jobs* werden gestoppt, bevor nicht unterstützte *Services* weiteres Material tragen. Arbeitsschiffe keh-

ren zurück, solange ihre Route noch eine bestätigte *Observation* besitzt. Bestand wechselt von versprochener Ausgabe zurück in den Zustand unbestätigt oder nicht verfügbar, wenn niemand seinen *Access* bestätigen kann.

Die Reihenfolge sollte jene Funktionen schützen, mit denen die Gruppe weiter beobachten und reduzieren kann. Bewahre genügend Routenwissen, um die Menschen ausserhalb zu berücksichtigen. Bewahre den *Access* zu Aufzeichnungen und entscheidendem *Replacement*. Erhalte die Fähigkeit, offene Aufträge zu schliessen oder ehrlich festzuhalten, dass die gegenwärtige Schicht sie nicht schliessen kann. Entferne optionale Tiefe, bevor diese Funktionen zu Hintergrundannahmen werden.

Degradation braucht sichtbare Zustände. «Reduziert» sollte benennen, welche Bewegung, welcher *Service*, welche *Reserve* oder welche Beobachtung nicht länger unterstützt wird. Der Rest von *Home* darf nicht als gescheitert gelten, nur weil eine Funktion beendet wurde. Präzision lässt die Gruppe innerhalb einer kleineren, wahrheitsgetreuen Grenze weiterarbeiten.

Dasselbe Prinzip gilt für das Format des *Handover*. Lassen sich sieben vollständige Felder nicht für jede routinemässige *Observation* pflegen, reserviere sie für offene Verpflichtungen und Ausnahmen. Stabile Hintergrundinformationen können in ihrer eigenen kontrollierten Referenz bleiben. Gegenwärtiges Handeln darf nicht in Beschreibungen ohne Verantwortlichen oder nächste Prüfung ertrinken.

Eine minimale Routine kann stark bleiben. Sie bewahrt die Menschen ausserhalb und die für ihre Rückkehr erforderlichen *Connections*. Daneben erhält sie die *Services* für offene *Work*, die Zugriffspfade zu entscheidender *Reserve* sowie jene *Unknowns*, welche eine Verringerung blockieren. Andere Beobachtungen können warten, bis wieder Kapazität vorhanden ist. Die Gruppe benennt, was nicht länger gepflegt wird, damit sein Fehlen im Bericht nicht mit einem unveränderten Zustand verwechselt wird.

Die *Degradation* sollte an ihrem Anfang und ihrem Ende angekündigt werden. Ankommende Piloten müssen wissen, welche Versprechen entfernt wurden, nicht nur, dass die Aktivität ruhig wirkt. Kehrt die Verantwortung zurück, wird der wiederhergestellte Umfang anhand frischer Evidenz aufgeführt. So schwankt die Routine nicht zwischen unsichtbarer Vernachlässigung und der ebenso unsichtbaren Annahme, alles sei wieder aufgenommen worden.

Eine Verringerung kann rückgängig gemacht werden, nachdem Evidenz und Verantwortung zurückgekehrt sind. Ein neuer Beobachter nimmt die *Chain* an, ein Verantwortlicher für den *Service* bestätigt dessen Unterstützung oder der eintreffende Nutzer prüft einen Zugriffspfad. Erst dann genehmigt die Gruppe ausgehend vom gegenwärtigen Zustand eine neue Verpflichtung. Wieder vorhandene Besetzung beweist nicht, dass jede alte Aussage erneut aktuell geworden ist.

Angesichts des Berichts aus der Eröffnung entscheidet sich der nächste Pilot gegen die geplante Frachtbewegung. Der Rückkehrzweig wird nur für den Piloten erneuert, der sich bereits ausserhalb befindet. Der neue *Industry Job* wartet, bis jemand die Unterstützung seines *Service* annimmt. Das *Replacement* bleibt als in Arbeit gekennzeichnet. *Home* ist kleiner geworden, als es der erste Bericht vermuten liess, doch jedes verbleibende Versprechen besitzt nun Evidenz und eine verantwortliche Person.

Dieses kleinere *Home* kann handeln.

Feldprinzip

Routine trägt begrenzte Wahrheit durch die Zeit.

Halte für jede offene Verpflichtung *Observation*, *Source* und *Timestamp* fest. Ergänze ihre Bedeutung, das *Unknown*, die verantwortliche Person und die nächste Prüfung. Verlange eine Annahme, wenn die Verantwortung wechselt. Nimmt sie niemand an, reduziere das Versprechen, statt es dem Zufall zu übergeben.

Schliesse Ausnahmen, indem du den erwarteten Zustand wiederherstellst und prüfst, einen neuen Zustand annimmst, die Verpflichtung beendest oder die nächste Handlung einer bestätigten verantwortlichen Person übergibst. Bewahre nur ausführbare Ausnahmen im aktuellen *Handover*. Verwende den Namen einer Person nicht als Ersatz für *Access*, Verfügbarkeit oder Evidenz.

Der Bericht der Eröffnungsszene lieferte eine genaue Geschichte und eine unsichere Anweisung. Nachdem die Routine das Alter seiner Aussagen offengelegt hatte, konnte der nächste Pilot bewahren, was unterstützt war, und beenden, was es nicht war. Es brauchte keine Krise, um *Home* kleiner zu machen.

Das zeigt den Wert von *Degradation* vor dem *Failure*. Kapitel 14 folgt ihren sichtbaren Signalen – *Power State*, Verfügbarkeit von *Services*, verändertem *Access* und einer dünner werdenden *Reserve* – und fragt, was reduziert werden sollte, bevor aus diesen Signalen eine Krise wird.

Kapitel 14 - *Failure* vor Krise

Die *Structure* sah seit dem letzten Abdocken unverändert aus.

Ihre Hülle stand weiterhin auf dem *Grid*. Der anfliegende Pilot konnte noch andocken, und nach seiner Ankunft war das vertraute Inventar sichtbar. Nichts in dieser Szene kündigte eine Krise an.

Im Innern war ein versprochener *Service* nicht verfügbar. Der erwartungsgemäss weiterlaufende Auftrag war pausiert. Die letzte *Fuel Observation* hatte ihre nächste Prüfung ohne bestätigte verantwortliche Person überschritten, und ein Pilot konnte den in der Vorwoche geprüften Lagerpfad der *Corporation* nicht länger nutzen. Neben der sichtbaren *Structure* wirkte jede Veränderung klein. Zusammen bedeuteten sie, dass *Home* die *Work* nicht mehr tragen konnte, welche seine Routine weiterhin versprach.

Failure beginnt oft als Abweichung zwischen Erscheinung und *Capability*. Dieses Kapitel diagnostiziert weder einen Angriff noch eine Reparatur und erklärt nicht jede Veränderung zum Notfall. Es liest den beobachtbaren Zustand, folgt den betroffenen Abhängigkeiten und reduziert Verpflichtungen, bevor die Gruppe alle diese Abhängigkeiten gleichzeitig benötigt.

Degradation zeigt Signale

Degradation ist sichtbar, wenn die *Baseline* konkret bleibt.

Kapitel 5 hielt das gewöhnliche *Home* mit datierten Beobachtungen von System, *Chain*, Infrastruktur und *Work* fest. Kapitel 8 und 13 ergänzten Verantwortliche und nächste Prüfungen. Diese Aufzeichnungen erlauben es, einen gegenwärtigen Unterschied zu benennen, ohne sich auf ein allgemeines Gefühl zu stützen, dass etwas nicht stimmt.

Das nützliche Signal ist begrenzt. Ein online erwarteter *Service* wird offline beobachtet. Eine in *Full Power* erwartete *Structure* wird in *Low Power*

beobachtet. Nicht mehr verfügbar ist eine *Shared Location*, die für einen *Character* online erwartet wurde. Die nutzbare *Reserve* fällt unter ihren Auslöser für das *Replacement*. Eine verantwortliche Person verpasst die Prüfung, welche eine offene Verpflichtung stützen sollte.

Diese Signale erklären ihre Ursachen nicht selbst. Eine bewusste Handlung der Verwaltung, ungenügende Unterstützung oder ein anderes Ereignis kann einen *Service* offline hinterlassen. Ein Zugriffspfad kann sich verändern, weil eine *Corporation Role*, ein *Profile* oder eine *Access List* geändert wurde oder weil der frühere Test einen anderen Kontext abdeckte. Eine *Reserve* kann durch gewöhnliche Nutzung abnehmen. Die Ursache sollte von der Person mit der passenden Befugnis beobachtet und untersucht werden; die Auswirkung auf die *Capability* lässt sich bereits jetzt festhalten.

Signale erhalten ihre Bedeutung durch das, was von ihnen abhängt. Ein offline befindlicher *Service* ohne offene oder geplante *Work* kann Wartung, aber nur wenig unmittelbare Verringerung verlangen. Derselbe *Service* unter einem installierten Auftrag und einem versprochenen *Replacement* verändert zwei Verpflichtungen. Ein veränderter Zugriffspfad zu einem Überschuss unterscheidet sich von einem veränderten Pfad zur einzigen nutzbaren *Reserve*. Das Feld wird anhand der Konsequenz gelesen, nicht anhand der Grösse einer Warnung in der Bedienoberfläche.

Die Abfolge ist ebenso wichtig wie der Schweregrad. Eine verpasste Beobachtungsprüfung kann eine Unterstützungsbedingung als *Unknown* hinterlassen. Neue *Work*, die danach genehmigt wird, fügt der Unsicherheit weitere Verpflichtung hinzu. Eine anschliessend beobachtete Veränderung des *Service* betrifft dann mehr Material und Menschen, als dies früher der Fall gewesen wäre. Die zuletzt sichtbare Pause kann wie der *Failure* wirken, obwohl das vermeidbare Wachstum zwischen der ersten verpassten Prüfung und dem letzten neuen Versprechen stattgefunden hat.

Die Aufzeichnung sollte Veränderungen deshalb verbinden, statt einzelne Warnungen zu sammeln. Die verpasste Prüfung des *Fuel*, die spätere *Observation* von *Low Power* und der pausierte Auftrag bilden eine Abhängigkeitsfolge. Eine veränderte *Access List* und eine gescheiterte Handlung an der *Reserve* bilden eine weitere. Jede Folge kann bei ihrer frühesten noch offenen Wahl reduziert werden, statt auf ein einziges dramatisches Ereignis zu warten, das alles erklärt.

Ein Signal braucht *Source* und Zeit. «Ausfall der *Structure*» verbindet viele mögliche Zustände und gibt der nächsten Person keinen Test. «Um 19:40 zeigte die Verwaltung der *Structure* den erforderlichen *Service* offline; der offene Auftrag besitzt keine bestätigte Unterstützung mehr» trennt Evidenz von Konsequenz. Die Aussage lässt sich prüfen, zuweisen und schliessen.

Auch fehlende Signale sind bedeutsam. Eine Prüfung des *Fuel* ohne Bericht ist keine Evidenz dafür, dass *Fuel* fehlt. Sie belegt, dass die Unterstützung unbestätigt ist. Die Reaktion kann dennoch darin bestehen, neue *Work* zu pausieren, weil das Versprechen eine aktuelle *Observation* voraussetzte. Ein *Unknown* ist ein operativer Zustand, wenn die nächste Handlung von Gewissheit abhängt.

In der *Structure* aus der Eröffnung ist der pausierte Auftrag nicht der erste *Failure*. Die verpasste Prüfung, der nicht verfügbare *Service* und der veränderte *Access* boten frühere Möglichkeiten zur Verringerung des unterstützten Umfangs. Eine Krise beginnt, wenn sich solche Signale unter einem unveränderten Versprechen ansammeln dürfen.

Low Power zählt

Der *Power State* beschreibt eine bestimmte Beziehung zwischen einer *Structure* und ihren *Services*.

Eine *Upwell Structure* mit mindestens einem online geschalteten *Service Module* befindet sich in *Full Power*. Ohne ein online geschaltetes *Service Module* befindet sie sich in *Low Power*. Diese Begriffe bewerten weder

die Kompetenz der *Corporation* noch beschreiben sie jede *Capability*, welche die *Structure* weiterhin besitzen kann. Es sind begrenzte Zustände mit operativen Konsequenzen.

Die erste Lesart von *Low Power* ist deshalb sachlich. Bestätige den Zustand anhand einer passenden *Source* und versieh ihn mit einem *Timestamp*. Stelle fest, welche *Service Modules* online erwartet wurden und welche *Work* von ihnen abhing. Benenne den Verantwortlichen für Wartung oder Verwaltung, der den Unterschied untersuchen kann. Leite nicht allein aus dem *Power State* die gesamte Ursache ab.

Die zweite Lesart betrifft Versprechen. Andocken und die Sichtbarkeit von Vermögenswerten können im Zustand *Low Power* erhalten bleiben. Das gilt auch für gewisse Handlungen. Diese verbleibenden Funktionen machen die offline befindlichen *Services* nicht verfügbar. Pläne sollten *Service* für *Service* reduziert werden. Stoppe neue Aufträge, die nicht unterstützte Funktionen von *Industry* benötigen. Klassifiziere Ausgaben neu, deren durchgehender Pfad nicht mehr besteht. Sage den Nutzern, welche bestimmte *Capability* geendet hat.

Diese Präzision vermeidet zwei schlechte Schlüsse. «Fortbestehende Präsenz der *Structure*» kann dazu verleiten, weiterhin von Funktionen abzuhängen, die sie nicht mehr bereitstellt. «*Home* ist verloren» kann unnötige Bewegung auslösen, obwohl nützlicher *Access* und Zeit verbleiben. Der beobachtete Zustand stützt für sich allein keine der beiden Aussagen.

Anhaltender Mangel an Verbrauch von *Fuel* kann eine *Upwell Structure* über *Low Power* hinaus in den Zustand *Abandoned* führen. Dieser Zustand entfernt weitere Fähigkeiten und verändert die mit der *Structure* verbundenen Konsequenzen. Für diese Entscheidung ist kein genauer Timer nötig. Ein anhaltend nicht unterstützter Zustand ist nicht statisch, und eine weitere Verzögerung kann Möglichkeiten entfernen, die bei der ersten *Observation* von *Low Power* noch bestanden.

Die in Kapitel 3 festgelegte Wiederherstellungsgrenze macht diese Entwicklung für eingelagerte Werte besonders wichtig. Ein Inventar, das in einer sich verschlechternden *Structure* noch sichtbar ist, garantiert keine spätere Wiederherstellung. Verringere die Konzentration, solange *Access*, Zeit und *Ownership* vorhanden sind, nicht erst, nachdem der Zustand eine Krisenkennzeichnung geliefert hat.

Beobachte *Abandoned* als eigenen begrenzten Zustand. Leite weder ab, wer ihn verursacht hat, noch ob aktiver feindlicher Druck besteht. Verbinde ihn mit den Funktionen und Vermögenswerten, deren Ergebnis sich verändert hat, stoppe weitere Einlagerungen und bewege die Entscheidung in Richtung Wiederherstellung oder bewussten *Exit*. Diese Handlungen folgen dem sichtbaren Verlust an *Capability*, ohne sich in Verteidigungsdoktrin zu begeben.

Der *Power State* verändert auch die Dringlichkeit der Verantwortung. Eine unbeantwortete Prüfung des *Fuel*, während *Services* online bleiben, ist bereits ein *Unknown* ihrer Unterstützung. Sobald kein *Service* online ist und die *Structure* in *Low Power* beobachtet wird, betrifft der Unterschied nicht mehr nur einen künftigen Schwellenwert. Die gegenwärtige Funktionalität hat sich verändert. Ihre verantwortliche Person muss den unterstützten Zustand entweder wiederherstellen und prüfen oder die abhängigen Verpflichtungen schliessen.

Das Kapitel legt weder Mengen von *Fuel* noch Verbrauchspläne fest. Diese Werte hängen von den eingebauten *Services* und aktuellen Mechaniken ab. Die operative Aufzeichnung nennt den *Service*, der Unterstützung braucht, die ihn beobachtende Person, die von ihm getragene *Work* und den Zeitpunkt der nächsten Prüfung. Eine aktuelle spezialisierte *Source* liefert alle erforderlichen Werte zum Zeitpunkt der Handlung.

Für den Piloten aus der Eröffnung erklärt *Low Power* nicht, weshalb der Auftrag pausiert wurde. Der Zustand zeigt der Gruppe, dass ihre frühere Aussage über gepflegte *Services* nicht mehr aktuell ist. Das genügt, um

keine weitere Abhängigkeit hinzuzufügen, während befugte Personen die Ursache untersuchen.

Services können verschwinden

Ein eingebautes *Service Module* ist nicht dasselbe wie ein *Service*, der online ist.

Service Modules stellen Funktionen einer *Structure* bereit und verbrauchen *Fuel*, solange sie online sind. Ein Modul kann mit der *Structure* verbunden bleiben, während seine Funktion nicht verfügbar ist. Ein Blick auf das *Fitting* oder die Erinnerung an den gewöhnlichen Zweck der *Structure* kann die Beobachtung des von der *Work* benötigten Zustands des *Service* nicht ersetzen.

Verschwindet ein *Service* aus der Nutzung, beginne mit den abhängigen Handlungen. Ermittle Aufträge, die nicht beginnen können, und installierte *Work*, die möglicherweise pausiert ist. Suche Ausgabe- oder Zugriffspfade, welche die Funktion voraussetzten. Prüfe Fristen für das *Replacement* erneut, die von ununterbrochener Unterstützung ausgingen. Daraus entsteht eine kleinere und klarere Reaktion als aus der Erklärung, die gesamte Infrastruktur sei ausgefallen.

Trenne als Nächstes Nutzer von Verantwortlichen für die Wartung. Ein Nutzer kann beobachten, dass eine Funktion nicht verfügbar ist, ohne das Recht zu besitzen, ein Modul online zu schalten, *Fuel* zu prüfen oder die Konfiguration der *Structure* zu verändern. Sein Bericht darf nicht abgewertet werden, weil er den Zustand nicht reparieren kann. Er liefert Evidenz aus dem Nutzungspfad. Ein Verantwortlicher für Wartung oder Verwaltung beobachtet danach innerhalb seiner Befugnis das zugrunde liegende Modul und dessen Unterstützung.

Das *Handover* verbindet beide Beobachtungen. «Nutzer konnte die erwartete Aktivität nicht beginnen» beschreibt die Auswirkung. «Erforderlicher *Service* offline beobachtet» beschreibt den Zustand der Infrastruktur. «*Fuel* noch nicht bestätigt» bleibt ein *Unknown*, bis die passen-

de Person es prüft. Werden diese Zeilen getrennt gehalten, erscheint eine plausible Ursache nicht als Tatsache im Bericht.

Aufträge brauchen ihren eigenen Zustand. Offizielle Hinweise zu *Structures* unterscheiden zwischen einem *Service*, der offline geht und damit verbundene *Work* pausieren kann, und der Entfernung des *Service* oder Zerstörung der *Structure*, wodurch sie abgebrochen werden kann. Die Gruppe hält fest, was die Bedienoberfläche für den tatsächlichen Auftrag zeigt. Ein Auftrag, der pausiert ist, darf nicht als abgeschlossen, abgebrochen oder mit garantierter Fortsetzung beschrieben werden.

Kehrt der *Service* zurück, prüfen die Nutzer den vollständigen Pfad erneut. Der Onlinezustand ist notwendig, löst aber veränderten *Access*, den Ort der Eingaben oder das *Ownership* der Ausgabe möglicherweise nicht. Nach der Wiederherstellung bestätigt die für den Auftrag verantwortliche Person, was fortgesetzt wurde, und passt die erwartete Entgegennahme an. Wiederherstellung ist Evidenz, kein Befehl, jede verschobene Verpflichtung gleichzeitig wieder aufzunehmen.

Kann die Unterstützung innerhalb des angenommenen Zeitraums nicht wiederhergestellt werden, wird die Verringerung zum Abschluss. Brich Pläne nur gemäss einem befugten und aktuellen Verfahren ab oder verschiebe sie. Führe nicht gebundene Eingaben in einen ehrlichen Inventarzustand zurück. Markiere die Ausgabe als nicht verfügbar und wähle einen alternativen Pfad für das *Replacement*, falls einer besteht. Hier geht es um die Verwaltung von *Capability*, nicht um Reparaturdoktrin.

Access kann wechseln

Access ist eine aktive Beziehung zwischen Person, *Ownership* und konfigurierten Regeln.

Structure Profiles bestimmen angebotene *Services*, während *Access Lists* mitbestimmen, wer sie nutzen darf. *Corporation Roles* steuern weitere Handlungen, einschliesslich bestimmter Beziehungen zu Vermögens-

werten der *Corporation* und *Industry*. Der Erfolg eines *Characters* in der Vorwoche beweist nicht, dass all diese Beziehungen heute unverändert sind.

Das erste Signal einer Verschlechterung des *Access* ist oft eine gescheiterte Handlung. Der Pilot kann sehen, aber nicht entnehmen. Er erreicht die *Structure*, kann aber einen *Service* nicht nutzen. Er kann eine Abteilung der *Corporation* öffnen, aber nicht den im Auftrag genannten Ort. Halte die gescheiterte Handlung und ihren Kontext fest, bevor du der Person allgemein fehlenden *Access* zuschreibst.

Genauigkeit ist wichtig, weil Regeln für den *Access* zwischen *Characters*, *Corporations*, *Alliances* und *Everyone* unterscheiden können, wobei spezifischere Einträge gegenüber allgemeineren Vorrang haben. Mehrere einem *Service* zugeordnete Listen können zusammenwirken. Deshalb erklärt eine breite Aussage wie «die *Corporation* ist zugelassen» möglicherweise nicht das Ergebnis für einen bestimmten *Character*. Die verwaltende Person prüft den konfigurierten Pfad; der Nutzer testet die vorgesehene Handlung.

Ein gescheiterter *Access* kann Konzentration offenlegen. Kann eine einzige nicht betroffene verwaltende Person jeden entscheidenden Gegenstand freigeben, kann *Home* während deren verfügbarer Zeit weiterbestehen und dabei allgemein fähig wirken. Die Routine sollte die vorübergehende Abhängigkeit benennen und entweder einen geprüften zweiten Pfad wiederherstellen oder reduzieren, für wen die *Reserve* Unterstützung verspricht.

Löse nicht jedes Problem des *Access*, indem du Befugnisse ausweitest. Ermittle zuerst die kleinste Handlung, welche die Verantwortung benötigt. Lasse eine befugte Person die eng begrenzte *Corporation Role*, das *Profile*, die *Access List* oder den Pfad des *Ownership* berichtigen und prüfe ihn danach mit einer unwichtigen Handlung. Breite Berechtigungen können den ursprünglichen Gestaltungsfehler verbergen und die Konsequenz künftiger Veränderungen vergrößern.

Der Abschluss des *Access* umfasst auch Menschen, die ihn nicht länger behalten sollten. Eine Übertragung, ein Weggang oder eine beendete Gastbeziehung sollte die Prüfung der betroffenen Liste und geteilten Informationen auslösen. Alten *Access* bestehen zu lassen, weil er *Work* gegenwärtig nicht blockiert, bewahrt eine Abhängigkeit ohne verantwortliche Person. Kapitel 15 wird diese Pfade im Verlauf eines bewussten Aufbruchs schliessen; die Routine sollte sie jedes Mal schliessen, wenn ihr Zweck endet.

Die gescheiterte Lagerhandlung aus der Eröffnung beweist weder, dass der Gegenstand verschwunden ist, noch dass jede *Corporation Role* falsch ist. Sie bleibt ein begrenzter *Failure* des geprüften Pfads. Die Gruppe schützt die *Reserve* vor weiteren Verpflichtungen, bis dieser Pfad wiederhergestellt ist oder ein kleineres Versprechen ihn ersetzt.

Reserve zeigt Fragilität

In der *Reserve* werden mehrere langsame *Failures* gemeinsam sichtbar.

Die Pause eines *Service* verzögert eine Ausgabe. Verändert sich der *Access*, wird bestehender Bestand unbrauchbar. Eine alternde Route verzögert importiertes *Replacement*. Fehlt die verantwortliche Person, verzögert sich die nächste Prüfung. Jede Abhängigkeit kann für sich allein erträglich wirken. Erreicht die nutzbare *Reserve* den vereinbarten Auslöser, wird ihre gemeinsame Wirkung zu einer Lücke der *Capability*, in der weniger einfache Entscheidungen bleiben.

Kapitel 12 unterscheidet in der Aufzeichnung der *Reserve* zwischen nutzbaren, reservierten, eintreffenden, in Arbeit befindlichen und unbestätigten Zuständen. *Degradation* verschiebt Bestand zwischen diesen Zuständen, bevor die Menge null erreicht. Ein Gegenstand hinter gescheitertem *Access* wird für den vorgesehenen Nutzer unbestätigt. Eine versprochene Ausgabe aus einem pausierten Auftrag bleibt in Arbeit, nicht nutzbar. Fracht, die jenseits einer nicht unterstützten Route wartet, bleibt eintreffend und ist keine lokale *Reserve*.

Diese Zustandsänderungen sollten die gemeldete *Capability* sofort reduzieren. Der physische Gegenstand kann weiterhin bestehen, und der Auftrag könnte später fortgesetzt werden. Keine der beiden Möglichkeiten beantwortet den heutigen Bedarf. Eine ehrliche *Reserve* beschreibt, was die gegenwärtigen Menschen und Pfade freigegeben können.

Der Horizont des *Replacement* legt die Dringlichkeit offen. Der nächste nutzbare Gegenstand kann von einem wiederhergestellten *Service*, veränderten Berechtigungen und einer nicht erneuerten Route abhängen. Dann kann der langsamste glaubwürdige Pfad den Nutzungszeitraum des verbleibenden Bestands überschreiten. Damit ist der Auslöser erreicht, selbst wenn ein letzter Gegenstand übrig ist.

Fragilität zeigt sich auch, wenn alle Alternativen dieselbe Bedingung teilen. Mehrere Aufträge für *Replacement* in derselben *Structure* sind nicht unabhängig, wenn derselbe *Service* sie trägt. Bestand in mehreren Abteilungen der *Corporation* kann weiterhin von einer einzigen Person mit der passenden *Corporation Role* abhängen. Fracht auf verschiedenen Schiffen kann weiterhin von derselben alternden *Connection* abhängen. Zähle unabhängige Pfade, nicht nur getrennte Objekte.

Die Reaktion beginnt mit dem Schutz dessen, was verbleibt. Stoppe optionalen Verbrauch oder Bewegungen, welche dieselbe *Reserve* beanspruchen. Bestätige, wer den gegenwärtigen Bestand freigegeben kann. Priorisiere die *Capability*, welche Beobachtung, Rückkehr oder die Fähigkeit zur Verringerung anderer Verpflichtungen bewahrt. Beginne den glaubwürdigen Pfad des *Replacement*, solange der letzte nutzbare Gegenstand noch Zeit verschafft.

Alles zu horten ist unnötig. Überschüsse ohne Zweck können die Konzentration in der *Structure* vertiefen und einen späteren *Exit* erschweren. Eine *Reserve* ist dafür da, eine benannte *Capability* während eines angenommenen Zeitraums zu stützen. Material ausserhalb dieses Zwecks sollte nach dem Plan der Gruppe verteilt, bewegt oder freigegeben werden und nicht als vage Beruhigung zählen.

Prüfe die *Reserve* während der gewöhnlichen Routine, ohne den entscheidenden Gegenstand zu verbrauchen. Ein befugter Nutzer kann Sichtbarkeit und Entnahme mit einem unwichtigen Objekt im selben Pfad bestätigen. Dabei prüft die verantwortliche Person, ob die Aufzeichnung auf den richtigen Kontext des *Ownership* verweist und ob der Auslöser des *Replacement* weiterhin eine aktuelle nächste Prüfung besitzt. Diese kleine Probe legt *Failures* bei *Access* und *Handover* offen, während die tatsächliche *Reserve* unberührt bleibt. Sie zeigt auch, ob die behauptete Alternative unabhängig ist oder stillschweigend von derselben Person und Berechtigung abhängt.

Frühzeitig reduzieren

Verringerung ist die Entscheidung, die Wiederherstellung des Feldes nicht weiter zu erschweren.

Wenn sich Signale ansammeln, liegt es nahe, alles zu reparieren, während die gewöhnliche *Work* weiterläuft. Das bewahrt den Anschein und teilt die Aufmerksamkeit. Eine kontrollierte Verringerung verhindert zuerst neue Abhängigkeiten und schliesst oder unterstützt danach die bereits offenen Verpflichtungen.

Beginne unter einem unbestätigten *Service* keine neuen Aufträge. Halte neue Fracht innerhalb eines nicht unterstützten Rückkehrzweigs. Ziehe Versprechen aus der *Reserve* zurück, die von einem ungeprüften Zugriffspfad abhängen. Rufe Arbeitsschiffe zurück, deren Zweck nicht länger wichtiger ist als ihre Freigabe. Diese Handlungen bewahren Wahlmöglichkeiten, ohne zu behaupten, die zugrunde liegende Ursache sei feindlich oder dauerhaft.

Erfasse als Nächstes die Menschen und Vermögenswerte, die bereits gebunden sind. Wer befindet sich noch ausserhalb? Welcher Auftrag ist pausiert? Welche Ausgabe wird erwartet, und welcher Bestand ist unzugänglich? Wer kann jede Bedingung beobachten und wer kann

handeln? Weise eine nächste Prüfung zu oder beende das Versprechen ausdrücklich.

Die Verringerung folgt der Konsequenz, nicht der Bequemlichkeit. Bewahre die *Capability*, die *Chain* zu beobachten, den gegenwärtigen Zustand mitzuteilen, auf entscheidende *Reserve* zuzugreifen und *Work* zu schliessen. Optionale Produktion, Bewegung von Überschüssen und Expansion warten. Die genaue Reihenfolge folgt dem Zweck der Gruppe, doch der Grund für jede bewahrte Funktion muss sichtbar sein.

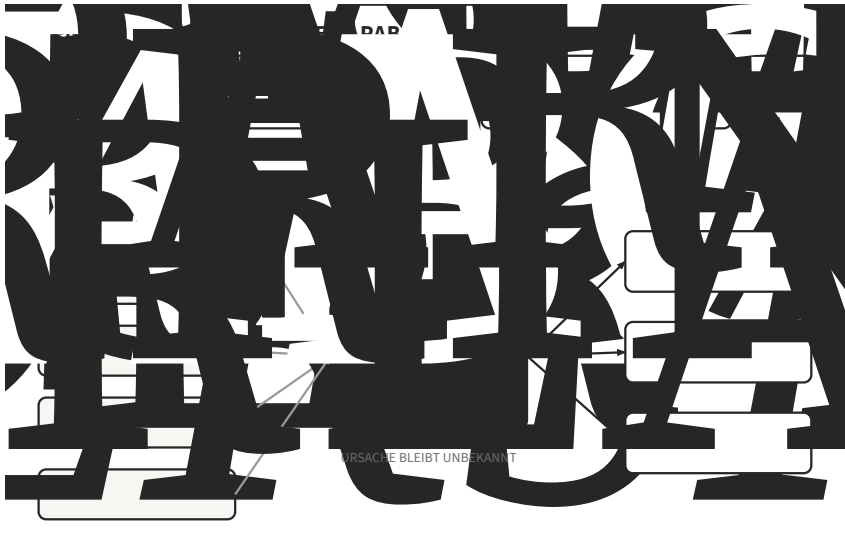
Der kleinere Zustand braucht seine eigene *Baseline*. Halte fest, welche *Services* verfügbar bleiben, welche Nutzer den *Access* geprüft haben, welche *Reserve* nutzbar ist und welche *Connections* ausstehende Rückkehr tragen. Entferne inaktive Versprechen aus dem aktuellen *Handover*. Andernfalls könnte die nächste Schicht ihre *Work* vom Anschein eines gewöhnlichen *Home* aus neu beginnen statt von der reduzierten Wirklichkeit.

Kehrt die *Capability* zurück, erfolgt die Ausweitung Abhängigkeit für Abhängigkeit. Beobachte den *Service*, prüfe den Nutzungspfad, bestätige die *Reserve* und weise Verantwortliche zu, bevor neue *Work* geöffnet wird. Die Wiederherstellung eines Signals entfernt nicht die anderen *Unknowns*, die sich daneben angesammelt haben.

Schrumpft die *Capability* weiter, bereitet die Verringerung die nächste Entscheidung vor. *Home* muss möglicherweise *Services* schliessen, Wert verschieben oder die Verpflichtung ganz beenden. Das sind geplante Entscheidungen, solange *Access* und Zeit verbleiben. Kapitel 15 folgt diesem Pfad. Es geht nicht um eine taktische Evakuierung unter Angriff, sondern um das bewusste Ende, das durch früh erkanntes Scheitern möglich wird.

Kein dramatischer Anblick war in der Eröffnungsszene nötig. Es genügte die Verbindung aus pausiertem Auftrag, Zustand *Low Power*, verändertem *Access* und alternder Aufzeichnung der Unterstützung. Die Gruppe stoppt neue *Work*, erfasst offene Verpflichtungen und stellt eine

kleinere *Baseline* wieder her. Sie hat verhindert, dass eine verschlechterte *Capability* den Namen *Home* ausleiht.



DEGRADATION ZÄHLT DURCH NICHT MEHR GETRAGENE CAPABILITY

REPARIEREN · VERRINGERN · VERLASSEN BLEIBEN GLEICHWERTIG

Abbildung AFH-B2-FIG-014 — Degradation wird durch getragene Capability begrenzt.

Degradation zählt durch die *Capability*, die sie nicht mehr trägt, nicht durch eine erfundene Ursache.

Feldprinzip

Lies *Failure* als Veränderung der unterstützten *Capability*.

Beobachte *Power State*, Zustand des *Service*, Zugriffspfade, Zustände der *Reserve* und *Ownership*, ohne ihre Ursache zu erfinden. *Full Power*, *Low Power* und *Abandoned* sind begrenzte Zustände einer *Structure*, keine allgemeinen Urteile. Ein eingebautes Modul ist kein *Service*, der online ist, und ein vorhandener Gegenstand ist bei gescheitertem *Access* keine nutzbare *Reserve*.

Verbinde jedes Signal mit der davon betroffenen *Work*. Stoppe neue Verpflichtungen, erfasse das bereits Offene, bewahre Beobachtungs- und

Freigabepfade und errichte eine kleinere aktuelle *Baseline*. Stelle den Umfang nur anhand frischer Evidenz und geprüftem *Access* wieder her. Die ruhige *Structure* aus der Eröffnung war bereits von ihren Versprechen abgewichen. Diese Abweichung vor aktivem feindlichem Druck zu erkennen bewahrte die Wahl, zu reparieren, zu reduzieren oder zu gehen. Das nächste Kapitel nimmt die letzte Möglichkeit ernst und zeigt, wie *Home* enden kann, ohne zur Improvisation zu werden.

Frühe Verringerung löst nicht jede Abhängigkeit. Dadurch beansprucht neue *Work* keine Unterstützung, die nicht mehr besteht. Die Verringerung hält die verbleibende *Capability* lesbar und gibt befugten Verantwortlichen Zeit, den nächsten begrenzten Zustand zu wählen. Dieser Zeitraum ist der praktische Unterschied zwischen einer als Evidenz verwalteten *Degradation* und einem als Krise erlebten *Failure*.

Kapitel 15 - Bewusst aufbrechen

Der Aufbruch wurde an einem gewöhnlichen Tag beschlossen.

Die *Chain* war beobachtet worden. Auch die *Structure* blieb zugänglich; ihre unterstützten *Services* waren bekannt, und kein aktiver feindlicher Druck bestimmte den Zeitplan. Weiterhin konnten die Piloten entscheiden, welche Schiffe zuerst bewegt wurden, welche Aufträge endeten und welche Aufzeichnungen bis zum letzten *Handover* verfügbar blieben.

Diese Ruhe liess den Aufbruch verfrüht erscheinen.

Home funktionierte noch, doch die Gruppe konnte den versprochenen Umfang im kommenden Zeitraum nicht länger aufrechterhalten. Weniger Menschen hatten die routinemässige Verantwortung angenommen. Das *Replacement* hing von einem *Service* ab, dessen Unterstützung enger geworden war. In einer *Structure* hatte sich zu viel Wert für einen Zweck angesammelt, den die Gruppe nicht mehr fortführen wollte. Zu bleiben würde einen bekannten Niedergang in einen späteren Notfall verwandeln.

Ein bewusster *Exit* beginnt, bevor das Gehen die einzige mögliche Reaktion ist. Er beendet neue Verpflichtungen und trennt, was bewegt werden muss, von dem, was bleiben darf. Danach schliesst er Aufträge und *Access* und prüft das Ergebnis. Wird die *Structure* stillgelegt, gehört diese Handlung als exponierter Teil zum *Exit*. Sie ist kein Knopf, der den ganzen Aufbruch ausführt.

Exit beginnt früh

Die erste Entscheidung für den *Exit* betrifft nicht den Beginn der Stilllegung. Sie bestimmt, was *Home* ab jetzt nicht mehr verspricht.

Eine Operation kann aufbrechen, während Schiffe, *Services* und Lagerung noch funktionieren. Diese verbleibende *Capability* ist die Ressour-

ce, die Ordnung ermöglicht. Piloten können Wert entsprechend seinen Konsequenzen bewegen, anstatt ihn in einer improvisierten Konzentration zusammenzuführen. Verantwortliche für *Industry* können Aufträge schliessen oder ablehnen, bevor weitere Eingaben gebunden werden. Verantwortliche für die Verwaltung können den *Access* prüfen, solange sowohl alte als auch empfangende Eigentümer verfügbar sind. Beobachtungen der *Chain* können geplante Bewegungen stützen, ohne sie garantieren zu müssen.

Die Entscheidung braucht einen begrenzten Grund. «Vielleicht gehen wir eines Tages» kann den nächsten Auftrag nicht stoppen. «Die Gruppe beendet dieses *Home*, weil sie die angenommenen Verpflichtungen für *Service*, *Handover* und *Reserve* nach dem benannten Zeitraum nicht aufrechterhalten kann» verbindet den Aufbruch mit der gegenwärtigen *Capability*. Zugleich verhindert diese Aussage, dass jede vorübergehende Unannehmlichkeit die grössere Frage erneut öffnet.

Lege einen Endzustand fest, bevor einzelne Bewegungen gewählt werden. Behält die *Corporation* die *Structure* für einen kleineren Zweck, überträgt sie die Verantwortung durch ein befugtes Verfahren oder legt sie die *Structure* still? Welche Menschen und Vermögenswerte sollen nicht länger von diesem *Home* abhängen? Welche Aufzeichnungen müssen nach der Bewegung des letzten Schiffs verfügbar bleiben? Die Antworten bestimmen, ob eine Aufgabe geschlossen, übertragen oder beibehalten wird.

Auch der *Exit* besitzt eine Grenze. Dieses Kapitel behandelt einen geplanten Aufbruch, solange die Gruppe noch beobachten, wählen und ein *Handover* ausführen kann. Taktische Evakuierung, Verteidigung der *Structure* und die Reaktion auf eine Vertreibung bleiben ausserhalb seines Umfangs. Aktiver feindlicher Druck verändert Zeit, Informationen und Prioritäten und gehört zu *Survivor*. Eine Krise als «schnelleren geplanten *Exit*» zu bezeichnen würde diesen Unterschied verbergen.

Ein früher Aufbruch kann verschwenderisch wirken, weil funktionierende Kapazität verbleibt. Diese Kapazität wird nicht verschwendet. Sie wird für einen sauberen Abschluss eingesetzt. Die Fähigkeit, eine Route zu wählen, Berechtigungen freizugeben und Fracht entgegenzunehmen, ist genau das, was bei einem späten Aufbruch zuerst verloren geht.

Benenne eine verantwortliche Person für den *Exit* und weitere für jeden Abhängigkeitsstrom. Eine Person pflegt den gesamten Endzustand. Andere können Menschen und Schiffe, *Industry Work*, eingelagerte Vermögenswerte, geteilte Informationen, *Services* der *Structure* und den Abschluss des *Access* verantworten. Mehrere Funktionen können bei einem *Character* liegen, doch keine darf im allgemeinen Satz «wir ziehen aus» verschwinden.

Der gewöhnliche Tag aus der Eröffnung wird zum ersten Tag im *Exit*, nicht weil etwas Dramatisches geschah, sondern weil die Gruppe beschloss, die noch sichtbaren Möglichkeiten zu bewahren.

Ein Plan für den *Exit* sollte seine unumkehrbaren Punkte zeigen. Die Ankündigung des Aufbruchs kann verändern, ob Menschen weitere Vermögenswerte bringen. Der Abbruch eines Auftrags kann den Zustand des Materials verändern. Die Bewegung des letzten lokalen Scanners kann eine beobachtende *Capability* entfernen. Der Beginn der Stilllegung legt einen Prozess der *Structure* offen, dessen Abschluss weiterhin eine Abholung verlangt. Setze unmittelbar vor jede dieser Veränderungen eine Prüfung, solange der vorherige Zustand noch bewahrt werden kann.

Umkehrbarkeit bedeutet keine Unentschlossenheit. Die Gruppe genehmigt den Endzustand einmal und prüft danach, ob die aktuelle Evidenz weiterhin jeden Schritt auf dem Weg dorthin stützt. Eine veränderte *Chain* kann die Reihenfolge der Bewegungen ändern, ohne den Grund für den Aufbruch neu zu öffnen. Ein fehlender Empfänger kann eine Vermögensklasse verzögern, ohne neue Produktion zu erlauben. Diese

Trennung hält den Plan anpassungsfähig, ohne dass Bequemlichkeit seine Grenze auflöst.

Neue Verpflichtungen stoppen

Ein Aufbruch scheitert, wenn neu hinzukommende Verpflichtungen die abgehende Kapazität überholen.

Füge keine *Work* mehr hinzu, deren Abschluss über den Plan für den *Exit* hinausreicht. Ressourcenoperationen können Fracht erzeugen, die noch Transport und Lagerung braucht. Weitere *Industry Jobs* können Materialien an einen *Service* binden, dessen Ende geplant ist. Die Gewährung von Gastzugang schafft einen weiteren Pfad, den Verantwortliche später entfernen müssen. Schiffe, die in *Home* eintreffen, ergänzen die Rückkehrbilanz um weitere Hüllen und Piloten.

Der Stopp verlangt nicht, jede gegenwärtige Handlung sofort zu beenden. Er markiert den Punkt, nach dem jede neue Verpflichtung einen ausdrücklichen Nachweis braucht, dass sie innerhalb des Endzustands geschlossen wird. Eine kleine Aufgabe, die den Aufbruch abschliesst, darf fortgeführt werden. Optionale Produktion oder Ansammlung erhält nicht allein deshalb eine Erlaubnis, weil die *Structure* online bleibt.

Erstelle eine Aufzeichnung der offenen Verpflichtungen. Beginne mit Schiffen und Menschen ausserhalb und ergänze installierte oder vorbereitete Aufträge sowie Material im Transport. Halte *Services* mit davon abhängiger *Work* fest. Nimm Gastzugänge, geteilte Ordner, Container und ungelöste Pfade des *Ownership* auf. Benenne für jeden Eintrag Abschlussbedingung, verantwortliche Person und nächste Prüfung. Die Aufzeichnung ist kein vollständiges Vermögensinventar; sie erfasst jene Versprechen, die den *Exit* offen halten können.

Verhindere danach stille Ergänzungen. Die Menschen, welche Aufträge genehmigen, gemeinsame Vermögenswerte bewegen oder *Access* gewähren, müssen wissen, dass sich der Zustand verändert hat. Eine Notiz, welche ausschliesslich die für den *Exit* verantwortliche Person

sieht, kann Handlungen anderswo nicht steuern. Die Ankündigung nennt die gestoppten neuen Verpflichtungen und das Verfahren zur Genehmigung einer notwendigen Ausnahme.

Ausnahmen sollten den *Exit* verkleinern und nicht darin ein neues *Home* schaffen. Ein Auftrag zur Herstellung eines für die letzte Bewegung benötigten Werkzeugs kann gerechtfertigt sein, wenn Unterstützung durch den *Service*, Entgegennahme und Abschluss in den Plan passen. Ein gewinnbringender Auftrag ohne Funktion für den Aufbruch fügt zeitliche und materielle Konsequenzen hinzu, selbst wenn er abgeschlossen werden könnte. Der Unterschied liegt im Zweck, nicht im Ertrag.

Bestehende Aufträge brauchen sachliche Zustände. Einige können abgeschlossen werden, bevor *Services* schliessen. Andere können pausiert sein und eine Entscheidung verlangen. Die Entfernung eines passenden *Service* oder die Zerstörung der *Structure* kann betroffene *Work* abbrechen; nimm nicht an, ein offener Auftrag werde während der Stilllegung stillschweigend zur Ausgabe. Die befugte verantwortliche Person prüft den aktuellen Auftrag und entscheidet sich für Abschluss, gegebenenfalls Abbruch oder einen anderen unterstützten Pfad.

Wenn der Zustrom stoppt, wird die Bilanz des Aufbruchs kleiner. Jeder geschlossene Auftrag, jedes eingetroffene Schiff und jeder entfernte Zugriffspfad bedeutet eine Bedingung weniger, welche der letzte Tag tragen muss.

Die Aufzeichnung braucht Zustände, die nicht mit Fortschrittsprozenten verwechselt werden können. Ein Eintrag ist offen, gehalten, übertragen, geschlossen oder als verbleibend angenommen. «Fast abgeschlossen» kann die einzelne Berechtigung oder den Empfänger verbergen, die weiterhin eine ganze Abhängigkeit offen halten. Ein gehaltener Eintrag nennt den Grund, weshalb keine Handlung erfolgt, und die Evidenz, die ihn freigeben wird. Geschlossen bedeutet, dass der

letzte Nutzer oder Eigentümer das Ergebnis bestätigt hat, nicht nur, dass eine Bewegung versucht wurde.

Prüfe die Bilanz vor der Genehmigung jeder Ausnahme. Fügt eine vorgeschlagene Handlung einen weiteren offenen Eintrag hinzu, ohne einen folgenreicheren zu schliessen, wächst der *Exit*. Das kann gelegentlich notwendig sein, verlangt aber eine ausdrückliche verantwortliche Person und ein Ende, statt sich die allgemeine Befugnis des Aufbruchs auszuleihen.

Vermögensklassen trennen

Vermögenswerte verlassen *Home* gemäss Konsequenz, *Ownership* und beabsichtigtem Endzustand.

Eine einzige Inventarsumme kann diese Reihenfolge nicht liefern. Trenne zuerst nutzbare *Reserve*, persönliches Eigentum und Eigentum der *Corporation*. Unterscheide Eingaben für *Industry* von fertigen Ausgaben. Halte installierte oder an die *Structure* gebundene Komponenten, entbehrliche Überschüsse und Aufzeichnungen fest. Ergänze die Menschen und Schiffe, deren Standort weiterhin Teil der Bilanz ist.

Über persönliche Vermögenswerte entscheiden ihre Eigentümer, doch der *Exit* darf nicht stillschweigend versprechen, sie zu bewegen. Informiere die Eigentümer über den Endzustand. Nenne zugleich den Zeitpunkt, ab dem der gegenwärtige Lagerpfad nicht länger unterstützt wird. Halte von allem, was verbleibt, nur die operative Konsequenz fest; verwandle den gemeinsamen *Exit* nicht in eine Überwachung privater Inventare.

Vermögenswerte der *Corporation* brauchen einen befugten Eigentümer und ein Ziel. Das Ziel muss nutzbaren *Access* umfassen, nicht nur den Namen einer anderen *Structure*. Prüfe empfangende Hangars und Container mit unwichtigen Gegenständen, bevor entscheidender Bestand bewegt wird. Bewahre den Kontext des *Ownership*, den die erwarteten Nutzer des Ergebnisses benötigen.

Für das *Replacement* entscheidende Vermögenswerte bewegen sich in einer Abfolge, welche die *Capability* für den *Exit* erhält. *Scanning*, Routenbeobachtung, Transport, *Access* und Abschluss können bis spät im Plan von lokalen Schiffen oder Werkzeugen abhängen. Werden alle entscheidenden Gegenstände zuerst bewegt, kann ein ausgedünntes *Home* den verbleibenden Aufbruch nicht abschliessen. Bleiben alle bis zuletzt, konzentriert sich die Konsequenz. Bestimme einen Arbeitssatz und einen wiederhergestellten Satz, damit auf beiden Seiten der Bewegung *Capability* besteht.

Vermögenswerte von *Industry* verlangen den Abschluss des Auftragspfads. Reservierte, aber nicht installierte Eingaben können in einen zugänglichen Inventarzustand zurückkehren. Installierte *Work* braucht eine aktuelle Entscheidung ihrer verantwortlichen Person. Eine fertige Ausgabe muss entgegengenommen und einem Ziel zugewiesen werden. Ein *Blueprint* darf nicht als frei gezählt werden, nur weil die Ausgabe noch wartet.

Überschüsse sollten ihre Priorität nicht aus der Menge beziehen. Material ohne fortbestehenden Zweck kann bewegt werden, wenn Kapazität und Evidenz der Route es erlauben, durch eine befugte Entscheidung übertragen werden oder als angenommener Verlust zurückbleiben. Der Plan für den *Exit* benennt diese Wahl. Optionales Volumen darf Menschen, entscheidende *Reserve* oder den Abschluss von *Access* nicht verzögern.

Diese Wiederherstellungsgrenze gilt auch für zurückgelassene Vermögenswerte. «Für *Asset Safety* zurücklassen» ist hier kein allgemeines Ziel im *Exit*. Jede mit der *Structure* verbundene Vermögensklasse, die zurückbleibt, verlangt ausdrückliche Annahme.

Die Bewegungsaufzeichnung folgt Kapitel 11. Sie begrenzt gleichzeitige Konsequenzen, berücksichtigt Signale von Masse und Lebensdauer der *Connection*, bestätigt die Entgegnahme und aktualisiert Inventare an

Quelle und Ziel. Die Ankündigung des Aufbruchs reserviert die *Chain* nicht.

Jede Klasse braucht einen Punkt, nach dem ihr alter Ort nicht länger als unterstützt gilt. Bis dahin pflegt die verantwortliche Person *Access* und Zustand. Danach trägt alles Verbleibende ein ausdrückliches Ergebnis: spätere Abholung nach einem neuen Plan, befugte Übertragung oder angenommener Rest. Damit halten die letzten wenigen Objekte nicht das gesamte *Home* gedanklich offen, nachdem sein operativer Zweck geendet hat.

Lass einfache Fracht folgenreiche Fracht nicht verdrängen, nur weil sie schnell bewegt werden kann. Die Reihenfolge schützt zuerst Menschen und die für die Fortführung des Aufbruchs nötigen Werkzeuge, stellt danach am Ziel die *Capability* wieder her und verwendet erst dann verbleibende Routenkapazität für optionales Volumen. Veränderungen der aktuellen *Chain* können diese Reihenfolge anpassen, doch jede Änderung sollte die für den Abschluss erforderlichen Funktionen bewahren.

Stilllegung schafft *Exposure*

Stilllegung ist kein Verschwinden.

Das gegenwärtige Verfahren für *Upwell Structures* beschränkt die Handlung der Stilllegung auf den *CEO* oder einen *Director* der besitzenden *Corporation*. Der Prozess wird über einen Zeitraum angekündigt, statt sofort abgeschlossen zu sein. Vor dem Abschluss kann er abgebrochen werden, wodurch der Prozess zurückgesetzt wird. Diese Tatsachen machen Befugnis, Zeitpunkt und Bestätigung zu Teilen des Plans für den *Exit*.

Der befugte *Character* sollte den Prozess nicht allein deshalb beginnen, weil «das meiste weg ist». Bestätige die Entscheidungen zu den Vermögensklassen, offene Aufträge, verbleibende Menschen, empfangenden *Access* und die verantwortliche Person für das Endergebnis. Halte fest,

wer die Entscheidung abbrechen oder abschliessen kann und welche Evidenz dies rechtfertigen würde.

Während der Stilllegung bleiben die *Structure* und alles, was noch von ihr abhängt, veränderten Bedingungen ausgesetzt. *Access* kann sich ändern. Routen können sich schliessen. Ereignisse ausserhalb des geplanten Prozesses können die *Structure* nicht verfügbar machen. Verteidigung während dieses Zeitraums wird hier nicht gelehrt. Das exponierte Intervall liefert einen weiteren Grund, Abhängigkeiten vor der Aktivierung zu reduzieren.

Auch der Abschluss hinterlässt *Work* im Raum. Nach dem offiziellen Verfahren bleibt die *Structure* selbst zur Aufnahme verfügbar. Ein Container kann *Modules*, Ladungen, den *Quantum Core* und *Fighters* zur Abholung enthalten. *Rigs* kehren nicht auf dieselbe Weise zurück. Das genaue aktuelle Ergebnis muss vor der Handlung geprüft werden. Für die Planung gilt: Jemand braucht ein Schiff, *Access*, Zeit und eine Route. Damit lässt sich entgegennehmen, was der Prozess hinterlässt.

Neben diesem Ergebnis gilt dieselbe Ausnahme von *Asset Safety*. Eine Stilllegung verwandelt ungelöste Lagerung nicht in eine garantierte private Zustellung.

Behandle die Abholung als eigene Operation. Benenne die befugte verantwortliche Person der Entscheidung, den Beobachter, den empfangenden Piloten, das Ziel und die Bestätigung. Lege fest, was die Gruppe tut, wenn die vorgesehene Route oder der Empfänger beim Abschluss nicht verfügbar ist. Bewahre weder die einzige Kopie der Aufzeichnung noch den benötigten *Access* oder die empfangende *Capability* in der endenden *Structure* auf.

Vermeide genaue Timer im beständigen Manuskript, wenn sie für die Lektion nicht notwendig sind. Die handelnde Person verwendet das aktuelle offizielle Verfahren und den sichtbaren Zustand des Clients. Das Buch bewahrt die Abhängigkeiten: Befugnis, angekündigter Prozess,

exponierter Zeitraum, Abholung und Ergebnis der Vermögenswerte im *Wormhole*.

Die Stilllegung erfolgt deshalb spät im *Exit*. Sie beendet ein einzelnes Objekt vom Typ *Structure*, nachdem die Gruppe bereits die meisten Verpflichtungen beendet hat, die dieses Objekt zu *Home* machten.

Vergleiche den Prozess bei jeder Prüfung mit dem Endzustand. Bestätige, dass kein neuer Auftrag und kein neues Versprechen von *Access* hinzugekommen ist, die Abholung weiterhin eine verantwortliche Person besitzt und die empfangende Lagerung noch funktioniert. Scheitern diese Bedingungen, entscheidet die befugte verantwortliche Person, ob der umgebende Plan berichtigt oder der Prozess abgebrochen wird, so lange dies noch möglich ist. Eigendynamik ist keine Evidenz dafür, dass der Abschluss weiterhin die am besten unterstützte Handlung darstellt.

Die letzte Abholung sollte wie jedes andere *Handover* eine Rückbestätigung besitzen. Der empfangende Pilot nennt, was abgeholt werden kann, das Ziel und die Bedingung, welche die Verantwortung beendet. Die für den *Exit* verantwortliche Person hält fest, was tatsächlich entgegengenommen wurde und was verblieb. Damit ersetzt das sichtbare Verschwinden der *Structure* nicht den Bericht über die Vermögenswerte.

Geteilten Access schliessen

Ein *Exit* bewegt Informationen und Berechtigungen ebenso wie Vermögenswerte.

Routen, taktische *Locations*, *Sites* in Bearbeitung und historische *Observations* können in *Shared Location Folders* verbleiben. Der *Access* zu einer *Structure* kann über *Profiles* und *Access Lists* gewährt werden. *Corporation Roles* können Handlungen an Vermögenswerten oder in *Industry* erlauben. Gastbeziehungen können von Regeln abhängen, deren ursprünglicher Eigentümer nicht länger anwesend ist.

Beginne mit einer Einteilung des verbleibenden Zwecks. Bewahre Informationen, welche Menschen bei ihrer Rückkehr oder bei der Abholung des Endergebnisses benötigen. Übertrage fortbestehende Aufzeichnungen an ihre nächste Gruppe oder Referenz. Abgelaufene Arbeitsinformationen sollten geschlossen werden. Alles bei der Ankündigung zu entfernen würde den *Exit* blind machen; alles auf unbestimmte Zeit bestehen zu lassen würde *Access* ohne Zweck bewahren.

Jeder geteilte Gegenstand oder Zugriffspfad braucht einen letzten Nutzer und eine Endbedingung. Der Rückkehrordner bleibt online, bis alle benannten Piloten erfasst sind. Die empfangende Aufzeichnung wird massgeblich, bevor die alte schliesst. Vorübergehender Gastzugang endet nach der angenommenen Bewegung und ihrer Bestätigung. Eine für die letzte Abholung verwendete *Corporation Role* wird nach der Entgegennahme überprüft.

Der Abschluss folgt der Genauigkeit. *Access Lists* können breite und konkrete Einträge verbinden, und spezifischere Einträge können allgemeine übersteuern. Die Entfernung eines Namens beweist nicht, dass der gesamte Nutzungspfad geschlossen ist, wenn Einträge für *Corporation*, *Alliance* oder *Everyone* weiterhin eine relevante Beziehung gewähren. Eine befugte verwaltende Person prüft das konfigurierte Ergebnis, während gegebenenfalls ein für den Test verwendeter *Character* den beabsichtigten Verlust oder Erhalt des *Access* bestätigt.

Lösche keine Evidenz, welche die Gruppe weiterhin zur Erklärung ihrer Entscheidungen braucht. Eine historische Routenaufzeichnung kann in der *Shared Map* verbleiben, ohne sich als aktuelle *Chain* darzustellen. Ein Inventar kann bewahren, was als verbleibend angenommen wurde, ohne dessen Wiederherstellbarkeit zu behaupten. Der Bericht zum *Exit* unterscheidet archivierte Geschichte von aktiver Anweisung.

Das *Ownership* der Werkzeuge ist wichtig. Ein persönliches Dokument oder Konto kann für die Gruppe verschwinden, wenn sein Eigentümer geht, selbst wenn der *Access* während des Aufbruchs offen blieb. Über-

trage die benötigte Aufzeichnung in einen befugten fortbestehenden Kontext, bevor du den alten Pfad schliesst. Das Feldhandbuch schreibt nicht vor, welches Werkzeug sie aufnehmen soll; es verlangt vom nächsten Eigentümer die Bestätigung ihrer Verfügbarkeit.

Entferne schliesslich vorübergehende Berechtigungen und halte Datum und Umfang des Abschlusses fest. Behalte keine breiten Rechte als undokumentierte Rückfallebene. Wird später *Access* benötigt, sollte er als neuer ausdrücklicher Zweck unter dem dann bestehenden Zustand beginnen.

Ordne den Abschluss von aussen nach innen. Beende optionale Gast- und Arbeitszugänge, sobald ihre letzte Handlung bestätigt ist. Bewahre die engen Pfade für Rückkehr, Entgegennahme und Abholung der *Structure*, bis ihre Verantwortlichen sie schliessen. Übertrage die massgebliche Aufzeichnung, bevor der alte Bearbeitungspfad verschwindet. Diese Reihenfolge hält den *Exit* beobachtbar, ohne jede historische Berechtigung bis zur letzten Minute aktiv zu lassen.

Prüfe das Ergebnis nach der Entfernung wenn möglich von beiden Seiten. Eine verwaltende Person sieht den konfigurierten Zustand; der frühere Nutzer bestätigt, dass der alte Pfad nicht mehr funktioniert oder der beibehaltene Pfad weiterhin seinen Zweck erfüllt. Eine Konfigurationsänderung ohne Prüfung des Nutzungspfads ist eine Handlung, aber noch kein Abschluss.

Ende überprüfen

Der Aufbruch endet, wenn kein verborgenes Versprechen mehr auf das alte *Home* verweist.

Die Überprüfung beginnt bei den Menschen. Erfasse die im Bericht zum *Exit* genannten Piloten und Schiffe. Bestätige, ob alle ihr Ziel erreicht haben, einem neuen Eigentümer übergeben wurden oder als angenommene offene Ausnahme verbleiben. «Alle sollten draussen sein» ist keine Evidenz.

Folge danach der *Work*. Kein Auftrag sollte weiterhin als erwartete Ausgabe gezählt werden, wenn er nicht eine verantwortliche Person und eine unterstützte Entgegennahme besitzt. Nicht länger versprochene *Services* sind als geschlossen festgehalten. Eingaben und Ausgaben besitzen einen abschliessenden Zustand des *Ownership*. Die Aufzeichnung des *Replacement* beschreibt die fortbestehende *Capability* ausserhalb der alten *Structure*.

Folge danach den Vermögenswerten nach Klasse. Bestand der *Corporation* hat geprüften *Access* erreicht oder wurde als verbleibend angenommen. Persönliche Eigentümer haben die Grenze erhalten, und kein gemeinsamer Plan garantiert stillschweigend ihr Eigentum. Installierte Komponenten und das Ergebnis jeder Stilllegung besitzen benannte Resultate der Abholung. Die Inventare an *Source* und Ziel stimmen bezüglich der erfolgten Bewegung überein.

Als Nächstes folgen die Informationen. Aktive *Shared Locations*, die für die Rückkehr benötigt wurden, sind nach ihrem letzten Nutzer geschlossen oder übergeben worden. Historische Aufzeichnungen stellen alte *Connections* nicht mehr als aktuell dar. *Profiles*, *Access Lists*, Gasteinträge und vorübergehende *Corporation Roles* wurden von befugten Personen geprüft. Fortbestehende Eigentümer haben die übernommenen Aufzeichnungen bestätigt.

Behandle die *Structure* als letztes Objekt, nicht als einziges. Eine beibehaltene *Structure* erhält eine neue *Baseline*, welche ihre *Services*, ihren *Access*, ihre Vermögenswerte, Verantwortlichen und nächste Prüfung des Endes nennt. Eine befugte Übertragung endet, wenn der neue Eigentümer die Verantwortung bestätigt. Nach einer Stilllegung werden Abholung und Ergebnis der Vermögenswerte im *Wormhole* anhand von Beobachtung statt Erwartung festgehalten.

Die Überprüfung sollte angenommene Reste einschliessen. Ein sauberer *Exit* verlangt nicht die Behauptung, jeder Gegenstand sei die Bewegung wert gewesen. Er verlangt Wissen darüber, was bewusst

zurückgelassen wurde, in welchem Kontext des *Ownership* und mit welcher erwarteten Konsequenz. Unbestätigte Reste bleiben ein *Unknown* und halten den entsprechenden Teil des Aufbruchs offen.

Schliesse die Verantwortung für den *Exit* mit einer abschliessenden Aussage: Das frühere *Home* unterstützt keine nicht aufgeführten Menschen, *Work*, Vermögenswerte, Zugriffspfade oder Aufzeichnungen. Jede verbleibende Ausnahme besitzt ausserhalb des geschlossenen Plans eine eigene verantwortliche Person. Die Aussage wird durch ihre *Sources* und Zeit begrenzt; sie ist keine Garantie für die Zukunft des Systems.

Ein zweiter Leser sollte die abschliessende Aussage bis zu ihrer Evidenz zurückverfolgen können. Er muss nicht jede Bewegung wiederholen. Er prüft, ob offene Einträge Verantwortliche, geschlossene Einträge eine Bestätigung und angenommene Reste eine Benennung besitzen. Jede Bedingung, die nur von der Erinnerung der für den *Exit* verantwortlichen Person abhängt, kehrt vor der Abschlussfreigabe des Aufbruchs in die Bilanz zurück.

Der gewöhnliche Tag endet ohne Drama. Die Gruppe nutzte funktionierende *Capability*, um Abhängigkeiten zu schliessen, nicht um die Entscheidung zu verschieben. Als die *Structure* aufhörte, *Home* zu sein, war das meiste von *Home* bereits bewegt, beendet oder ehrlich zurückgelassen worden.

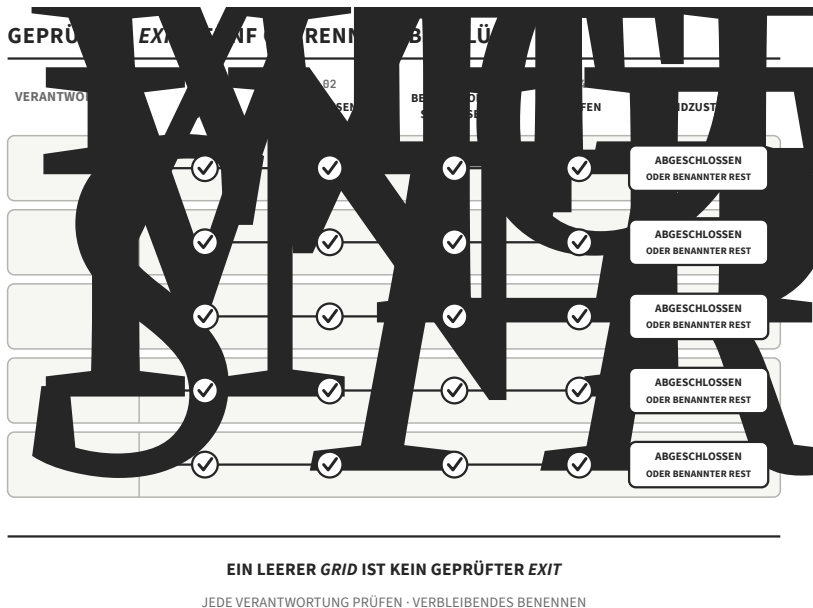


Abbildung AFH-B2-FIG-015 — Ein geprüfter *Exit* schliesst getrennte Verantwortungen ab. Ein bewusster *Exit* schliesst Menschen, *Work*, Vermögenswerte, Information und *Structure* als getrennte Verantwortungen ab.

Feldprinzip

Brich auf, solange der Aufbruch noch eine Abfolge von Entscheidungen ist.

Stoppe neue Verpflichtungen, erfasse offene *Work* und trenne Vermögenswerte nach *Ownership*, Zweck und Konsequenz. Bewahre die *Capabilities*, die für Beobachtung, Bewegung, Entgegennahme und Abschluss nötig sind. Behandle die Stilllegung als befugte, angekündigte und exponierte Operation, deren Ergebnis weiterhin eine Abholung verlangt.

Schliesse *Shared Locations*, *Profiles*, *Access Lists*, *Corporation Roles* und Gastzugänge gemäss ihrem letzten gültigen Zweck. Überprüfe Menschen, *Work*, Vermögenswerte, Informationen und die *Structure*

getrennt. Benenne, was verbleibt, statt fehlende Evidenz als sauberen *Exit* zu bezeichnen.

Die Entscheidung aus der Eröffnung fiel vor der Krise und bewahrte deshalb die Kontrolle. Diese Grenze schliesst die Lehre von *Explorer* über ein gepflegtes *Home* ab. Kapitel 16 kehrt zur zweiten Nacht des Piloten zurück und fragt, welche *Capability* verbleibt, wenn der nächste Druck keine gewöhnliche *Degradation* mehr ist, sondern eine bewusste Handlung durch jemand anderen.

Kapitel 16 - Bereit für *Survivor*

Dasselbe System empfängt den Piloten in der zweiten Nacht erneut.

Die *Structure* steht noch dort, wo der vorherige Flug endete. In ihrem Innern wartet das Scanschiff. Auch die alte *Chain* ist sichtbar, und die Nachricht neben einer *Connection* lautet noch immer *bei der Prüfung ruhig*.

Nichts an diesem Bild muss falsch sein.

Diesmal verlangt der Pilot vom Bild nicht, allein wegen seiner Vertrautheit aktuell zu sein. Das Scanschiff besitzt einen Kontext des *Ownership*, einen Zugriffspfad, einen Zweck und einen Zustand des *Replacement*. Zur *Structure* gehören *Services*, Verantwortliche für die Wartung und beobachtete Bedingungen statt einer allgemeinen Eigenschaft namens verfügbar. Die *Chain* besitzt *Sources*, *Timestamps*, Verantwortliche und *Unknowns*. Neben der Nachricht stehen ein Berichterstatter und eine Grenze. Jede Verpflichtung hat eine beabsichtigte Art zu enden.

Das Feld ist nicht gewiss geworden. Es ist lesbar geworden.

Genau diese Lesbarkeit ist die *Capability*, für deren Aufbau *Explorer* geschrieben wurde. Bevor die Leser aktiven feindlichen Druck untersuchen, müssen sie unterscheiden können, was *Home* unterstützt, was sich verschlechtert hat, was *Unknown* bleibt und was die Gruppe bewusst beendet hat. Andernfalls wird jede Unterbrechung wie ein Feind und jedes vertraute Objekt wie Sicherheit wirken.

***Capability* vor Konflikt**

Ein Konflikt vereinfacht die Abhängigkeiten von *Home* nicht. Er übt zusätzlichen Druck auf sie aus.

Eine Gruppe, die nicht sagen kann, welche Route eine Rückkehr stützt, wird nicht erkennen, was eine unerwartete Routenänderung verändert hat. Verbindet sie *Ownership*, *Access* und Fähigkeit, wird sie nicht wissen,

ob eine gescheiterte Handlung aus Konfiguration, Abwesenheit oder Einwirkung von aussen folgt. Zählt sie versprochene Ausgabe als *Reserve*, begegnet sie begegnet Druck mit einer *Capability*, die sie nie besass. Operative Klarheit muss deshalb bestehen, bevor der Konflikt zum Thema wird. Sie beginnt mit einer Verpflichtung. Ein Zweck ist eng genug, um zu enden. Jemand hat die Verantwortung angenommen. Der aktuelle Zustand ist für jene Menschen sichtbar, von denen eine Handlung erwartet wird. Die erste Konsequenz des *Failure* ist bekannt, und ein vorbereitetes Ende verhindert, dass Gewohnheit zur dauerhaften Verpflichtung wird.

Home selbst wird als Geflecht von Beziehungen gelesen. Die Systemwahl folgt dem Zweck statt einer universellen Rangordnung. Vermögenswerte werden nach der von ihnen geschaffenen *Capability* und der durch sie erzeugten Konzentration gruppiert. Der gemeinsame Zustand besitzt *Stewardship*. Eine *Baseline* trennt den erwarteten Zustand von der gegenwärtigen *Observation*, ohne zu behaupten, erwartet bedeute sicher.

Behandle die *Chain* als datiertes Modell. *Connections* erhalten *Sources*, *Timestamps* und Zweck. Eine direkte *Observation* bleibt von *Reported* Informationen getrennt. Ein *Unknown* bleibt bestehen, wo keine der beiden die gegenwärtige Entscheidung stützt. Mehrere Menschen können mit der *Shared Map* auf Grundlage derselben begrenzten Aussagen arbeiten, doch sie verwandelt diese Aussagen nicht in beständige Strassen.

Infrastruktur wird anhand ihrer Abhängigkeiten verfolgt. In dieser Beziehung stehen *Structure*, *Service Module*, Unterstützung durch *Fuel* und *Power State*. Auch Zugriffsregel und abhängiger Auftrag gehören in diese Beziehung, sind aber nicht mit den übrigen Teilen austauschbar. Eine sichtbare Hülle beweist keinen *Service*. *Full Power* beweist nicht jeden Nutzungspfad. *Ownership* beweist nicht, dass ein *Character* handeln kann. Die Gruppe prüft genau die Handlung, welche ihr Plan verlangt.

Work beginnt mit *Exposure* statt mit Ertrag. Der Zweck entscheidet, ob eine Gelegenheit zu einer Operation wird. Schiffe, Aufmerksamkeit, Zeit, Route, Inventar und Menschen erhalten Abbruchbedingungen. Die Operation endet erst nach Rückkehr, Lagerung und Bericht. Bewegung begrenzt die gleichzeitige Konsequenz, und *Replacement* wird erst gezählt, wenn der vorgesehene Nutzer es erreichen kann.

Über die Zeit übergibt die Routine diese Beziehungen an die nächste Person. Ein offener Eintrag behält *Observation*, *Source* und *Timestamp*. Zusätzlich nennt er Bedeutung, *Unknown*, verantwortliche Person und nächste Prüfung. Ausnahmen werden geschlossen. Nicht unterstützter Umfang verschlechtert sich sichtbar. Wenn die Gruppe *Home* nicht länger tragen will, beginnt der Aufbruch früh genug, um *Work* zu stoppen, Vermögenswerte zu trennen, *Access* zu schliessen und das Ende zu überprüfen.

Das sind keine Konflikttaktiken. Es sind die Fähigkeiten, die jede spätere Taktik überprüfbar machen. Ohne sie kann eine Handlung auf ihrem *Grid* erfolgreich sein, während die Gruppe jene *Capability* verliert, die sie bewahren wollte.

Der Pilot in der zweiten Nacht besitzt nun eine Möglichkeit zu fragen, was wahr ist, bevor er fragt, was sich ihm entgegenstellen könnte.

Bereitschaft lässt sich anhand von Entscheidungen statt Erklärungen prüfen. Gib dem Piloten einen alten Routenbericht und eine wertvolle Bewegung: Erneuert er die massgebliche *Observation* vor der Verpflichtung? Gib der Gruppe ein sichtbares *Replacement* hinter einer ungeprüften Berechtigung: Zählt sie den Gegenstand oder prüft sie den Nutzungspfad? Lass einen *Service* seine bestätigte Unterstützung verlieren: Fügt sie *Work* hinzu, stellt sie Evidenz wieder her oder reduziert sie den Umfang? Biete eine geordnete Gelegenheit zum Aufbruch: Kann sie Abhängigkeiten schliessen, ohne auf Panik zu warten?

Keine Antwort garantiert das Überleben. Jede zeigt, ob die Gruppe den Unterschied zwischen Objekt und *Capability*, Bericht und *Observation*,

Anwesenheit und Unterstützung bewahren kann. Genau dieses Instrument wird *Survivor* benötigen, wenn das Feld bewusst irreführend wird.

Verbleibende *Exposure* benennen

Lesbarkeit entfernt *Exposure* nicht. Durch sie kann sich *Exposure* nicht in Selbstsicherheit verbergen.

Nützliche *Sources* können in der alten *Chain* verbleiben, während die heutige Bewegung dennoch eine frische *Observation* verlangt. *Access* zum Scanschiff kann bestehen, während dieses noch immer das einzige lokale *Replacement* für seine Funktion ist. Gegenwärtige *Services* kann die *Structure* bereitstellen, während zu wenige bestätigte Verantwortliche ihre künftige Unterstützung tragen. Ein geordnetes Inventar kann weiterhin Wert unter dem Ergebnis für Vermögenswerte im *Wormhole* konzentrieren.

Benenne diese Bedingungen, ohne Urteile daraus zu machen. «In *Home* verbleibt nur ein geprüfter Scanpfad» beschreibt *Capability* und Konzentration. «Der Rückkehrzweig ist aus der vorangegangenen Schicht *Reported* und für diese Bewegung unbestätigt» beschreibt Evidenz und Zweck. «*Service* online beobachtet; die nächste Prüfung seiner Unterstützung hat keine bestätigte verantwortliche Person» beschreibt gegenwärtige Funktion und künftiges *Unknown*.

Die verbleibende *Exposure* umfasst auch Menschen. Eine benannte verantwortliche Person kann mehrere entscheidende Prüfungen tragen. Ein Empfänger kann nur während eines Teils der Bewegung verfügbar sein. Geteiltes operatives Wissen kann in einer persönlichen Aufzeichnung liegen. Der Bericht muss private Zeitpläne nicht erklären. Er hält fest, wo einem Versprechen eine bestätigte Person fehlt, und reduziert den Umfang entsprechend.

Ein Teil der *Exposure* wird angenommen. Bewusst kann die Gruppe eine einzelne *Structure* behalten, eine einzige Route nutzen oder ein spezialisiertes Schiff binden, weil der Zweck dies rechtfertigt. *Explorer*

betrachtet diese Annahme nicht als verdächtig. Die Annahme muss zeigen, was nicht verfügbar würde und welche vorbereitete Handlung verbleibt.

Ein Teil der *Exposure* sollte enden. Überschüsse können gehen, bevor sie die Konzentration vertiefen. Ein *Service* ohne unterstützten Zweck kann schliessen. Ein Gastzugang kann nach seiner letzten Bewegung ablaufen. Alte Zweige der *Chain* können historisch werden, statt optisch aktuell zu bleiben. Beenden ist eine *Capability*, weil es Aufmerksamkeit freigibt und verringert, was späterer Druck erreichen kann.

Die abschliessende Aufzeichnung der *Exposure* sollte klein bleiben. Sie nennt offene Abhängigkeiten, ihre Verantwortlichen und die nächste Entscheidung. Die Aufzeichnung führt nicht jeden möglichen Verlust auf. Ebenso wenig stellt sie sich jeden künftigen Akteur vor. Spekulation kann grenzenlos wachsen; das gegenwärtige Feld kann es nicht. Beginne mit dem, was ein aktueller Zweck von *Home* tatsächlich zu tragen verlangt.

Jede Zeile sollte in eine von drei Richtungen führen. Annehmen bedeutet, dass der Zweck die beobachtete *Exposure* weiterhin rechtfertigt und eine verantwortliche Person die nächste Prüfung angenommen hat. Reduzieren bedeutet, dass die Gruppe eine kleinere *Capability* behält, solange ein *Unknown* oder eine fehlende verantwortliche Person verbleibt. Schliessen bedeutet, dass der Zweck geendet hat und die Abhängigkeit freigegeben oder übertragen wird. Eine Zeile ohne Richtung ist eine Beschreibung, noch keine operative Entscheidung.

Diese kleine Disziplin verhindert, dass eine Prüfung zum Inventar von Sorgen wird. *Exposure* wird nicht um ihrer selbst willen gesammelt. Sie wird mit einer Entscheidung verbunden, welche die Gruppe treffen kann, solange die gegenwärtige Evidenz noch eine Handlung stützt.

In der zweiten Nacht erkennt der Pilot den Rückkehrzweig als erste offene Abhängigkeit. Die alte Nachricht bleibt als *Reported Source* nützlich, kann das heutige *Unknown* aber nicht schliessen. Die *Observation*

kann mit dem Scanschiff wiederhergestellt werden, und die Bewegung wartet, bis diese *Work* abgeschlossen ist. Nichts ist gescheitert. Die *Exposure* hat lediglich eine verantwortliche Person und eine Reihenfolge erhalten.

Grenze erkennen

Failure und *Threat* sind keine Synonyme.

Failure ist der Verlust oder die *Degradation* einer Bedingung, die der Zweck verlangt. Ein *Service* geht offline. Ein Zugriffspfad erlaubt die beabsichtigte Handlung nicht länger. Eine *Reserve* kann nicht freigegeben werden. Eine Aussage über eine Route altert über die Bewegung hinaus, die sie stützen sollte. Die operative Auswirkung kann beobachtet werden, unabhängig davon, ob ihre Ursache ein Fehler, Vernachlässigung, gewöhnliche Veränderung oder etwas noch Unbekanntes ist.

Threat führt Absicht und *Capability* von ausserhalb des gepflegten Plans ein. Ein anderer Akteur könnte Bedingungen beobachten, beeinflussen, verweigern oder ausnutzen wollen, von denen die Gruppe abhängt. Die Feststellung dieser Beziehung verlangt Evidenz und andere Fragen. *Survivor* wird diese Fragen aufgreifen. *Explorer* endet, bevor er Akteure klassifiziert, Gegenmassnahmen lehrt oder eine Reaktion im Kampf vorschreibt.

Diese Grenze schützt das Urteilsvermögen in beide Richtungen. Wird jeder offline befindliche *Service* als feindliche Handlung bezeichnet, ersetzt die Gruppe die Diagnose durch Alarm und übersieht möglicherweise ihren eigenen *Failure* der Wartung. Wird jede Veränderung als gewöhnlich vorausgesetzt, kann die Gruppe Evidenz dafür ignorieren, dass ein anderer Akteur das Feld bewusst formt. Eine ehrliche *Observation* lässt keinen der beiden Schlüsse vor seiner *Source* eintreffen.

Eine operative Reaktion kann beginnen, bevor die Ursache geklärt ist. Stützt eine *Observation* der Route eine wertvolle Bewegung nicht mehr, stoppe die Bewegung und stelle die Evidenz wieder her. Gescheiterter

Access zu einer entscheidenden *Reserve* verlangt einen Stopp weiterer Abhängigkeiten und eine Prüfung des Pfads. Ein nicht unterstützter *Service* verlangt, dass der Zustand des offenen Auftrags festgehalten und sein Versprechen reduziert wird. Diese Handlungen bewahren *Capability* unter vielen Ursachen. Sie behaupten nicht, die Existenz eines *Threat* zu beantworten.

Sobald Evidenz für bewussten Druck für die nächste Entscheidung wichtig wird, hat sich das Lernproblem verändert. Das neue Problem betrifft, was ein anderer Akteur wissen, wollen und verändern kann; wie das Verhalten der Gruppe beobachtbar wird; und welche Reaktion das Feld verändert. Das sind keine grösseren Varianten von Prüfungen des *Fuel* oder des *Access*. Es sind die Eröffnungsfragen eines anderen Bandes.

Der Pilot sollte die Grenze laut nennen können. «Die *Connection* wurde für diese Bewegung nicht frisch beobachtet» ist eine gegenwärtige Tatsache. «Jemand hat sie geschlossen, um uns eine Falle zu stellen» ist eine Erklärung, die Evidenz verlangt. «Die Bewegung wartet, während wir die *Chain* wiederherstellen» ist eine operative Reaktion. Die Trennung der drei Aussagen bewahrt sowohl Ruhe als auch Neugier.

Diese Grenze verspricht nicht, dass die Gruppe jede feindliche Handlung rechtzeitig erkennt. Sie stellt sicher, dass der gewöhnliche Zustand lesbar genug ist, um einen bedeutsamen Unterschied zu bemerken.

Aufzeichnung bewahren

Ein abgeschlossener Band sollte den erneuten Eintritt ins Feld erleichtern.

Die Aufzeichnung am Ende von *Explorer* ist keine eingefrorene Beschreibung von *Home*. Sie ist ein kontrollierter Satz aus *Sources*, aktuellen *Observations*, angenommenen *Unknowns*, Verantwortlichen und Enden. Aus der Aufzeichnung erfährt der nächste Pilot, welche Evidenz eine Handlung stützt und welche Aussage erneuert werden muss.

Bewahre die *Baseline* getrennt von der gegenwärtigen Abweichung. Die *Baseline* nennt, was *Home* im angenommenen Zeitraum nach Absicht der Gruppe unterstützen soll. Gegenwärtige Einträge zeigen, welche Teile beobachtet, reduziert oder geschlossen wurden. Historische Berichte erklären die Abfolge, ohne einen alten Zustand als Anweisung darzustellen.

Bewahre auch das Vokabular. *Home* ist das System und die Verpflichtung, welche die Gruppe zur Unterstützung eines fortbestehenden Zwecks tragen soll. Vorübergehende *Connections* bilden das aktuelle Modell der *Chain*. Ein *Grid* ist das lokale Feld, in dem Objekte und Handlungen beobachtet werden. Eine *Source* bezeichnet die Herkunft einer Aussage. Eine *Observation* ist eine begrenzte Aussage, die mit dieser *Source* und einer Zeit verbunden ist. Dabei bezeichnet *Unknown* eine bewahrte Lücke, weil die Entscheidung sie nicht ehrlich füllen kann.

Diese Begriffe verhindern bequeme Ersetzungen. *Home* ist nicht allein die *Structure*. Keine *Chain* ist eine dauerhafte Route. Ein *Grid* ist nicht das gesamte System. Ein Bericht ist nicht allein deshalb eine direkte *Observation*, weil ihm vertraut wird. *Unknown* bedeutet keine Gefahr, und fehlender Nachweis ist kein Beweis dafür, dass sich nichts verändert hat.

Bewahre *Ownership* und Zugriffspfade auf der Ebene, auf der Menschen sie nutzen. *Ownership* einer *Corporation*, persönliches *Ownership*, *Structure Profiles*, *Access Lists*, *Corporation Roles* und geprüfte Handlungen der Nutzer beantworten unterschiedliche Fragen. Die fortbestehende Aufzeichnung muss nicht jede Konfiguration wiedergeben. Sie muss die befugte verantwortliche Person den Pfad finden und prüfen lassen, von dem eine aktuelle *Capability* abhängt.

Bewahre den Abschluss. Abgeschlossene *Work* folgt Ergebnis, Rückkehr, Lagerung und Bericht. Abgelaufene *Locations* und alte *Connections* werden historisch oder verlassen die aktuelle Ansicht. Beendeter *Access* behält seinen letzten Zweck und seine Bestätigung. Angenommene

Reste bleiben benannt, damit sie nicht als unerklärlicher *Failure* im Inventar zurückkehren können.

Eine Freigabeprüfung schützt die Aufzeichnung davor, zu einer weiteren dauerhaften Verpflichtung zu werden. Schliesse Verantwortliche, deren Handlungen geendet haben, und archiviere *Sources*, die nur noch Geschichte erklären. Setze die nächste Prüfung für die wenigen Aussagen, welche die gegenwärtige *Work* weiterhin stützen. Die Aufzeichnung bleibt pflegbar, weil der Abschluss einer Verpflichtung neben ihrer operativen *Exposure* auch ihre routinemässigen Kosten entfernt.

Ein anderer Pilot sollte eine Aussage infrage stellen können, ohne die Person infrage zu stellen, die sie gemacht hat. *Source* und *Timestamp* zeigen, wo nachzusehen ist; die Bedeutung zeigt, weshalb Aktualität wichtig ist; das *Unknown* zeigt, welche Annahme der Bericht verweigert. So wird die Berichtigung zu einem Teil gewöhnlicher *Stewardship* und nicht zum Beweis für gescheitertes Vertrauen.

Bewahre schliesslich die Quellengrenze des Buchs selbst. Die Mechaniken von EVE verändern sich. Das Manuskript verwendet für die wenigen benötigten Aussagen aktuelle offizielle *Sources* und vermeidet Tabellen oder Werte, deren Pflege zum verborgenen Risiko für den Leser würde. Wenn eine aktive Handlung von einem aktuellen Detail abhängt, prüft die verantwortliche Person das aktuelle offizielle Verfahren, statt diesen Band als Abbild des Clients zu behandeln.

Diese Aufzeichnung kann nun ihren Autor überdauern. Ein anderer Pilot erkennt, was beobachtet wurde, weshalb es wichtig ist, was *Unknown* bleibt und wer die nächste Prüfung verantwortet. Nimmt diese Person nicht an, schrumpft die unterstützte Bewegung. Die Aufzeichnung ist zu einem Teil der *Capability* geworden statt zu einem Denkmal vergangener *Work*.

Nach vorn übergeben

Erneut dockt der Pilot mit dem Scanschiff ab.

Diesmal wiederholt der Flug die erste Nacht nicht blind. Der Pilot benennt die Bewegung, welche die *Chain* unterstützen muss, und prüft den *Access* zum Schiff. Danach nimmt er die *Stewardship* des massgeblichen Zweigs an und beobachtet die für die nächste Entscheidung benötigten *Sources*. Alte Aufzeichnungen weisen den Weg zur Suche. Sie entscheiden nicht, was dort ist.

Das Ergebnis kann gewöhnlich sein. Die *Connection* besteht weiterhin, die aktuellen Informationen stützen die beabsichtigte Bewegung, und die gemeinsame Aufzeichnung erhält einen frischen *Timestamp*. Der Pilot kann zurückkehren, das Scanschiff freigeben und die vorübergehende Verantwortung schliessen.

Das Ergebnis kann stattdessen *Degradation* offenlegen. Eine *Connection* ist verschwunden, die Route muss neu aufgebaut werden und die Bewegung wartet. Zur Erklärung des operativen Zustands braucht es keinen Feind. Die Gruppe folgt dem Pfad des *Failure*, den sie bereits versteht: Menschen bewahren, neue Verpflichtungen reduzieren, Beobachtung zuweisen und entscheiden, ob der Zweck die *Work* weiterhin rechtfertigt.

Es gibt eine dritte Möglichkeit, doch *Explorer* entfaltet sie nicht. Evidenz kann zeigen, dass ein anderer Akteur nicht nur anwesend ist, sondern die Bedingungen, von denen *Home* abhängt, bewusst verändert. Dann braucht es mehr als eine gepflegte *Baseline*. Das Verständnis muss sich auf Beobachtung durch andere sowie auf Absicht, Druck und Täuschung erstrecken. Es muss auch Reaktion und Überleben umfassen, ohne die hier gelernte Disziplin zu verlieren.

Hier übergibt *Explorer* das Feld an *Survivor*.

Der nächste Band wird Abhängigkeiten nicht durch Kampf ersetzen. Er wird fragen, was geschieht, wenn jemand anderes diese Abhängigkeiten ebenfalls lesen kann. Eine Verpflichtung kann zum Signal werden, und Routine kann Zeitabläufe offenlegen. *Failure* kann hervorgerufen oder

ausgenutzt werden. Wie bewahrt eine Gruppe ihr Urteilsvermögen, wenn das Feld einen Akteur enthält, der es zu formen versucht?

Diese Fragen können nun gestellt werden, ohne jedes *Unknown* in Angst zusammenfallen zu lassen. Der Leser besitzt ein *Home* für den Vergleich und eine *Chain*, deren Alter sichtbar ist. Zugriffspfade können geprüft werden, während die *Reserve* Versprechen von *Capability* unterscheidet. Ein *Exit* kann beginnen, bevor die Wahl verschwindet.

Diese zweite Nacht endet dort, wo der erste Band über das Leben in *Home* enden musste. Gewissheit, Dauerhaftigkeit und Sieg fehlen. Es bleibt ein Feld, das von jemand anderem gelesen werden kann als der Person, die es zuletzt berührt hat.

Feldprinzip

Ein ehrliches *Home* ist die Grundlage des Überlebens.

Benenne Verpflichtungen, bevor der Druck sie für dich benennt. Halte *Observation* mit *Source*, Zeit und Zweck verbunden. Trenne *Ownership*, *Access*, Verantwortung und tatsächliche *Capability*. Schliesse *Work* nach Rückkehr und Entgegennahme. Reduziere nicht unterstützten Umfang, bevor *Failure* zur Krise wird, und geh, solange der Aufbruch noch überprüft werden kann.

Failure zeigt, dass eine erforderliche Bedingung den Zweck nicht länger stützt. *Threat* fragt, ob ein anderer Akteur diese Bedingung bewusst formen will und dazu in der Lage ist. Beantworte die zweite Frage nicht, indem du Evidenz erfindest, welche die erste nicht enthält.

Mit seiner Rückkehr in der zweiten Nacht hat der Pilot Anoikis nicht sicher gemacht. Er hat gelernt, was das Bleiben von Schiffen, Menschen, Informationen und Infrastruktur verlangt. Auch die Anforderungen von *Work* und Zeit sind für ihn sichtbar geworden. Was unsicher bleibt, ist sichtbar. Für offene *Work* ist entweder eine verantwortliche Person benannt oder ihr Umfang wird reduziert. Beendete *Work* kann geschlossen werden.

Der Kreislauf von *Explorer* ist vollständig.

Survivor beginnt, wenn das Feld zurückblickt.

Glossar und Index

Dieses gemeinsame Register definiert die kontrollierte Sprache von *Explorer* und verweist auf jene Abschnitte, in denen ein Gedanke eingeführt oder angewandt wird. Die Verweise wählen hilfreiche Erklärungen aus, nicht jedes Vorkommen. Die genannten Kapitel und Abschnitte bleiben über Ausgaben hinweg gültig; der endgültige Drucksatz kann Seitenzahlen ergänzen.

Abandoned

Ein Zustand einer *Upwell Structure*, der nach anhaltendem Ausbleiben des Verbrauchs von *Fuel* durch *Service Modules* erreicht wird. Er entfernt weitere *Capabilities* und unterscheidet sich sowohl von *Full Power* als auch von *Low Power*. Siehe: Kapitel 8, *Bedeutung von Power*; Kapitel 14, *Low Power zählt*.

Access

Die gegenwärtige Fähigkeit eines bestimmten *Characters*, einen Ort, Vermögenswert, *Service*, eine Aufzeichnung oder Handlung zu erreichen oder zu nutzen. *Access* allein beweist weder *Ownership* noch administrative Befugnis oder fortbestehende *Capability*. Siehe: Kapitel 4, *Access ist bedingt*; Kapitel 7, *Vier getrennte Fragen*; *Den Pfad prüfen*.

Access List

Eine Liste, die über ein *Structure Profile* zugewiesen wird und mitbestimmt, welche *Characters* einen bestimmten *Service* nutzen können. Ihre Wirkung hängt von den massgeblichen Einträgen, den anderen zugewiesenen Listen und dem geprüften *Character* ab. Siehe: Kapitel 7, *Profiles bieten Services*; *Access Lists wählen*; Kapitel 14, *Access kann wechseln*.

Asset Safety

Die Mechanik zur Wiederherstellung berechtigter Vermögenswerte in vielen *Upwell Structures*. *Upwell Structures* in *Wormhole Space* erzeugen nicht das gewöhnliche Ergebnis von *Asset Safety*; betroffene Vermögens-

werte können stattdessen in den Raum ausgestossen werden. *Siehe:* Kapitel 3, *Andere Rückgewinnung*; Kapitel 12, *Lagerung ist Risiko*; Kapitel 15, *Vermögensklassen trennen*.

Baseline

Eine datierte und mit *Sources* versehene Darstellung der gewöhnlichen Bedingungen, die einen bestimmten Zweck in *Home* unterstützen sollen. Eine *Baseline* macht Veränderungen vergleichbar; sie macht den festgehaltenen Zustand nicht dauerhaft. *Siehe:* Kapitel 4, *Baseline definieren*; Kapitel 5, *Gewöhnliches als Evidenz*; *Veränderungen brauchen Bedeutung*.

Bookmark

Eine gespeicherte Position, deren Nutzen von Bezeichnung, *Access*, Aktualität und gegenwärtigem Zweck abhängt. In *Explorer* werden *Bookmarks* ausserdem zu gepflegten gemeinsamen Vermögenswerten, deren Bedeutung ablaufen kann, bevor die Aufzeichnung entfernt wird. *Siehe:* Einleitung, *Anwesenheit wird System*; Kapitel 6, *Handover bewahrt Zweifel*; Kapitel 15, *Geteilten Access schliessen*.

Capability

Capability verbindet einsatzfähige Menschen, Wissen, Schiffe und Vermögenswerte. Auch *Access*, *Services* und Routen müssen den genannten Zweck jetzt unterstützen. Besitz ohne einen funktionierenden Abhängigkeitspfad begründet keine *Capability*. *Siehe:* Einleitung, *Der Arbeitsrhythmus*; Kapitel 13, *Routine muss schrumpfen*; Kapitel 16, *Capability vor Konflikt*.

Chain

Die gegenwärtig unterstützte Form des Zugangs über *Wormhole Connections*. In *Explorer* ist die *Chain* zugleich eine gepflegte Abhängigkeit von *Home*, *Work*, *Transport* und schichtübergreifender Rückkehr. *Siehe:* Einleitung, *Anwesenheit wird System*; Kapitel 6, *Karten veralten*; Kapitel 11, *Zeit wird geteilt*.

Verpflichtung

Eine Person, ein Vermögenswert, Versprechen, eine Aufgabe oder Abhängigkeit, deren Konsequenz nach der unmittelbaren Handlung fortbesteht. Eine Verpflichtung erzeugt Gewicht, wenn Wiederherstellung, *Replacement* oder Abschluss spätere *Work* erfordern. *Siehe*: Einleitung, *Das Gewicht*; Kapitel 1, *Verpflichtung benennen*; Kapitel 3, *Was zurückbleibt*.

Connection

Eine gegenwärtige Verbindung durch ein *Wormhole* zwischen zwei Orten. Lebensdauer, Masse, Endpunkte und operative Bedeutung bleiben auf diese *Connection* begrenzt und gelten nicht für die gesamte *Chain*. *Siehe*: Kapitel 2, *Connections lesen*; Kapitel 6, *Zeit verändert Routen*; Kapitel 11, *Masse wird geteilt*.

Corporation Role

Eine von der *Corporation* gewährte Fähigkeit, die bestimmte Handlungen erlaubt. *Roles* sind nicht austauschbar mit *Ownership* an Vermögenswerten, Verwaltung von *Profiles* oder *Access Lists* und geprüftem Nutzerzugang. *Siehe*: Kapitel 7, *Roles ermöglichen Handeln*; Kapitel 14, *Access kann wechseln*; Kapitel 15, *Geteilten Access schliessen*.

Degradation

Eine sichtbare Verringerung der Unterstützung für einen Zweck vor einem vollständigen *Failure*. *Degradation* kann sich als alterndes Wissen, engerer *Access*, verringerte *Reserve*, pausierte *Work* oder fehlender *Service* zeigen. *Siehe*: Kapitel 13, *Routine muss schrumpfen*; Kapitel 14, *Degradation zeigt Signale*; *Frühzeitig reduzieren*.

Abhängigkeit

Eine Bedingung, Ressource, Person, Berechtigung oder vorausgehende Handlung, die für das Fortbestehen einer *Capability* oder Verpflichtung erforderlich ist. Abhängigkeiten werden als Pfade verfolgt, damit sich eine fehlende Bedingung nicht hinter einem sichtbaren Ergebnis ver-

birgt. *Siehe:* Kapitel 8, *Failure wirkt weiter; Abhängigkeiten zuweisen*; Kapitel 14, *Frühzeitig reduzieren*.

Exit

Der festgelegte Pfad mit den Bedingungen, unter denen eine begrenzte Aktivität oder Verpflichtung enden kann, solange Wiederherstellung möglich bleibt. Ein *Exit* gehört in den Plan, bevor seine Nutzung dringend wird. *Siehe:* Kapitel 10, *Exit vorbereiten; Nach Evidenz stoppen*; Kapitel 15, *Exit beginnt früh*.

Explorer

Die in diesem Band entwickelte operative Rolle bezeichnet einen Piloten. Er kann einen Zweck in *Wormhole Space* begründen, pflegen, nutzen, reduzieren und bewusst beenden, ohne Unsicherheit in Gewissheit oder *Threat* zu verwandeln. *Siehe:* Einleitung, *Die Grenze*; Kapitel 16, *Grenze erkennen; Nach vorn übergeben*.

Exposure

Die Konsequenz, die Veränderung oder Verlust erreichen kann. Sie entsteht, während Aufmerksamkeit, Schiffe, Vermögenswerte, Zeit oder Routen gebunden sind. *Exposure* kann begrenzt werden, selbst wenn sie sich nicht entfernen lässt. *Siehe:* Kapitel 3, *Konzentration verändert Verlust*; Kapitel 9, *Work hinterlässt Spuren*; Kapitel 11, *Exposure aufteilen*.

Failure

Der Zustand, in dem ein erforderlicher Teil seinen genannten Zweck nicht länger unterstützt. *Failure* liefert für sich allein keine Evidenz für die Absicht oder *Capability* eines anderen Akteurs. *Siehe:* Kapitel 14, *Degradation zeigt Signale; Frühzeitig reduzieren*; Kapitel 16, *Grenze erkennen*.

Full Power

Der Zustand einer *Upwell Structure*, der besteht, wenn mindestens ein *Service Module* online ist. Dieser Zustand beweist nicht, dass jeder beabsichtigte *Service* verfügbar ist. *Siehe:* Kapitel 8, *Bedeutung von Power*; Kapitel 14, *Low Power zählt*.

Grid

Das unmittelbare Feld, in dem der *Client* nahe Objekte und Handlungen darstellt. Ein übersichtlicher *Grid* ist eine begrenzte *Observation*, kein Beweis für ein unverändertes System, eine unveränderte *Chain* oder einen unveränderten Unterstützungspfad. *Siehe*: Kapitel 5, *Ebenen von Home*; Kapitel 9, *Work hinterlässt Spuren*; Kapitel 16, *Grenze erkennen*.

Handover

Die angenommene Übertragung von aktueller Evidenz, Bedeutung, *Unknowns*, *Ownership* und nächster Handlung zwischen Menschen oder Schichten. Das Senden einer Nachricht oder Zuweisen eines Namens genügt erst, wenn die empfangende Seite die Aufzeichnung nutzen kann. *Siehe*: Kapitel 4, *Handover ohne Annahme*; Kapitel 6, *Handover bewahrt Zweifel*; Kapitel 13, *Jede Schicht übernimmt*.

Home

Die gewählte operative Basis, deren Systeme, *Connections*, Vermögenswerte, Menschen und Infrastruktur einen bestimmten Zweck unterstützen. *Home* wird gepflegt und bleibt bedingt; es ist weder dauerhaft noch eine garantierte Rückkehr. *Siehe*: Einleitung, *Anwesenheit wird System*; Kapitel 2, *Zweck vor Class*; Kapitel 15, *Ende überprüfen*.

Industry Job

Eine bestimmte Aktivität von *Industry*, die *Blueprint*, Eingaben, *Facility*, Ausgabe und verantwortliche Person über die Zeit bindet. Ihre Kontinuität hängt von nutzbarem *Access* und dem massgeblichen *Service* ebenso ab wie vom Material selbst. *Siehe*: Kapitel 8, *Failure wirkt weiter*; Kapitel 12, *Industry braucht Kontinuität*; Kapitel 15, *Vermögensklassen trennen*.

Low Power

Der Zustand einer *Upwell Structure*, in dem kein *Service Module* installiert oder online ist. Er bezeichnet eine begrenzte Bedingung der Infrastruktur, nicht ihre Ursache oder den Zustand jeder anderen *Capability*. *Siehe*: Kapitel 8, *Bedeutung von Power*; Kapitel 14, *Low Power zählt*.

Observation

Eine begrenzte Aussage über Evidenz, verbunden mit der *Source*, Zeit und den Bedingungen, unter denen sie entstand. Ihre operative Reichweite endet dort, wo diese Bedingungen die gegenwärtige Entscheidung nicht mehr unterstützen. *Siehe: Einleitung, Der Arbeitsrhythmus; Kapitel 5, Jede Observation altert; Kapitel 13, Wissen altert.*

Observed

Der Informationszustand für eine Aussage, die direkt durch eine benannte aktuelle *Observation* begründet wurde. Er bleibt durch *Source*, Zeit und Reichweite dieser *Observation* begrenzt. *Siehe: Kapitel 4, Baseline definieren; Kapitel 6, Handover bewahrt Zweifel; Kapitel 16, Capability vor Konflikt.*

Verantwortliche Person

Die benannte Person oder Rolle, welche die nächste Prüfung, Entscheidung, Handlung oder den Abschluss verantwortet. Diese Verantwortung überträgt nicht zwingend das *Ownership* am Vermögenswert oder die Befugnis über das betroffene System. *Siehe: Kapitel 4, Steward bestimmen; Kapitel 13, Berichte brauchen Verantwortliche; Kapitel 14, Frühzeitig reduzieren.*

Ownership

Die Beziehung, welche bestimmt, wessen Vermögenswert, Aufzeichnung, Entscheidung oder fortbestehende Verantwortung betroffen ist. *Ownership* bleibt von gegenwärtigem *Access* und der Fähigkeit zur Ausführung einer erforderlichen Handlung getrennt. *Siehe: Kapitel 3, Was zurückbleibt; Kapitel 7, Vier getrennte Fragen; Kapitel 12, Ownership formt Access.*

Profile

Die Konfiguration einer *Structure*, welche die anderen Spielern angebotenen *Services* festlegt und diese mit zugewiesenen *Access Lists* verbindet. Mehr als eine *Structure* kann dasselbe *Profile* verwenden, sodass eine Veränderung eine grössere Reichweite besitzen kann. *Siehe: Kapitel*

2, *Bestehende Infrastruktur*; Kapitel 7, *Profiles bieten Services*; Kapitel 14, *Access kann wechseln*.

Replacement

Die Wiederherstellung nutzbarer *Capability* nach Verbrauch, Verlust oder Verpflichtung. *Replacement* umfasst Beschaffung, Bewegung, *Access* und Einsatzbereitschaft; es endet nicht bereits, wenn ein Gegenstand irgendwo anders vorhanden ist. *Siehe*: Kapitel 3, *Entfernung des Replacement*; Kapitel 12, *Vor Leere ersetzen*; Kapitel 14, *Reserve zeigt Fragilität*.

Bericht

Eine übermittelte Darstellung, die ihre *Observation*, *Source* und ihren *Timestamp* sichtbar hält. Sie nennt ausserdem operative Bedeutung, verbleibendes *Unknown*, verantwortliche Person und nächste Prüfung. Ein Bericht erneuert die beschriebene Evidenz nicht. *Siehe*: Kapitel 13, *Berichte brauchen Verantwortliche*; Kapitel 16, *Aufzeichnung bewahren*.

Reported

Der Informationszustand für eine Aussage, die von einem anderen Piloten, Werkzeug oder einer Aufzeichnung stammt und nicht als eigene aktuelle *Observation* des empfangenden Piloten begründet wurde. *Siehe*: Kapitel 1, *Die zweite Nacht*; Kapitel 6, *Handover bewahrt Zweifel*; Kapitel 16, *Capability vor Konflikt*.

Reserve

Zugängliche Kapazität, die für einen bestimmten künftigen Bedarf bereitsteht. Eine *Reserve* muss von bereits verbrauchtem, gebundenem oder versprochenem Material und von anderswo wartendem Material ohne nutzbaren Zugriffspfad unterscheidbar bleiben. *Siehe*: Kapitel 3, *Exposure begrenzen*; Kapitel 12, *Vor Leere ersetzen*; Kapitel 14, *Reserve zeigt Fragilität*.

Service

Eine durch Infrastruktur bereitgestellte Funktion, die Handlungen wie Andocken oder *Industry* unterstützt. Ein angebotener *Service* ist für

einen bestimmten *Character* im benötigten Augenblick nicht zwingend einsatzfähig. *Siehe:* Kapitel 7, *Profiles bieten Services*; Kapitel 8, *Services brauchen Mittel*; Kapitel 14, *Services können verschwinden*.

Service Module

Das eingebaute Modul einer *Upwell Structure*, welches den entsprechenden *Service* bereitstellt und online *Fuel* verbraucht. Seine Anwesenheit beweist nicht, dass es online, mit *Fuel* versorgt oder für den vorgesehenen Nutzer zugänglich ist. *Siehe:* Kapitel 8, *Services brauchen Mittel*; *Bedeutung von Power*; Kapitel 14, *Services können verschwinden*.

Shared Location

Eine über einen *Shared Location Folder* veröffentlichte *Location* mit eigenem *Access*, Onlinezustand und möglichem Ablaufdatum. Das Teilen erweitert die Verfügbarkeit; es hält die zugrunde liegende Position oder Bedeutung nicht dauerhaft aktuell. *Siehe:* Kapitel 4, *Access ist bedingt*; Kapitel 5, *Jede Observation altert*; Kapitel 13, *Wissen altert*.

Shared Map

Eine gemeinsame Veröffentlichung von Systemen, *Connections* und ihren gegenwärtigen Aussagen. Sie unterstützt gemeinsame *Work* nur, solange *Source*, *Timestamp*, Zustand und ungelöster Zweifel sichtbar bleiben. *Siehe:* Kapitel 6, *Karten veralten*; *Handover bewahrt Zweifel*; Kapitel 16, *Aufzeichnung bewahren*.

Source

Die Herkunft und der Beobachtungspfad einer Aussage, einschliesslich Bedienoberfläche, Person, Werkzeug oder Aufzeichnung, über die sie gewonnen wurde. Autorität entfernt die Notwendigkeit nicht, eine *Source* zu benennen. *Siehe:* Kapitel 5, *Jede Observation altert*; Kapitel 13, *Wissen altert*; Kapitel 16, *Verbleibende Exposure benennen*.

Steward

Die Person, die einen begrenzten Teil des gemeinsamen Zustands pflegt, bis er bestätigt, übergeben, an die Gruppe zurückgegeben oder geschlossen ist. Ein *Steward* kann die *Work* koordinieren, ohne jede

Handlung persönlich auszuführen. *Siehe:* Kapitel 4, *Steward bestimmen*; *Handover ohne Annahme*; Kapitel 5, *Wesentliches beobachten*.

Structure

Eine Installation des Typs *Upwell Structure*, die dieser Band als ein Element von *Home* behandelt. Ihre Hülle, ihr *Profile*, ihre *Access Lists* und *Service Modules* bleiben getrennt. *Fuel*, Zugriff auf Vermögenswerte und abhängige *Work* vervollständigen ihren operativen Zustand. *Siehe:* Kapitel 2, *Bestehende Infrastruktur*; Kapitel 8, *Services brauchen Mittel*; Kapitel 15.

Structure Browser

Die Bedienoberfläche von EVE Online, welche zugängliche *Structures* und angebotene *Services* zeigt. Befugte *Corporation Roles* können zusätzliche Verwaltungsinformationen zu ihren *Structures* sehen. *Siehe:* Kapitel 5, *Ebenen von Home*; Kapitel 8, *Zustand sichtbar machen*; Kapitel 14, *Low Power zählt*.

Timestamp

Die mit einer *Observation*, Aussage oder Veränderung verbundene Zeit. Das Bearbeiten oder Weiterleiten einer Aufzeichnung erneuert einen *Timestamp* nicht, sofern die zugrunde liegende Bedingung nicht erneut beobachtet wird. *Siehe:* Kapitel 5, *Jede Observation altert*; Kapitel 6, *Karten veralten*; Kapitel 13, *Wissen altert*.

Unknown

Informationen, die nicht auf dem von der gegenwärtigen Entscheidung benötigten Niveau begründet sind. *Unknown* bedeutet weder Abwesenheit noch Gefahr; es ist eine sichtbare Grenze dessen, was die verfügbare Evidenz unterstützen kann. *Siehe:* Kapitel 4, *Baseline definieren*; Kapitel 6, *Handover bewahrt Zweifel*; Kapitel 16, *Grenze erkennen*.

Work

Eine zielgerichtete Tätigkeit, die Aufmerksamkeit und *Capability* über Vorbereitung, Ausführung, Lagerung, Rückkehr und Abschluss bindet. *Work* beginnt vor der ersten sichtbaren Handlung auf ihrem *Grid* und

endet, nachdem ihre Konsequenzen erfasst worden sind. *Siehe: Kapitel 9, Work hinterlässt Spuren; Kapitel 10, Zweck wählen; Kapitel 12, Lagerung ist Risiko.*

Wormhole Space

Die beständigen Sonnensysteme, die über vorübergehende *Wormhole Connections* statt über eine dauerhafte Route von *Stargates* erreicht werden. In diesem Band ist es die Umgebung, in der *Home* und seine fortbestehenden Verpflichtungen gepflegt werden. *Siehe: Einleitung, Anwesenheit wird System; Kapitel 2, Zweck vor Class; Kapitel 15, Bewusst aufbrechen.*

Weiterführende Literatur

EVE Online verändert sich fortlaufend. Diese Quellen vertiefen die Konzepte dieses Bandes, doch massgebend bleiben der *Live Client* und die aktuelle offizielle Dokumentation. Die unten aufgeführten Quellen wurden am 20. August 2026 erneut geprüft. Sie dienen der weiteren Recherche; die in *Explorer* durchgehend angewandte Disziplin bei *Source* und *Timestamp* ersetzen sie nicht.

Grundlagen der *Wormholes*

- *Wormholes*, EVE Online Support — vorübergehende *Connections*, ihre Entdeckung sowie die ungefähren Angaben zu Lebensdauer und Masse, die einem Piloten zur Verfügung stehen.
- *EVE Academy - Wormhole Basics*, EVE Online — eine offizielle Einführung in *Wormhole Space* und den gewöhnlichen Kontext von C1 bis C6.
- *Scanning*, EVE Online Support — *Sensor Overlay*, *Probe Scanner*, *Cosmic Anomalies*, *Cosmic Signatures* und die Verwaltung von Scanergebnissen.
- *Patch Notes - Version 23.02*, EVE Online — die Einführung und Bedeutung von *Reliable Lifetime* und *Expired*. Prüfe spätere *Patch Notes*, bevor du davon ausgehst, dass diese Bezeichnungen unverändert geblieben sind.

Infrastruktur und Zugang

- *Structure Management*, EVE Online Support — *Structure Browser*, zugängliche *Services*, Verwaltungsinformationen und *Power State*.
- *Profiles*, EVE Online Support — die Beziehung zwischen einer *Structure*, den angebotenen *Services* und den zugewiesenen *Access Lists*.
- *Access Lists*, EVE Online Support — Mitgliedschaft, Verwaltung und die detaillierten Einträge, anhand deren der Zugang bestimmt wird.

- *Service Modules*, EVE Online Support — die Komponenten, welche die Funktionen einer *Structure* bereitstellen und *Fuel* benötigen, solange sie online sind.
- *Roles Listing*, EVE Online Support — die getrennten Berechtigungen der *Corporation* für die Verwaltung von *Structures*, *Industry* und den Zugang zu Hangars.
- *Asset Safety*, EVE Online Support — die gewöhnliche Wiedererlangung von Assets und das abweichende Ergebnis für *Upwell Structures* in *Wormhole Space*.
- *Upwell Structure Deployment and Unanchoring*, EVE Online Support — Berechtigung zur Verankerung, Einschränkungen des Standorts, Stilllegung und Abholung.
- *Upwell Structures Vulnerability States* und *Upwell Structures - Abandoned State*, EVE Online Support — die Unterschiede zwischen *Full Power*, *Low Power*, *Abandoned* und anderen Zuständen einer *Structure*. *Explorer* verwendet diese Seiten, um *Capability* zu verstehen, nicht um die Verteidigung von *Structures* zu lehren.

Work und Aufzeichnungen

- *EVE Academy - Ships and Career Paths*, EVE Online — Orientierung in *Exploration*, *Resource Acquisition*, *Industry* und *Transport*, ohne eine einzelne Tätigkeit als allgemeingültige Antwort zu behandeln.
- *Manufacturing*, EVE Online Support — die grundlegende Beziehung zwischen einem *Blueprint*, Materialien, einer geeigneten *Facility* und der Ausgabe.
- *Activities and Job Types*, EVE Online Support — die verschiedenen unter *Industry* zusammengefassten Tätigkeiten und Aufgabenkategorien.
- *Industry User Interface*, EVE Online Support — die sichtbare Beziehung zwischen Eingaben, *Blueprint*, *Facility*, Ausgabe, Dauer und Kosten.
- *Shareable Bookmarks: Alliance Bookmarks and More!* — EVE Online — die offizielle Einführung in *Personal Location Folders* und *Shared Location Folders* sowie deren Zugriffsstufen, Onlinezustand und Ablauf.

Die Seite ist eine historische Einführung der Funktion; das aktuelle Verhalten des *Clients* muss weiterhin anhand des aktuellen Stands bestätigt werden.

Daten und Veränderung

- *Static Data* und *Using Static Data*, EVE Developer Documentation — die gepflegten Datenpakete und die Grenzen dessen, was Daten zu *Wormhole Class* und *Effect* belegen können.
- Die aktuellen EVE Online *Patch Notes* — Änderungen an Bezeichnungen, Mechaniken und Verhalten des *Clients* nach den oben genannten Quellen.

Veröffentlichungsdaten, Bezeichnungen im *Client* und Datenstände gehören zu jeder präzisen Aussage. Ein vertrauter Seitentitel macht eine ältere Aussage nicht aktuell, und ein von der Gruppe täglich verwendetes Community-Tool wird dadurch nicht zu einer primären *Source*.

Danksagung

Jedes *Home* in Anoikis wird zum Teil von jemandem geprägt, der schon früher dort war.

Jemand hat eine Route benannt oder die *Probes* ersetzt. Es wurde geprüft, ob ein Bericht zum *Fuel* noch aktuell war. Oder jemand sprach offen aus, dass eine alte Karte die nächste Entscheidung nicht mehr tragen konnte. Aus dem grössten Teil dieser Arbeit entsteht keine dramatische Geschichte. Sie gibt jemand anderem die Chance, eine zu erleben.

Die Autorin dankt den *Capsuleers*, die gemeinsame Aufzeichnungen unverfälscht halten. Sie geben unfertige Arbeit weiter, ohne Zweifel zu verbergen. Sie erklären *Failure*, ohne einen Schuldigen zu erfinden. Neuen Piloten zeigen sie zuerst, weshalb eine Bedingung wichtig ist. Erst dann sagen sie ihnen, welchen Knopf sie drücken sollen. *Explorer* verdankt dieser stillen Form der Kompetenz seine Gestalt.

Dank gilt auch allen, die den offiziellen EVE-Online-Support, die Entwicklerdokumentation und die *Patch Notes* verfassen und pflegen. Ihre Arbeit hat es ermöglicht, veränderliche Mechaniken und Bezeichnungen anhand einer sichtbaren *Source* zu prüfen, statt sie als Überlieferung aus der Erinnerung abzurufen. Die Verantwortung für Auswahl, Begrenzung und Auslegung dieser Quellen trägt weiterhin dieses Buch.

The Anoikis Field Manual richtet sich an Leserinnen und Leser, die bereit sind, angesichts einer Konsequenz innezuhalten. Sie sind ebenso bereit, ein unbequemes *Unknown* zu bewahren und der nächsten Person eine tatsächlich nutzbare Aufzeichnung zu hinterlassen. Dieser Band wird im selben Geist angeboten: nicht als das letzte Wort über das Leben in *Wormhole Space*, sondern als sorgfältige Darstellung, die hinterfragt werden darf, wenn sich das Feld verändert.

Jeder verbleibende Fehler liegt bei Autorin und Herausgeber. Jede Korrektur sollte dort beginnen, wo auch die Feldarbeit beginnt: Zeige die *Source*, benenne die Bedingung und prüfe erneut.

Über den Verlag

Unsettling Whisper

A Corporation of Quietly Eerie

Unsettling Whisper veröffentlicht *The Anoikis Field Manual* als inoffizielle, von Fans geschaffene und nicht kommerzielle Reihe für die EVE-Online-Community.

Die Reihe stammt von Irdyna Ferydo und folgt über *Pathfinder*, *Explorer*, *Survivor*, *Navigator* und *Commander* einer zunehmenden operativen Verpflichtung. Jeder Band wird abgeschlossen, bevor die Arbeit am nächsten beginnt.

Redaktioneller Kontakt: irdynaferdydo@gmail.com

