FiveFold

TECHNICAL LIBRARY / GUIDE 03 / OCTOBER 2026

MORIAH

Operating Handbook | Companion Guide

Explore the substance of the internal operating handbook without exposing privileged procedures.

A reader's guide to nine documented chapters, their actual numbered sections, operating responsibilities, and the limits of what documentation establishes.

by FiveFold Ventures / October 2026

What this companion is for

The internal Moriah Stack Operating Handbook is an owner, operator, administrator and engineer-facing manuscript about the work of keeping a substantial AI operating environment controlled and understandable. It sits alongside technical canon, accepted proof and specialized operating records; it does not replace them.

This companion shows the shape and substance of that documented work. The chapter and numbered-section titles follow the source manuscript. Explanations describe the subjects a reader would encounter and why the distinctions matter. The protected procedures themselves remain in the controlled handbook.

Why the layers matter. Models produce reasoning and language. The operating environment around that reasoning must separately manage identities, source eligibility, authorized actions, saved work, evidence and continuity. The handbook documents responsibilities at those boundaries.

Who should read this. Technical evaluators assessing engineering discipline; organization leaders considering governance and administration; engineers looking for ownership structure; and operators trying to understand support and release maturity.

How to navigate. Start with the Technical Architecture & Capabilities Guide for foundation and capabilities; use this Companion for operating responsibilities and handbook coverage. Linked contents and bookmarks preserve chapter and subsection order.

Reading the maturity language

Documented foundation. The manuscript explains a concept, procedure or intended owner; this alone does not prove current behavior.

Historically accepted behavior. Dated engineering evidence may prove a bounded operation in a specific environment; current customer configuration is a separate question.

Configuration dependent. A supported route may still require consent, licensing, keys, scope, tenancy, user rights or target-specific validation.

Separately unproven. Full user journeys, protected cross-organization gates and some release stages have independent acceptance requirements.

Future-facing material. Later capability slots and enterprise patterns are documented as optional directions, not generally delivered functionality.

Source identity. This review uses the preserved October 18 authoritative Markdown manuscript and index snapshot. No definitive physical page count exists for the evolving manuscript; a separate October 10 reading-copy PDF contained 96 pages. The source crosswalk and hashes are held in the internal review record.

Contents

Choose any chapter below. Each chapter explains its original numbered sections and the operating understanding they provide. In a PDF viewer, the bookmark pane also opens individual subsections.

01 Owner Quickstart

For founders, owners and principal operators · 8 numbered subsections

02 Operator Runbook

For trusted platform operators and support leads · 13 numbered subsections

03 Admin and Organization Operations Guide

For organization administrators and access-governance reviewers · 18 numbered subsections

04 Recovery and Release Guide

For reliability, change-control and technical diligence readers · 15 numbered subsections

05 UI and Working Surfaces Guide

For product operators, administrators and evaluators · 11 numbered subsections

06 Sessions and Support Guide

For support, acceptance and operations teams · 11 numbered subsections

07 Engineer / Technical Onboarding Guide

For incoming engineers and authorized technical operators · 14 numbered subsections

08 Enterprise and Future Lanes Guide

For solution architects, enterprise evaluators and product planners · 11 numbered subsections

09 Glossary and Reference Pointers

For all readers locating definitions and authoritative sources · 9 numbered subsections

Additional reading: Currentness and supplements • What this document does not disclose • Release and claim boundaries

CHAPTER 01 / Owner Quickstart

Owner Quickstart

An owner-facing orientation to the system, its operating surfaces, the responsibilities around change, and the circumstances that require specialist support. The emphasis is on recognizing system health and ownership rather than becoming an infrastructure engineer.

What the reader learns. An evaluator sees how continuity and system responsibility are made legible to the person accountable for the product. The material identifies the difference between knowing which area to inspect and having permission to make a change.

Interpretation boundary. The chapter is an operating orientation, not a promise that every function is universally deployed or a substitute for privileged access procedures.

Inside this chapter

The following 8 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What the stack is
Introduces the system as a governed application, data and operating environment, rather than equating Moriah with an individual model.

2. Where it lives
Identifies the different classes of source, runtime and operating custody an owner needs to locate; actual infrastructure locations remain private.

2A. Public presence and brand operations
Separates responsibility for the public website, search presence and brand-facing operations from internal application ownership.

3. Main services and surfaces
Describes the role of services and surfaces in the working environment and why component visibility is not the whole user experience.

4. How to tell if it is up
Explains the difference between basic availability signals and proof that a specific product workflow completes correctly.

5. What to do before making changes
Emphasizes checking authoritative state, scope and permission before any operational change is considered.

6. What to do after making changes
Explains post-change verification, evidence retention and handoff as distinct from merely observing that a change was made.

7. When to stop and escalate
Defines cases where ownership, incomplete evidence or an unsafe assumption should trigger escalation rather than improvisation.

CHAPTER 02 / Operator Runbook

Operator Runbook

The day-to-day operating model: establish reliable state, interpret service and user-surface signals, work within a defined scope, record evidence and escalate when a check does not support the claim. Its procedures are deliberately more specific than a public reader needs.

What the reader learns. An engineer or operating partner learns the sequence of judgment behind routine operations: current authority first, bounded validation second, and recorded outcome before release or handoff.

Interpretation boundary. Published here is the purpose of each operating stage; shell commands, endpoint details, privileged paths and maintenance procedures remain inside the controlled runbook.

Inside this chapter

The following 13 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. Core operating posture
Sets the routine posture: identify what is actually running, what is authorized and what question the operator is resolving.

2. Non-negotiable rules
Names invariants around protected state, trustworthy evidence and the separation of source edits from live behavior.

3. Canonical places to know
Explains why operational work has authoritative repositories, control records and evidence owners; exact privileged locations are omitted.

4. Routine posture checks
Frames a bounded first review of source, release and service posture before treating a problem as product behavior.

5. Health, version, and UI checks
Distinguishes availability, version identity and visible UI reachability from functional acceptance of an authenticated task.

6. How to read compose status
Teaches how service inventories, health state and listening surfaces are interpreted without implying functional coverage.

7. Bounded proof-pack and milestone validation flows
Explains when targeted regression evidence is appropriate and why broader release gates are selected by authority.

8. How to interpret pass/fail posture
Separates completed, partially completed and unproven results, including the importance of real return codes.

9. Where outputs and artifacts go
Describes evidence and artifact custody so that results remain attributable and available for later review.

10. Recovery drill and release discipline
Places recovery preparation and release actions behind bounded ownership rather than ad hoc intervention.

11. Safe operating sequence
Summarizes the order of authorization, inspection, a bounded operation, proof and disposition.

12. When to stop and escalate
Identifies conditions in which an operator stops and returns a precise unresolved fact to its owner.

12. Release-control quick reference
Introduces the release-control concept of candidate, acceptance stage and preserved predecessor; it does not reproduce an operator procedure.

CHAPTER 03 / Admin and Organization Operations Guide

Admin and Organization Operations Guide

The people, organization and configuration layer of the platform, including roles, membership lifecycle, administration surfaces, connection ownership and provider-credential boundaries. Administrative advice and actual permissioned execution are carefully separated.

What the reader learns. An organization can see which decisions belong to a tenant administrator, which belong to an individual connector user, and why service readiness depends on consent, scope, credentials and current acceptance.

Interpretation boundary. A documented control or setup interface does not by itself prove that a given customer tenant has been configured or that a formal cross-organization security gate is complete.

Inside this chapter

The following 18 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What this chapter is for
Defines administration as a combination of organization membership, access control, setup decisions and support responsibilities.

2. The basic admin model
Distinguishes platform authority, organization authority and the privileges of ordinary members.

3. What admin actions exist today
Surveys membership, organization and connection-management action families without implying universal operational readiness.

4. What admins can do
Describes legitimate day-to-day responsibilities such as reviewing members, permissions and readiness within authorized scope.

5. What admins should avoid doing casually
Identifies changes that should not be treated as casual settings adjustments because they affect trust or access.

6. Membership behavior in plain language
Explains how membership identity, organization association and lifecycle state interact conceptually.

7. Minimum role understanding
Provides a practical way to understand differences in roles and why titles alone are insufficient authorization.

8. Admin portal surfaces
Maps the classes of administrative views, including members, access, preferences, connections and status information.

9. How admin functions support real users and organization operations
Connects administration to onboarding, support, predictable ownership and safe organizational use.

10. Safe admin operating sequence
Frames administrative work as a permissioned process that can be independently reviewed and confirmed.

11. When to stop and escalate
Explains when administrators need specialist assistance or explicit authority beyond the interface.

12. Admin/provider setup orientation status
Defines guidance for providers and organization configuration separately from execution proof.

13. Guidance versus action boundary
Makes the distinction between instructional recommendations and a real, permissioned change explicit.

14. Microsoft 365 / Entra setup concepts (plain-language orientation)
Introduces the concepts of identity provider, organizational consent and application permissions relevant to a Microsoft tenant.

15. My Connections versus Admin Controls (orientation)
Separates the personal connection experience from organization-wide administrative controls and entitlement ownership.

16. Provider fuel/BYOK and connector readiness
Explains model/provider usage responsibility, customer credentials and connection readiness without asserting universal BYOK operation.

17. Safe support fields and never-paste boundary
Defines the difference between safe support metadata and sensitive secrets or private records that should never be circulated.

18. Who to involve and when
Assigns the appropriate decision to the tenant administrator, provider account owner or FiveFold/platform owner.

CHAPTER 04 / Recovery and Release Guide

Recovery and Release Guide

An overview of backup and recovery responsibilities, release evidence, risk tiers, restoration planning, staging and isolation. It explains why technical recovery, application correctness and full business continuity are different levels of proof.

What the reader learns. Technical readers see a proof-driven release system with explicit ownership, recovery preparation, limited change scope and decision points for rollback or escalation.

Interpretation boundary. Implementation-specific recovery steps, infrastructure layout, authentication recovery and incident details are excluded. Internal recovery exercises are not a blanket disaster-recovery certification.

Inside this chapter

The following 15 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What this chapter is for
Clarifies the purpose of recovery and controlled release: protection against loss, drift and unsupported confidence.

2. The basic recovery and release posture
Introduces bounded release authority, verification and recovery readiness as connected operating disciplines.

3. How the stack is protected today
Distinguishes protection of infrastructure, application data and customer work; mechanisms carry separate acceptance scopes.

4. Backup and restore drill
Describes why restore exercises and recovery plans must be tested for the particular layer they purport to protect.

5. What is recoverable today
Explains how recoverability depends on which state was copied, where it is held and how it has actually been verified.

6. What is only partially recoverable or still partial
Calls out partial recovery boundaries instead of presenting an infrastructure restore as complete business recovery.

7. Release proof pack
Outlines the role of evidence packages in authorizing a release and retaining a record of the result.

8. Release risk classes
Classifies changes by consequence so evidence and rollback expectations scale with operational risk.

9. Deployment isolation trigger classes
Explains circumstances in which isolation may be needed to protect one workload or deployment from another.

10. What shared posture means today
Describes shared service/deployment posture as a deliberate operating choice with defined limits.

11. What stronger isolation means conceptually
Explains what stronger separation is meant to provide, without presenting an optional architecture as already deployed.

12. What is still open
Separates unfinished, partial and not-yet-validated recovery or release requirements.

13. Safe recovery and release sequence
Summarizes a guarded release/recovery decision path at the level of responsibility and proof, not commands.

14. Release Control Lane — Current Stack Function, Operating Model, and Why It Exists
Explains the release-control lane as the owner of candidate identity, evaluation, staged acceptance, predecessor protection and outcome records.

15. When to stop and escalate
Specifies conditions where recovery or release authority must stop and return to the appropriate owner.

CHAPTER 05 / UI and Working Surfaces Guide

UI and Working Surfaces Guide

A map of the visible work journey: asking, reviewing response and source status, bringing documents into work, saving results, finding retained material and creating actionable work products. Dedicated support controls are differentiated from ordinary daily use.

What the reader learns. Readers learn the product vocabulary and why a visible card, an admitted readable source, a model-generated answer and a saved item represent separate pieces of state.

Interpretation boundary. Moriah supports durable Active Thread Save and Resume, including bounded accepted continuation journeys. A saved-response viewer, sharing permission, or changed-source restore is a separate operation; a screenshot alone does not prove every such workflow.

Inside this chapter

The following 11 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What this chapter is for
Positions the interface as an end-to-end work environment rather than a collection of unrelated screens.

2. The main UI idea
Explains how ordinary user controls should expose useful work without hiding material trust or state differences.

3. Ask and response
Introduces the Ask surface, generated responses and the limits of interpreting available sources as sources actually used.

4. Save, recent, open, and retrieve
Distinguishes saving work, finding it again, opening saved content and restoring a working context as separate functions.

5. Uploads, inbox, and downloads
Maps document upload, Inbox and download behavior to readable material, metadata and source eligibility.

6. Decision brief, next actions, and checklist
Explains how brief, action and checklist outputs can turn work into follow-through without granting external write authority.

7. Support bundle, safe snapshot, and bounded session surfaces
Separates support diagnostics and bounded evidence functions from normal content-creation controls.

8. What users should expect from the main working surfaces
Describes the intended relationship among Ask, sources, outputs and retained work during a real working session.

9. How the UI is intended to be used at a practical level
Shows the user journey from question to useful output to an explicitly retained record where supported.

10. What should not be treated casually
Highlights actions and evidence states that should not be represented as interchangeable or automatically safe.

11. When to stop and escalate
Identifies cases where visible state, expected content or permissions require further examination rather than user guesswork.

Reading the user journey as an evaluation

The chapter is most useful when read as a sequence of separate claims. An operator can ask a question and receive an answer; source cards can indicate different eligibility or use states; an output can be transformed into a brief or next-action format; and eligible work may be saved. Those steps are related, but none substitutes for the acceptance of the next step.

A technical evaluator should examine the actual Active Thread Save → Resume → Continue journey under a named user and work scope; bounded journeys have been accepted. Then evaluate separately saved responses, reopened viewer content, changed-source permissions, and teammate access on their own evidence. These distinct checks strengthen the core continuity story rather than making it merely hypothetical.

CHAPTER 06 / Sessions and Support Guide

Sessions and Support Guide

The purpose of bounded support sessions and evidence packages: capture only the information needed to understand a specific run, describe what happened, preserve provenance and retain privacy and review boundaries.

What the reader learns. Evaluators see how problems are investigated without conflating a support snapshot with unrestricted telemetry or using artifact existence as proof of correctness.

Interpretation boundary. Support material and diagnostic detail remain controlled. The handbook is not a claim that every support command or browser-capture tool is available to every customer.

Inside this chapter

The following 11 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What this chapter is for
Defines a support session as a deliberately bounded record of a specific run or troubleshooting question.

2. What a bounded session is
Separates session purpose, time window and scope from general activity monitoring.

3. What the main session controls are for
Describes the intended roles of start, finish, summary and support-oriented controls.

4. What gets captured in a bounded session
Explains the categories of evidence retained for a run while preserving user/privacy limits.

5. What bounded sessions are not
States what a support session does not do: it is not persistent surveillance or unrestricted diagnostics.

6. Where the artifacts go
Identifies the existence of controlled artifact custody without publishing internal paths or collection procedures.

7. What support bundle and safe snapshot are for
Explains support bundles and safe snapshots as scoped evidence aids, not complete forensic copies.

8. What diagnose is for
Frames diagnosis as a limited operational assessment rather than an automatic repair capability.

9. What is proven today versus still partial
Separates documented and demonstrated support functions from those requiring additional operating-environment proof.

10. How these flows should be used in practice
Outlines responsible use of bounded evidence and review during troubleshooting or acceptance.

11. When to stop and escalate
Identifies privacy, authorization or unclear-result conditions that require escalation.

CHAPTER 07 / Engineer / Technical Onboarding Guide

Engineer / Technical Onboarding Guide

The entry point for technical work: source-of-truth order, architecture navigation, code/runtime distinction, change scope, regression ownership and handoff responsibilities. Its aim is safe continuity of engineering, not broad access.

What the reader learns. A technical reviewer learns that a developer is expected to identify the owning contract and proof surface, protect existing state, and establish exactly what a patch or release demonstrates.

Interpretation boundary. Privileged paths, local environment assumptions, test runners and operator instructions are not reproduced; engineer onboarding is access-scoped, not a public self-service deployment manual.

Inside this chapter

The following 14 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What kind of stack this is
Describes the product as an application, data, policy and operations stack with multiple technical owners.

2. Where technical truth starts
Explains why current contracts, accepted proof and repository canon precede historical summaries or assumptions.

3. The most important paths to know
Distinguishes the main classes of technical owner and runtime custody; exact internal paths are not disclosed.

4. What docs matter first
Maps the technical documentation families an authorized engineer should understand before modifying an owned area.

5. What Atlas is for
Introduces Atlas as the technical owner map used to locate canonical boundaries and responsibility.

6. What the live stack looks like at a basic level
Frames live topology at a conceptual level while keeping network endpoints and environment specifics internal.

7. First technical posture checks
Explains how to confirm initial posture without treating a healthy endpoint as feature acceptance.

8. Non-negotiable technical rules
Emphasizes permission boundaries, non-destructive operation, evidence integrity and protected source material.

9. Compose and runtime safety
Separates runtime operations from repository source work and the need to preserve existing live state.

10. Source of truth versus helper docs
Shows which documents govern current truth and which are explanatory, supporting or historical.

11. How to approach a change safely
Describes small coherent changes, owner authorization and isolation of technical work from other teams.

12. How to prove a change
Connects a change to the right tests, source/runtime proof and bounded acceptance decision.

13. What not to do
Warns against unsupported fixes, recreated proof, guessed source identity and weakening a gate to obtain green.

14. When to stop and escalate
Explains when ambiguous ownership or insufficient proof requires escalation to the responsible technical or change-control owner (termed a Control Tower internally).

CHAPTER 08 / Enterprise and Future Lanes Guide

Enterprise and Future Lanes Guide

The boundary between a stable Moriah operating foundation and optional enterprise extension. It discusses configurable capability slots, isolation choices, connector families and the process for selecting and accepting future work.

What the reader learns. A solutions architect can examine how the product intends to accommodate new providers, tools and organization requirements while keeping identity, policy, provenance and retention under governance.

Interpretation boundary. Some manuscript passages are future-facing historical planning language. Historical Microsoft 365 read workflows have separate bounded acceptance, but current customer setup and new write actions still require their own proof.

Inside this chapter

The following 11 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. What this chapter is for
Explains the distinction between a stable operating foundation and optional new customer-facing capabilities.

2. What is fixed versus what can evolve
Separates trust-bearing invariants from areas where models, connectors or user experience can evolve.

3. The product should stay one product
Argues for one coherent work environment instead of independent products with incompatible identity and state.

4. What enterprise should mean here
Defines enterprise expectations in terms of ownership, permission, service and proof boundaries.

5. Future capability slots
Introduces potential capability families as scoped slots whose presence is not equal to deployment.

6. What must never become configurable
Identifies authorization, provenance and durable-work principles that should remain controlled centrally.

7. Microsoft 365 and enterprise connectors
Discusses Microsoft 365 and enterprise connection patterns; some language reflects an earlier planning snapshot, not the complete accepted implementation history.

8. What belongs now, later, and never
Distinguishes design work worth documenting now, potential later execution and approaches the product should not adopt.

9. What future growth should protect against
Warns about fragmentation, ungoverned configuration and the loss of trustworthy work identity as the platform expands.

10. How future lanes should be promoted
Describes the process by which proposed capability work becomes authorized implementation and then accepted behavior.

11. When to stop and escalate
Sets the escalation boundary where a new product scope or customer integration needs owning authority.

Reading future capability language correctly

The enterprise chapter is deliberately a boundary map rather than a general availability catalog. A provider interface may be implemented before a particular organization has approved credentials. A read-only connector may be useful without write permission. A proposed workflow can be a legitimate planning item without a released executor. These differences are central to estimating integration scope and operational risk.

The manuscript's Microsoft 365 paragraph reflects an earlier planning stage. Separate historical engineering acceptance records document bounded authenticated Outlook and OneDrive read-and-reason workflows; they do not establish that a new tenant is connected today or that message sending, broad SharePoint access, or tracker writeback have been accepted. For a prospective customer, the right evaluation starts with one permitted data source, a concrete task and a testable result.

CHAPTER 09 / Glossary and Reference Pointers

Glossary and Reference Pointers

The handbook's navigation vocabulary: operating and trust terms, user-facing concepts, document/source state, runtime and governance language, and routes to deeper owners. It is the bridge between introductory reading and specialized documentation.

What the reader learns. Readers can interpret terms such as source eligibility, active work, saved work, proof status and operating ownership consistently, without treating similar-looking words as equivalent capabilities.

Interpretation boundary. A definition explains what a term means; it is not proof that a particular provider or deployment currently meets the definition.

Inside this chapter

The following 9 numbered sections retain the manuscript's titles and order. The descriptions explain scope—not privileged instructions.

1. How to use this chapter
Explains that the glossary is for precise language and navigation, not as evidence that a runtime feature is available.

2. Fast where-do-I-go map
Maps common reader questions to the handbook chapter or authoritative technical owner most likely to answer them.

3. Core handbook and canon terms
Defines the key documents and terms used to distinguish canonical instructions from orientation material.

4. Operating and trust terms
Distinguishes privacy, identity, operating authority and trust-related ideas that shape allowed work.

5. User-facing product terms
Defines user-facing terms around Ask, sources, response, save and other work surfaces without equating distinct states.

6. Platform and runtime terms
Defines technical and runtime categories so operators can discuss source, target and proof boundaries accurately.

7. Document-posture and governance terms
Clarifies what makes a document active canon, current reference or historical material.

8. Role-based pointer map
Points different reader roles to the correct source family without publishing privileged links or data.

9. When this chapter is not enough
States when a glossary definition is inadequate and a domain owner or additional evidence must be consulted.

Currentness, supplements and scope

The nine numbered chapters are the stable reading structure. The maintained internal manuscript also has introductory material and dated updates about specific engineering, administrative, evidence and release questions. Those notes should not be mistaken for additional major chapters or for current universal feature acceptance.

What the source supplements add

Proof and source authority. Additional notes cover how a source becomes eligible, readable, selected or used, and why an accepted local test is different from a complete authenticated work journey.

Models, documents and evidence. Dated notes describe provider readiness, extraction/output-engine work and authenticated capture at specific checkpoints. No provider, file format, workflow or media package is automatically available or approved for publication.

Administration, security and release. Dated notes describe bounded administrative, helper, release and continuity work. They are neither a third-party security certification nor current universal deployment acceptance.

Where historical language needs caution

Some passages in the maintained manuscript reflect the period when they were written. For example, early enterprise-connector planning language predates later accepted read-only Outlook/OneDrive workflows. Conversely, acceptance of a bounded save or source operation is not proof of every Open/readback, cross-user or writeback scenario. Specific claims belong to the newest authoritative source and matching evidence.

What the companion deliberately withholds

This public-facing guide does not reproduce privileged commands, internal addresses or infrastructure mapping, credentials and recovery materials, incident records, operational secrets, raw logs, proprietary test fixtures or private customer data. It also does not replace the Technical Architecture & Capabilities Guide, which explains product construction and technical capabilities rather than handbook organization.

Controlled-distribution statement

The internal operating handbook is controlled documentation. This companion describes its coverage and reading order; it is not the full handbook, a demonstration that every documented function is live, a security audit, a formal enterprise certification, or a promise of any particular customer integration.