ODA3 INSTITUTE
ADOPTION / PRACTITIONER FIELD GUIDE

Implementing GAISSF™

A Practitioner’s Guide to Operationalizing GAISSF Controls

Doc ID: ODA3-2026-08-WHP-COM-017
Classification: Public — GEL v1.0

ODA3 INSTITUTE

IMPLEMENTING GAISSF™

A Practitioner’s Guide to Operationalizing GAISSF Controls

Doc ID: ODA3-2026-08-WHP-COM-017
Document Type: WHP — Adoption / Practitioner Field Guide
Framework Context: GAISSF Ecosystem
Classification: Public — GEL v1.0
Status: CORRECTED FINAL R10
Original Publication Date: 13 August 2026
Correction Date: 18 August 2026
Publisher: ODA3 Institute

Publication Boundary

This is an informative practitioner field guide for implementing GAISSF™ controls in enterprise systems. It does not create, amend, replace or supersede the authoritative GAISSF requirements, controlled assessment methodology, certification decision logic, scheme rules, legal obligations, sector-specific safety requirements or organizational release authority.

The implementation patterns, navigation routes, worksheets, examples and practitioner checkpoints in this guide are practical aids. They do not create a mandatory implementation sequence, implementation score, readiness designation, assessment result, certification eligibility or commercial-delivery authority.

Where this guide conflicts with a current controlling ODA3 source, that controlling source governs.

Framework Baseline

COM-017 uses the current public practitioner convention **GAISSF™ — Global AI Safety and Security Framework**.

For this guide:

- GAISSF™ v1.0 uses a **59-control / nine-domain** architecture.
- The current reconciled Foundational baseline is **52 canonical controls in D1–D8**.
- The seven D9 controls are additional where physical AI or cyber-physical actuation is within the applicable scope.
- GAISSF D9 remains retained for continuity; **PAI-SF™** supersedes and expands the detailed physical-AI security-assurance layer represented by D9 where PAI-SF applies.
- Implementation records are candidate evidence sources only; **COM-015** owns public evidence-pack preparation. Any later assessment activity is governed separately under controlled ODA3 methodology and applicable authorization.
- **GEL v1.0** governs use of GAISSF Ecosystem licensed material. Public availability does not create unrestricted Commercial Use or ODA3 partner, assessment or certification authority.

The detailed source register, source precedence and historical-name treatment are provided in **Section 98**.

Part I — Implementation Foundations

Use this part once to understand the implementation model, authority boundaries and handoffs. Then use Section 9 to enter the guide at the point that matches your role or implementation problem.

1. Executive Summary

GAISSF™ becomes useful only when an applicable requirement is translated into something that actually exists and operates in the enterprise: a decision right, architecture boundary, access-control rule, deployment gate, human-oversight mechanism, supplier obligation, monitoring path, recovery capability or other functioning control arrangement.

This guide explains that translation.

The practitioner model used throughout COM-017 is:

> **Scope → Decompose → Translate → Design → Implement → Integrate → Verify → Operate → Observe → Change → Improve**

The model is informative. It is not a new normative GAISSF lifecycle and does not require every organization to execute work in the same order.

Three principles govern the guide:

1. **Implement the requirement in the real system.** Do not reduce implementation to copying control text into policy, buying a product or completing a checklist.
2. **Use the enterprise you already have.** Extend functioning IAM, SDLC, MLOps, GRC, SOC/CSIRT, supplier, privacy, resilience and safety processes where they are adequate rather than creating a parallel GAISSF bureaucracy.
3. **Preserve the assurance boundary.** Implementation produces candidate evidence. COM-015 explains evidence preparation. Any later assessment activity requires a separately governed and authorized process. Implementation itself is not conformity, certification or an assessment outcome.

The guide pays particular attention to the implementation areas where conventional security and governance approaches often need AI-specific extension: model and data provenance, runtime adversarial behavior, agent identity and tool authority, RAG source control, provider opacity, human oversight, material change, physical-effect paths and incident reconstruction.

You do not need to read all 100 sections linearly. Section 9 provides role-based, situation-based and lifecycle routes through the guide.

2. Purpose and Audience

COM-017 answers one question:

> **How does an enterprise actually implement GAISSF controls and operationalize the framework in real systems?**

It is written primarily for CISOs, AI Security Leads, AI Governance Leads, Security and Enterprise Architects, system/service owners, control owners, engineering leads, AI/ML and MLOps teams, platform and DevSecOps teams, GRC teams and internal assurance functions.

Additional functions may include data governance, privacy, legal/compliance, procurement, supplier management, SOC/CSIRT, resilience, business continuity, OT/ICS, safety engineering and clinical or other specialist functions where the system context requires them.

The document is not primarily an assessor manual. Any reference to later assessment exists only to keep implementation work traceable and to prevent implementation-side checks from being misrepresented as independent assurance conclusions.

---

3. Publication and Authority Boundary

COM-017 is an informative public practitioner guide. It does not create, modify, replace or supersede:

- GAISSF™ v1.0 normative requirements;
- the GAISSF Control Catalogue;
- controlled scope, applicability or profile decisions;
- organization-specific engineering, architecture or risk decisions;
- sector-specific safety, quality, privacy, cybersecurity or legal requirements;
- UAIF™ incident-classification rules;
- AI-IRF™ incident-response requirements;
- PAI-SF™ detailed physical-AI requirements;
- COM-015 evidence-preparation guidance;
- public assessment-participation or certification guidance;
- controlled assessor methodology, scoring or certification decision logic.

Where this guide conflicts with a current authoritative GAISSF source, the authoritative source governs.

Implementation is not conformity. A configured tool, completed workflow, policy, architecture diagram, test result or operational record does not by itself establish control effectiveness, evidence sufficiency, assessment readiness, certification, regulatory compliance, product safety or operational assurance.

---

4. Where COM-017 Sits in the Practitioner Suite

The practitioner handoff is:

scope / applicability
→ **implementation**
→ **evidence generation**
→ **COM-015 evidence preparation**
→ **separately governed assessment activity, where authorized**
→ **bounded reporting / remediation, where applicable**

COM-009 provides an introductory implementation-management spine. COM-010 provides hypothetical application scenarios. COM-011 through COM-013 cover incident orientation,

classification and response. COM-014 provides the detailed physical-AI practitioner layer. COM-015 prepares evidence. Any later assessment activity sits outside the public practitioner-guide sequence and is governed separately.

COM-017 owns the **implementation layer** between a determined requirement and a functioning enterprise control arrangement.

---

5. GAISSF™ Implementation in Plain Terms

Implementation is the act of converting an applicable requirement into a control that has:

- a defined scope;
- an accountable owner;
- one or more operating mechanisms;
- known dependencies;
- a place in the enterprise architecture or workflow;
- a way to detect failure or material change;
- a method for implementation-side checking;
- an operating and maintenance owner;
- a change and remediation path;
- naturally generated records.

A control can be implemented through governance, technical and operational mechanisms in combination. For example, a requirement concerning consequential autonomous actions may need a governance decision defining permitted authority, a technical authorization mechanism enforcing that authority and an operational process for monitoring exceptions and changes.

Implementation quality therefore depends on the **control system as a whole**, not on one artifact or product.

---

6. Implementation Is Not Conformity

An organization may accurately say that it has implemented a particular mechanism or is implementing GAISSF™ controls when that statement is factually supportable and properly scoped.

That statement does not establish that:

- every applicable requirement is satisfied;
- the mechanism operates effectively across the relevant period;
- contrary evidence is absent;
- an independent assessment would reach a favorable conclusion;
- the organization is certified or approved by ODA3 Institute;
- legal or regulatory obligations are satisfied.

Implementation status should therefore remain a management statement about work performed and mechanisms deployed, not a substitute for a later assurance determination.

---

7. Implementation Is Not Assessment

Implementation teams may verify that a technical or operational mechanism behaves as intended. Examples include checking that a policy decision blocks a prohibited tool action, confirming that rollback restores an approved configuration or validating that a provider-version change generates the expected internal alert.

Those checks are **implementation-side verification**. They are not assessor determinations.

COM-017 does not publish or create:

- public control scoring;
- assessor confidence scales;
- assessment result categories;
- finding-severity thresholds;
- universal sampling rules;
- certification eligibility criteria;
- certification decision logic.

A practitioner tool in this guide does not become an assessor disposition simply because it is well documented.

---

8. Implementation Artifacts Are Candidate Evidence

Implementation naturally produces records: architectures, approvals, configurations, version records, deployment manifests, test outputs, monitoring events, supplier records, change tickets, training records, incident-readiness exercises and corrective-action records.

Those records are **candidate evidence sources**.

COM-017 does not decide whether they are sufficient or admissible for a later assessment. It does, however, encourage implementation teams to preserve enough identity, scope, ownership, version and change context that the records can later be prepared properly under COM-015.

The handoff is:

> **implementation artifact → candidate evidence source → COM-015 evidence preparation → authorized handoff, where applicable**

Build sound controls because the system needs them. Do not manufacture artifacts solely to create a favorable review appearance.

---

9. How to Use This Guide

COM-017 is designed as a **reference field guide**, not a document that every practitioner must read from page 1 to page 100.

Start with a system or implementation scope for which there is a governed basis for determining applicable GAISSF requirements. Then choose the route below that matches the work in front of you.

9.1 Implementation Path at a Glance

Implementation task	Primary sections	Practitioner question	Typical working output
Bound the system	10–20	What exactly are we implementing against, and where can the requirement fail?	working scope, inventory, component/dependency view, assumptions and unknowns
Translate requirements	21–32	What system-specific condition should exist, who owns it, and how will it work?	implementation objective, mechanism, owner, verification, monitoring and exception record
Integrate with the enterprise	33–44	Which existing enterprise processes can carry the control, and where is AI-specific extension needed?	IAM/SDLC/ MLOps/GRC/ procurement/SOC/ change integration
Apply AI-specific patterns	45–54	Which model, runtime, agentic, RAG, provider, human or physical-AI patterns are relevant?	architecture/control refinements for the applicable AI risk surface
Deploy and operate	55–63	How will the implementation enter production, remain owned, and respond to change or degradation?	release/rollback plan, operating ownership, monitoring and change triggers
Design for incident readiness	64–72	Can responders identify, reconstruct, preserve, contain and recover the system when something goes wrong?	incident-readiness capabilities and interfaces
Remediate and improve	73–78	How are gaps, failed tests, incidents and changes converted into controlled improvement work?	implementation backlog, corrective action and verification
Prepare the evidence handoff	79–83	What records arose naturally from implementation, and where does COM-017 stop?	candidate implementation artifacts routed to COM-015
Use practitioner tools	84–92	Which worksheet or register helps the	working implementation

Implementation task	Primary sections	Practitioner question	Typical working output
		current task?	records and review aids
Check boundaries / claims	93–100	Are we overclaiming, misusing licensing language or crossing into controlled methodology?	bounded external/internal statements and source traceability

This path is informative rather than mandatory. Material change may send a team back to scope, architecture, implementation design or another earlier step.

9.2 Role-Based Entry Routes

Role / function	Start here	Then focus on
CISO / AI Security Lead	1–9	10, 21, 27, 33, 39–43, 59–63, 73–78, 92–97
AI Governance / GRC / Risk	4–9	10, 20–23, 27, 32, 39–40, 51–53, 59–63, 73–83, 86–92
Security / Enterprise Architect	10–20	21, 28–31, 33–44, relevant 45–54, 55–72, 85–90
AI/ML / MLOps / Platform Engineering	11–16	21–31, 35–38, 43, 45–50, 55–63, 65–70, 86, 88–90
System / Service / Control Owner	10–12	21–32, 55–63, 73–83, 86, 88, 90, 92
Procurement / Supplier / Third-Party Risk	14, 18	28, 40, 49, 62, 69, 80–82, 87, 89
SOC / CSIRT / Incident Readiness	13–16	41, 60, 64–72, 91; use COM-011, COM-012 and COM-013 for active incident work
OT / Safety / Physical-AI Team	19, 26	42, 54, 57, 63, 71, 90–92; then use COM-014 for the detailed PAI-SF layer
Evidence / Internal Assurance Team	6–8	20–22, 30–31, 79–83, 86–92; then use COM-015 for the public evidence-preparation layer

The routes overlap deliberately because implementation is cross-functional.

9.3 Situation-Based Routes

If the immediate problem is:

- **“We are starting GAISSF implementation for the first time.”**
  Read Sections **1–9**, then work through **10–32** for one bounded system before scaling the pattern.

- **“We already have a production AI system and need to operationalize controls.”**
  Start at **10–20**, then use **33–63**, relevant AI-specific sections **45–54**, and **64–72** for incident readiness.
- **“The system is agentic or can take external actions.”**
  Prioritize **16, 21, 26, 28–31, 35, 47, 60, 64–70, 85–91**.
- **“We rely heavily on a foundation-model / SaaS provider.”**
  Prioritize **14, 18, 28, 40, 49, 62, 69, 80–82, 87, 89**.
- **“The system uses RAG or enterprise knowledge sources.”**
  Prioritize **15, 35, 38, 43, 48, 60, 65–68, 85–90**.
- **“AI can materially influence physical action.”**
  Start with **19 and 54**, then use **COM-014 — Securing Physical AI** for the detailed PAI-SF implementation layer. Sections **63 and 71** remain important for degraded operation and safe containment interfaces.
- **“We are preparing evidence.”**
  Use **79–82** only for the implementation-to-evidence handoff, then move to **COM-015**. Do not turn COM-017 into the evidence-pack methodology.
- **“We are preparing for or participating in an assessment.”**
  Follow the separately authorized assessment process applicable to the engagement. COM-017 does not provide assessor workflow, assessment-participation instructions or assessment-result logic.
- **“An incident is happening now.”**
  Use **COM-011, COM-012 and COM-013, together with the applicable AI-IRF™ process**. Sections **64–72** explain what should have been implemented beforehand; they are not the active-response playbook.

9.4 Practitioner Tool Map

Tool	Section	Best used when	What it helps make visible
Implementation Scope Worksheet	84	starting or materially changing scope	bounded system, owners, providers, data/tools, physical interfaces and open questions
AI System Decomposition Worksheet	85	mapping architecture and trust boundaries	components, authority, state, dependencies, failure modes, monitoring and recovery
Control Implementation Record	86	translating an applicable control into operating work	objective, owners, mechanisms, dependencies, verification, monitoring and limitations
Shared Responsibility Matrix	87	providers/shared services carry material control	enterprise/provider mechanisms, visibility limits and

Tool	Section	Best used when	What it helps make visible
		responsibility	fallback/exit
Implementation Backlog	88	gaps or improvements need delivery ownership	action, owner, dependency, verification and project status
Dependency Register	89	several controls depend on the same service/provider/person/process	common-mode assumptions, monitoring, failure consequence and fallback
Change-Trigger Register	90	defining when implementation must be reconsidered	model/provider/tool/data/permission/infrastructure/use/physical changes
Incident-Readiness Implementation Check	91	design review, pre-deployment or major change	identification, reconstruction, preservation, containment, escalation and recovery capability
Implementation Review Checklist	92	internal implementation-quality review	whether key implementation dimensions remain visible before handoff

These tools are optional working aids. They do not create a mandatory GAISSF implementation package or an ODA3 readiness state.

9.5 Route Out of COM-017 When the Problem Changes

Stop using COM-017 as the primary guide when the work has moved into another governed layer:

- **applicability/profile uncertainty** → return to the authoritative GAISSF scoping/applicability source;
- **active incident classification** → UAIF™ / COM-012;
- **active incident response** → AI-IRF™ / COM-011 / COM-013;
- **detailed physical-AI implementation** → PAI-SF™ / COM-014;
- **evidence-pack preparation** → COM-015;
- **assessment participation** → the separately governed and authorized assessment process, where one exists;
- **legal, regulatory, safety or sector approval** → the applicable controlling authority/process;
- **commercial third-party delivery** → GEL v1.0 and any separately applicable written ODA3 authorization requirements.

For each applicable control, the recurring practitioner questions remain:

1. What is actually in scope?
2. Where can the requirement fail?

3. What system-specific condition should exist?
4. Which mechanism or combination of mechanisms can create that condition?
5. Who owns the decision and who operates the mechanism?
6. What does the control depend on?
7. How will the implementation team know the mechanism is functioning?
8. How will degradation or material change become visible?
9. What naturally generated records should be preserved?
10. What would require the implementation to be revisited?

Examples in this guide are illustrative. Select mechanisms appropriate to the real architecture, risk, sector, applicable obligations and authoritative requirement.

Part II — Establish the Implementation Boundary

Use this part to build a working implementation boundary before selecting mechanisms. The output is a traceable view of the system, components, providers, data, agents/tools, shared controls, physical-effect paths, assumptions and unknowns.

10. Start from the Authoritative Scope

Implementation should begin from an authoritative scope/applicability determination, not from whichever controls are easiest to implement.

As a practical starting point, the implementation team should be able to identify the legal entity, business process, AI system, deployment environment, users, geographies, providers, interfaces and lifecycle stages relevant to the working boundary. This is not a universal minimum-data requirement. If the scope or profile determination is still provisional, record that state rather than treating it as settled.

A change in intended use, autonomy, data access, provider, deployment environment or physical consequence can invalidate earlier assumptions. Scope is therefore a controlled implementation input, not a one-time administrative exercise.

---

11. Build the AI System Inventory

The inventory should be detailed enough to support control placement and ownership. Useful inventory elements include:

Inventory element	Practitioner detail
System / service	Name, purpose, owner, business process and lifecycle state
Models	Provider, model family/version, local/hosted state and configuration
Prompts / orchestration	System instructions, workflow logic, routers and policies
Data	Training, fine-tuning, retrieval, operational and feedback data

Inventory element	Practitioner detail
Agents	Identity, sponsoring user/service, goals, memory and delegation
Tools / APIs	Reachable capabilities, permissions and downstream effects
Providers	Models, datasets, SaaS, libraries, platforms and critical support services
Infrastructure	Environments, networks, identities, secrets, compute and regions
Human roles	Users, reviewers, operators, approvers and escalation authorities
Physical interfaces	Sensors, actuators, machinery, vehicles or other effect paths where relevant

Do not assume the CMDB, model registry, application inventory or procurement register alone contains the full AI boundary. Reconcile them where needed.

---

12. Define the System Boundary

A useful system boundary answers more than “which application is this?” It identifies the path through which risk can enter, propagate or be controlled.

Document:

- user and machine entry points;
- authentication and session boundaries;
- application and orchestration layers;
- model endpoints and routing;
- data/retrieval stores;
- agent memory and state;
- tools and downstream systems;
- external provider connections;
- monitoring and policy enforcement services;
- recovery or rollback dependencies;
- physical actuation paths where applicable.

The boundary should include material dependencies even when they are owned by another team or supplier. Ownership does not remove a dependency from the implementation model.

---

13. Decompose Components and Trust Boundaries

A trust boundary exists wherever identity, authority, data, instruction, state or control responsibility changes.

Typical boundaries include:

- human-to-AI interaction;

- application-to-model service;
- application-to-retrieval source;
- model/agent-to-tool;
- enterprise-to-provider;
- development-to-production;
- control plane-to-data plane;
- monitoring source-to-assurance repository;
- AI decision-to-physical action.

For each material boundary, document the assets crossing it, the identities involved, permitted actions, relevant controls, monitoring, failure modes, owner and recovery path.

A diagram that shows boxes without authority, data flow or trust transitions is usually insufficient for implementation design.

---

14. Identify Models, Providers and Versions

AI behavior can change because the model, provider, model configuration, system prompt, routing policy or runtime wrapper changes. Treat those elements as controlled implementation dependencies.

A practical record should identify, where available:

- provider and service;
- model name/family and version or equivalent release identifier;
- approved endpoint or registry entry;
- configuration and policy baseline;
- evaluation baseline associated with the approved state;
- deployment history;
- change-notification or detection method;
- rollback/fallback options;
- known provider visibility limitations.

Where a provider does not expose a stable model version, record the limitation and compensate through the organization’s own behavioral monitoring, change detection or contractual arrangements as appropriate.

---

15. Identify Data and Retrieval Dependencies

Data implementation should cover the full path from source to use, not merely the storage location.

Identify:

- data origin and owner;
- classification and permitted purpose;
- provenance and lineage;
- ingestion and transformation steps;
- access controls;

- retrieval/indexing layers;
- tenant or user separation;
- freshness and update behavior;
- feedback or adaptation paths;
- retention and deletion constraints;
- supplier or licensing dependencies;
- integrity/poisoning concerns.

For RAG systems, treat source authorization, index integrity, context assembly and model behavior as separate control surfaces. Incorrect retrieval is not automatically proof of poisoning, and poisoned source content is not the only explanation for a harmful answer.

---

16. Identify Agents, Tools and External Actions

Agentic systems require a boundary that extends beyond the model response. Identify:

- sponsoring user or service;
- agent identity;
- objective and instruction source;
- planning/orchestration component;
- memory/state;
- tools and APIs;
- credentials and permissions;
- delegated sub-agents;
- human-approval points;
- transaction/action constraints;
- resulting state changes;
- telemetry and containment paths.

A system that can call an API, create code, alter a repository, send a message, approve a payment or change a physical state should be implemented around **action authority**, not just output filtering.

---

17. Identify Shared Enterprise Controls

Many GAISSF requirements may depend on enterprise controls that already exist. Examples include:

- IAM and privileged access;
- secure SDLC and change management;
- network segmentation;
- secrets management;
- logging/observability;
- vulnerability management;
- privacy governance;
- procurement and supplier management;
- incident response;

- business continuity;
- records management.

Reuse these controls when their scope and operation genuinely support the AI system. Do not duplicate a functioning process merely to create a GAISSF-branded copy.

Equally, do not assume a general enterprise control automatically covers AI-specific subjects such as model versions, agent permissions, retrieval sources, AI provider changes or physical actuation.

---

18. Identify Supplier and Inherited Controls

For every material supplier dependency, distinguish what is implemented by the provider from what remains the enterprise's responsibility.

Use four practitioner descriptors:

- **Enterprise-controlled** — the organization directly implements and operates the mechanism.
- **Provider-controlled** — the provider operates the mechanism and the organization relies upon it.
- **Shared** — both parties contribute mechanisms necessary to the control objective.
- **Visibility-limited** — material provider-controlled behavior cannot be independently observed by the enterprise.

These are implementation-responsibility descriptors, not assessment result categories.

They are also **not one-to-one equivalents of COM-015 evidence-source states**. For example, a provider-controlled mechanism may still generate customer-observed evidence; a shared implementation responsibility may produce provider-provided, customer-observed or shared/interface evidence; and a visibility-limited control may remain only partially evidenced. Preserve the distinction between **who operates the control** and **who can evidence what happened**.

Record evidence/access limitations, change-notification paths, provider support routes, fallback/exit arrangements and what happens if a provider-controlled mechanism becomes unavailable or unverifiable.

---

19. Identify Physical-Effect Paths

Where AI-enabled sensing, inference, autonomy or command can create or materially influence physical consequence, implementation analysis should extend through the effect path rather than stopping at the model or application boundary.

Map the material physical consequence path using the current COM-014 practitioner sequence where applicable:

environment → sensing/perception → inference → autonomy state → command authorization → command → actuator state → safety-controller state → operator action → physical outcome → recovery/revalidation

Architecture teams may add intermediate components such as planners, gateways, networks, controllers or policy engines, but should not omit the authorization, safety-controller, operator and outcome interfaces when they materially affect the consequence path.

This mapping is a handoff trigger to PAI-SF™ where applicable. COM-017 does not replace the detailed physical-AI requirements or specialist functional-safety engineering.

Do not assume an IT containment action is safe merely because it disconnects a component. Network isolation, power-cycling or loss of telemetry can itself create a hazard in a cyber-physical system.

---

20. Record Assumptions, Unknowns and Constraints

Implementation designs are often built with incomplete provider visibility, evolving architecture and uncertain operating conditions. Record those constraints instead of silently converting them into certainty.

Typical items include:

- provider version opacity;
- unavailable telemetry;
- incomplete data lineage;
- inherited control assumptions;
- unverified third-party representations;
- technical limitations on rollback;
- privacy/legal limits on logging;
- unresolved system ownership;
- physical operating assumptions;
- dependencies on future tooling or contracts.

An assumption should have an owner and a review trigger where it is material. An unknown that affects risk or control behavior should be visible in the implementation backlog.

Part II checkpoint: before moving into control design, a practitioner should be able to point to the working system boundary, material components/dependencies, provider and agent/tool interfaces, physical-effect path where relevant, and the assumptions or unknowns that could change implementation. This is an implementation checkpoint, not a formal applicability or assessment determination.

Part III — Translate Requirements into Implementation

Use this part to turn applicable requirements into system-specific objectives, owners, mechanisms, dependencies, implementation-side verification, monitoring and exception handling.

21. From Requirement to Implementation Objective

Do not rewrite the authoritative requirement into a new quasi-standard. Translate it into a **system-specific implementation objective**.

Use the chain:

> **Authoritative requirement → affected system condition → implementation objective → mechanism → owner → operation → verification → monitoring/change trigger → generated artifact**

For example, if a requirement concerns bounded agent authority, the implementation objective might be that an agent can only execute an approved class of actions using attributable identity and an enforceable permission set. The selected mechanism could combine per-service identity, tool allowlists, transaction constraints and approval policy.

That example does not mean every system must use the same technology or exact mechanism.

---

22. The Control Implementation Record

A practical implementation record can be maintained per applicable control or per traceable control grouping where organizational tooling supports reliable one-to-many relationships. The record below is a practitioner aid, not a new mandatory GAISSF record format.

Field	Recommended practitioner content
Requirement / Control	Authoritative GAISSF reference
Scope element	System, component, supplier, process or location affected
Implementation objective	System-specific condition sought
Accountable owner	Function accountable for implementation
Operating owner	Function/service operating or maintaining the mechanism
Mechanism	Governance, technical and operational controls used
Dependencies	Shared services, suppliers, identities, data, tools and people relied upon
Shared-responsibility state	Enterprise-controlled, provider-controlled, shared or visibility-limited
Internal implementation state	Local project-management status only; not an ODA3 assessment state
Verification method	Engineering/operational check
Monitoring	How failure, degradation or change becomes visible
Change triggers	Events requiring review
Generated artifacts	Candidate implementation artifacts
Assumptions / limitations	Known gaps or constraints
Related actions	Remediation or backlog references

Avoid status labels that look like conformity outcomes. Simple internal labels such as Planned, In Progress, Implemented, Blocked or Retired may be used if the organization clearly treats them as project/operational states rather than GAISSF assessment results. In this guide, **Implemented** means only that the organization reports the planned mechanism as deployed/operating for the stated scope; it does not establish effectiveness, evidence sufficiency, conformity or assessment readiness.

23. Governance Implementation

Governance implementation creates the authority structure within which technical and operational controls can function.

Typical mechanisms include:

- system and control ownership;
- risk-acceptance authority;
- policy and standards;
- approval rights;
- exception governance;
- procurement requirements;
- escalation routes;
- management review;
- user or stakeholder obligations;
- change approval.

A governance control is not implemented merely because a policy exists. Verify that the policy has a real owner, is reflected in decisions, is connected to workflows and can be enforced or escalated when breached.

---

24. Technical Implementation

Technical implementation places enforceable mechanisms into architecture and runtime paths.

Examples include:

- identity and authorization;
- model/provider allowlists;
- version pinning or release registries;
- data lineage and integrity controls;
- retrieval authorization;
- content/output controls;
- tool/action constraints;
- sandboxing;
- rate/transaction limits;
- secure configuration;
- telemetry;
- rollback;
- deterministic or independently assured safety constraints where applicable.

A tool purchase is an implementation input, not an implementation conclusion. The implementation question is whether the mechanism is appropriately scoped, configured, integrated, operated and monitored for the control objective and system context.

---

25. Operational Implementation

Operational implementation keeps the control alive after deployment.

It includes:

- recurring review;
- monitoring and alert handling;
- supplier oversight;
- exception handling;
- incident readiness;
- change management;
- maintenance;
- operator training;
- exercises;
- corrective action;
- decommissioning and retirement.

An otherwise sound technical control can fail operationally if no one reviews its alerts, provider changes are ignored, the approved version drifts or recovery procedures are unavailable.

---

26. Human Factors Across Implementation

Human oversight is not satisfied merely by naming a reviewer.

Where human authority is material, align implementation with the five-dimension practitioner pattern used across COM-014 and COM-015:

1. **Awareness** — does the person know that review or intervention is required?
2. **Comprehension** — can the person understand the relevant system state, decision or risk?
3. **Time** — is there enough time to act before the decision or consequence becomes effectively irreversible?
4. **Authority** — is the person actually permitted to intervene, approve, stop, escalate or override?
5. **Effectiveness** — can the intervention materially constrain, change or reverse the relevant system outcome?

Supporting conditions may include access to the required interface/control, competence to interpret system behavior, escalation support when uncertainty remains, manageable workload and safeguards against obvious automation-bias or rubber-stamp review.

Human factors may influence governance, technical interface design and operational procedure simultaneously. COM-017 does not create universal pass/fail thresholds for these dimensions.

---

27. Assign Accountable and Operating Owners

Separate accountability from execution where needed.

Typical functions include:

Function	Implementation role
Executive sponsor	Authority, resources and cross-functional support
System/service owner	Intended use, service risk and system-level coordination
Control owner	Accountable for the control within scope
Control operator	Executes/maintains the mechanism
Enterprise/security architect	Architecture placement, trust boundaries and shared controls
AI/ML engineering / MLOps	Model, data, evaluation, deployment and runtime controls
Platform / DevSecOps	Infrastructure, CI/CD, identity, secrets and telemetry
Data/privacy	Data use, minimization, lineage, access and privacy constraints
Supplier owner	Provider requirements, support, evidence, change and exit
SOC/IR	Detection, incident integration, containment and recovery interfaces
Business owner	Consequence, continuity and business decision authority
Legal/compliance	Applicable obligations and legal interpretation
Safety/OT/clinical	Specialist safety authority where relevant
Evidence custodian	Repository/record custody, not evidence sufficiency

Small organizations may combine functions. Combining functions does not remove the need to define authority and handoffs.

---

28. Map Dependencies and Preconditions

A control rarely operates alone. Record what must be true for it to work.

Examples:

- an authorization control may depend on accurate identity and tool metadata;
- a provider-change alert may depend on contractual notice or behavioral monitoring;
- a retrieval control may depend on source metadata and tenant isolation;
- a rollback procedure may depend on an archived known-good model and compatible data/schema state;
- a human-approval control may depend on a server-side enforcement point rather than a UI-only prompt;
- a physical safe-state control may depend on independent local logic and available sensor state.

Dependency mapping makes silent common-mode failures visible.

---

29. Select Implementation Mechanisms

Select mechanisms based on the actual architecture, risk and authoritative requirement rather than a fixed technology pattern.

Useful selection questions are:

- Can the mechanism enforce the intended boundary?
- Is it positioned before the risky action occurs?
- Does it rely on the same component it is intended to constrain?
- Can it be bypassed through another path?
- Can failure be detected?
- Can the mechanism be maintained when providers or models change?
- Is its operation compatible with privacy, safety and legal constraints?
- Can it fail in a controlled way?

Prefer simpler mechanisms when they achieve the required outcome reliably. Complexity that cannot be operated or observed may create new assurance problems.

---

30. Define Implementation-Side Verification

Verification asks whether the mechanism behaves as intended in the implementation context.

Examples include:

- confirming that a denied agent action is blocked at the server-side policy point;
- testing negative retrieval across tenants/roles;
- checking that an unauthorized model version cannot be promoted;
- exercising rollback in a non-destructive environment;
- confirming that provider-change alerts reach the assigned owner;
- testing whether a human approval actually prevents pre-approval execution;
- validating safe containment under specialist safety authority for physical systems.

Verification should preserve failed and inconclusive results. It should also record the environment and version tested.

Do not use this section to infer later assessment sufficiency.

---

31. Define Monitoring and Change Triggers

Monitoring should answer two implementation questions:

1. **How will we know the control or its dependency has degraded?**
2. **How will we know the system has changed enough that the implementation must be revisited?**

Potential signals include model/provider changes, new tools, permission changes, source updates, drift, repeated policy bypass, supplier incidents, unexplained behavior change, monitoring loss, user complaints, near misses or changes to physical operating conditions.

The organization should define its own thresholds and decision criteria. COM-017 does not prescribe universal operational thresholds.

---

32. Manage Exceptions Without Rewriting Requirements

An implementation exception is a governed deviation or temporary inability to implement a requirement as intended. It should not silently become a new local interpretation of GAISSF.

Record:

- affected requirement and scope;
- reason;
- risk and consequence;
- compensating mechanisms where applicable;
- accountable owner and risk authority;
- review/expiry condition;
- remediation action;
- monitoring;
- change triggers.

A permanent workaround should be re-examined as a control-design problem rather than renewed indefinitely without new analysis.

Part III checkpoint: the team should now be able to explain, for each implementation item in scope, the objective sought, who owns it, which mechanisms carry it, what it depends on, how implementation-side verification will occur, what change/degradation looks like and which limitations remain open.

Part IV — Integrate GAISSF into Enterprise Architecture

Use this part to place implementation into existing IAM, SDLC, MLOps, data/privacy, GRC, procurement, SOC, resilience and change processes rather than building a parallel GAISSF operating stack.

33. Reuse Existing Enterprise Controls First

Most enterprises already operate security, privacy, risk, quality, incident, supplier and change processes. GAISSF implementation should reuse functioning controls where they genuinely support the requirement and scope.

Avoid two opposite errors:

- **duplicate bureaucracy** — creating a second approval, risk register or incident process solely because GAISSF is being adopted;
- **false reuse** — assuming a general control automatically covers model, data, agent, tool or physical-AI specifics.

Map intent and operation, not labels. A similarly named policy may still require AI-specific extension.

34. Enterprise and Security Architecture

Architecture integration should show where controls sit relative to trust boundaries and failure paths.

A practical layered view may include:

- governance and assurance plane;
- security/policy enforcement plane;
- user/channel layer;
- application and orchestration layer;
- model-service layer;
- data/knowledge layer;
- agent/tool/action layer;
- platform/infrastructure layer;
- physical actuation/environment layer where relevant.

This is a logical pattern, not a mandatory topology. The important property is traceability from requirement to boundary, mechanism, owner, monitoring and recovery.

---

35. IAM and Privileged Access

Extend IAM beyond human users.

Identify and govern:

- service identities;
- model endpoints;
- agents;
- tools;
- automation accounts;
- privileged administrators;
- deployment identities;
- provider credentials;
- emergency/recovery access.

For agentic systems, authorization should consider capability as well as identity: objective, tool, action type, resource, transaction, destination, time and delegation rights may all be relevant.

Avoid shared unrestricted credentials where attributable identity and revocation are required.

---

36. AI/ML Engineering and MLOps

Integrate GAISSF into the model/data lifecycle instead of treating it as a post-build review.

Implementation touchpoints can include:

- model and dataset provenance;
- approved registries;
- reproducible build/release records;
- evaluation baselines;
- fine-tune/adapter integrity;
- configuration control;
- release promotion;
- drift/change detection;
- rollback;
- retirement.

Link implementation work to the same engineering backlog and release workflow used for product development. Control tasks should not live only in a GRC spreadsheet disconnected from the team that operates the system.

---

37. SDLC / DevSecOps / CI-CD Integration

Where appropriate, implement controls as gates or checks in existing engineering workflows.

Potential integration points include:

- source/dependency review;
- data ingestion checks;
- model/package provenance;
- prompt/orchestration change review;
- authorization tests;
- infrastructure/configuration validation;
- secret scanning;
- release manifests;
- rollback readiness;
- post-release monitoring.

A failed gate should route to the organization’s governed exception or remediation process rather than be bypassed informally.

Avoid importing old implementation-source numeric pass criteria into COM-017 unless they are verified and currently authoritative.

---

38. Data Governance and Privacy

AI implementation should connect model and application behavior to existing data-governance rules.

Address:

- permitted purpose;
- data minimization;
- classification;

- access;
- provenance;
- lineage;
- retrieval authorization;
- tenant separation;
- retention/deletion;
- feedback and adaptation use;
- provider disclosure;
- logging/preservation constraints.

Do not treat comprehensive logging as a universal good. Prompts, outputs and retrieval context can themselves contain sensitive, regulated or privileged information. Design observability and evidence preservation with minimization and access controls.

---

39. GRC and Risk Governance

GRC systems can provide useful structure for ownership, applicability, exceptions, risk decisions, actions and evidence references, but they should not become the place where implementation is merely asserted.

For each control, connect the GRC record to the real operating mechanism and accountable technical/business owner.

Useful integrations include:

- scope and applicability;
- control ownership;
- risk acceptance;
- exception expiry;
- remediation backlog;
- supplier dependency;
- change impact;
- management review;
- claims review.

Do not create a parallel “GAISSF risk” divorced from the organization’s existing enterprise risk system when those risks can be governed coherently within it.

---

40. Procurement and Supplier Management

Provider-dependent AI controls frequently fail because procurement captures commercial terms but not operational assurance dependencies.

Where relevant, contracts or supplier processes should address:

- service/model identification;
- material change notification;
- incident notification/support;
- data use and retention;

- subprocessor/dependency visibility;
- security/support obligations;
- evidence access;
- portability/exit;
- continuity/fallback;
- decommissioning and post-termination access.

These are implementation patterns, not a universal contract clause set. Legal/commercial terms remain context-specific.

---

41. SOC / CSIRT Integration

GAISSF implementation should fit around the organization's existing SOC/CSIRT process.

Add AI-specific observability and response interfaces rather than creating a parallel incident system unless separation is operationally necessary.

Ensure the environment can support:

- affected model/provider/version identification;
- agent/tool action attribution;
- prompt/configuration/retrieval change reconstruction where relevant;
- provider case/support references;
- containment of identities/tools/connectors/models;
- evidence preservation;
- rollback/recovery;
- physical-safety escalation where applicable.

UAIF™ and AI-IRF™ govern their respective incident functions. COM-017 does not create new incident record types or severity scales.

---

42. Resilience and Business Continuity

AI control implementation should define what happens when a model, provider, retrieval store, policy service, tool, monitoring pipeline or physical component becomes unavailable or unreliable.

Possible implementation patterns include:

- fallback model/service;
- queueing;
- controlled degradation;
- abstention;
- human takeover;
- feature restriction;
- provider switch;
- cached approved policy where appropriate;
- rollback to a known-good state;
- physical safe-state logic under specialist authority.

The resilience design should consider the **control consequence** of degradation, not just service availability.

---

43. Change and Configuration Management

Maintain an approved baseline for material model, data, prompt, orchestration, tool, policy, identity, provider and infrastructure components.

A change record should identify:

- what changed;
- old/new version or state;
- affected scope;
- affected controls/dependencies;
- testing or validation required;
- rollback or fallback;
- approval authority;
- implementation artifacts that need refresh;
- public or contractual claims affected where relevant.

Emergency change may shorten normal sequencing under organizational policy, but it should not eliminate accountability, traceability, preservation of the pre-change state or retrospective review.

---

44. Retirement and Decommissioning

Retirement is a control activity, not merely a project closure.

Confirm:

- users and integrations are identified;
- processing and external actions are stopped;
- credentials, tool permissions and supplier access are revoked;
- data is retained, returned or disposed according to applicable requirements;
- models/prompts/configurations are archived where needed;
- incident and implementation records are preserved appropriately;
- public inventories/claims are updated;
- residual contractual, legal or support obligations are assigned.

Where a legacy mechanism is being replaced, avoid premature retirement before the replacement is operationally established.

Part IV checkpoint: for each material mechanism, the team should be able to identify where it lives in the existing enterprise operating model, which function operates it, which AI-specific extension was needed and how handoffs to adjacent enterprise processes are maintained.

Part V — AI-Specific Implementation Patterns

Read only the patterns relevant to the system. These sections help extend conventional controls for AI-specific risk surfaces; they do not replace the authoritative GAISSF Control Catalogue or create a fixed implementation architecture.

45. Model and Dataset Integrity

Current GAISSF Domain D1 addresses **Model Integrity & Adversarial Robustness**. At implementation level, useful control surfaces include:

- dataset provenance;
- source integrity;
- model/package identity;
- signed or controlled artifacts where appropriate;
- fine-tune/adapter provenance;
- controlled build and promotion;
- behavioral baseline;
- version/change detection;
- rollback/recovery.

Illustrative implementation pattern: an enterprise-hosted model is admitted into a controlled registry only after its provider/source, package hash or equivalent identity, configuration, evaluation baseline and approval status are recorded. Production deployment references the registry entry rather than an ad hoc file path. A provider/package change triggers re-evaluation before broad use.

The example is not the only valid implementation and does not establish a universal cryptographic mechanism.

---

46. Runtime Security and Adversarial Defense

Current GAISSF Domain D2 addresses **Runtime Security & Adversarial Defense**. Implementation should place security mechanisms in the path where untrusted instructions, multimodal content, tool requests or cross-context information can influence behavior.

Patterns may include:

- input/content inspection;
- instruction separation;
- external policy enforcement;
- capability restrictions;
- tool authorization;
- sandboxing;
- output controls;
- rate/abuse detection;
- adversarial testing;
- runtime monitoring.

Illustrative implementation pattern: a customer-facing assistant treats retrieved web content as untrusted data rather than authoritative instruction. The application enforces tool policy outside the model, so a malicious retrieved instruction cannot itself grant new tool authority.

Detection of an injection attempt is not proof that compromise or harmful action occurred.

---

47. Agentic AI and Autonomous Actions

Current GAISSF Domain D3 addresses **Agentic Risk & Autonomous System Security**. Agent implementation should be based on bounded authority, attributable identity and controllable action paths.

Use the decomposition:

sponsoring principal → agent identity → objective → model → memory/state → plan/orchestration → tool permissions → delegated authority → approvals → actions → resulting state → telemetry → containment

Implementation patterns may include:

- per-agent or per-service identity;
- least-agency/least-privilege design;
- tool and action allowlists;
- parameter constraints;
- transaction/value/destination limits;
- step/time limits;
- controlled delegation;
- memory isolation;
- pre-action human/policy approval for contextually consequential actions;
- circuit breakers, credential revocation and rollback.

Worked example — internal engineering agent: an agent may read approved repositories and draft pull requests but may not merge to protected branches or retrieve production secrets. The agent uses an attributable service identity, tool policy is enforced outside the model, and merge approval remains with an authorized human/workflow. Tool requests, authorization decisions and resulting repository changes are recorded. This example does not create a universal human-approval requirement for all agents.

---

48. RAG and Knowledge-Source Controls

RAG security should be implemented across the complete path:

source → ingestion → integrity/approval → index → retrieval authorization → context assembly → model → output → downstream use

Useful patterns include:

- approved/attributable sources;
- authorization before retrieval;
- tenant separation;

- source metadata and freshness;
- integrity/change review;
- instruction isolation;
- context minimization;
- output attribution/citation where appropriate;
- monitoring of source/index changes.

Worked example — policy assistant: employees can query internal policy documents, but retrieval filters by role and business unit before content enters the model context. Each answer retains source identifiers and policy version. A source update triggers controlled re-indexing. If a user receives an incorrect answer, source compromise, retrieval failure, stale content, model behavior and ordinary content error remain competing hypotheses until evidence supports a conclusion.

---

49. Supply Chain and External Model Providers

Current GAISSF Domain D4 addresses **Supply Chain & Third-Party AI Security**.

Provider implementation should make external trust boundaries explicit. Useful mechanisms include:

- approved-provider inventory;
- endpoint/service verification;
- data minimization before transmission;
- contractual change notification;
- provider incident/support path;
- model/service change detection;
- dependency/vulnerability management;
- fallback or exit;
- documentation of unavailable provider internals;
- separation of provider representation from enterprise observation.

Worked example — hosted model API: the enterprise records the approved endpoint and service tier, restricts data classes that may be sent, detects material response-format/behavior change, maintains a provider escalation contact and has a bounded fallback mode. The enterprise does not describe the provider's internal controls as independently verified unless it actually has a basis to do so.

---

50. Content / Output Integrity and Human Review

Current GAISSF Domain D5 addresses **Content Safety & Output Integrity**.

Implementation depends on intended use. Patterns may include:

- harmful-output controls;
- leakage prevention;
- user disclosure;
- structured output/schema validation;

- downstream decision checks;
- citation/grounding;
- abstention or escalation;
- human review where appropriate;
- feedback/complaint mechanisms.

Human review should be positioned where it can actually change the outcome. A “review” that happens after an irreversible action is not equivalent to pre-action oversight.

Do not equate incorrectness with maliciousness, legal harm or incident severity.

---

51. Governance and Human Oversight

Current GAISSF Domain D6 addresses **Governance, Accountability & Human Oversight**.

Implementation should connect governance statements to real decision rights. Useful arrangements include:

- named system/control owners;
- approved intended use;
- explicit authority for consequential decisions;
- exception and residual-risk governance;
- incident readiness;
- deprecation/retirement ownership;
- supplier governance;
- resilience ownership;
- human override/escalation capability.

Illustrative oversight test: ask whether the arrangement provides **awareness, comprehension, time, authority and effectiveness**. If the assigned human does not know intervention is required, cannot understand the relevant state, lacks time or authority, or cannot materially change the outcome, a named reviewer may be only nominal oversight.

---

52. Human and Societal Risk Interfaces

Current GAISSF Domain D7 addresses **Human & Societal Harms**.

Implementation can involve security, user, workforce and organizational controls around AI-enabled deception, social engineering, authentication, manipulation, harmful-use patterns and user recourse.

Useful patterns include:

- out-of-band verification for sensitive requests;
- staff/user awareness tied to realistic AI-enabled threats;
- escalation for suspected impersonation or manipulated media;
- complaint/recourse paths;
- monitoring of misuse patterns;
- integration with fraud, communications or trust-and-safety teams where relevant.

Training attendance alone does not establish that people can recognize or respond to the relevant condition. Use realistic exercises or performance evidence where appropriate.

---

53. Regulatory-Alignment Implementation Boundary

Current GAISSF Domain D8 addresses **Regulatory Alignment & Compliance**.

COM-017 does not determine whether a law or regulation applies. Implementation should instead connect the organization's controlled obligations register to the system, relevant controls, owners and operating records.

Useful practices include:

- identify applicable legal/regulatory/contractual obligations through competent internal or external authority;
- link obligations to system/control implementation only where the relationship is supportable;
- preserve requirements outside GAISSF scope;
- maintain jurisdiction/version information;
- update mappings when obligations or system use changes;
- avoid automatic equivalence claims.

A GAISSF control implementation may support an obligation without proving legal compliance.

---

54. Physical AI and the PAI-SF™ Handoff

Current GAISSF Domain D9 addresses **Physical AI Safety**. Under the reconciled GAISSF v1.0 profile treatment used by this guide, the seven D9 controls remain part of GAISSF and become additional mandatory controls whenever physical AI or cyber-physical actuation is within the applicable scope. D9 applicability must therefore be considered and justified; D9 is not treated as deleted.

ODA3's current physical-AI practitioner architecture then applies a second, distinct rule: **PAI-SF™ supersedes and expands the detailed physical-AI security-assurance layer represented by D9 while D9 remains retained for citation and framework continuity.** This does not create two independent assurance obligations merely because both frameworks are referenced. Preserve the D9-to-PAI-SF mapping and use PAI-SF for the deeper physical-AI implementation layer where it applies.

COM-017's job is to identify and operationalize that handoff, not reproduce PAI-SF's 12-domain / 41-control architecture.

Where the governed scope/applicability analysis identifies AI-enabled sensing, inference, autonomy or command that can create or materially influence physical consequence, initiate the physical-AI handoff. Specialist treatment should then address the complete perception-to-effect chain, including safe-state and degraded-mode behavior, independent override, actuator/sensor integrity, runtime telemetry, the physical operating environment and functional-safety interfaces as applicable.

Worked example — warehouse mobile robot: a remote security action that severs connectivity could remove operator visibility or change behavior while the robot is in motion. An appropriate implementation would therefore route material cyber-containment decisions through the relevant

safety/operational authority, provide locally enforceable safe behavior where the system design requires it, and define recovery/revalidation appropriate to the system. COM-017 does not authorize return to service.

Part V checkpoint: use only the AI-specific patterns that correspond to the real system. The implementation record should show which AI-specific surface required extension and which conventional enterprise control already carried the rest.

Part VI — Deploy and Operate

Use this part from release planning through steady-state operation. The practical focus is reversibility, ownership, monitoring, material change, provider dependency and controlled degraded/suspended/retired states.

55. Implementation Through Deployment

A control design is not operational until it is deployed into the environment that matters.

Deployment should preserve:

- approved scope;
- model/data/configuration identity;
- control placement;
- rollback/fallback;
- monitoring;
- incident readiness;
- known limitations;
- owner handoff;
- change traceability.

Use the organization’s release/change process. GAISSF does not require a separate deployment system.

---

56. Pilot and Limited-Exposure Patterns

Limited exposure can reduce the consequence of implementation defects while producing useful operational learning.

Possible constraints include:

- user population;
- traffic or transaction volume;
- regions;
- data classes;
- tools/actions;
- autonomy;
- features;
- physical operating conditions;

- enhanced support/monitoring.

These are patterns, not universal requirements. COM-017 does not prescribe a pilot duration, sample size, success threshold or fixed exposure limit.

A pilot should be designed to reveal relevant failure modes, not merely produce a low-risk demo environment unlike production.

---

57. Staged Release and Rollback

Useful staged-release patterns include canary deployment, blue-green release, feature flags, shadow operation and progressive enablement.

Choose a pattern based on reversibility and consequence. For example, shadow operation can be useful where the new system must be observed without allowing it to affect a real decision. It may be inappropriate if merely processing live data creates unacceptable privacy or safety exposure.

Rollback planning should consider:

- last known good release;
- model/data/schema compatibility;
- transactions already executed;
- supplier dependency;
- monitoring during rollback;
- evidence preservation;
- recovery validation.

Rollback can be impossible or incomplete for irreversible transactions or physical effects; design alternative containment accordingly.

---

58. General-Production Handoff

Before broad production operation, confirm that the organization has named owners for:

- system/service operation;
- models and data;
- control operation;
- monitoring;
- provider dependencies;
- incident escalation;
- resilience/recovery;
- open remediation and exceptions.

The operational team should know the approved baseline and the conditions under which capability should be restricted, escalated or changed.

A deployment checklist is a management aid, not authorization to deploy merely because all boxes are marked complete. Actual release authority remains with the organization and any applicable safety/legal processes.

59. Operational Ownership

After deployment, implementation becomes an operating responsibility.

Control owners should know:

- what normal operation looks like;
- which events indicate degradation;
- who responds;
- which dependencies are external;
- when the control must be retested or redesigned;
- how changes are approved;
- what records should be preserved;
- how exceptions are escalated;
- how retirement is handled.

Avoid handover by assumption. A system can reach production while no team accepts ownership of model change, provider monitoring or agent authorization. Make those handoffs explicit.

---

60. Monitoring and Observability

Design monitoring around the risks and actions the system can produce.

Relevant signal planes can include:

- security events;
- model behavior;
- data/retrieval state;
- agent plans/tool calls/actions;
- user complaints and overrides;
- supplier events;
- physical safety signals;
- control/evidence pipeline failures.

For each important signal define the source, owner, interpretation, response path and known blind spots.

Absence of alerts does not establish absence of compromise, harm or control failure. Monitoring itself can fail or be incomplete.

---

61. Material Change and Reimplementation

Material change can invalidate a previously sound implementation.

Review triggers may include:

- model/provider version;

- prompt or orchestration logic;
- new tool or connector;
- changed agent authority;
- new retrieval/data source;
- architecture/infrastructure change;
- user or jurisdiction expansion;
- new high-impact use;
- physical operating change;
- supplier incident/change;
- loss of observability;
- material threat or legal change.

A trigger does not automatically mean a formal reassessment is required. It means the organization should reconsider the affected implementation under its applicable governance and methodology.

---

62. Supplier Change and Dependency Failure

Provider and dependency changes should have defined operational responses.

Examples:

Supplier signal	Implementation response
Model/service change	Revisit behavioral baseline, compatibility and assumptions
Security incident	Assess exposure, activate provider/support path and preserve evidence
Availability degradation	Use fallback, controlled degradation or suspension as appropriate
Assurance/evidence expiry	Request updated information or increase independent observation
Subprocessor change	Revisit data, jurisdiction and dependency assumptions
Opaque failure	Record visibility limitation and bound operation based on available evidence

Do not assume provider culpability merely because behavior changed.

---

63. Degraded Operation, Suspension and Retirement

Organizations may need more than binary “on/off” operation.

Useful local states include:

- **degraded operation** — capability restricted with increased oversight;
- **suspended function** — affected feature disabled while preserving relevant state/evidence;
- **withdrawn capability** — access/integration removed;

- **retired system** — system decommissioned and residual obligations closed.

These are practitioner operational concepts, not GAISSF assessment states.

For physical AI, degraded/safe operation and return-to-service decisions remain subject to PAI-SF and specialist safety authority.

Part VI checkpoint: the operational owner should be able to identify the approved baseline, monitoring path, provider dependencies, material-change triggers, rollback/fallback options, degraded/suspended states and ownership for open exceptions or remediation.

Part VII — Design for Incident Readiness

Use this part before an incident to engineer identification, reconstruction, evidence preservation, containment, provider escalation and recovery into the system. For active response, move to COM-011, COM-012 and COM-013, together with AI-IRF™.

64. Why Incident Readiness Starts During Implementation

When an AI incident occurs, response teams can only use the identity, telemetry, versioning, containment and recovery capabilities the system was designed to provide.

Therefore implementation should make the system **incident-ready** without duplicating COM-011, COM-012 or COM-013, or AI-IRF™.

COM-017 does not classify the event. A signal, observation or event under triage remains within the organization's governed intake/triage process until the applicable UAIF™ classification is supportable; COM-017 does not introduce `Observation` or any other new UAIF `record_type`. Protective action may still begin while classification remains provisional.

The implementation question is:

> If this control or AI capability fails, will responders have enough visibility, authority and recovery capability to act proportionately without destroying important evidence or creating additional harm?

---

65. Detection and Attribution

Ensure potentially material behavior can become visible and can be associated with the affected components.

Depending on system context, responders may need to identify:

- system/service;
- session/user;
- agent identity;
- model/provider/version;
- prompt/configuration/policy version;
- tool/API;
- retrieval source/index;

- provider case/reference;
- physical component.

Do not log data simply because it might be useful. Balance attribution with privacy, legal and security requirements.

---

66. Reconstruction and Version Traceability

Incident reconstruction may depend on the ability to determine what changed and what action occurred.

Implementation may therefore need records of:

- relevant release identifiers;
- model/provider versions;
- prompt/policy/orchestration versions;
- tool requests and authorization decisions;
- retrieval/source identifiers;
- agent state transitions;
- resulting system changes;
- provider notices;
- physical command/telemetry sequence where applicable.

Record clock/correlation limitations where systems cannot be synchronized precisely.

---

67. Evidence-Preservation Capability

Preservation should be possible without assuming that every incident requires full forensic collection.

Design for:

- controlled log/record holds where appropriate;
- version snapshots or release manifests;
- integrity-preserving export;
- restricted access to sensitive content;
- provider evidence requests;
- documentation of evidence that cannot lawfully, safely or technically be retained.

Preservation does not necessarily mean copying full prompts, outputs or sensitive context into a general ticketing system. Controlled reference or protected collection may be safer.

Human safety and prevention of material physical harm take priority where preservation conflicts with immediate safety action.

---

68. Containment Surfaces

During implementation, identify what can actually be constrained if behavior becomes unsafe or compromised.

Potential containment surfaces include:

- user/session;
- identity/token;
- tool/API;
- connector/data source;
- agent;
- model/model route;
- workflow;
- feature flag;
- provider dependency;
- physical command path.

Containment should be as targeted as practical while respecting safety and operational constraints.

The fact that a capability can be disabled does not mean disabling it is always safe.

---

69. Provider Escalation and Dependency Containment

Provider incidents can create response dependencies outside enterprise control.

Predefine:

- support/escalation channels;
- account/service identifiers;
- required provider evidence or notices;
- version/change questions;
- fallback/substitution options;
- ability to restrict the dependent capability;
- decision authority if provider support is unavailable.

If the enterprise cannot identify the provider version, obtain evidence or constrain the dependency, record that as an implementation limitation before an incident occurs.

---

70. Rollback and Recovery Interfaces

Recovery is not merely restarting a service.

Implementation should define:

- known-good state;
- recovery dependencies;
- configuration/model/data consistency;

- authorization to restore;
- validation after restoration;
- residual uncertainty;
- restrictions that remain in place;
- provider/supplier involvement.

For digital systems, recovery may restore a model or configuration while retaining enhanced monitoring. For physical systems, return to operation may require separate physical inspection, safe-state verification or other specialist validation.

---

71. Physical-Safety Constraints on Containment

Default IT containment can be unsafe for cyber-physical systems.

Examples of potentially hazardous actions include:

- severing network links that carry safety-critical telemetry;
- power-cycling machinery with stored energy or uncontrolled momentum;
- disabling a controller without a safe fallback;
- forcing an agent/device offline while it is responsible for maintaining a stable physical state.

Physical containment should be governed by the appropriate safety/OT/clinical authority. COM-017 does not define return-to-service authority.

---

72. Learning Back into the Control Baseline

Incident and near-miss learning should feed implementation rather than remain only in the incident record.

Ask:

- Which control or dependency failed or was absent?
- Was the control design wrong, bypassed or poorly operated?
- Did a provider assumption fail?
- Was monitoring insufficient?
- Was authority unclear?
- Did the same weakness exist in other systems?
- What architecture, process, training or supplier change is required?

Preserve uncertainty where root cause remains unproven. Do not force a single cause merely to close the incident.

Part VII checkpoint: incident readiness is materially stronger when responders can identify the affected system/version, reconstruct consequential actions, preserve proportionate evidence, constrain the relevant authority path, reach providers, recover under defined authority and route physical containment through the applicable safety/operational process.

Part VIII — Remediation and Improvement

Use this part when implementation gaps, failed tests, incidents, provider changes or review findings need to become owned corrective work and verified improvement.

73. Build the Implementation Backlog

Implementation gaps, unknowns and improvements should be managed in the same delivery system used by the teams responsible for fixing them where practical.

A backlog item should include:

- affected requirement/control;
- system/scope;
- problem or implementation need;
- owner;
- risk/consequence rationale;
- dependencies;
- proposed action;
- verification method;
- change/release needs;
- generated artifacts;
- status.

Avoid creating a separate assurance backlog disconnected from engineering and operations.

---

74. Prioritize Without Inventing Universal Thresholds

COM-017 does not create a universal priority scale.

Organizations can prioritize using context such as:

- potential safety/security/privacy/rights consequence;
- exposure and blast radius;
- control dependency;
- recurrence;
- detectability;
- reversibility;
- provider concentration;
- implementation complexity;
- legal/contractual urgency;
- feasibility of interim containment.

Document the rationale. Do not prioritize solely by ease of closure, artifact count or visual dashboard status.

---

75. Correction, Containment and Corrective Action

Distinguish the purpose of the action:

- **Correction** — remove the immediate implementation defect.
- **Containment** — limit exposure or consequence while the issue remains active.
- **Corrective action** — address the supported cause so recurrence is less likely.
- **Preventive/systemic improvement** — apply the lesson to plausible related weaknesses or other systems.

A workaround may be useful containment without being a durable corrective action. Record the distinction.

---

76. Validate the Fix

Closure should be based on changed behavior, not merely on ticket completion.

Implementation-side validation can include:

- re-running the original failed condition;
- testing negative/edge paths;
- comparing current state with the approved baseline;
- checking that compensating controls are no longer required;
- confirming monitoring detects recurrence;
- verifying provider/dependency assumptions;
- checking side effects on related controls.

Where root cause remains uncertain, say so and define the residual monitoring or restriction required.

---

77. Review Related Controls and Systems

AI failures are often systemic because models, providers, data sources, identity paths and policy services are shared.

When a defect is found, ask whether it affects:

- other systems using the same model/provider;
- shared retrieval/data source;
- common agent/tool policy;
- shared authentication/authorization service;
- deployment pipeline;
- monitoring infrastructure;
- supplier contract;
- similar human-review workflow;
- physical systems with the same component.

Do not assume identical impact across all dependent systems; verify context separately.

78. Feed Incidents, Tests and Findings into Improvement

Improvement inputs include incidents, near misses, failed tests, user complaints, provider changes, operational degradation, internal review findings, external assessment findings and new threat intelligence.

Good improvement practice:

- preserves negative evidence;
- separates symptom from cause;
- updates policy, process, training and tooling when needed;
- validates the changed control;
- checks transferability to other systems;
- monitors recurrence.

Do not automate a broken control merely to increase maturity or efficiency.

Part VIII checkpoint: improvement work should have an owner, consequence rationale, action, dependencies, implementation-side verification and a clear link back to the affected system/control. Closure of a project ticket does not by itself prove control effectiveness.

Part IX — Implementation Outputs and Evidence Handoff

Use this part to preserve implementation records with enough context for later evidence preparation. COM-017 stops at candidate evidence; COM-015 begins the evidence-pack preparation layer.

79. What Implementation Naturally Produces

Examples of natural implementation artifacts include:

- scope and system-boundary records;
- inventories;
- architecture diagrams and decisions;
- control implementation records;
- RACI/ownership records;
- approved configurations;
- model/provider/version records;
- data-flow and trust-boundary records;
- agent/tool permission definitions;
- supplier records;
- deployment and rollback records;
- runbooks;
- change records;
- implementation-side test outputs;
- monitoring outputs;
- incident-readiness exercises;

- exceptions/risk decisions;
- corrective-action records.

The list is illustrative. A control may produce different records depending on architecture and operating model.

---

80. Candidate Evidence vs Defensible Evidence

A natural implementation artifact may later support an assurance claim, but only if its identity, scope, period, source, integrity, status, limitations and relationship to the requirement are sufficiently understood.

COM-017 therefore asks implementation teams to preserve context without deciding sufficiency.

Examples:

- a configuration export can show a setting but may not prove operation across time;
- a policy can show intended governance but may not prove technical enforcement;
- a provider attestation can describe provider scope but may not prove customer-side use;
- a test result can show one tested condition but not absence of rare failures outside the tested set.

COM-015 owns the evidence-preparation discipline.

---

81. Preserve Provenance Without Building the Evidence Pack Here

Implementation teams should avoid records that lose their source context.

Where practical, preserve:

- system/release identity;
- relevant version;
- date/time or period;
- responsible owner/operator;
- source system/process;
- link to the implementation decision or change;
- transformation/redaction note where relevant;
- known limitations.

A screenshot pasted into a presentation may be easy to produce but weakly attributable. Prefer source-system records or controlled exports when appropriate.

Do not recreate COM-015’s evidence index or assessor-facing package inside COM-017.

---

82. Handoff to COM-015

When implementation has produced a stable set of candidate artifacts, use **COM-015 — Building an AI Assurance Evidence Pack** to:

- establish evidence identifiers;
- organize evidence objectives;
- index scope, period, provenance and limitations;
- handle superseded records;
- surface gaps and contrary evidence;
- prepare a defensible handoff.

A complete implementation record is not automatically a complete evidence pack.

Do not force COM-017 implementation-responsibility labels into COM-015 evidence-source labels. The two answer different questions: **who operates or controls the mechanism** versus **who can provide or directly observe the evidence**. Preserve both views where provider/shared responsibility is material.

---

83. Boundary After Evidence Preparation

COM-015 owns the public evidence-preparation handoff. COM-017 does not extend the public practitioner journey into assessment participation.

Where an assessment is separately authorized, its engagement terms, controlled methodology and designated assessment authority govern what happens after evidence preparation.

COM-017 does not predict what the assessor will conclude, which evidence will be sufficient or whether certification is available or achievable.

Implementation outputs may be strong, weak, incomplete or contradicted by later evidence. That possibility is a feature of evidence-based assurance, not a defect in the implementation process.

If implementation changes after an evidence handoff or during later assessment activity, preserve the timing. Later remediation should not be rewritten as though it operated throughout an earlier assessed period. Material scope or architecture change should be surfaced and routed through the governed scope/applicability and evidence-handoff processes rather than silently absorbed into the record.

Part IX checkpoint: the handoff should contain candidate implementation artifacts with enough identity, scope, period/version, ownership and provenance context for COM-015 to prepare them properly. COM-017 does not decide evidence sufficiency.

Part X — Practitioner Tools

Select the tools that help the current task; do not treat all nine as a mandatory package. Each is an internal implementation-management aid, not an assessment workpaper or readiness gate.

84. Implementation Scope Worksheet

Best used when: implementation begins, scope materially changes, or different teams have conflicting views of the system boundary. **Working output:** a bounded implementation-scope record, not a formal applicability determination.

Field	Entry
System / service	
Business purpose / intended use	
Accountable owner	
Users / affected functions	
Models / providers	
Data / retrieval sources	
Agents / tools / external actions	
Infrastructure / environments	
Suppliers / shared services	
Geographies / jurisdictions	
Physical interfaces / consequences	
Applicable GAISSF source/profile reference	
Material assumptions	
Open scope questions	
Change triggers	

Use rule: this worksheet is an implementation-management artifact. It does not itself determine formal applicability or assessment scope.

---

85. AI System Decomposition Worksheet

Best used when: architecture, trust boundaries, delegated authority or common dependencies are not yet visible enough to design controls. **Working output:** a component-and-boundary view that can feed control translation and dependency analysis.

Component / boundary	Identity / authority	Data / state	Permitted actions	Dependencies	Failure mode	Monitoring	Owner	Recovery
User/ channel								
Application/ orchestration								
Model/ provider								
Retrieval/data								

Component / boundary	Identity / authority	Data / state	Permitted actions	Dependencies	Failure mode	Monitoring	Owner	Recovery
Agent/ memory								
Tool/ API/ action								
Platform / infrastructure								
Supplier boundary								
Physical -effect path								

Add/remove rows to match the real architecture.

---

86. Control Implementation Record

Best used when: an applicable requirement needs to become owned implementation work.
Working output: a traceable control-design/operation record; it is not an assessor workpaper.

Field	Entry
Requirement / control reference	
Scope element	
Implementation objective	
Accountable owner	
Operating owner	
Governance mechanism	
Technical mechanism	
Operational mechanism	
Human-factor dependency	
Shared-responsibility state	
Dependencies / preconditions	
Internal implementation state	
Implementation-side verification	
Monitoring / detection	
Change triggers	
Generated artifacts	
Assumptions / limitations	

Field	Entry
Related backlog / remediation	

Boundary: internal implementation state is not an ODA3 assessment state.

---

87. Shared Responsibility Matrix

Best used when: a provider, integrator, shared platform or enterprise service carries part of the mechanism or limits visibility. **Working output:** an explicit responsibility and observability boundary.

Control / objective	Enterprise mechanism	Provider mechanism	Shared dependency	Enterprise observation	Provider representation	Visibility limitation	Fallback / exit	Owner

Use the matrix to expose, not hide, provider opacity.

---

88. Implementation Backlog

Best used when: gaps, constraints, failed checks or improvements need delivery ownership. **Working output:** implementation work connected to the affected control and system rather than a separate assurance-only backlog.

Item ID	Requirement / control	Scope	Problem / need	Consequence rationale	Owner	Action	Dependencies	Verification	Status

Suggested internal status values may be simple project terms such as Planned, In Progress, Blocked, Implemented or Closed. They are not assessment results.

---

89. Dependency Register

Best used when: several controls or systems rely on the same identity service, provider, dataset, tool, infrastructure component, human authority or safety mechanism. **Working output:** a view of common-mode assumptions and failure consequences.

Dependency ID	Type	Supports	Owner / provider	Critical assumption	Failure consequence	Monitoring	Fallback / containment	Review trigger

Dependency types may include identity, model, provider, data, tool, infrastructure, human authority, legal process or physical safety mechanism.

---

90. Change-Trigger Register

Best used when: defining which changes should reopen implementation analysis. **Working output:** explicit triggers linked to owners, review/actions and implementation-side verification.

Trigger	Affected component / control	Why material	Implementation review / action	Owner	Related verification	Record / artifact
Model/ provider change						
Prompt/ orchestration change						
New tool / connector						
New data / retrieval source						
Changed autonomy / permissions						
Infrastructure change						
Supplier incident/change						
Monitoring loss						
New use / population / jurisdiction						
Physical operating change						

91. Incident-Readiness Implementation Check

Best used when: reviewing design before release, after a material architecture/provider change, or after an incident reveals an investigability/containment gap. **Working output:** a pre-incident capability check, not an incident-response procedure.

Use this as a design review, not an incident-response checklist.

- ☐ The affected system/model/provider/version can be identified where required.
- ☐ Agent/tool actions are attributable where consequential.
- ☐ Relevant configuration/change history can be reconstructed.
- ☐ Evidence/state can be preserved proportionately without uncontrolled duplication.
- ☐ Sensitive/privileged records have appropriate access handling.
- ☐ Identity/session/tool/model/workflow containment surfaces are known.
- ☐ Provider support/evidence routes are known.
- ☐ Rollback/fallback is available or the limitation is explicit.
- ☐ Monitoring blind spots are documented.
- ☐ Recovery authority and validation responsibilities are defined.
- ☐ Physical containment is routed to the relevant specialist safety/operational authority where applicable.
- ☐ Incident learning can feed control and architecture changes.

No item above establishes that an incident-response capability is effective under every scenario.

---

92. Implementation Review Checklist

Best used when: performing an internal implementation-quality review before operational handoff, material change review or evidence preparation. **Working output:** unresolved implementation questions and actions; checklist completion is not an ODA3 readiness gate.

- ☐ Scope and applicable source are explicit.
- ☐ System components and trust boundaries are documented.
- ☐ Models/providers/versions are identified to the extent possible.
- ☐ Data and retrieval dependencies are mapped.
- ☐ Agent/tool authority is explicit where applicable.
- ☐ Shared and supplier controls are visible.
- ☐ Physical-effect paths are identified where applicable.
- ☐ Each applicable control has an accountable owner.
- ☐ Implementation objectives are traceable to authoritative requirements.
- ☐ Governance, technical and operational mechanisms are distinguished.
- ☐ Human oversight is functional, not merely nominal, where relevant.
- ☐ Dependencies and failure modes are documented.
- ☐ Implementation-side verification is defined.
- ☐ Monitoring and material change triggers are defined.
- ☐ Deployment/rollback/recovery are considered.
- ☐ Incident-readiness prerequisites are considered.
- ☐ Provider representations are separated from enterprise observations.
- ☐ Limitations and unknowns remain visible.
- ☐ Candidate implementation artifacts are preserved with context.
- ☐ No implementation statement is converted into an unsupported conformity, certification, compliance or safety claim.

Part X checkpoint: use the completed tools to improve internal implementation visibility, not to declare a GAISSF score, readiness state, pass/fail result or certification status.

Part XI — Boundaries, Source Traceability and Next Reading

Use this part to check anti-patterns, claims, licensing, controlled-methodology boundaries and source traceability before publishing or relying on implementation statements.

93. Common Implementation Anti-Patterns

Anti-pattern	Why it fails	Better practice
Treating GAISSF as a checklist	Hides scope, ownership and operation	Connect requirements to system mechanisms and owners
Copying control text into policy	Shows intent, not functioning implementation	Add architecture, workflow, technical and operating mechanisms
Buying a security tool and declaring the control implemented	Tool may cover only one control surface	Verify scope, configuration, workflow, monitoring and ownership
GRC-only implementation	Work remains disconnected from engineering/operations	Link control records to actual system owners and backlogs
Provider attestation = customer assurance	Hides inherited and shared responsibilities	Separate provider representation from enterprise observation
Shared unrestricted agent credentials	Weak attribution and revocation	Use bounded service/agent identity where appropriate
Retrieval authorization after content enters context	Unauthorized data may already influence behavior	Enforce authorization before retrieval where architecture permits
Unversioned prompts/policies	Behavior cannot be reconstructed reliably	Use controlled configuration/versioning
Logging without review	Produces data, not a functioning detective control	Assign alert ownership and response path
Evidence collected only before review	Does not represent normal operation	Generate records through normal workflows
Permanent exceptions	Normalize unresolved control weakness	Use governed review and remediation
Rollback untested	Recovery may fail when needed	Test rollback/fallback proportionately
Monitoring added after deployment	Early failures are invisible	Design observability before release
Treating every change as harmless	Baselines drift silently	Apply material-change review
Treating every anomaly as an incident	Creates false conclusions	Preserve triage/classification discipline
IT containment applied blindly to physical AI	May create physical hazard	Use specialist safety authority and PAI-SF handoff

94. Claims Firewall

Public or internal implementation statements should remain factual and scoped.

Examples of scoped factual formulations that may fall within GEL v1.0's factual-alignment safe harbor when accurate, supportable and otherwise permitted include:

- "implementing GAISSF™ controls for [named system/scope]";
- "uses GAISSF™ as an internal AI safety and security control framework";
- "implements GAISSF control [reference] through [named mechanism]";
- factual "maps to" or "aligns with" statements where the basis is documented.

Do not infer from implementation alone:

- GAISSF compliance;
- GAISSF certification;
- ODA3 endorsement;
- regulator recognition;
- legal compliance;
- safety approval;
- zero risk;
- guaranteed control effectiveness;
- guaranteed future performance.

Claims should match the real scope and current implementation state.

---

95. GEL and Commercial-Delivery Boundary

COM-017 is public under GEL v1.0. Public availability does not create unrestricted commercial-delivery rights.

GEL v1.0 permits non-public implementation within an organization's own systems as Internal Use, subject to the licence terms. It distinguishes that from Commercial Use. Commercial Use includes, among other things, consulting, advisory or implementation services; audits, assessments or compliance services; certification/accreditation services; SaaS or software integration; paid training; managed services; redistribution in commercial products; derivative commercial frameworks; and client delivery or other monetized services.

An organization's Internal Use rights do **not** automatically authorize an external consultant, contractor, systems integrator or other service provider to use the licensed material commercially on the organization's behalf. Where the contractor's activity is Commercial Use, the applicable separate commercial licence/authorization must be held. Public redistribution rights likewise do not grant downstream Commercial Use rights.

Software, scripts, tools, APIs or automated systems that reproduce, operationalize or are substantially derived from substantive GAISSF Ecosystem material may also engage GEL derivative-work/commercial-use restrictions. Factual interoperability with separately identifiable licensed material is not the same thing as unrestricted embedding of GAISSF control logic.

Accordingly, this guide does not itself authorize third parties to provide GAISSF-based consulting, implementation services, managed services, assessments/audits, certification/accreditation, paid training, software/SaaS embedding, derivative commercial frameworks or client delivery. Such activity is governed by the applicable GEL terms and by any separately applicable written

commercial-delivery, service, partner or empanelment requirements in force for that activity. This guide does not itself establish that a particular partner or empanelment programme is active, available or granted to any party.

Factual interoperability or alignment statements do not imply endorsement, certification, partner status or official authorization.

---

96. Controlled-Methodology Firewall

COM-017 does not publish, reconstruct or invent:

- assessor scoring;
- evidence-confidence scales;
- control-rating scales;
- assessment result categories;
- public capability/maturity scoring used for assessment;
- aggregation formulas;
- mandatory-control confidence floors;
- sampling formulas or mandatory sample sizes;
- finding-severity rules;
- universal implementation deadlines or SLAs;
- universal minimum operating periods;
- certification thresholds or decision logic;
- certification eligibility rules;
- certificate validity periods;
- surveillance cycles;
- assessor judgement/override levels;
- proprietary assessor procedures;
- live assessment authority;
- current certification availability;
- regulator recognition.

Where older implementation sources contain such operational or assurance constructs, COM-017 uses only the public-safe implementation principle required for this guide unless later authoritative verification supports a different treatment.

For avoidance of doubt:

- the **Control Implementation Record** in Section 22 is not an assessor workpaper;
- an **implementation-side verification method** is not a public GAISSF test protocol, scoring input or assessor procedure;
- an internal **Implemented** state is not a control determination;
- a remediation or change-trigger record is not a finding-severity or closure decision;
- checklist completion does not create an ODA3 readiness designation, partner status, assessment authorization or certification gate.

---

97. Notably Absent

This guide deliberately does **not** establish:

1. that implementation equals conformity;
2. that deployment proves control effectiveness;
3. that policy alone implements technical or operational controls;
4. that purchasing an AI-security product establishes GAISSF implementation or conformance;
5. that GAISSF requires a specific vendor, product, cloud, GRC, SIEM, SOAR, gateway or observability platform;
6. that every organization must implement a control identically;
7. that any reference architecture is universally appropriate;
8. that automation is inherently superior to manual control;
9. that every control must be automated;
10. that every system requires universal continuous monitoring;
11. that continuous monitoring means continuous assessment;
12. that an implementation artifact is automatically sufficient evidence;
13. that evidence quantity establishes evidence quality;
14. that a provider certificate/attestation proves customer-side implementation;
15. that a provider representation is independently verified evidence;
16. that completion of COM-017 creates assessment readiness;
17. that completion of COM-017 creates certification eligibility;
18. that a checklist establishes operational assurance;
19. that a control implementation establishes legal or regulatory compliance;
20. that implementation eliminates residual AI risk;
21. a required rollout duration, pilot duration or minimum operating period;
22. a mandatory universal implementation sequence;
23. universal prompt logging, output logging or indefinite telemetry retention;
24. that every anomaly is an incident;
25. a new UAIF™ taxonomy or record type;
26. a replacement for AI-IRF™ response procedures;
27. that GAISSF D9 alone satisfies PAI-SF™ where PAI-SF applies;
28. that GAISSF or PAI-SF replaces functional safety;
29. that IT-style containment is always safe for cyber-physical systems;
30. deployment authorization or physical return-to-service authority;
31. client outcome data, proprietary deployment history or independently measured implementation-effectiveness statistics that ODA3 Institute does not possess;
32. universal validation across all industries, jurisdictions or deployment environments;
33. that COM-009's 30-day planning structure is a required rollout duration or readiness gate;
34. that similarity to a COM-010 worked scenario establishes applicability, profile choice or control sufficiency;
35. that a local signal, observation or triage state is a UAIF™ `Observation` record type or confirmed Incident;
36. that COM-017's incident-readiness design is a replacement for COM-011/013 response guidance or AI-IRF™;
37. that an implementation-responsibility descriptor is equivalent to a COM-015 evidence-source state or an assessment result;
38. that later remediation can be presented as though it operated throughout an earlier evidence or assessment period;

39. an official GAISSF implementation-completion score, implementation maturity level or readiness designation;
40. a mandatory COM-017 implementation record format, minimum implementation package or universal implementation checklist;
41. that an internal status of **Implemented** establishes operating effectiveness, evidence sufficiency or conformity;
42. that implementation-side testing or verification reproduces a controlled assessor procedure;
43. that use of this public guide grants ODA3 partner, empanelment, assessor, auditor, certification-body or commercial-delivery status.

These are boundaries, not omissions to be filled by implication.

---

98. Source and Traceability Register

This register identifies the principal ODA3 sources used to control terminology, architecture and document boundaries in COM-017. It is a traceability aid, not a substitute for the underlying authoritative sources.

98.1 Authoritative GAISSF Sources

Source	Role in COM-017	Publication/source treatment
GAISSF-NOR-001 — Framework Standard	Primary normative framework authority	**Final Publication Edition.** Establishes framework-level interpretation and the authoritative 59-control / nine-domain baseline.
GAISSF-NOR-004 — Control Catalogue	Normative control library	**Final Publication Edition.** Provides the canonical control library and domain/control architecture; framework-level inconsistency resolves to NOR-001.
GAISSF-NOR-008 — GAISSF v1.0 Change Log	Change, supersession and reconciliation provenance	**Final Publication Edition.** Used for historical traceability, supersession and the final Foundational/D9 reconciliation notice.
GAISSF Ecosystem License (GEL) v1.0	Licensing and use boundary	Governs Internal Use, Commercial Use, attribution, factual-alignment claims, trademark restrictions and reserved certification/audit authority.

Historical GAISSF files may preserve the earlier expansion **Global AI Security & Safety Framework** and/or earlier mark conventions. COM-017 uses the current market-facing convention **Global AI Safety and Security Framework / GAISSF™** without rewriting historical source titles.

98.2 Verified GAISSF Domain Architecture

Domain	Current public title	Controls
D1	Model Integrity &	9

Domain	Current public title	Controls
	Adversarial Robustness	
D2	Runtime Security & Adversarial Defense	6
D3	Agentic Risk & Autonomous System Security	7
D4	Supply Chain & Third-Party AI Security	7
D5	Content Safety & Output Integrity	6
D6	Governance, Accountability & Human Oversight	7
D7	Human & Societal Harms	5
D8	Regulatory Alignment & Compliance	5
D9	Physical AI Safety	7

The total is **59 controls across nine domains**.

For the profile treatment used by COM-017, the final reconciliation notice in GAISSF-NOR-008 controls: **52 canonical controls in D1–D8 form the Foundational baseline; the seven D9 controls are additional mandatory controls whenever physical AI or cyber-physical actuation is within the assessed scope.**

98.3 Physical-AI Source Boundary

The principal physical-AI sources used by COM-017 are:

- **ODA3-2026-07-CSX-PAISF-004 — GAISSF Domain 9 to PAI-SF Control Crosswalk** — controlled mapping dependency used for D9-to-PAI-SF provenance; it is not treated as a public safety certification or regulatory approval.
- **ODA3-2026-07-NOR-PAISF-005 — Physical AI Security Framework Standard** — framework-level PAI-SF source defining the physical-AI security-assurance scope and the 12-domain / 41-control architecture.
- **ODA3-2026-08-WHP-COM-014 — Securing Physical AI: A Practitioner’s Guide to PAI-SF™** — current public practitioner boundary used by COM-017.

COM-017 follows the public practitioner position that PAI-SF™ supersedes and expands the detailed physical-AI security-assurance layer represented by GAISSF D9 while D9 remains retained for continuity. PAI-SF does not replace functional safety, product safety, sector engineering, regulator authorization or physical return-to-service authority.

98.4 GAISSF Implementation Source Family

COM-017 synthesizes, rather than republishes, the established implementation family:

- **GAISSF-IMP-009 — Adoption Guide** — adoption governance, ownership and anti-checklist discipline.
- **GAISSF-IMP-010 — Implementation Handbook** — implementation records, architecture/workflow integration and implementation anti-patterns.

- **GAISSF-IMP-011 — Deployment Playbook** — staged rollout, rollback, production handoff and material-change treatment.
- **GAISSF-IMP-012 — Migration Guide** — existing-control reuse, shared/supplier controls and anti-duplication.
- **GAISSF-IMP-013 — Reference Architecture** — trust boundaries, layered architecture, RAG/agent/provider/edge patterns, observability and recovery.
- **GAISSF-IMP-014 — Operational Handbook** — run-state ownership, monitoring, supplier oversight, change, corrective action and retirement.
- **GAISSF-IMP-015 — FAQ** — public implementation interpretation and boundary clarification.
- **GAISSF-IMP-016 — Troubleshooting Guide** — diagnosis, containment, root-cause and remediation patterns.
- **GAISSF-IMP-017 — Evidence Collection Guide** — implementation-to-evidence handoff.
- **GAISSF-IMP-018 — Continuous Improvement Guide** — corrective action, root-cause and cross-system learning.

Where preserved implementation sources contain older profile shorthand, domain summaries, fixed cadences, severity/health/maturity scales, VTS criteria or certification-program constructs, those items are not treated as current public COM-017 authority unless they remain consistent with the controlling current source state.

98.5 WHP Practitioner Publications

Document	Current role in COM-017	Boundary preserved
ODA3-2026-08-WHP-COM-009 — The First 30 Days with GAISSF™	Introductory implementation planning	Thirty days remains illustrative pacing, not an implementation or readiness requirement.
ODA3-2026-08-WHP-COM-010 — Applying GAISSF™: Worked Scenarios	Hypothetical application reasoning	Scenario similarity does not determine real-system applicability, evidence sufficiency or outcome.
ODA3-2026-08-WHP-COM-011 — When AI Breaks: The ODA3 Guide to AI Incident Response	Short incident-response orientation	COM-017 uses incident-readiness prerequisites only; it does not reproduce the active-response process.
ODA3-2026-08-WHP-COM-012 — Classifying AI Incidents: A Practical Guide to UAIF™	UAIF™ classification and uncertainty discipline	COM-017 does not create UAIF record types, severity logic or legal-reportability determinations.
ODA3-2026-08-WHP-COM-013 — AI Incident Response Field Guide: Using UAIF™ and AI-IRF™ for Real-World AI Incidents	UAIF™ / AI-IRF™ operational integration	COM-017 designs prerequisites for response; it does not convert response checkpoints, observability tiers or timed windows into GAISSF requirements.
ODA3-2026-08-WHP-COM-014 — Securing Physical AI: A Practitioner's Guide to PAI-SF™	Physical-AI practitioner implementation and safety interface	COM-017 performs the D9/PAI-SF handoff and leaves detailed physical-AI treatment, safety engineering and return-to-service authority outside.
ODA3-2026-08-WHP-COM-015 — Building an AI Assurance Evidence Pack	Evidence preparation	COM-017 stops at candidate implementation artifacts; COM-015 owns evidence

Document	Current role in COM-017	Boundary preserved
		indexing, provenance, limitations, contradictions and handoff preparation.
Controlled assessment-methodology family	Separately governed assessment activity	COM-017 does not publish assessment-participation procedure, assessor procedure, scoring, confidence, sampling, findings or certification decision logic. Public evidence preparation ends with COM-015.

The WHP publication state of a practitioner guide does not itself change the maturity, authorization or certification state of an underlying framework or controlled methodology.

98.6 Naming, Marks and Licensing

Current public practitioner usage in COM-017 is:

- **GAISSF™ — Global AI Safety and Security Framework**
- **UAIF™ — Unified AI Incident Framework**
- **AI-IRF™ — AI Incident Response Framework**
- **PAI-SF™ — Physical AI Security Framework**
- **GAISSF Ecosystem** — umbrella term for the integrated ODA3 framework ecosystem

Historical source titles remain identifiable as historical source titles and are not silently rewritten.

Under GEL v1.0, factual alignment statements may be made where accurate and otherwise permitted, but they do not themselves imply ODA3 certification, endorsement, audit, approval or official status. Public availability and Internal Use do not create unrestricted third-party Commercial Use rights. Commercial consulting, implementation, assessment, audit, certification, paid training, managed services, software/SaaS embedding and other defined Commercial Use remain governed by GEL v1.0 and any separately applicable written authorization.

---

99. Next Reading

Recommended practitioner sequence after this guide:

1. **Building an AI Assurance Evidence Pack — ODA3-2026-08-WHP-COM-015** — prepare implementation records into a versioned, defensible evidence handoff.
2. **When AI Breaks: The ODA3 Guide to AI Incident Response — ODA3-2026-08-WHP-COM-011** — short operational orientation for potentially material AI events.
3. **Classifying AI Incidents: A Practical Guide to UAIF™ — ODA3-2026-08-WHP-COM-012** — incident classification and uncertainty discipline.
4. **AI Incident Response Field Guide: Using UAIF™ and AI-IRF™ for Real-World AI Incidents — ODA3-2026-08-WHP-COM-013** — deeper classification/response integration.
5. **Securing Physical AI: A Practitioner's Guide to PAI-SF™ — ODA3-2026-08-WHP-COM-014** — detailed physical-AI implementation where physical/cyber-physical scope applies.
6. **The First 30 Days with GAISSF™ — ODA3-2026-08-WHP-COM-009** — introductory implementation planning; the 30-day structure is illustrative.

7. **Applying GAISSF™: Worked Scenarios — ODA3-2026-08-WHP-COM-010** — hypothetical application reasoning; scenario similarity does not determine applicability.

Assessment participation is not part of the public Next Reading path. Where assessment activity is separately authorized, the applicable engagement terms and controlled sources govern.

For exact framework requirements, control statements, versions and controlled scheme state, consult the current authoritative GAISSF Ecosystem sources rather than relying on a WHP guide alone.

---

100. Publication Metadata, GEL Attribution and Final Publication Statement

Publication Metadata

Document ID: ODA3-2026-08-WHP-COM-017
Title: *Implementing GAISSF™*
Subtitle: *A Practitioner's Guide to Operationalizing GAISSF Controls*
Document Type: WHP — Adoption / Practitioner Field Guide
Framework Context: GAISSF Ecosystem
Classification: Public — GEL v1.0
Publisher: ODA3 Institute
Copyright: © 2026 ODA3 Pvt Ltd
Publication date: 13 August 2026
Correction date: 18 August 2026
Publication state: CORRECTED FINAL R10

Suggested citation

ODA3 Institute (2026), *Implementing GAISSF™: A Practitioner's Guide to Operationalizing GAISSF Controls*, ODA3-2026-08-WHP-COM-017.

No DOI, ISBN, accreditation number, certification number or regulator-recognition identifier is assigned by this guide.

GEL Attribution and Use Boundary

GAISSF™, UAIF™, AI-IRF™, and PAI-SF™ are frameworks of ODA3 Institute and are referenced under the GAISSF Ecosystem License (GEL) v1.0.

This attribution does not itself authorize unrestricted commercial delivery, assessment, audit, certification, endorsement, paid training, software embedding or client delivery. Applicable GEL terms and any separately applicable ODA3 authorization / partner rules govern those uses.

Final Publication Statement

This document is the **CORRECTED FINAL R10** edition of **ODA3-2026-08-WHP-COM-017 — Implementing GAISSF™**.

R10 removes public routing to the withdrawn assessment-participation publication and clarifies that the public practitioner journey ends at COM-015 evidence preparation. It does not change GAISSF requirements or add assessment methodology.

FINAL status applies to this practitioner guide only. It does not create or alter framework maturity, live assessment authority, certification availability, assessor authorization, partner or empanelment status, regulator recognition, product/sector approval, physical return-to-service authority or commercial-delivery rights.