GAMUT AI

Administrator & Workspace Operations Manual

A complete customer guide to secure tenant administration, workspace governance, access control, subscriptions, integrations and agentic runtime operations.

Edition 2026-07-22
Comprehensive administrator edition

About this manual

This manual is generated from Gamut's maintained public guidance after verification against current product behaviour. It is written for tenant administrators, workspace owners, security administrators, authorised integration owners and people accountable for operating Gamut safely.

It explains the administrator's responsibilities and the customer-visible workflows needed to configure, govern and review the service. The live tenant remains authoritative for the plan entitlements, permissions and controls available to a particular organisation and user.

> Gamut applies role, workspace, tenant and subscription boundaries together. An interface option is not an authority grant: sensitive actions remain subject to server-side permission, entitlement, scope and step-up checks.

Administrator operating principles

Principle	Administrator expectation
Named accountability	Use individual accounts and keep owners, approvers and reviewers identifiable.
Least privilege	Grant only the tenant role and workspace access required for current duties.
Separation of duties	Use the dedicated Runtime Policy Approver role for independent approval instead of assigning broad administration rights.
Tenant and workspace isolation	Confirm the selected tenant and workspace before granting access, changing settings or reviewing records.
Entitlement integrity	Treat the tenant subscription as a hard product boundary; roles do not unlock capabilities outside the plan.
Strong authentication	Use MFA and complete step-up verification for sensitive administrative and runtime actions.
Auditability	Retain the request, decision, owner, scope, evidence and review point for privileged changes.

Authoritative tenant roles

Role	Purpose and boundary
Administrator	Tenant-scoped administration, user management and the customer admin console.
Runtime Policy Approver	Approval-only review and publication of runtime access policies in assigned workspaces; no drafting or general administration.
Advanced	Assessment and advanced product capabilities within the tenant plan and assigned workspaces.
Standard	Core assessment capabilities within the tenant plan and assigned workspaces.
External Auditor	Read-only assurance access with a restrictive product ceiling; no writes, exports or administration.
Subscriber	Account access without product-operating authority.

> Role labels describe the upper permission boundary. Effective access is the intersection of role, tenant, workspace assignment, subscription, record scope and any required security gate.

Contents

Part 1: Administrator orientation and secure launch

Establish the administrator's operating model, create the first governed workspace and prepare users for controlled work.

1. What is Gamut AI

Gamut AI is an AI governance lifecycle platform that helps organisations discover AI, classify risk, assess against multiple frameworks, and produce audit-ready evidence.

Gamut AI is an **AI governance lifecycle platform**. It gives organisations one place to discover the AI they use, understand and classify the risk it carries, assess it against the frameworks that matter, and produce the evidence that boards, buyers, auditors and regulators ask for.

It is built for organisations that need to demonstrate structured governance over their AI systems, banks, financial institutions, regulated enterprises and any organisation whose AI adoption has outgrown spreadsheets and fragmented point tools.

The problem Gamut solves

AI adoption usually runs ahead of governance. Teams buy AI tools, embed models in products, and build agentic workflows faster than risk, compliance and security functions can track them. The result is familiar:

- No reliable inventory of where AI is used, who owns it, or what data it touches.
- Risk decisions made informally and inconsistently, with no traceable rationale.
- Policy that exists on paper but cannot be evidenced when someone asks for proof.
- Evidence scattered across documents, email threads and screenshots, gathered in a panic at audit time.

Gamut replaces that with a single, governed operating layer that takes an AI system from first discovery through to board-level reporting and continuous improvement.

What Gamut gives you

- **A standing AI inventory**: a register of AI systems, use cases, owners, suppliers, data context and lifecycle status.
- **Consistent risk classification**: deterministic risk tiering driven by a configurable governance weighting profile of purpose, impact, human oversight, data exposure and operating context.
- **Multi-framework assessment**: assess a system against GTSAF, the EU AI Act, NIST AI RMF, ISO/IEC 42001 and 42005, NAGF, ACRS, the Agentic Trust Framework (ATF) and MAESTRO.
- **Evidence as a first-class object**: evidence requests, artefacts, quality tracking, findings and remediation, captured as governance work happens.
- **Audit-ready outputs**: workpaper-grade exports and board reporting that trace from requirement to evidence to finding.
- **AI assistance, governed**: an AI Consultant grounded in your own assessment records, plus policy drafting, all proxied server-side and metered by plan.
- **Governed agentic AI**: Agentic CISO, Gamut Gateway and Gamut Claw extend governance to AI that takes action, not just AI that answers questions.

Design principles

Gamut is built around a small number of deliberate choices:

- **Evidence, not just policy.** Governance is only credible when it can be proven. Gamut is organised around producing defensible evidence, not generating documents.
- **Traceability end to end.** Every record connects: AI system -> risk decision -> control expectation -> evidence -> finding -> remediation.
- **Governance is a lifecycle, not an event.** Gamut keeps governance current as systems and obligations change, rather than producing a point-in-time snapshot.

- **Trust by construction.** AI analysis is proxied server-side so model keys are never exposed to the browser, tenants are isolated, and access is controlled by role.

How this documentation is organised

- **Getting started**: the fastest path from sign-in to your first assessed AI system.
- **Platform**: the modules that make up Gamut and how they work.
- **Governance frameworks**: the frameworks Gamut supports and how routing works.
- **Agentic stack**: Agentic CISO, Gateway and Claw.
- **Administration**: workspaces, users, SSO, plans and audit.
- **API reference**: integrate Gamut with the rest of your stack.

> **New to AI governance?:** Start with The governance lifecycle for the mental model, then Core concepts & glossary for the vocabulary used throughout these docs.

2. Who Gamut is for

Gamut serves organisations running AI governance and the auditors who review it, across risk, compliance, security, procurement and the board.

Gamut is built for two audiences that have to trust the same records: the **organisation running AI governance** and the **auditor reviewing it**. The platform is designed so that both work from one source of truth.

For organisations running AI governance

Gamut gives AI governance, risk, compliance, security, procurement and business teams a shared operating layer for responsible AI adoption.

- One place to see AI systems, owners, risks and evidence.
- A practical route from AI literacy to board reporting.
- A governance lifecycle that supports EU AI Act readiness and wider assurance needs.
- Clear outputs for leadership, risk, compliance, security and procurement teams.

Roles that use Gamut

Role	What they get from Gamut
Chief Risk / Compliance Officer	A defensible view of AI risk across the organisation and board-ready reporting.
AI governance / responsible AI lead	A repeatable operating model across the lifecycle and across frameworks.
CISO / security	Control expectations, evidence and the agentic stack for AI that takes action.
Procurement / vendor risk	Structured assessment of vendor and third-party AI before and during use.
System / model owners	A clear record of their AI system, its risk tier and the controls expected of it.

For auditors reviewing AI governance

Gamut helps auditors and reviewers assess governance activity through structured records, evidence quality, findings and remediation history.

- Structured evidence instead of scattered files and informal explanations.
- Traceability from AI system record to risk decision, control expectation, evidence and finding.
- Consistent review packs for internal audit, client assurance and external review.
- Clearer follow-up on findings, exceptions, remediation and residual risk.

Where Gamut fits

Gamut is most valuable when AI adoption has outgrown informal tracking but governance still has to satisfy external scrutiny, buyers running vendor due diligence, boards asking for assurance, internal audit testing controls, and regulators assessing readiness against the EU AI Act and equivalent regimes.

> **Tip:** If you are evaluating Gamut for a specific obligation, for example EU AI Act readiness, start with the relevant framework page, which explains what Gamut captures and produces for that regime.

3. Core concepts & glossary

The vocabulary used throughout Gamut, AI systems, intake, risk tiers, frameworks, controls, evidence, findings, model cards and the agentic stack.

This page defines the core objects and terms used across Gamut and throughout this documentation. They are deliberately consistent: the same words mean the same thing in the product, the API and these docs.

Core objects

AI system : A registered unit of AI use, an internal application, an embedded model, a vendor AI tool or an agentic workflow. The AI system is the anchor that every other record connects back to. See Registry & Discovery.

Use case : The business purpose an AI system serves. One AI system can support multiple use cases, each with its own context and impact.

Intake : The structured capture of context for an AI system or use case, purpose, users, data, oversight, suppliers and geography, used to determine the level of governance required. See Intake & risk tiering.

Risk tier : The classification of an AI system's risk, derived from intake. Risk tiers drive which controls and frameworks apply.

Framework : A structured set of controls or requirements, for example GTSAF, the EU AI Act, NIST AI RMF or ISO/IEC 42001. See Frameworks overview.

Domain : A grouping of related controls within a framework. GTSAF, for example, organises its controls across 17 domains.

Control : A specific governance expectation that can be assessed and evidenced. Controls belong to domains within a framework.

Assessment : The act of scoring an AI system against a framework's controls, recording the rationale and current state. See Assessments & control testing.

Control test : Evidence that a control is operating effectively, beyond simply being designed. Control testing produces the proof that backs an assessment.

Evidence : An artefact that supports a governance claim, a document, export, screenshot, log or record. Evidence is requested, captured, quality-checked and linked to controls and findings. See Evidence & findings.

Finding : A recorded deficiency, gap or exception identified during assessment or audit, tracked through to remediation and closure.

Remediation : The work that closes a finding, with its progress visible to the right stakeholders.

Model card : A document of a model's technical and ethical characteristics, intended use, data, limitations, performance and oversight. See Model cards.

Policy : An AI governance policy, which Gamut can help draft with AI assistance. See Policy generation.

The agentic stack

Agentic AI : AI that takes action, calling tools, invoking APIs and running workflows, rather than only producing text. Agentic AI needs runtime governance, not just design-time assessment.

Agentic CISO : Gamut's capability for governing agentic AI, the agent register, ATF assessment, tool and data governance, approvals and runtime evidence. See Agentic CISO.

Gamut Gateway : The policy decision and enforcement layer that applies governance policy to agent actions at runtime. See Gamut Gateway.

Gamut Claw : The secure agent execution layer that requests and runs work only through Gateway-controlled paths. See Gamut Claw.

ATF (Agentic Trust Framework) : The framework that defines trust, control and evidence expectations for agentic AI, which Gateway and Claw implement at runtime. See ATF.

ATF level : An agent's autonomy ceiling, L1 Intern, L2 Junior, L3 Senior or L4 Principal. Gateway enforces a different action boundary at each level, from read-only at L1 to strategic autonomy at L4.

ACRS (Agentic Capability Risk Score) : A score of an agent's capability risk across dependency, action, access and harm dimensions, producing a band that sets how tightly to govern. See ACRS.

Connector : A governed adapter through which Gateway performs a tool, model or data call. The connector holds the credential and endpoint policy; the agent never does. See Connector catalog.

Tool permission : The business authorisation in Agentic CISO that an agent may use a given tool. It must align with a registered connector before Gateway allows the call.

Approval gate : A configured requirement that a named human approves a sensitive or mutating action before Gateway will allow it.

Gateway decision : The verdict Gateway returns for a requested action, allow, require approval, degrade or block, with a full decision path and a signed, short-lived authorisation token on allow.

Claw task : A governed unit of agent work of a defined type (for example governed reasoning, evidence gap analysis, incident triage), run under a lease with bounded steps and redacted output. See Gamut Claw.

Trajectory event : A journaled record of what an agent did at runtime, hash-chained and fed back to Agentic CISO as runtime evidence.

Platform terms

Workspace / tenant : An isolated environment for one organisation. Each workspace keeps its own users, systems, assessments and evidence separate from every other. See Workspaces & tenancy.

Role : The set of permissions a user holds within a workspace, controlling what they can see and do. See Users & roles.

Entitlement / plan : The feature tier available to a workspace, which gates frameworks and capabilities. See Plans & entitlements.

Audit log : The immutable record of state-changing actions in a workspace, used for accountability and review. See Audit log.

4. Quickstart

Get from sign-in to your first assessed AI system in Gamut, with the shortest path through the governance lifecycle.

This quickstart takes you from signing in to your first assessed AI system. It is the fastest useful path through Gamut; later pages go deeper on each step.

> **Access:** Gamut is available at **run.gamutassure.com**. If you do not yet have a workspace, ask your administrator to invite you, or start a trial from the sign-in page.

Before you start

You will move faster if you have one real AI system in mind to register, an internal tool, an embedded model or a vendor AI product. Have a rough idea of:

- What it does and who uses it.
- What data it touches.
- Who owns it.
- Which obligation you care about most (for example EU AI Act readiness).

Step 1: Sign in to your workspace

Sign in at run.gamutassure.com. You land in your **workspace**, an environment isolated to your organisation. Everything you create stays within it. If you belong to more than one workspace, confirm you are in the right one.

Step 2: Register an AI system

Open **AI System Records** and add your AI system. Give it a name, owner and a short description of what it does. This creates the anchor record that everything else connects to.

-> Full walkthrough: Register your first AI system.

Step 3: Complete intake and set a risk tier

Run **AI Use Case Intake & Approval** for the system: capture purpose, users, data exposure, human oversight, supplier involvement and geography. Gamut uses this, through the **Risk Tiering Engine** and **ACRS**, to set a **risk tier** and route the system to the controls and frameworks that apply.

-> More detail: Intake & risk tiering.

Step 4: Run an assessment

Open the framework routing suggested for the system, for example GTSAF or EU AI Act readiness, and score its controls on each framework's answer scale, recording your rationale as you go.

-> Full walkthrough: Run your first assessment.

Step 5: Capture evidence and findings

As you assess, use the **Evidence Tracker** to raise evidence requests and attach artefacts, and the **Findings Register** to record any gaps. Evidence and findings link back to the controls they relate to, so your assessment is defensible rather than just complete.

-> More detail: Evidence & findings.

Step 6: Produce a report

Open the **Board Dashboard** or generate a **Workpaper Pack** to see your AI inventory, risk picture, evidence quality and open findings in one place, suitable for leadership and review.

-> More detail: Reporting & exports.

Next steps

5. Register your first AI system

Walk through capturing an AI system in AI System Records, the anchor record that intake, assessment, evidence and every finding connects back to.

The **AI system** is the anchor of everything in Gamut. It is registered in **AI System Records**, and registering one well makes every later step, intake, risk tiering, assessment, evidence and reporting, faster and more consistent. Nothing else can attach to a system until the record exists: the registry is what intake and assessment feed from.

Work inside a workspace

Everything you create belongs to a **workspace** (also called an assessment), the container that scopes your systems, intake, evidence and reports. Make sure you have created or opened a workspace before you begin; AI System Records lists the systems within the workspace you are in.

What counts as an AI system

Register anything that uses AI and carries governance relevance:

- Internal applications and services that embed a model.
- Vendor or third-party AI tools your teams use.
- GenAI assistants and copilots in active use.
- Agentic workflows that take action through tools and APIs.

If in doubt, register it. An under-governed AI system is a bigger risk than an extra record. If a system was surfaced by the AI Discovery Inbox, you can promote it into AI System Records rather than entering it from scratch.

Fields to capture

A system record is structured. You do not need every field on day one, but the more you capture, the better intake and routing work later:

Group	Fields
Identity	Name, system ID, version, lifecycle stage (development through retired).
Ownership	Owner and owner email, department, responsible-AI contact.
Nature	AI technique, model type, autonomy level, human-oversight model, vendor, deployment type and environment.
Data & exposure	Data classification, personal-data involvement, data sources, users, affected persons, geographies, regulatory exposure.
Governance	Risk tier, ACRS score and review dates (these are set or confirmed later, in intake and risk tiering).

Steps

1. Open **AI System Records** in your workspace.
2. Add a new system record (or promote one from the AI Discovery Inbox).
3. Enter the **identity** and **ownership** fields: name, owner, owner email, department.
4. Describe the **nature** of the system: what technique and model type it uses, its autonomy level and how humans stay in oversight.
5. Record the **data and exposure** context: data classification, personal data, who is affected, and where it operates.
6. Save. The system now appears in your inventory and is ready for intake.

Good practice

- **One owner, always.** Ownership is what makes governance actionable.
- **Describe purpose, not implementation.** Reviewers care about what the system does and why.
- **Register early.** Capturing a system at proposal stage is far cheaper than reconstructing its history later.
- **Let discovery feed the registry.** A registry that misses systems produces governance that only looks complete.

Next

With the system registered, complete AI Use Case Intake and set a risk tier, then run your first assessment.

6. Run your first assessment

Ground a system through intake and risk tiering, route it to the right frameworks, then score its controls with rationale, evidence and findings.

An **assessment** scores a registered AI system against the controls of a framework and records why. This is where governance becomes defensible: not just a decision, but a decision with a documented rationale and evidence behind it. In Gamut, a good assessment is the end of a short chain that grounds the system first.

Ground the system before you score it

A framework score is only as good as the context behind it. Before assessing, work through the chain that feeds it:

1. **AI System Records** establishes what the system is.
2. **AI Use Case Intake & Approval** explains what it does, who it affects, what data and decisions are involved, and records an accountable approval. See Intake & risk tiering.
3. **Risk Tiering Engine** and **ACRS Risk Assessment** set the depth of governance expected, confirming the risk tier and capability band.
4. **Assessment Plan & Assurance Routing** routes the system to the frameworks it actually needs.

By the time you open a framework, the system is grounded and routed, so you are scoring against the right controls rather than guessing which apply.

Choosing a framework

Routing will suggest the frameworks that fit. Common starting points:

- **EU AI Act readiness**: to demonstrate readiness for the EU AI Act.
- **GTSAF**: for depth, 358 controls across 17 domains.
- **NIST AI RMF** or **ISO/IEC 42001**: if you align to those standards.
- **ATF**: for systems that take action as agents.

You can assess the same system against more than one framework; Gamut keeps each distinct while sharing the underlying system, evidence and findings. See Frameworks overview.

Scoring controls

Open the framework and work through its controls, domain by domain. Each framework uses the answer scale that fits it:

Framework	How controls are scored
GTSAF, ATF	Each control is answered **Yes / No / N/A**. ATF adds a depth rating: how embedded a Yes is, or how far a No has progressed.
EU AI Act, NAGF	Each requirement is **Compliant / Non-compliant / N/A**, with a depth rating.
ISO/IEC 42005	Each requirement is scored on a **maturity scale**.
MAESTRO	Each threat is scored for **likelihood and impact**, which combine into a risk band.

For each control, whatever the scale:

- Record the **answer or score**.
- Capture the **rationale**, why you reached that conclusion.
- Attach or request evidence where relevant.
- If a control genuinely does not apply, mark it **N/A with a documented justification**. An unjustified N/A stays in scope and counts as not met.
- Raise a **finding** for any gap, deficiency or exception.

Scoring is **coverage-inclusive**: controls you have not assessed count as not-yet-met, so the score reflects real progress rather than only the parts you have looked at. See how scoring works for the model that applies across every framework.

Capture evidence, tests and findings

As you score, the supporting registers are one click away:

- **Evidence Tracker**: raise evidence requests and attach artefacts against controls.
- **Testing Centre**: record control tests with design and operating effectiveness.
- **Findings Register**: track gaps through root cause, remediation and validated closure.

After the assessment

7. Invite your team

Bring colleagues into Gamut and assign roles from the real catalogue, a tenant role for organisation-wide access plus additive assessment-level roles for the work they touch.

Gamut is a shared operating layer, so it works best with the right people in place, governance leads, risk and compliance, security, procurement and the owners of individual AI systems.

Inviting people

Administrators invite colleagues by email. Each invitation is tied to a **role** that determines what the person can see and do once they accept.

1. Open **Administration -> Users**.
2. Select **Invite user** and enter their email address.
3. Choose the **tenant role** appropriate to their responsibilities (see below).
4. Send the invitation. The recipient receives an email with a secure link to join.

Choosing a tenant role

The tenant role sets what someone can do across the organisation. Assign the least privilege needed for their job:

Role	Use it for
Administrator	People who manage users, SSO, plans and the admin console.
Advanced	Governance, risk and security staff who need all modules, AI analysis, the agentic stack and exports.
Standard	Contributors who need core frameworks and limited AI, without control testing, workpapers or the agentic stack.
Subscriber	Login only, no product access (useful while access is being arranged).

What a role can actually use is also capped by the workspace's plan: a user needs both the role permission and the plan entitlement.

Scoping access to specific assessments

Beyond the tenant role, you can add someone to an individual assessment with an **additive workspace role**, so they get broad read access organisation-wide but write access only where they work:

- **Lead Assessor**: full read/write on the assessment, can share, delete, approve and export.
- **Contributor**: create and update records, no delete or export.
- **Reviewer**: read plus notes and comments.
- **Viewer**: read-only.

This is the practical way to bring in a system owner, an auditor or a reviewer without granting them broad authority. See Users & roles for the full model.

A starting allocation

- **System / model owners**: Standard tenant role, added as Contributor on the assessments covering their systems. They also get read/update on objects they own automatically.
- **Governance, risk and compliance**: Advanced tenant role, Lead Assessor where they run the work.
- **Reviewers and auditors**: Standard or Subscriber tenant role, Reviewer or Viewer on the relevant assessments.
- **Administrators**: Administrator tenant role.

Single sign-on

For larger organisations, connect your identity provider so people sign in with corporate credentials and access follows your existing joiner / mover / leaver process. SSO governs authentication; roles still govern authorisation. See Single sign-on (SSO).

Accountability

Granting or revoking a role is an audited action, and every state-changing action in the workspace is recorded in the audit log. Suspending a user immediately revokes their access and ends their active sessions.

Next

Part 2: Tenant, identity, access and commercial administration

Operate identity, role, workspace, entitlement, billing, auditor and security controls using least privilege and separation of duties.

8. Workspaces & tenancy

Each organisation uses Gamut in an isolated tenant, with assessments as workspaces inside it, keeping AI systems, evidence and users separated by design.

Gamut is multi-tenant. Each organisation operates in its own **tenant**, an isolated environment whose AI systems, assessments, evidence and users are kept separate from every other organisation. Inside a tenant, each **assessment** acts as a workspace that work and membership attach to.

Isolation by construction

Tenant isolation is a deliberate security property, enforced at the data layer rather than by application filtering alone: each tenant's records live in their own normalised database schema. Your governance records, evidence and users are never visible to another organisation, and access is always scoped to the tenant you belong to. See Security & data handling.

What lives in a tenant

- The AI system registry and discovery results.
- Assessments, evidence and findings.
- The agent register and agentic governance.
- Users and roles.
- Tenant configuration, including SSO and entitlements.

Assessment-level membership

Within a tenant, membership and roles can be scoped to an individual assessment. A user carries a tenant role for what they can do organisation-wide, and an additive workspace role, Lead Assessor, Contributor, Reviewer or Viewer, on the specific assessments they are added to. This is how you give someone broad read access but write access only where they actually work.

The **Runtime Policy Approver** role is also workspace-scoped. It reveals submitted Runtime Access Policies only for assigned workspaces and does not grant general editing authority.

View-only workspaces

Locking a workspace in **View only** mode preserves its governance record for inspection while blocking changes. Existing Runtime Access Policies can still be opened and reviewed, including their boundaries and history. Policy drafting, editing and lifecycle actions remain unavailable until the workspace is deliberately unlocked by an authorised user.

Belonging to more than one tenant

Some people work across more than one organisation or environment, for example advisers and auditors. Where that applies, a user can belong to multiple tenants and switch between them, always acting within the one currently selected and never carrying data across the boundary.

Tenant status

A tenant has a status that controls access. An active tenant operates normally; a suspended tenant is locked out entirely until reactivated. Provisioning, suspension and reactivation are audited account lifecycle actions.

Next

9. Users & roles

Gamut governs access with a fixed catalogue of roles across three scopes, enforced server-side as a permission set, combined with plan entitlements, and additive at the assessment level.

Access in Gamut is governed by **roles**. A role maps to a precise set of namespaced permissions (for example `system.update`, `finding.export`, `agent.approve`), and every state-changing action checks for the permission it requires. Roles are not free-form; they are a defined catalogue, enforced the same way in the interface and behind every API call.

Three scopes of role

Roles exist at three scopes, and a user's effective access is the combination.

Tenant roles

What a user can do across the organisation:

Role	Access
Administrator	Full tenant access, user management and the admin console.
Runtime Policy Approver	Approval-only access to inspect and decide submitted Runtime Access Policies in explicitly assigned workspaces.
Advanced	Advanced-tier frameworks, AI analysis, Agentic CISO, simulation and exports; Enterprise runtime and licensed ISO access remain separately gated.
Standard	Core frameworks, bounded AI and workpaper/report capabilities. No control testing, AI Consultant or agentic stack.
Subscriber	Login only. No product access.

Workspace (assessment) roles

Additive permissions scoped to a single assessment, layered on top of the tenant role:

Role	Access within the assessment
Lead Assessor	Full read/write, can share and delete, can approve agents and export.
Runtime Policy Approver	Read supporting agentic context and approve or reject submitted Runtime Access Policies only.
Contributor	Create and update records. Cannot delete or export.
Reviewer	Read access plus notes and comments. Cannot delete.
Viewer	Read-only.

Permission and entitlement: both must pass

A role granting a permission is necessary but not always sufficient. Sensitive product capabilities are gated by **both** an RBAC permission **and** a plan entitlement. A tenant administrator is deliberately not exempt: administration authority never unlocks a capability the tenant has not purchased. So "can this user do X" is answered by two questions: does their role grant the permission, and does the plan include the feature.

Object ownership

Independently of role, the **owner** of an object gets read and update on that object, for systems, risks, findings, evidence and agents. Ownership never escalates to delete or to audit-sensitive actions; it is a deliberately narrow grant so owners can maintain their own records without broad access.

Least privilege

Assign the least privilege needed for someone to do their job. It is easier to grant more access later than to recover from over-sharing, and the additive workspace roles mean you can give someone broad read access at the tenant level and write access only on the assessments they work on.

Runtime Policy Approver

Runtime policy decisions can change the authority exercised by AI agents. Gamut therefore provides a dedicated **Runtime Policy Approver** rather than requiring reviewers to hold an administrator role.

When creating an approver account, the administrator selects the exact workspace the person may review. Additional workspace assignments must be granted explicitly. A policy is visible only when it belongs to an active, assigned workspace.

The role can:

- inspect submitted Runtime Access Policies and relevant supporting context;
- use the pending-approval inbox;
- approve and publish a pending policy;
- reject a pending policy with a recorded reason.

The role cannot:

- draft, edit, delete or submit a policy;
- suspend or revoke active authority;
- create or change agents, tools, connections, approval gates or data flows;
- manage users, roles, workspaces or tenant settings;
- operate Gateway or bypass plan entitlements.

The restriction is enforced behind the interface as well as in it. Account-security actions such as changing the approver's own password and enrolling MFA remain available.

Both approval and rejection require the approver's password and current MFA code through a short-lived step-up verification. The person who created or submitted the policy cannot approve that version. Every assignment and decision is audited.

See Runtime Access Policies for the review checklist and policy lifecycle.

Managing users

Administrators manage users from **Administration -> Users**:

- **Invite** colleagues by email and assign a role. See Invite your team.
- **Adjust** roles as responsibilities change.
- **Assign approvers** to the exact workspace containing the policies they may review.
- **Suspend** a user to immediately revoke access. An inactive user fails every permission check, and suspension ends their active sessions.

Sessions

Administrators can review active sessions and revoke them individually, useful for offboarding or responding to a concern. Sessions also expire on their own over time.

Privileged controls and break-glass

The **admin console** holds the most sensitive controls, behind both a permission gate and a **step-up re-authentication**. They include the runtime break-glass switches:

- **Disable Gateway Runtime** (the Gateway kill switch): blocks every agent action across the tenant, fail-closed, until reactivated. See Gateway.
- **Suspend Agent Runtime**: halts a single agent's actions until it is reactivated.
- Gateway approval review, connector and integration configuration.

Every one of these is audited. They are deliberately kept out of the normal app and behind step-up so a break-glass action is always a deliberate, attributable decision.

Accountability

Granting or revoking a role is itself an audited action, and every state-changing action a user takes is written to the audit log, so access and activity remain accountable and reviewable.

Single sign-on

For organisations with an identity provider, SSO governs authentication (proving who someone is) while roles continue to govern authorisation (what they can do).

Next

10. Account security and MFA

Protect human accounts with MFA, step-up verification, secure sessions and a controlled recovery process.

Every person should use their own account. Shared accounts weaken attribution, separation of duties and incident response.

Enrol MFA

Use the **MFA** control in the sidebar to begin enrolment. Scan the displayed authenticator setup with an approved application, enter the current code and store any recovery material according to your organisation's security process. Do not send setup secrets or recovery codes through chat or email.

MFA enrolment protects future sign-ins and supports **step-up verification** for sensitive actions. Runtime Policy Approvers must enrol MFA before they can approve or reject policies. MFA does not hide their assigned pending policies; it prevents the decision until strong verification is complete.

Step-up actions

Actions such as security changes, write-token creation, trial activation and sensitive approvals may require the current password and/or authenticator code. Reauthentication is deliberately short-lived and applies to the exact operation being performed.

Account lifecycle

- Invite users to the correct tenant and role.
- Verify identity through the approved organisational process.
- Require MFA for privileged and approval roles.
- Review sessions, tokens and role assignments after job changes.
- Suspend access promptly when no longer required.
- Use administrator-supported recovery rather than creating a replacement shared account.

If compromise is suspected

Suspend the account, revoke sessions and API tokens, reset credentials, review the Audit Trail and investigate affected records. Restore access only after identity and device security are confirmed.

Next

11. Single sign-on (SSO)

Connect your identity provider to Gamut with OpenID Connect so people sign in with corporate credentials and access follows your existing identity process.

Gamut supports **single sign-on (SSO)** via **OpenID Connect (OIDC)**, so people sign in with their corporate credentials and your existing identity process governs access.

Why use SSO

- **One set of credentials.** People sign in with the account they already use.
- **Centralised control.** Access follows your joiner / mover / leaver process in your identity provider.
- **Stronger security.** You apply your own MFA and conditional-access policies at the identity provider.

How it works

SSO is configured per workspace by an administrator, using the standard OpenID Connect authorization code flow:

1. In your identity provider, register Gamut as an application and obtain the connection details (discovery URL, client ID and client secret).
2. In Gamut, open the **Administration -> SSO** panel and enter those details.
3. Save the configuration. Once configured, the sign-in page offers a **Sign in with SSO** option.
4. People authenticate at your identity provider and are returned to Gamut, where their account is provisioned from the verified identity.

Connection secrets are stored encrypted. Gamut verifies identity tokens cryptographically on each sign-in.

Configuring SSO

SSO is configured per tenant by an administrator:

1. In your identity provider, register Gamut as an **OpenID Connect application** and note the issuer (discovery) URL, client ID and client secret. Set the redirect (callback) URL to the one Gamut provides for your tenant.
2. In Gamut's tenant administration, enter the **issuer URL, client ID and client secret**, and choose how provider identities map to Gamut users (for example which email domains are allowed and the default role for a newly provisioned user).
3. **Test** a sign-in with a provider account, then enable enforcement so users authenticate through the provider.

Credentials are held server-side, never exposed to the browser, and the sign-in is CSRF- and nonce-protected. Roles and entitlements still apply on top of SSO, see roles with SSO.

Supported providers

Any standards-compliant OpenID Connect identity provider can be used, for example the major enterprise identity platforms. If your provider supports the OpenID Connect authorization code flow with discovery, it will work with Gamut.

Roles with SSO

SSO governs authentication, proving who someone is. Their authorisation in Gamut is still controlled by roles within the workspace.

Next

12. Plans & entitlements

Current Gamut plan pricing, framework access, quotas, billing behavior and server-enforced entitlements, capped by both plan tier and user role.

A workspace's **plan** determines which frameworks, modules, quotas and billing options it can use. **Entitlements** are enforced server-side, not only hidden in the interface. A capability is available only when the workspace plan allows it and the signed-in user's role allows it.

Current plan tiers

Free

- **Price:** $0.
- **Scale:** 1 paid seat, up to 3 AI systems and 3 intake records, 1 active assessment at a time, and up to 3 assessment creations per rolling 12 months.
- **Framework access:** Choose 1 at signup: GTSAF, NIST AI RMF, EU AI Act or NAGF.
- **Included:** Dashboard, Learning Hub, AI System Records, AI Use Case Intake, assessment, risk register, model cards, CSV export and a limited platform-funded AI-analysis allowance.
- **Not included:** AI Consultant, reports, evidence tracker, findings register, policy generation, audit-trail module, Discovery, Agentic CISO, Gateway and Claw.

Standard

- **Price:** $499/month, or $4,990/year.
- **Scale:** 5 paid seats, up to 25 AI systems and 50 assessments per year.
- **Framework access:** GTSAF, EU AI Act, NIST AI RMF and NAGF.
- **Included:** Free capabilities plus policies, AI analysis, bounded policy generation, reports, report export, evidence tracker, findings register, workpaper packs and unlimited free read-only auditor seats.
- **Not included:** Control testing, AI chat, Agentic CISO, Gateway simulation, Gateway, Claw and Discovery.

Advanced

- **Price:** $1,499/month, or $14,990/year.
- **Scale:** 25 paid seats, up to 200 AI systems and 200 assessments per year.
- **Framework access:** GTSAF, EU AI Act, NIST AI RMF, NAGF, MAESTRO, ACRS and ATF.
- **Included:** Standard capabilities plus control testing, AI chat, policy generation, Agentic CISO and Gateway simulation.
- **Not included:** Production Gateway enforcement, Claw runtime execution and Discovery, which remain Enterprise-only.

Enterprise

- **Price:** Starting at $40,000/year, contact us.
- **Scale:** 100 paid seats, effectively unlimited AI systems and effectively unlimited assessments.
- **Framework access:** GTSAF, EU AI Act, NIST AI RMF, NAGF, MAESTRO, ACRS and ATF, plus ISO/IEC 42001 and ISO/IEC 42005 when licence-confirmed.
- **Included:** Full product access, including Discovery, Agentic CISO, Gateway, Claw, production runtime enforcement, OpenID Connect single sign-on and contracted data-residency options.
- **Independent runtime review:** the dedicated Runtime Policy Approver role is available for workspace-scoped policy decisions without granting administrator authority.

ISO/IEC 42001 and ISO/IEC 42005 are not automatic plan entitlements. They are enabled only after a valid licence confirmation is recorded.

Standard annual billing saves $998 compared with monthly billing. Advanced annual billing saves $2,998 compared with monthly billing.

Free signup framework choice

New public signups start on **Free**. During signup the user chooses one starting framework from GTSAF, NIST AI RMF, EU AI Act or NAGF. The Free workspace is locked to that selected framework.

GTSAF is the recommended default when the user does not have an immediate framework-specific need: it has the deepest control coverage and maps outward to the other frameworks. If the user needs a specific regulatory or regional view first, they can choose NIST AI RMF, EU AI Act or NAGF instead.

The ACRS score used by intake can still be calculated to route risk. The standalone ACRS assessment module is not included on Free unless the plan is upgraded or a valid boost is active.

Advanced Trial Boost

Free and Standard workspaces can activate a one-time **14-day Advanced Trial Boost** from the billing page. The boost:

- temporarily grants Advanced-tier features;
- includes 50 AI Consultant interactions;
- preserves all data created during the boost after it expires;
- can be used once per workspace;
- requires administrator step-up MFA before activation.

After the boost expires, the workspace returns to its prior plan and the normal entitlements apply.

Quotas

Plan	AI analysis quota	Policy generation quota	Audit retention
Free	20 daily, 20 monthly	0 daily, 0 monthly	90 days
Standard	20 daily, 200 monthly	2 daily, 10 monthly	365 days
Advanced	50 daily, 500 monthly	100 daily, 1,000 monthly	1,095 days
Enterprise	999 daily, 9,999 monthly	999 daily, 9,999 monthly	2,555 days

Policy generation has a separate entitlement and quota. If the `policy_generation` entitlement is off, the policy quota is zero even if general AI analysis is available.

Agentic access

The agentic stack is split deliberately:

- **Advanced** can use Agentic CISO, ATF assessment and Gateway simulations.
- **Enterprise** is required for production Gateway enforcement, Claw dispatch, Claw schedules, BYO runtime credentials, Gateway Event Centre runtime actions and Discovery.

This lets teams design and assess agentic governance before allowing live runtime execution. Runtime-sensitive actions stay behind the Enterprise `gateway` entitlement and are checked again server-side at the access point.

Runtime Access Policy approval is also Enterprise-gated. Assigning an approver role does not create a plan entitlement and cannot make production Gateway enforcement available on another plan.

Two caps, whichever is lower

Effective access is the **intersection** of two limits:

- the workspace **plan tier**;
- the user's **role product ceiling**.

A Standard-role user is capped at Standard capabilities even on an Enterprise workspace. An Advanced-role user can use Advanced capabilities but cannot reach Enterprise-only Gateway or Claw runtime execution. External auditor users are read-only across the assurance surface and do not inherit runtime privileges from the tenant plan.

How gating is enforced

Every gated capability binds a feature entitlement to one or more RBAC permissions. Requests are checked server-side, including API calls, asset access, report and workpaper scopes, framework writes, Gateway actions and runtime dispatch. The UI is not the security boundary.

The access model uses role-based access plus entitlement dual-gating, row-level tenant isolation, fail-closed checks, CSRF protection, rate limiting, step-up MFA for sensitive changes and full audit records for state-changing actions.

Administering your plan

Administrators can review billing and the current plan from **Billing**. Standard and Advanced checkout use secure payment flow. Enterprise is handled by contract and invoice.

> **Note:** The live workspace's returned quota and entitlement payload remains authoritative for that tenant and user.

Next

13. Audit log

Gamut writes an audit entry before every state-changing action returns, with a defined set of high-sensitivity actions that mandate an audit record, giving each workspace a reviewable history.

Gamut records state-changing actions in an **audit log**. It gives every workspace an accountable record of what happened, who did it and when, which is essential for a platform whose whole purpose is defensible governance.

Written before the action returns

Auditing is not an afterthought in Gamut. Every state-changing API handler writes its audit entry before it returns, so a successful change and its audit record are inseparable. The log captures the actions that change governance state, creating or updating AI systems, running assessments, managing evidence and findings, changing users and roles, agentic governance decisions, rather than routine read-only activity.

Actions that mandate an audit record

A defined set of high-sensitivity permissions **require** an audit entry. These are the actions where after-the-fact accountability matters most:

- **Access control**: granting or revoking roles.
- **Risk decisions**: formal risk acceptance, and risk-register export.
- **Findings**: deleting or exporting findings.
- **Exports**: report, workpaper, evidence and audit-log exports.
- **Gateway**: altering enforcement decisions, deleting simulations and changing runtime policy lifecycle state.
- **Policy**: generating, publishing or deleting policy documents.
- **Agents**: deleting an agent, and ATF promotion or approval-gate sign-off.
- **Account and billing**: tenant lifecycle, billing changes and SSO configuration.

Marking these as audit-required in the permission model means the obligation travels with the permission, it cannot be forgotten when a new endpoint is added.

Why it matters

The audit log is part of what makes Gamut governance trustworthy:

- **Accountability.** Every entry is attributable to a user and a time.
- **Reviewability.** Internal audit and reviewers can see how the governance record came to be what it is.
- **Integrity.** A consistent record of changes supports investigation and assurance.

It complements the evidence and findings model: evidence proves the state of controls, while the audit log proves the history of actions. Exporting the audit log is itself an audited action.

Runtime evidence for agents

For agentic AI, the equivalent record is the runtime evidence captured as every agent action passes through Gateway and flows back to Agentic CISO, alongside the tamper-evident, hash-chained journal that Claw keeps per task. Together, the audit log and runtime evidence give a complete account of both human and agent activity.

Runtime policy decision history

Each Runtime Access Policy also exposes its authoritative history. The combined policy history and audit log record:

- draft creation and controlled updates;
- submission for independent review;
- approval and publication;
- rejection and its reason;
- suspension, revocation, expiry or replacement;
- the identity responsible for each decision;
- successful and failed approver-verification events.

Policy content is frozen while pending or authoritative, so the reviewed version cannot be silently edited after the decision.

Next

14. Billing, plans and Trial Boost

Review effective entitlements, manage billing and activate a time-limited Advanced Trial Boost without changing permanent plan rights.

The Billing area shows the workspace plan, quotas and available upgrade or trial actions. The server-returned entitlement profile for the current tenant and user is authoritative.

Before changing a plan

Review required frameworks, system and assessment volume, seats, AI usage, reporting, Agentic CISO, Discovery and production runtime needs. A higher plan expands the tenant ceiling; the user's role still limits effective access.

Advanced Trial Boost

Eligible Free and Standard workspaces can activate one 14-day Advanced Trial Boost. Activation is an administrator action protected by step-up MFA. The boost temporarily grants Advanced capabilities and a bounded AI Consultant allowance; it does not grant Enterprise-only production Gateway, Claw or Discovery.

When the boost expires, records remain but unavailable modules become read-only or inaccessible according to the restored plan. Export or review important outputs before expiry and do not design an operating dependency on a temporary entitlement.

Billing controls

- Use the approved checkout or contractual process.
- Confirm workspace, plan, term and price before payment.
- Restrict billing administration to authorised users.
- Review entitlement state after checkout or contract changes.
- Contact support when payment and effective entitlement do not reconcile.

Next

15. API keys and model providers

Configure personal model-provider keys securely and distinguish them from Gamut API tokens and agent runtime credentials.

My API Keys stores user-authorised model-provider configuration for supported AI features. These keys are different from Gamut bearer API tokens and from Gateway-managed runtime credentials.

Credential	Purpose	Custody
Model-provider key	Run permitted AI Assist, Consultant or generation features.	Stored and used server-side for the authorised user context.
Gamut bearer token	Integrate with supported Gamut API resources.	Created by the user and stored by the external integration.
Gateway connection credential	Invoke an approved external tool or provider at runtime.	Held by the governed Gateway connection, never the agent.

Configure safely

1. Obtain a dedicated organisational provider key where possible.
2. Restrict provider-side project, spend and capability.
3. Add it through **My API Keys**; do not paste it into assessment notes or chat.
4. Run a low-risk test and confirm the selected provider.
5. Monitor usage and rotate according to policy.
6. Remove the key promptly when no longer required or suspected exposed.

The browser does not receive the stored provider secret during ordinary AI use. Entitlement, permission and usage quotas still apply even when the user supplies the provider key.

Privacy decisions

Configuring a key does not itself authorise sending every workspace record. Use scores-only privacy mode for framework AI Assist when detailed context is unnecessary. The AI Consultant uses the permitted workspace context shown on its screen and should not be used where that context is not appropriate for the configured provider.

Next

16. Auditor and independent review access

Invite read-only reviewers and dedicated Runtime Policy Approvers without granting administrator authority.

Gamut provides separate access patterns for assurance review and runtime-policy decisions. Neither requires an administrator role.

Auditor access

Auditors receive a read-only assurance profile constrained by tenant, workspace, plan and role. They can inspect the permitted systems, assessments, risks, policies, evidence, findings, tests, workpapers and audit records, but cannot mutate governance records or export where the role does not permit it.

Invite the reviewer to the exact workspace and period required. Review the invitation status and remove access when the engagement ends. Free-plan review invitations may be time and count limited; paid-plan behaviour follows the current entitlement.

Runtime Policy Approver

The dedicated approver role is for independent decisions on submitted Runtime Access Policies in assigned Enterprise workspaces. The approver can inspect policy context and approve or reject. The role cannot draft, edit, delete, submit, suspend or revoke policy; administer users; operate Gateway; or mutate unrelated product records.

MFA must be enrolled before an approval or rejection. The creator or submitter cannot approve the same policy. The plan must include production Gateway and policy approval; assigning the role does not create an entitlement.

Review checklist

- Correct tenant and workspace assignment.
- Minimum role for the review purpose.
- Individual account and verified identity.
- MFA for sensitive approval.
- No creator-approver conflict.
- Access expiry and removal recorded.

Next

Part 3: Workspace governance and assurance operations

Understand the records, routing, evidence, reporting and assurance workflows that administrators enable and oversee.

17. Launch Center and Learning Hub

Use guided readiness and learning resources to move from account setup to repeatable governance operations.

The **Learning Hub** provides role-appropriate guidance. The **Launch Center** provides an operational readiness view for authorised platform operators. They support adoption but do not replace the governance records, approvals and evidence held elsewhere in Gamut.

Learning Hub

Use the Learning Hub when onboarding administrators, assessors, reviewers and agentic governance teams. Complete the recommended path for the work the person will perform, then validate learning through supervised use of a non-production or demonstration workspace.

Learning completion is not evidence that a control operates. Where competence is a control requirement, retain the organisation's approved training, role assignment and practical assurance evidence.

Launch Center

Launch Center is visible only to appropriately authorised platform operations users. It summarises readiness and directs operators to incomplete setup or control work. It should be treated as a navigation and readiness aid, not as a bypass around tenant administration, entitlement, MFA or approval requirements.

When an item is incomplete:

1. Open the identified area.
2. Confirm the accountable owner and required evidence.
3. Complete the underlying configuration or governance decision.
4. Return to verify that readiness reflects the saved state.

Do not publish screenshots or operational configuration from Launch Center into customer-facing evidence unless it is necessary, appropriately redacted and approved.

Recommended onboarding sequence

1. Understand tenancy, roles and plan entitlements.
2. Register a system and complete intake.
3. Review routing, ACRS and the assessment plan.
4. Perform a framework assessment with evidence.
5. Record risks, findings and remediation.
6. Generate a traceable report.
7. For agentic operation, complete the dedicated Agentic CISO and Gateway path.

Next

18. AI System Records

Create and maintain the authoritative inventory record that connects intake, routing, assessments, evidence, risks, model cards and reporting.

An **AI System Record** is the stable identity for the system Gamut governs. Intake describes a particular use and determines routing; the system record identifies the product, service, model or agentic workflow to which that work belongs. Keeping those responsibilities separate prevents an assessment result from becoming detached from its owner, version or operating context.

What to record

Complete enough detail for another reviewer to identify the same system without relying on local knowledge:

Area	Record
Identity	Name, unique identifier, version and lifecycle state.
Accountability	Business owner, technical owner, responsible-AI contact and operating department.
Purpose	Intended use, users, affected people and decisions or actions supported.
Technology	Model or AI type, supplier, deployment arrangement, environment and agentic capability.
Data	Sources, classifications, personal-data involvement and geographic exposure.
Operation	Autonomy, human oversight, integrations, access and dependency.
Governance	Registration status, risk information, review dates and linked assurance records.

Do not use a marketing product name alone where several deployments, versions or purposes have different risks. Create boundaries that match how change, ownership, access and evidence are managed.

Lifecycle

1. **Create** the record as soon as a proposed or discovered system enters governance.
2. **Complete** identity, ownership and operating context before relying on routing.
3. **Open or link intake** for the use case being approved.
4. **Assess** the system through the routed framework modules.
5. **Maintain** version, supplier, data, purpose, access and deployment changes.
6. **Reassess** when a material change invalidates prior scope or evidence.
7. **Retire** the record without deleting the history needed for accountability.

Relationship to intake

The system record and intake are connected but are not interchangeable. The record supplies durable facts such as system identity and ownership. Intake supplies the decision context used for risk, legal routing, framework applicability and assurance depth. Where the two disagree, investigate and correct the source record rather than selecting whichever result is more convenient.

Use **Sync system record** when the intake screen identifies that the linked record needs updating. Confirm the saved state before leaving the page. A system with incomplete intake can remain in the inventory, but its routing should be treated as provisional or incomplete.

Downstream effects

The selected system scopes framework assessments, AI-assisted assessment outputs, risks, model cards, evidence, findings, reports and reassessment history. Renaming a system should not create a new governance identity. A genuinely different product, materially different deployment or independently governed use may require a separate record.

Review checklist

- [ ] The record identifies one defensible system boundary.
- [ ] Named owners are current and able to act.
- [ ] Purpose, users and affected people are specific.
- [ ] Data, geography, supplier and deployment facts are current.
- [ ] Autonomy and human oversight describe actual operation.
- [ ] The correct intake and assessments are linked.
- [ ] Material changes have triggered reassessment.
- [ ] Retired systems retain the required audit history.

Next

19. Registry & Discovery

Maintain a standing inventory of AI systems and surface shadow AI across the organisation through a governed discovery pipeline of sources, rules, candidates and reconciled artifacts.

The **Registry** is your standing inventory of AI systems. **Discovery** is how you find AI that is in use but not yet registered, so the inventory stays honest.

The Registry

The Registry holds every AI system the organisation governs. Each record is structured, with the fields governance and audit actually need:

- **Identity and ownership**: name, system ID, version, owner, owner email, department, and a responsible-AI contact.
- **Nature of the system**: AI technique, model type, autonomy level, human-oversight model, vendor, deployment type and environment.
- **Data and exposure**: data classification, personal-data involvement and detail, data sources, users and affected persons, geographies and regulatory exposure.
- **Governance state**: risk tier, conformity status, registration status, ACRS score, and review dates (deployment, last review, next review).
- **Lifecycle stage**: from `development` through in-use to retired.

The Registry is the anchor of the platform: assessments, evidence, findings, model cards and reports all attach to a registry record. See Register your first AI system for the steps.

Discovery

The hardest part of AI governance is knowing what exists. Discovery is a governed pipeline for surfacing AI systems, GenAI tools and emerging agentic workflows that are in use but not yet registered. It is built from four connected object types:

Object	What it is
Sources	**Where signals come from:** connector-based collectors or manual imports, with a cadence, owner and run history.
Rules	Normalisation rules that map raw signals to a canonical app and vendor by domain or pattern, and tag sanction status and risk level.
Candidates	Suspected AI in use, each with a signal type, a detector, a confidence level, guessed owner/vendor/environment, and a review decision.
Artifacts	The underlying evidence (usage signals, code references, endpoints, model references) with a fingerprint and a reconciliation status.

A discovery **run** executes a source, produces candidates and artifacts, and records a summary and count. Candidates move through a review (`new` to a decision), and artifacts carry a **reconciliation status** so each piece of evidence is either tied to a known asset or flagged as unreconciled.

From discovery to registry

Reviewed candidates are promoted into the Registry, where they go through intake and risk tiering like any other system. Discovery finds the shadow AI; intake classifies and routes it; the Registry holds it. Nothing is governed until it is a registry record, and discovery is how things get there.

> **Note:** Discovery is an enterprise capability. Availability depends on your plan & entitlements.

Setting up discovery

1. Open **Discovery** and add a **discovery source** (for example a cloud or identity signal feed, or a manual import). Give it a name and type, and provide the import or connection detail it needs.
2. Optionally set a **schedule** (cadence) so the source is scanned automatically; otherwise run it on demand. A scheduled source shows its next and last run and overdue status.

3. Review the **candidates** the scan surfaces in the Discovery Inbox: each carries a source, confidence and rationale.
4. For a real, in-use system, **promote it to the Registry** (which starts the standard governance workflow); otherwise dismiss it with a logged reason.
5. In the **Registry**, complete each system record: identity, ownership, nature, data and exposure, and governance fields.

This is what feeds intake: a current registry makes intake and everything downstream complete.

Why this matters

An inventory that misses systems produces governance that looks complete but is not. Registry and Discovery together give you an inventory you can defend: comprehensive, owned, current, and with a documented trail showing how each system was found and brought under governance.

Next

20. Discovery operations

Operate AI Discovery from source coverage and rules through candidates, alerts, reconciliation, promotion and assurance reporting.

Discovery helps identify AI that may not yet be registered. It produces **signals requiring human review**, not automatic declarations that a person or team is using an unauthorised system.

Operating model

Stage	Question
Coverage	Which business areas, identities, cloud services, repositories or imports are observed?
Detection	Which built-in or approved custom rules interpret the signals?
Candidate review	Is the signal a real AI system, duplicate, permitted tool or false positive?
Reconciliation	Does it match an existing system, supplier, model or prior candidate?
Governance action	Should it be promoted, linked, dismissed, monitored or escalated?
Assurance	Can coverage, decisions and unresolved gaps be explained to management or audit?

Sources and credentials

Define the source owner, purpose, scope, cadence and lawful basis before collection. Use the smallest read scope that can produce the required signal. Store connection credentials through the approved tenant-scoped integration mechanism; do not paste credentials into candidate notes or imports.

Test a source before relying on its coverage. A healthy connection does not prove full coverage if important accounts, regions, repositories or environments are excluded.

Rules and scans

Built-in and custom rules normalise raw indicators into reviewable candidates. Document what a rule detects, expected false positives, risk labels and the owner who can change it. Preview an import or rule change before a live scan where that option is available.

Monitor last run, next run, errors, volume and unusual changes. A sudden zero may indicate a clean environment, but it can also indicate failed collection. A sudden spike may reflect a deployment, rule change or duplicate ingestion.

Candidate and alert decisions

For each candidate:

1. Inspect source, detector, confidence, rationale and supporting artifacts.
2. Search for an existing system or related candidate.
3. Confirm owner and intended use with an accountable person.

4. Promote a real system into the Registry or link it to the matching record.
5. Dismiss a false positive with a reason that another reviewer can understand.
6. Escalate material unknown use, prohibited tools or sensitive-data exposure.

Alerts should have a named disposition and owner. Bulk actions are appropriate only when the same decision basis genuinely applies to every selected item.

Reconciliation and playbooks

Reconciliation prevents duplicate inventory and preserves the chain from raw artifact to governed record. Suggested matches assist review but do not replace confirmation. Discovery playbooks can create or link follow-up risks and evidence work; confirm the target system and workspace before execution.

Assurance reporting

A defensible Discovery report explains sources, coverage, run health, rule set, candidates, decisions, unresolved artifacts, overdue alerts and limitations. It must not imply that "no candidates" means "no shadow AI" unless coverage and detector effectiveness support that conclusion.

Review checklist

- [ ] Sources have owners, purpose, scope and review dates.
- [ ] Credentials use least privilege and are not exposed in records.
- [ ] Rules are tested and false-positive assumptions documented.
- [ ] Failed and overdue scans are investigated.
- [ ] Candidates have attributable decisions.
- [ ] Promotions enter normal intake and routing.
- [ ] Unresolved artifacts and coverage gaps are reported.

Next

21. Intake & risk tiering

Capture the context of an AI system, classify its risk through structured signals and an ACRS capability score, and route the right frameworks and reviews to it.

Intake captures the context an AI system needs to be governed. **Risk tiering** turns that context into a structured classification that helps decide how much governance the system needs. Authoritative routing combines Intake with the linked System Record to determine which frameworks need assessment or screening. Together they form the front of the governance lifecycle.

What intake captures

Intake is a structured record, not a free-text form. Alongside the descriptive context, purpose, users, owner, business unit, vendor, deployment type and environment, data sources and geographies, it captures a set of **risk signals** as explicit flags:

- **Personal data** and **special-category data** involvement.
- **High-risk** use and **automated decision-making**.
- **Public-facing** exposure.
- **Retrieval characteristics: RAG**, retrieval sources, **vector store**, **long context** and **external retrieval**.
- Context-specific signals such as **cultural significance**, **heritage relevance**, **local language significance**, **creative-sector** use and **community impact**.
- Cross-framework routing facts covering **decision impact**, **action capability**, **model origin**, **organisation roles**, material capabilities and **jurisdictions**.

These are routing inputs. The linked System Record supplies the standing system facts, while Intake supplies use-case and impact context. See Routing & applicability for the complete decision model.

From intake to risk tier

Intake produces a **risk tier** (a system carries `undetermined` until classified, then low through critical) and an **initial risk rating**. The risk tier is the pivot of the whole lifecycle: it prioritises attention and routes the system to the controls and frameworks that matter for its level of risk.

- Higher-risk systems attract deeper assessment and more demanding control expectations.
- Lower-risk systems are governed proportionately, without unnecessary overhead.

How signals become a tier is deterministic: a governance weighting profile of weighted dimensions and configurable thresholds does the mapping, so the same inputs always yield the same tier.

The ACRS score and confirmation

For systems with agentic or capability-driven risk, intake derives an **ACRS score** and a capability **band** from the signals captured. Each dimension is seeded by inference from the intake facts and can be refined with explicit assessor evidence, so an intake-anchored capability score exists straight away. Gamut reconciles the system's assigned risk tier against the ACRS band, so a mismatch between "how risky we called it" and "how capable it actually is" is surfaced rather than hidden.

You can open ACRS, complete the dimension assessment, and move on to the assessment plan without a mandatory gate. **Confirmation is an optional, audited sign-off**: it records an assessor-signed ACRS for the system and strengthens the basis for assurance depth, cadence and escalation. It is not required for GTSAF factual applicability: a complete, conflict-free authoritative Intake route is what makes the system selectable for a scoped GTSAF assessment.

If routing-critical facts change after confirmation, the route becomes **Reassessment required**. Review the changed facts, reconsider ACRS where necessary, and reconfirm only after the resulting framework routes and assessment scope have been reviewed.

Framework routing

Intake does not route in isolation. Gamut combines it with the linked System Record and produces:

- **Applicable frameworks**: frameworks positively triggered by current facts.
- **Screening-required frameworks**: frameworks that cannot safely be ruled out while facts remain unknown.
- **Organisational review**: a decision, such as ISO/IEC 42001 AIMS scope, that is not a simple system-level exclusion.
- **Required reviews**: the specific reviews it must undergo.
- A reviewable routing basis, quality gaps and any conflicts between the records.

This is what connects intake to the rest of the platform: a routed system arrives at assessment already pointed at the right frameworks.

Unknown material facts fail closed as **screening required**. They do not silently remove a framework. A confirmed route becomes **reassessment required** when its material basis changes.

Approval

An intake record carries an approval status (`draft` through approved) with a named approver. Risk classification is therefore an accountable decision, not an automatic one: a person signs off that the system is classified and routed correctly before it moves into deeper governance work.

Field groups and decision ownership

Group	Examples	Primary effect
Identity and purpose	System, use case, owner, business unit, intended and prohibited use	Establishes the assessed boundary and accountability.
People and decisions	Users, affected people, decision impact, appeal and human oversight	Risk, impact and legal routing.
Data and retrieval	Sources, classification, personal data, RAG, external retrieval and retention	Privacy, security, supplier and impact controls.

Group	Examples	Primary effect
Technology and supply	Model origin, vendor, deployment, integrations and dependencies	Threat, supplier, model and resilience routes.
Action and access	Autonomy, tools, permissions, targets and reversibility	ACRS, agentic and assurance depth.
Geography and regulation	Deployment, affected-person and market geography, regulated roles and sector	Jurisdictional and legal screening.
Lifecycle	Status, dates, review triggers and approver	Approval, reassessment and reporting.

The person completing intake should obtain facts from system owners, technical teams, data/privacy, security, legal and affected business functions as needed. "Unknown" is valid while investigating; it is not a convenient negative answer.

Saving and synchronising

The screen shows save or autosave status. Wait for a successful saved state before navigating away. If save fails, preserve the entered information, correct the displayed validation or service issue and retry. **Sync system record** updates the connected inventory record where supported; verify the result rather than assuming the two records now agree.

When authoritative facts are incomplete

Incomplete material facts produce a routing-incomplete or screening-required state. Gamut may show conservative starting values, but those are not a confirmed factual route. Complete the named facts, resolve contradictions, review ACRS and then confirm the route. Other descriptive intake fields remain valuable governance records even when they do not independently determine applicability.

Worked routing example

A marketing-content generator and a fraud-monitoring system are both registered AI systems, so both receive baseline governance. The marketing system may trigger supplier, retrieval and content controls while leaving many decision, privileged-access and severe-harm controls out of scope. The fraud monitor may process personal data, influence consequential decisions, depend on live systems and require stronger evidence, testing and review. ACRS and the Governance Weighting Profile can increase depth and escalation; they do not invent or remove the factual triggers.

Intake quality checklist

- [ ] System boundary and owner confirmed.
- [ ] Purpose, users, affected people and decision impact are specific.
- [ ] Data, retrieval, supplier, action and access facts reflect production reality.
- [ ] Geographic and regulated-role facts have an evidence basis.
- [ ] Unknowns and contradictions are resolved or explicitly escalated.
- [ ] Save and system-record synchronisation succeeded.
- [ ] ACRS, route, applicability and review streams were checked after changes.
- [ ] Approval and reassessment triggers are recorded.

Why consistency matters

The value of risk tiering is consistency. When every system is classified against the same signals and the same routing logic, risk decisions become comparable across the organisation and defensible to reviewers, who can trace each classification back to the intake that produced it. This is also the foundation for EU AI Act readiness, which is itself a risk-tiered regime.

> **Note:** Intake, risk and assessment capabilities vary by plan, volume and role. The current workspace plan and effective entitlements determine availability.

Next

22. Assessments & control testing

Score AI systems against framework controls and evidence that those controls are designed and operating effectively, with rationale, samples and traceable results.

An **assessment** evaluates its defined subject against a framework. Most subjects are AI systems; ISO/IEC 42001 uses an organisation-level AIMS scope and ATF uses a registered agent. **Control testing** goes a step further: it evidences that controls are not just designed, but operating effectively.

Assessments

An assessment connects an AI system to a framework and works through its controls, domain by domain. For each control you record:

- The **current state**, how well the control is met.
- The **rationale**, why you reached that conclusion.
- **Evidence** links, the proof that supports the conclusion.

Because an assessment records rationale and evidence per control, the result is defensible rather than just a score. See Run your first assessment for the steps.

Assessing against multiple frameworks

The same AI system can be assessed against more than one framework, for example GTSAF for depth and the EU AI Act for regulatory readiness. Gamut keeps each assessment distinct while sharing the underlying system, evidence and findings, so work done once is reused across frameworks rather than repeated. See Frameworks overview for how routing selects frameworks.

Running an assessment

1. Make sure the AI system is registered and has been through intake and risk tiering, so it is routed to the right frameworks.
2. Open the routed framework (for example GTSAF or the EU AI Act) for that system.
3. Work through the controls, scoring each on the framework's answer scale and recording your rationale. Each control's auditor advisory explains what to assess and how to score it.
4. Attach or request evidence, run control tests, and raise findings for gaps.
5. Roll the result up with reporting.

How scoring works

Gamut applies a consistent scoring discipline across every framework, so an assessment score is honest about both quality and completeness.

- **Each framework uses its own answer scale.** GTSAF uses Yes / No / N/A; ATF uses Fully met / Partially met / Not addressed; the EU AI Act and NAGF use Compliant / Non-compliant / N/A with a depth rating; ISO/IEC 42001 separates conformity from assurance; ISO/IEC 42005 uses maturity plus assurance; MAESTRO scores likelihood and impact. See Frameworks overview.
- **Scope-correct, with rollup.** System assessments remain bound to one system, ATF to one agent and ISO/IEC 42001 to the approved AIMS scope. Portfolio views do not transfer evidence between those subjects.
- **Applicability is separate from depth.** For GTSAF, system facts determine which conditional controls apply; ACRS and the governance weighting profile determine proportionate assessment rigour. A higher applicable count is not automatically a higher-risk conclusion.
- **Coverage-inclusive.** Controls you have not assessed count as not-yet-met, so a partly-finished assessment cannot report a high score.
- **Justified N/A only.** Marking an applicable control N/A requires a documented justification; a justified N/A leaves scope, while an unjustified one stays in scope and counts as not met.
- **Auditor advisory on every control.** Each control carries built-in guidance on what to assess, what evidence to expect, how to test it and how to score it, so assessors apply the scale consistently.

Control testing

Designing a control is not the same as operating it. A control test in Gamut captures the full testing record auditors expect:

- **Test objective** and **test procedure**, what was tested and how.
- **Sample method** and **sample size**, the basis for the conclusion.
- A **design effectiveness** score and an **operating effectiveness** score, kept separate, because a well-designed control can still fail in operation.
- A **test result**: `pass`, `partial`, `fail` or `not_tested`.
- **Exceptions**: a count and a summary of what failed.
- **Tester** and **reviewer**, test date and next test date, and **evidence references**.

This turns an assertion ("we review model outputs") into proof ("here are the reviews, sampled this way, on this cadence, by these people, with these exceptions"). Control tests produce the artefacts that back an assessment and that auditors rely on when they test your governance.

> **Note:** Control testing and workpaper-grade outputs are advanced capabilities. Availability depends on your plan & entitlements.

Findings

Where an assessment or control test reveals a gap, raise a **finding**. A finding carries a severity, a root cause, a recommendation and a management response, and it links back to the control, the test and the system it relates to, then tracks through to remediation and closure. See Evidence & findings.

AI assistance

Gamut can assist assessment work with AI, for example by helping analyse context or draft narrative. All AI analysis is proxied server-side, so model provider keys are never exposed to the browser. See AI assistance & data handling.

Next

23. Routing & applicability

How Gamut combines System Records and Use Case Intake into an authoritative, reviewable cross-framework route, handles unknowns and conflicts, and triggers reassessment after change.

Routing answers a narrower question than assessment:

> Given the approved facts about this AI system, which assessment frameworks require work, which require a scope review, and which can presently be treated as outside the route?

It does **not** decide that a control is satisfied, that a legal obligation has been met, or that a system is safe. It creates the defensible starting scope for those decisions.

The two records used

Gamut combines:

- The **System Record**, which is the standing description of the system, its ownership, technology, lifecycle, deployment, data and operating context.
- The **Use Case Intake**, which records why and how the system is being used, the people and decisions it affects, its action capability, jurisdictions and assessment-specific risk facts.

The records are linked by the system reference. Similar names are not treated as proof that two records describe the same system.

Where the same standing fact appears in both records, the System Record is treated as the primary record and a disagreement is surfaced for review. A conflict is not silently resolved in favour of the more convenient route.

Authoritative cross-framework routing facts

The structured routing section in Intake captures facts that materially change applicability:

Fact group	What to record	Why it matters
Decision impact	Advisory, operational, consequential, or legally significant	Helps identify impact-assessment, legal and assurance depth
Action capability	Read-only, recommend, write, execute, or administer	Identifies agentic and privileged-action exposure
Model origin	Internal, third-party, open-source/open-weight, or hybrid	Informs supplier, provider and lifecycle responsibilities
Organisation roles	For example provider, deployer, importer or downstream provider	Legal and value-chain duties depend on role
Capability flags	Privileged access, training activity and supplier dependency	Routes controls for access, change, data and third parties
Human-impact flags	Vulnerable populations, biometrics and emotion recognition	Triggers enhanced legal, rights and impact screening
Service and content flags	Critical service, synthetic content and deepfakes	Routes resilience, transparency and content obligations
Jurisdictions	Markets, users, affected people and output destinations	Establishes territorial and regulatory nexus

Use **Unknown** when the answer has not been established. Do not select "No" merely because evidence has not yet been obtained.

Information needed for a reliable route

A reliable route normally requires:

- A stable link to the System Record.
- System name, purpose and use-case boundary.
- Model type or AI technique.
- Autonomy and human oversight.
- Deployment context.
- Users and affected people.
- Data sources and classification.
- Relevant geographies.
- Decision impact and action capability.

Missing information remains visible as a routing-quality gap.

Routing states

Each framework receives one of four states:

State	Meaning	Assessor action
Applicable	Current facts positively trigger the framework or it is a universal baseline	Include it in the assessment plan
Screening required	Applicability cannot safely be ruled out because a material fact is unknown or needs specialist review	Complete the missing facts or scope review; do not treat it as excluded
Not applicable	Current facts provide a documented basis for leaving the framework outside the system route	Retain the basis and review it after change
Organisational review	Applicability is decided at organisation or management-system scope, not solely for one system	Resolve it through the appropriate organisational governance process

This is fail-closed scoping: uncertainty creates work; it does not create an exemption.

Overall routing status

The route itself carries a status:

Status	Meaning
Incomplete	Routing-critical information is missing
Conflict review	The linked records disagree on one or more material facts
Provisional	A route has been calculated but not confirmed by an accountable reviewer
Confirmed	The current basis has been reviewed and accepted
Reassessment required	Material facts changed after the previous confirmation

Confirmation applies only to the facts and route visible at that time.

What to do when reassessment is required

A route panel may remain green because the current facts are complete and conflict-free while also showing an amber reassessment warning. These messages answer different questions:

- **Green route panel:** the current route is calculable and has no missing-fact or conflict blocker.
- **Reassessment required:** the basis changed after the previous confirmation, so accountable review is required.

Review the changed System Record and Intake facts, reconsider the ACRS dimensions, inspect any changed framework routes and control scope, then approve the current ACRS position or adjust it where the evidence requires. Revisit affected assessments and reconfirm only when the current basis is understood.

How each framework is treated

Framework	Routing approach
GTSAF	Universal framework baseline; complete system facts determine per-control applicability, while ACRS and governance weighting adjust assurance depth
ACRS	Universal capability-risk baseline for the system; informs proportionate assurance depth
NIST AI RMF	Available as a universal risk-management baseline; context determines prioritisation
ISO/IEC 42001	Organisation-level AIMS scope decision informed by the AI portfolio
ISO/IEC 42005	Triggered by actual or foreseeable impacts, affected parties and sensitive uses; incomplete impact facts require screening
EU AI Act	Routed from territorial nexus, role, use and regulated characteristics; the module then performs authoritative legal classification
ATF	Routed when the system can act, use tools or resources, orchestrate work, or exercise delegated authority
NAGF	Routed when markets, users, affected people, outputs, operations or regulated roles establish a Nigerian nexus
MAESTRO	Routed when architecture, model, data, tools, orchestration, ecosystem or human interaction creates layer-specific exposure

Framework-level routing is deliberately separate from item-level applicability. Being routed to a framework does not mean every item within it applies.

Confirmation and change

Review the route before confirming it. Confirm that:

- The linked System Record is correct.
- Unknowns have been investigated or deliberately left for screening.
- Conflicts are resolved.
- The stated organisation roles are accurate.
- Jurisdictions cover deployment, users, affected people and outputs.
- Decision impact and action capability reflect real operation.
- The required assessment plan is proportionate.

When a routing-critical fact changes, Gamut marks the previous route for reassessment. Examples include:

- A system moves from recommendation to execution.
- A new market or affected population is introduced.
- A new supplier, model or training activity is added.
- Biometric, emotion-recognition or synthetic-content functionality is enabled.
- The system enters a critical service or consequential decision process.
- The organisation's legal role changes.

Reconfirm the route only after the changed basis and its assessment consequences have been reviewed.

Effect on assessment and scoring

Routing determines what work is presented and how it is prioritised. It does not award a favourable score.

- Unknown applicability is not counted as a pass.
- A route does not satisfy a requirement.
- A mapped control does not automatically satisfy another framework.
- GTSAF conditional controls are included or excluded from complete, conflict-free system facts; incomplete facts remain pending.
- ACRS and governance weighting may increase required assurance depth but cannot invent or remove factual control applicability.
- Legal modules perform their own detailed scope, role and requirement decisions after routing.

Understanding the GTSAF route

GTSAF distinguishes:

- **Applicability breadth:** the controls factually relevant to the selected system.
- **Assurance depth:** Baseline, Triggered or Enhanced rigour for each applicable control.
- **Assessment result:** what the answers, evidence and tests demonstrate.

The applicable count is not a risk ranking. A system using an external component may have more supplier controls than a higher-impact internal system, while the higher-impact system still requires much deeper assurance. GTSAF therefore also shows depth-weighted workload and normalised assurance intensity. See GTSAF system scope, applicability and assurance depth.

AI Assist and routing facts

In full-context mode, AI Assist may consider the selected system's current route, routing facts, quality gaps and conflicts together with the authorised assessment record.

In **Privacy Mode**, direct identifiers and free-text routing facts are excluded. The model receives only the minimum structured route, applicability signals, completeness state and assessment values needed for bounded assistance. Privacy Mode reduces disclosure; it does not change the route.

AI Assist cannot confirm a route, approve an exclusion, alter applicability or accept risk.

Reviewer checklist

- [ ] System Record and Intake are linked to the same system.
- [ ] Purpose and boundary are specific.
- [ ] Decision impact and action capability describe production reality.
- [ ] Organisation roles are complete.
- [ ] All four jurisdiction perspectives have been considered.
- [ ] Unknowns are not disguised as negative answers.
- [ ] Conflicts have been resolved.

- [ ] Screening-required frameworks have an owner and next action.
- [ ] Organisational decisions are not misrepresented as system-level exclusions.
- [ ] Route confirmation reflects the current basis.
- [ ] Material changes trigger reassessment.

Next

24. Evidence & findings

Capture governance evidence as work happens through request-based workflows, track its quality, and manage findings through root cause, remediation and validated closure.

Evidence and findings are where governance stops being a claim and becomes something you can prove. Gamut treats both as first-class objects, captured as work happens rather than gathered in a rush at audit time.

Evidence

Evidence is any artefact that supports a governance claim, a document, export, log, screenshot or record. In Gamut, evidence is managed as a request-based workflow rather than a folder of files. An **evidence request** carries:

- A **request title** and description, tied to a specific control.
- The **expected evidence type**, so the owner knows what will satisfy it.
- A named **owner** and **due date**.
- A **status** as it moves from `requested` to fulfilled.
- A **quality rating** and **review notes**, so weak or missing evidence is visible rather than assumed.

Linking evidence to controls is what creates traceability: a reviewer can move from a control to the evidence behind it without hunting through shared drives. And because requests are tied to controls and owners, the evidence base builds continuously as governance work happens, instead of being chased at the end of a cycle.

Findings

A **finding** records a gap, deficiency or exception identified during assessment, control testing or audit. Each finding is a structured record:

- A **title**, **description** and **finding category**.
- A **severity**: `low`, `medium`, `high` or `critical`.
- A **root cause** and a **recommendation**.
- A **management response**, a named **owner** and a **target date**.
- A **status** as it moves toward closure, with **closure notes**, a **validated by** and a **validated date** so closure is verified, not just asserted.
- Links to the **control test** and **evidence request** it arose from.

Recording findings honestly is more valuable than an unsupported pass: it gives leadership a true picture and gives reviewers confidence that governance is real.

Remediation

Remediation is the closure path for a finding. Keeping remediation visible, with an owner, a target date and a validated closure, turns findings from a list of problems into a managed improvement programme, and gives the Improve stage of the lifecycle something concrete to track over time.

Traceability

Evidence and findings complete the chain that makes Gamut defensible:

```
control -> control test -> evidence request -> finding -> remediation -> validated closure
```

Any conclusion in a report can be followed back through this chain to the underlying records, and every change along it is captured in the audit log.

Setting up evidence, tests and findings

1. From a scored control in an assessment, **request evidence**: set the control reference, what is needed, an owner and a due date. Requests track from open to satisfied.
2. **Attach evidence** to the request, or record it directly against the control.
3. **Run a control test** where you need to prove a control operates: capture the objective, procedure, sample method and size, the design and operating effectiveness scores, the result, and any exceptions.
4. Where there is a gap, **raise a finding**: severity, root cause, recommendation and management response, linked back to the control, test and system.
5. Convert a finding into a remediation item and track it to closure.

Owners and due dates make evidence and remediation overdue-trackable, and every step is captured in the audit log.

Next

25. System selection in assessments

Understand which systems appear in framework selectors, why a selection may be unavailable and how system-scoped results are kept separate.

Framework assessments are **system-scoped**. The system selector determines whose intake facts, routing decision, scores, evidence and AI Assist output the module loads. Selecting the wrong system can produce a well-formed but irrelevant assessment, so always verify the scope banner before scoring.

What selector states mean

State	Meaning	Action
Selectable	The system has the minimum confirmed records required by that module.	Select it and verify the displayed name and scope.
Greyed out	The system exists but a prerequisite, route or confirmation is incomplete.	Follow the displayed reason back to intake, ACRS or routing.
Not listed	The record is outside the current workspace, unavailable to the user or not eligible for the module.	Verify workspace, role, entitlement and system status.
All controls / framework overview	No system-specific record is loaded.	Use for catalogue review only; do not mistake it for a completed system assessment.

Some frameworks have additional legitimate prerequisites. ACRS confirmation can be required before GTSAF depth is final. ISO/IEC modules require an applicable licence confirmation. ATF is assessed per agent rather than as a generic system catalogue. Legal frameworks may require their authoritative scope gate before a conclusion can be confirmed.

Safe workflow

1. Complete and save the AI System Record.
2. Complete intake facts and resolve routing-incomplete prompts.
3. Review and, where required, approve the ACRS score.
4. Open the routed assessment from the plan or use the framework selector.
5. Confirm that the system name, scope and assessment status have changed.
6. Score, add evidence and run AI Assist only after that confirmation.

If the selector returns to the framework overview, do not continue entering system evidence. Refresh the routing plan and verify the prerequisite records. Results already saved for another system remain associated with that system; they should not be copied merely to make the new assessment look complete.

AI Assist and system scope

AI Assist output is stored against the selected system or agent and assessment item. Switching scope must not present the previous system's output as current. The output header should identify the analysed system. If it does not, stop and re-establish scope before relying on the analysis.

Troubleshooting checklist

- Correct client, workspace and system selected.
- Intake save shows a successful state.
- Required authoritative facts are resolved.
- ACRS is confirmed where the route requires it.
- The framework appears in the server-authoritative plan.
- The user's role and plan both permit the framework.
- The record is active and has not been retired.

See Routing & applicability for how prerequisites affect the assessment plan without changing subscription entitlements.

26. Reporting & exports

Produce board-level and workpaper-grade outputs from Gamut through eight governed pack types, each composed of the right sections and carrying run-level traceability metadata.

Reporting turns the records in Gamut into outputs people act on, a board view of AI governance maturity, and workpaper-grade exports for audit and assurance. Because reports are generated from the connected records, every conclusion can be traced back to the evidence behind it.

What reports show

A report draws the lifecycle together into a single, practical view:

- **AI inventory**: the systems in scope, their owners and lifecycle status.
- **Risk**: the classification picture across systems.
- **Framework scores**: per-system conformance, compliance or maturity, rolled up to the workspace using the same coverage-inclusive scoring as the source assessment at generation time.
- **Evidence quality**: how well governance claims are supported.
- **Findings**: open deficiencies and exceptions.
- **Remediation progress**: what is being fixed and how far along it is.
- **Readiness priorities**: where to focus next.

Pack types

Rather than one generic report, Gamut produces purpose-built **packs**, each composed of a different set of sections for a different audience:

Pack	Built for
Full assessment	**The complete picture:** every section, for a deep review.
Framework focus	A single framework in depth (for example GTSAF, EU AI Act, ATF).
Executive	**Leadership:** dashboard, estate, framework, evidence, findings, decisions, roadmap.
Board pack	**Board oversight:** adds estate, agentic and gateway posture to the executive view.
Board summary	A concise board-level summary.
Evidence pack	**Evidence-centred:** controls, evidence, findings and supporting workflow.
Control testing	The control-testing record for assurance work.

Pack	Built for
Agentic snapshot	**The agentic posture:** agents, ATF, gateway decisions and runtime evidence.

Each pack selects only the sections relevant to its purpose, so a board pack is not a 200-page audit file and an evidence pack is not a one-page summary.

Generating a report

1. Open **Reporting** in the workspace whose records you want to report on.
2. Choose a **pack type** for your audience (full assessment, framework focus, executive, board pack, evidence pack, control testing, or agentic snapshot).
3. Set the **framework scope** (all frameworks or a single one) and, optionally, enable the **scores-only AI narrative**.
4. **Generate** the pack. Each run gets a run ID and traceability metadata.
5. **Export** it (subject to the `export.report` / `export.workpaper` permissions) for boards, clients or audit.

Because reports are built from connected records, regenerate one whenever the underlying assessment, evidence or findings change rather than maintaining a separate document.

Two audiences, one source

Reporting serves two audiences at once, from the same records:

- **Boards and leadership** get a concise picture of AI governance maturity, exposure and the decisions required of them, without engineering detail.
- **Auditors and reviewers** get traceable, workpaper-grade outputs that connect each conclusion back to its underlying evidence.

Because both views draw on the same connected records, they can be reconciled to the same source state and generation time. Different pack purposes may summarise that state differently, and any pack becomes historical when the source records subsequently change.

Traceability metadata

Every generated report carries run-level metadata, a run ID, the generation timestamp, the assessment, the framework scope and the pack type, so any export can be located, reproduced and defended. A report is not an anonymous PDF; it is a dated, identified artefact tied to the records it was built from.

Optional AI narrative

A report can optionally include an AI-generated narrative summary. To protect confidentiality, that narrative is generated from **assessment statistics only**, scores, counts, framework scope and posture figures, and never from free-text governance content. No confidential record text is sent to the model, and the generation is proxied server-side so model-provider keys are never exposed. See AI assistance & data handling.

Exports

Reports are produced as governed HTML and can be exported for distribution and record-keeping, for board packs, client assurance, internal audit and external review, subject to the `export.report` and `export.workpaper` permissions. Because the underlying records are connected, an export reflects the traceable source state at its recorded generation time. Regenerate it after material assessment, evidence, finding or remediation changes.

From point-in-time to continuous

Traditional governance produces a report once a year. Because Gamut keeps records connected and current, reporting becomes something you run whenever you need it, supporting the Improve stage with a view of progress over time, not just a single snapshot.

Next

27. Policy generation

Draft AI governance policies, standards, procedures and guidelines in Gamut with AI assistance, grounded in your systems, risk and frameworks, under a draft-to-publish lifecycle and usage quotas.

Gamut can help you **draft AI governance documents** with AI assistance, turning the governance work you have already done into written policy, rather than starting from a blank page.

A library of document types

Policy generation is not a single template. It spans the document set a real AI governance programme needs, across four kinds of artefact:

- **Policies**: the overarching AI governance policy (purpose, scope, principles, roles, structure, risk approach, enforcement, review cadence), an AI acceptable-use policy (permitted and prohibited uses, bring-your-own-AI and shadow-AI controls), and jurisdiction-specific policies such as a NITDA-aligned Nigeria AI governance policy.
- **Standards**: data classification, privacy, risk and transparency standards.
- **Procedures**: risk and impact assessment, incident response, change, lifecycle, monitoring, bias, data handling, vendor and decommissioning procedures, and NITDA audit procedures.
- **Guidelines**: ethics, human-oversight and training guidelines.

Each is drafted against a structured outline so the output is consistent and complete, not a blank prompt.

Grounded in your governance

Generated documents are grounded in the context Gamut already holds, your AI systems, risk tiers and the frameworks you assess against, so policy reflects your actual estate rather than a generic template. The output is a **starting draft** to review and adapt, not a finished policy. Human ownership and sign-off remain essential; generation removes the blank-page problem and keeps policy consistent with the rest of your governance.

Draft to publish

Generated documents follow a lifecycle governed by RBAC:

- `policy.generate` produces a draft.
- `policy.publish` promotes a reviewed draft to a published document.
- `policy.delete` removes one.

So drafting, reviewing and publishing are separate, accountable steps, and a generated draft is never a published policy until a person with the right permission promotes it.

AI handling and quotas

As with all AI features in Gamut, policy generation is proxied server-side: model provider keys are never exposed to the browser (see Security & data handling). It is gated by the `policy_generation` entitlement and metered by daily and monthly generation quotas that depend on your plan & entitlements. When the entitlement is off, the quota is zero.

Generating a policy document

1. Confirm the workspace has the records to ground the draft (system, intake, risk and assessment work). The more complete these are, the more specific the draft.
2. Open **Policy generation** and pick a **document type** from the library.
3. **Generate** the draft. It is produced server-side from your governance records; provider keys are never exposed to the browser, and the call counts against the plan's AI quota.
4. **Review and own** the draft: it is a starting point for a human to finalise, never an auto-published artefact.
5. Publish or store the finished document through your normal process.

Generation requires the `ai_chat` entitlement and AI permission; if the entitlement is off, the quota is zero. See plans & entitlements.

Why generate policy inside Gamut

- **Consistency.** Generated documents reflect the same frameworks and risk model your assessments use, so policy and practice stay aligned.
- **Speed.** Drafting accelerates from days to minutes, leaving more time for review and tailoring.
- **Traceability.** Policy connects to the governance records it governs, rather than living as a detached document.

Next

28. Model cards

Document the technical and ethical characteristics of the models behind your AI systems, intended use, data, oversight, autonomy and risk, linked to the system that uses them.

A **model card** documents a model's technical and ethical characteristics in one place. Where an AI system record describes a use of AI, a model card describes the model itself, and the two are linked.

What a model card captures

A model card is created against a workspace and linked to a system. It records:

- **Model name** and the **linked system** it belongs to.
- **Model type** and **provider**.
- **Deployment environment**.
- **Purpose**: what the model is for.
- **Data sources**: the data it relies on, at an appropriate level.
- **Human oversight** model and **autonomy level**.
- **Risk tier**: the model's own risk classification.

Because a model card can be created directly from a system record, its fields are pre-filled from the system, the model type, provider, environment, purpose, data sources, oversight and autonomy flow through, so the card stays consistent with the system that uses it rather than drifting.

Creating a model card

1. Open **Model cards** and create a card, linking it to the AI system that uses the model.
2. Record the model identity and type, intended use and limitations, and the model-level characteristics relevant to impact.
3. Set the card's status and keep it current as the system's intake context changes.
4. Reference the card from ISO/IEC 42005 impact assessment and from reports.

Why model cards matter

Model cards give reviewers and decision-makers a consistent, comparable view of the models the organisation depends on. They support:

- **Assessment**: grounding control conclusions in documented model characteristics.
- **Reporting**: giving boards and auditors a clear view of model-level risk.
- **Procurement**: comparing vendor models on a like-for-like basis.

Relationship to assessments and frameworks

Model cards complement assessments: the assessment evaluates a system against controls, while the model card documents the model that system uses. Several frameworks, including ISO/IEC 42005 and the EU AI Act, expect model and system documentation of this kind, and model cards help you meet those expectations. Model-level detail can also be pulled into reports.

Next

29. Governance weighting profile

The deterministic policy profile that drives risk tiering in Gamut, six weighted governance dimensions, configurable thresholds and floor rules, owned and calibrated by accountable stakeholders.

The **governance weighting profile** is the policy that turns intake signals into a risk tier. It is what makes risk tiering **deterministic and explainable**: the same inputs always produce the same tier, and the reasoning can be traced to a profile you control rather than a black box.

The six dimensions

A profile scores a system across six weighted governance dimensions. The defaults are:

Dimension	Default weight
Business criticality	0.22
Autonomy level	0.14
Human impact	weighted
Regulatory sensitivity	weighted
Data sensitivity	weighted
External exposure	weighted

Each dimension carries a weight, and the weighted combination produces a governance score for the system. Weights are configurable, so an organisation can express its own risk appetite, for example weighting regulatory sensitivity more heavily in a regulated sector.

Thresholds and tiers

The score is mapped to a tier through configurable thresholds. The defaults are:

- **Moderate** at 45
- **High** at 65
- **Critical** at 80

Below the moderate threshold a system is treated as lower risk. Raising or lowering a threshold changes how readily systems are classified into each tier, again, a deliberate policy choice, not a hidden constant.

Floor rules

Profiles support **floor rules** (enabled by default): hard minimums that force a system to at least a given tier when a specific high-risk signal is present, regardless of the weighted score. This prevents a genuinely sensitive system from scoring its way under the radar because other dimensions are low.

A profile is policy, and is owned

A weighting profile is treated as governance policy in its own right. It carries a name, an owner, an approver, a review date and a status. The shipped default is explicitly marked as "current policy until formally approved or calibrated by accountable stakeholders", the intent is that an organisation reviews and signs off its own profile rather than silently inheriting the default.

Configuring the profile

The weighting profile is a workspace setting you own and calibrate:

1. Open the **governance weighting profile** settings for the workspace.
2. **Set the dimension weights** to reflect your risk appetite (for example weight regulatory sensitivity higher in a regulated sector). What matters is the relative emphasis.
3. **Set the tier thresholds**, the score at which a system becomes Moderate, High or Critical.
4. **Enable or adjust the floor rules** that force a minimum tier when a specific high-risk signal is present.
5. **Record the policy metadata**: a profile name, an accountable **owner** and **approver**, a **review date** and a **status**. Until your own profile is approved, the shipped default applies and is marked as interim policy.

Because the profile is deterministic, recalibrating it re-tiers every system consistently from the same inputs. Save the profile, then re-run intake to apply it.

Effect on applicability and assurance depth

The weighting profile affects:

- Governance tier.
- Required assurance depth.
- Review cadence.
- Escalation and approval expectations.

It does **not** add or remove a control solely because a weighted score is higher or lower. Factual control applicability remains grounded in the System Record and Use Case Intake: architecture, data, suppliers, users, impacts, jurisdictions and operating characteristics.

This separation prevents policy calibration from rewriting the system's factual control boundary. In GTSAF, a higher tier may move an already applicable control to Triggered or Enhanced depth without making an unrelated conditional control applicable.

Why determinism matters

Because tiering runs through an owned, weighted, threshold-based profile:

- **Decisions are consistent.** Two systems with the same profile inputs get the same tier.
- **Decisions are defensible.** A reviewer can see exactly which dimensions and thresholds drove a classification.
- **Policy is adjustable in one place.** Changing risk appetite means recalibrating the profile, not re-judging systems case by case.

Next

30. Risk Register

Record, own, treat and monitor AI risks from intake, assessment, discovery, findings and runtime governance.

The **Risk Register** turns identified exposure into an owned management decision. A framework score describes assessment posture; a risk record describes an uncertain event or condition, its impact, the treatment decision and who is accountable for it.

Sources of risk

Create or link risks from:

- intake and risk-tiering decisions;
- framework gaps or failed control tests;
- findings and evidence deficiencies;
- Discovery alerts and unregistered AI;
- supplier, model, data or regulatory change;
- agentic simulations, runtime denials and incidents;
- manual management or audit review.

Avoid one generic "AI risk" covering unrelated causes. A useful statement connects a cause, event and consequence, for example: "Because customer prompts may contain special-category data, sending them to an unapproved provider could cause unlawful disclosure and customer harm."

Minimum defensible record

Field	Purpose
Title and statement	Identifies the risk and causal scenario.

Field	Purpose
Linked system	Establishes scope and ownership boundary.
Category and source	Explains where the risk arose.
Inherent exposure	Records exposure before the claimed treatment.
Existing controls	Identifies what is already reducing likelihood or impact.
Residual exposure	Records the post-control judgement and its basis.
Treatment	Avoid, reduce, transfer or accept, with actions and evidence.
Owner and due date	Assigns accountable action and review timing.
Status	Tracks open, treating, accepted, closed or superseded state.
Review trigger	Defines when the judgement must be revisited.

Operating workflow

1. Link the risk to the correct system and source record.
2. Write a specific cause-event-consequence statement.
3. Assess exposure using the organisation's approved method.
4. Identify controls and evidence actually operating, not planned controls.
5. Select treatment, assign an owner and set dates.
6. Link remediation work, tests and findings.
7. Obtain the required acceptance or escalation decision.
8. Review after material change, failed treatment, incident or due date.

Closing a finding does not automatically eliminate its underlying risk. Close the risk only when the treatment and residual decision are supported, or mark it superseded when a replacement record preserves the history.

Governance Weighting Profile

The Governance Weighting Profile can change how the organisation prioritises review, escalation and assurance. It does not rewrite the factual system scope or erase a risk. Document any management judgement applied above the calculated result.

Quality checks

- The risk is system-specific and understandable without oral explanation.
- Inherent and residual judgements are not confused.
- Controls have linked evidence or tests.
- Acceptance is within the approver's authority and time-bounded where appropriate.
- Overdue actions and high residual risks are escalated.
- Changes in model, data, supplier, autonomy, geography or law trigger review.

Next

31. Remediation Roadmap

Track the closure of findings as sequenced, owned remediation items with priorities, target dates and overdue tracking, the managed improvement programme behind the Improve stage.

The **Remediation Roadmap** is where findings become a managed programme of work rather than a list of problems. Each gap raised during assessment, control testing or audit becomes a tracked remediation item with an owner, a priority and a target date, and the roadmap shows what is in progress, what is done and what is overdue.

What a remediation item holds

Field	Purpose
Title	What is being fixed.
Owner	Who is accountable for the fix.
Control reference & framework	The control and framework the item relates to.
Priority	How urgent the work is.
Status	Where the work has reached (through to done).
Target date	When it is due.
Description & notes	The detail and running commentary.

Remediation items link back to the findings they close, completing the traceable chain from a control gap to the work that resolves it.

Overdue tracking

The roadmap watches target dates. Any item that is not done and is past its target date is flagged as **overdue**, and the count surfaces as a warning in the interface so slippage is visible rather than buried. The roadmap also shows progress at a glance as a done-over-total ratio.

Where it sits in the lifecycle

Remediation is the closure path for findings and the concrete substance of the Improve stage of the governance lifecycle:

```
control -> control test -> evidence -> finding -> remediation item -> done
```

Keeping remediation visible, owned and dated turns governance from a point-in-time assessment into a continuous improvement programme, and gives reporting a real view of progress over time, not just a snapshot of open gaps.

The AI Consultant can help plan it

The AI Consultant's **Remediation Planning** mode can draft sequenced remediation actions with owners, dependencies and closure tests, grounded in the findings already recorded. As always, the output is a drafting accelerator for a well-prepared record, not a substitute for accountable sign-off.

Next

32. AI Consultant

An in-platform advisory tool grounded in your live assessment records, with task-specific modes for governance, risk, evidence, policy, reporting, agentic review, board summary and remediation planning.

The **AI Consultant** is an in-platform advisory tool that analyses and drafts from your **live assessment records**, the system context, framework scores, evidence entries, risk items and findings already recorded. It draws on what is in the workspace, not on general knowledge alone, so its output is specific to the governance situation in front of it.

Grounded, not general

Every consultation reads the current assessment at the moment of the query. That grounding is both the Consultant's strength and its limit:

- When the record is detailed, scored controls with notes, evidence linked to controls, named risk owners, a specific system description, the Consultant produces directly relevant analysis.
- When the record is thin, the output is generic, because there is nothing specific to draw from.

The practical consequence: the Consultant cannot compensate for an incomplete assessment. The screen shows a **Workspace context included** manifest with the current workspace and record counts for systems, intake, risks, evidence, tests, findings and agentic governance. Use that manifest to check what can ground the answer. A zero count or unavailable module is a limitation, even when the answer reads fluently. Treat the output as a drafting accelerator for well-prepared records, not as authoritative governance and not as a substitute for the work that produces the record.

Task-specific modes

The Consultant offers focused modes, each oriented to a specific governance activity. Matching the mode to the question produces sharper output than a general query:

Mode	What it produces
Governance Review	Ownership, accountability, approval gaps and operating-model weaknesses.
Risk Review	Risk statements, exposure summaries and treatment options from the assessment data.
Evidence Review	What evidence is present, what is missing, and which requests would strengthen assurance.
Policy Support	Generates or improves policy wording from the governance record.
Report Support	Report-ready narrative.
Agentic AI Review	**Scoped to agentic governance:** agent controls, approval gates, ATF readiness and Gateway records.
Board Summary	The assessment distilled into executive priorities and decisions needed.
Remediation Planning	Sequenced actions with owners, dependencies and closure tests.
General Advice	Open-ended questions, broader output than the task-specific modes.

The selected **output type** controls the intended artefact, for example Advisory Note, Risk Summary, Evidence Gap Review, Agentic AI Control Review, Remediation Plan, Board Briefing, Policy Improvement Note, Report Narrative or Executive Summary. Mode controls the analytical lens; output type controls the form of the draft.

Context and privacy boundary

The Consultant uses permitted workspace records, notes and other free text so that it can answer workspace-specific questions. The framework **scores-only privacy mode does not narrow Consultant context**. If scores-only mode is enabled, the Consultant displays a warning explaining this difference.

Before sending a question:

1. Check the **Workspace context included** manifest.
2. Confirm that the selected workspace is the intended one.
3. Do not paste secrets, credentials or unnecessary personal data into the question.
4. Do not use the Consultant if the permitted workspace context should not be sent to the configured model provider; use the relevant framework's scores-only AI Assist instead.
5. Review the provider and organisational data-handling terms that govern your deployment.

How it is governed

The Consultant is held to the same controls as every other AI feature in Gamut:

- **Server-side only.** Prompts and responses are proxied server-side; model provider keys are never exposed to the browser. See AI assistance & data handling.
- **Entitlement and permission gated.** It requires both the `ai_chat` entitlement and the AI Consultant permission, an Advanced-tier capability. See Plans & entitlements and Users & roles.
- **Usage-metered.** AI calls count against the plan's daily and monthly quotas.
- **Tenant-scoped.** It only ever sees the records of the workspace you are in.
- **Visible context boundary.** The screen identifies the categories and record counts available to the consultation before the question is sent.

Get the most from it

- **Complete the assessment first.** A half-finished assessment leaves the Consultant blind to large parts of the picture, and the output will sound more complete than it is.
- **Pick the right mode.** "What are the ownership gaps here?" belongs in Governance Review, not General Advice.
- **Treat output as a draft.** Review and own everything it produces before it becomes a governance artefact.

Next

33. Audit and assurance workflows

Operate Evidence Tracker, Testing Centre, Findings Register, Workpaper Packs, Board Dashboard and the Audit Trail as one traceable assurance chain.

Gamut separates six assurance records so evidence, testing, deficiencies, reporting and system activity are not collapsed into a single score.

Area	Primary question
Evidence Tracker	What proof has been requested, received, reviewed and linked?
Testing Centre	Did the control operate as designed for the sampled period and scope?
Findings Register	What deficiency, exception or uncertainty remains?
Workpaper Packs	Can another reviewer reproduce the assurance conclusion?
Board Dashboard	What exposure, trend and decision requires leadership attention?
Audit Trail	Who changed which governed record, and when?

Evidence Tracker

Define the exact claim, framework item, system, owner, requested artifact and period. Record source, version, receipt date, reviewer and sufficiency decision. An attachment alone is not sufficient: explain what it proves, what it does not prove and whether it is current.

Reject or qualify evidence that is generic, self-authored without corroboration, outside the assessment period, scoped to another system or contradicted by tests.

Testing Centre

A control test should identify objective, population, sample, procedure, expected result, actual result, exceptions and reviewer. Distinguish design adequacy from operating effectiveness. A policy can demonstrate design but usually cannot prove repeated operation.

Failed or inconclusive tests should create or update a finding and, where material, a risk. Retests must preserve the original result and show the corrective action assessed.

Findings Register

State the condition, expected criterion, cause, consequence, affected scope and supporting evidence. Assign severity, owner, due date and status. Management response, remediation and closure evidence remain separate decisions. Closing a task is not the same as validating the finding.

Workpaper Packs

Select the correct system, framework, period and purpose. Before export, confirm scope, evidence links, reviewer sign-off, open limitations and generation metadata. Packs are point-in-time artifacts; regenerate after material source changes.

Board Dashboard

Use it to communicate exposure, trends, overdue high-severity work, material exceptions and decisions. Avoid presenting coverage or maturity as legal compliance. Drill into source records before acting on an aggregate.

Audit Trail

Use the Audit Trail to investigate state changes and decision history. It supports accountability but does not prove the substantive correctness of the decision. Restrict access, export only what is needed and preserve retention requirements.

End-to-end closure

1. Assessment identifies a claim or gap.
2. Evidence is requested and reviewed.

3 A test evaluates operation.

4 Exceptions become findings and risks where appropriate.

5 Remediation is completed and retested.

6 A reviewer closes or qualifies the finding.

7 Workpapers and board reporting preserve the relevant conclusion and limitations.

Next

34. Assessment history and delta tracking

Establish baselines, compare assessment states and explain what changed without overwriting the original evidence or conclusion.

Assessment History, **Assessment Baselines & Delta Tracking** and **Automated Gap Narratives** answer a different question from the live assessment: **what changed since the selected baseline, why did it change and does the earlier conclusion remain defensible?**

Assessment History

Use history to locate prior assessment states, dates, systems, frameworks and conclusions. Confirm that two records refer to the same system boundary and comparable framework version before drawing a trend. A higher score can reflect real improvement, increased assessment completion or changed applicability; these are not the same outcome.

Assessment Baselines & Delta Tracking

A baseline should have a clear purpose, such as pre-deployment approval, annual assurance, post-incident state or certification-readiness review. Record its scope, cutoff date, framework version, assessor and known limitations.

Do not move the baseline merely to improve the reported trend. Create a new baseline when a major system boundary or methodology change makes direct comparison misleading.

Delta interpretation

Review changes in:

- system scope, purpose, data, supplier, geography or autonomy;
- routing and control applicability;
- assessment status and assurance depth;
- evidence freshness and test results;
- findings, accepted risks and remediation;
- human conclusions and review dates.

Automated Gap Narratives

An Automated Gap Narrative can summarise recorded changes, but it cannot determine whether the business accepted the change, whether evidence is credible or whether a legal conclusion is valid. Require assessor review and retain material caveats.

Defensible comparison checklist

- [ ] Same system identity or an explained boundary change.
- [ ] Comparable framework and methodology versions.
- [ ] Applicability changes separated from control improvement.
- [ ] Completion changes separated from assurance changes.
- [ ] Evidence period and freshness considered.
- [ ] Material regressions have an owner and treatment.

- [ ] The new conclusion records the decision basis and reassessment trigger.

Next

35. Cross-framework analysis and mappings

Interpret Cross-Framework View and framework mapping screens as traceability aids rather than automatic equivalence or compliance.

Cross-framework analysis shows where one recorded control, outcome or evidence item can support several governance views. It reduces duplicate work, but it does not make unlike requirements equivalent.

Available views

Gamut provides **Cross-Framework View**, **EU AI Act Mapping**, **NIST RMF Mapping**, **ISO 42001 Mapping**, **ISO 42005 Mapping** and **NAGF Compliance** views. Each uses the records available to the current workspace and system.

Use a mapping to answer:

- Which existing controls may support this requirement?
- Which evidence can be reused without changing its meaning?
- Where does the target framework require additional scope, legal analysis or assurance?
- Which gaps are unique to the target framework?

What a mapping does not prove

A mapped relationship does not by itself prove implementation, operating effectiveness, legal applicability, conformity or certification readiness. Frameworks differ in subject, terminology, unit of assessment, required evidence and decision authority.

For example, a GTSAF security control may support an EU AI Act risk-management obligation, but the legal role, applicable article, quality-management context and conformity evidence still require separate assessment.

Review workflow

1. Select the correct system and confirm its authoritative route.
2. Review the source item's score, evidence and assurance depth.
3. Read the target requirement in its own framework context.
4. Decide whether the evidence is reusable, partially supportive or irrelevant.
5. Record any target-specific evidence, gap or conclusion.
6. Revisit the mapping after scope, framework version or evidence changes.

Good reporting language

Use "mapped support", "potentially reusable evidence" or "traceability coverage". Avoid "compliant through inheritance", "automatically satisfied" or "certified by mapping".

Next

36. Certification and assurance readiness

Use readiness views to prepare governance records without presenting them as certification, legal advice or an external assurance opinion.

Cert Readiness (certification and assurance readiness) brings together scope, assessment coverage, evidence, tests, findings and management conclusions to show what still needs attention before an external review.

Readiness is an internal preparation view. It is not a certificate, conformity decision, legal opinion or guarantee of audit outcome.

Readiness dimensions

- **Scope:** the management system, AI system, sites, functions and exclusions are clear.
- **Coverage:** required clauses, controls or outcomes have been assessed.
- **Implementation:** documented controls are operating in the assessed environment.
- **Evidence:** artifacts are current, attributable and linked to the claim.
- **Testing:** material controls have appropriate design and effectiveness testing.
- **Findings:** gaps have owners, dates, treatment and closure evidence.
- **Governance:** conclusions, approvals, risk acceptance and review triggers are recorded.

Workflow

1. Confirm the target standard, scheme, period and scope.
2. Review the applicable framework module and any licence requirements.
3. Resolve routing and applicability before interpreting completion.
4. Review evidence quality and tests, not scores alone.
5. Identify open major issues, exclusions and dependencies.
6. Generate a readiness pack for internal review.
7. Agree corrective work and retest before external assurance.

Interpreting the result

A high percentage with weak evidence is not strong readiness. A lower result caused by newly identified scope can represent better governance than a narrow, superficially complete assessment. Explain the denominator, assessment period, assurance depth and unresolved limitations.

For ISO/IEC work, use a licensed copy of the relevant standard and an appropriately competent certification or assurance provider. Gamut does not reproduce controlled standard text or make the certification decision.

Next

37. Global Search

Find governed records across the current workspace while preserving tenant, role and entitlement boundaries.

Global Search helps locate systems, assessments and related governance records in the current workspace. Search results are constrained by the signed-in user's tenant, workspace access, role and entitlements. Search does not grant access to the underlying record.

Search effectively

- Use a stable system ID, control ID, owner, supplier or distinctive title where available.
- Open the result and verify system, workspace, framework and record status.
- Treat snippets as navigation aids, not the complete evidence or decision.
- Refine ambiguous searches rather than assuming the first result is correct.

Security and privacy

Do not use search to enumerate records beyond a legitimate work purpose. Results may contain sensitive governance metadata even when the underlying evidence is separately restricted. Report unexpected cross-workspace visibility immediately and do not investigate another tenant's data.

If an expected record is absent, verify the selected workspace, your role, module entitlement, record status and spelling. Administrators should correct access at the source rather than copying content into a less restricted area.

Next

Part 4: Agentic and runtime administration

Administer agents, identities, connectors, permissions, runtime policies, approvals, Gateway, Claw and operational evidence.

38. Agent Launchpad

Bring an agent into governance through a controlled sequence from registration and assessment to simulation and bounded operation.

Agent Launchpad guides an agent from proposed use to governed readiness. It coordinates the required records but does not override an incomplete gate.

Recommended sequence

1. Define the agent boundary, purpose, prohibited purposes and owners.
2. Register the agent and runtime identity.
3. Complete system intake, ACRS, ATF and relevant MAESTRO assessment.
4. Establish data movement, tools, connections and least-privilege permissions.
5. Draft and independently approve exact Runtime Access Policies.
6. Test simulations, approval, denial, timeout, retry and containment paths.
7. Resolve critical findings and accept residual risk through authorised governance.
8. Enable only the intended environment and monitor the first operating period closely.

Readiness does not equal authority

Launchpad progress indicates that governance work has been completed or remains outstanding. Live execution still requires Gateway to validate authenticated identity, connection, permission, policy, approval and request context at runtime.

Reassessment

Return to Launchpad after changes to purpose, owner, model, tools, data, supplier, environment, autonomy, delegation or risk. Suspend authority first when the change invalidates the trusted boundary.

Next

39. Agentic CISO

Agentic CISO is where Gamut governs agentic AI, the agent register, ATF assessment, tool and data governance, approvals, red-team evidence and runtime trajectory, the system of record that Gateway enforces and Claw executes against.

Agentic CISO is the governance home for agentic AI inside Gamut AI. It is the **system of record**: it owns the truth about which agents exist, what they are allowed to do, who is accountable for them, and what they have actually done. Gateway enforces that truth at runtime, and Claw (or a BYO runtime) executes the work. Nothing an agent does at runtime is valid unless it traces back to a record held here.

The separation of duties

```
Agentic CISO = governance system of record (this page)
Gamut Gateway = runtime policy decision and enforcement
Gamut Claw = secure execution layer
```

Agentic CISO never executes agent work and never holds provider credentials. Its job is to define and hold the governance state that Gateway reads on every single agent action. This is what keeps Gamut a trust plane rather than just another agent orchestrator.

What Agentic CISO holds

Governing an agent across its lifecycle takes more than an inventory entry. Agentic CISO brings together a connected set of records, each one a control surface that Gateway can enforce against:

Record	What it governs
Agent register	The inventory of agents, their owners, purpose, ATF level, risk tier and lifecycle status.
Org nodes & identity profiles	Where an agent sits in the organisation, the escalation path for high-impact actions, and the identity it acts under.
Runtime Access Policies (Access Matrix)	Exact, versioned authority for an agent, governed connection, action, resource, data class, environment, purpose and time boundary.
Tool permissions	The business authorisation binding an agent to the specific tools it may use.
Data flows	What classified data an agent may process, and how it must be handled.
Approval gates	The human approval required before sensitive or mutating actions run.
ATF assessments	The trust and autonomy assessment for each agent.
Incident playbooks	The pre-agreed response for prompt injection, rogue-agent and unauthorised-action events.
Red-team tests	Adversarial test scenarios and their results, evidence that the agent was challenged.
Gateway simulations	Recorded "what would Gateway decide" runs against a policy snapshot.
Memory & checkpoints	Governed agent memory and resumable state, kept inside the trust boundary.
Trajectory events	The stream of what agents actually did at runtime, fed back from Gateway.

Every one of these is workspace-scoped and access-controlled. Reading, changing, approving and simulating agentic governance are separate permissions, further constrained by the workspace plan. See users & roles and plans & entitlements.

The agent register

The agent register is the precondition for everything else. **No unregistered agent may take any autonomous action**. Gateway blocks it outright with a critical severity. A register entry carries, at minimum:

- A **human owner** and a **security owner**, both named. Gateway fails the action if either is missing.
- An **ATF current level** (L1 Intern through L4 Principal), which sets the agent's autonomy boundary.
- A **risk tier** (low through critical) and capability flags such as can spend money and can access customer data, which raise the bar on what controls Gateway demands.
- A **lifecycle status**. A `suspended` agent is blocked from all actions until it is reactivated through formal change control.

Whether an agent runs on Claw or an external runtime, it is registered here first. Registration is what makes a runtime identity issuable at all.

The two layers of tool authorisation

Agentic CISO and Gateway divide tool authorisation cleanly, and an agent may use a tool **only when both layers agree**:

- **Tool permissions in Agentic CISO** are the business authorisation layer: a deliberate decision that this agent may use this tool, with an approved capability scope, an audit-logging requirement, and an approval requirement for critical-risk tools.
- **Connector registration in Gateway** is the technical capability layer: the governed adapter that actually performs the call, holding the credential and the endpoint policy.

If a tool is permitted here but no connector exists, the action cannot run. If a connector exists but the tool is not permitted here, Gateway blocks it. Capability and authorisation are deliberately kept on different sides of the boundary.

Setting up an agent

Bringing an agent under governance is a short, ordered process. Each step is workspace-scoped and audited, and the Workflow Studio wizard will deep-link you to whichever of these screens still needs attention.

1. **Register the agent** in the Agent Register: give it a name, a **named human owner** and **security owner**, its purpose and **risk tier**, and its capability flags (can it spend money, take external action, reach customer data). Gateway refuses any action by an unregistered agent, or one missing an owner.
2. **Set the ATF level** (L1 Intern to L4 Principal), the autonomy ceiling Gateway enforces. Start low and promote through the ATF gates.
3. **Grant Tool Permissions** for each tool the agent needs: bind the agent to the tool adapter, set the allowed capability scope and data classes, **enable audit logging**, and require approval for critical-risk tools. A tool permission that is inactive, not runtime-enabled, or missing audit logging blocks the agent at preflight and at Gateway.
4. **Author Runtime Access Policies** for each material authority the agent needs. Bind the agent to the governed adapter and target, exact actions, resource and data boundaries, environment, purpose, dates, logging, monitoring and approval requirements.
5. **Validate and submit each policy**. Review structural validity, breadth, overlap and potential conflicts before freezing the version for independent approval.
6. **Independently approve runtime authority** through a workspace-scoped Runtime Policy Approver. Draft, pending, rejected, suspended, revoked and expired policies cannot permit runtime actions.
7. **Map the agent into the Org Chart** so high-impact actions have an escalation path back to the CEO, orchestrator, security owner or other accountable node. This proves accountability only: org chart placement does not grant inherited permissions.
8. **Configure the matching connector** so the tool can actually execute (the technical capability layer). Both layers must agree.
9. **Define approval gates** for any external, financial or code-modification action that needs human sign-off.
10. **Record data flows** for any classified data the agent processes.
11. **Simulate** the agent's intended action (see below) and resolve any control gaps before going live.

See Runtime Access Policies for the complete field model, lifecycle, approval workflow and troubleshooting guidance.

Runtime policy lifecycle

Runtime Access Policies move through controlled states:

```
Draft -> Pending approval -> Active
↑ ↓
└──── Rejected

Active -> Suspended or Revoked
Active -> Expired or Superseded
```

Submitted and authoritative versions are immutable. A change to active authority is proposed as a new draft and must pass validation and independent approval again. In a locked workspace, policies remain visible for audit and inspection while authoring and lifecycle changes remain disabled.

The **Validate & impact** view helps assess policy completeness, broad authority, overlap and potential allow/deny conflicts. Validation confirms that the policy can be reviewed; it does not replace human risk judgement.

How ATF level bounds autonomy

The ATF level on the register entry is not a label, it is an enforced ceiling. Gateway applies a different action boundary at each level:

Level	Boundary enforced at runtime
L1 Intern	Read, observe and report only. Any write or external action is blocked.
L2 Junior	Routine actions allowed; every external or financial action requires an approval gate.
L3 Senior	Operates within guardrails; financial and high-risk external actions require approval.
L4 Principal	Strategic autonomy; critical or top-secret data access still requires documented approval.

Raising an agent's autonomy is therefore a governed decision made here, with immediate runtime consequences. See ATF for the full level model and promotion gates.

Runtime evidence and accountability

Because every agent action flows through Gateway, Agentic CISO receives a continuous stream of **trajectory events**: what was requested, which controls passed or failed, what Gateway decided, and what happened. Combined with the journal that Claw keeps for each task, this turns agent activity from an opaque black box into the same reviewable, auditable evidence and findings model the rest of Gamut uses. Open risks or findings linked to an agent are surfaced back into the Gateway decision, so an agent with unresolved critical findings cannot quietly keep operating.

Gateway decisions also carry bounded policy evidence back into Gamut: the matched policy and version, access decision, org escalation path, matched approval gate, controls passed, controls failed and decision path. The Gateway Event Centre shows these events without exposing raw payloads or secrets.

Simulating before you authorise

Before an agent goes live, you can run a **Gateway simulation** from Agentic CISO: choose an agent (or a transient template agent), a tool, an action type, a data classification, a target system and an environment, and Gateway returns the full decision it would make, the controls passed, the controls failed, the missing evidence, the decision path, the runtime-policy match and the org escalation path, against a frozen policy snapshot with a 90-day validity and explicit retest triggers. This lets you close control gaps before any real action is ever attempted.

Break-glass: suspending an agent or halting the runtime

Governance includes the ability to stop. Gamut offers graduated controls, all audited:

- **Suspend one agent.** Set the agent's lifecycle status to `suspended` in the register, or use **Suspend Agent Runtime** in the admin console. Gateway then blocks every action by that agent, fail-closed, until it is reactivated through change control.
- **Stop one running step.** Use the **Stop** button in Workflow Studio, which cooperatively cancels that step's Claw task at the next safe checkpoint.
- **Halt all agentic execution (kill switch).** An administrator can flip the **Gateway kill switch** in the admin console (Gateway Status, Disable Gateway Runtime). Gateway then blocks every agent action, and Gamut itself fail-closes every dispatch and skips scheduled runs, so nothing executes until it is cleared. A separate Claw kill switch, set on the Claw service, halts the execution worker entirely. See Gateway.

Next

40. Agent identity, ownership and organisation

Establish accountable agent identities, human ownership and escalation relationships before granting runtime authority.

Agentic governance starts with a stable identity. An agent name or job title is not an identity and does not grant authority. The Agent Register, Identity & Ownership and Organisation Chart establish who or what the agent is, who is accountable and where escalation goes.

Agent Register

Create one record for each independently governed agent boundary. Record purpose, owner, operating status, runtime, model, environment, data access, tools, autonomy, criticality and review dates. Separate agents when they have materially different authority, owners, runtime identities or failure consequences.

An active register entry is necessary but not sufficient for execution. Tool Permissions, a ready connection, an active Runtime Access Policy and any required approval must also pass.

Identity & Ownership

Bind the governance record to the runtime identity used in authenticated requests. Record the business owner, technical owner and accountable executive or service owner as appropriate. Owners remain responsible for purpose, risk, access review, incidents and retirement; the agent cannot own its own risk acceptance.

Rotate or revoke runtime identity material through the approved service workflow. Never place raw credentials, service secrets or private keys in the Agent Register.

Organisation Chart

The chart explains reporting, delegation and escalation. It can help determine who must be consulted or approve a high-impact action, but position in the chart does not create runtime permission. Gateway evaluates exact policy and request context.

Lifecycle controls

1. Register the agent and owners.
2. Complete risk tiering and ATF assessment.
3. Configure least-privilege tools and governed connections.
4. Validate simulations and approval paths.
5. Activate only the authority required for the intended environment.
6. Review after change, incident, expiry or owner transfer.
7. Suspend identity and runtime authority before retirement.

Review checklist

- [ ] Unique agent boundary and runtime identity.
- [ ] Named, current human owners.
- [ ] Purpose and prohibited purposes defined.
- [ ] Environment and data boundaries recorded.
- [ ] Escalation path is usable but not treated as permission.
- [ ] Dormant, duplicate and retired identities are disabled.

Next

41. Runtime Access Policies

Define, validate, approve and maintain exact, versioned authority for AI-agent actions, with least privilege, independent review and execution-time enforcement through Gamut Gateway.

A **Runtime Access Policy** defines the exact authority an AI agent may exercise through Gamut Gateway. It answers a narrow question:

> May this agent perform this action, through this governed connection, on this resource, with this class of data, in this environment, for this purpose, at this time?

The answer is never inferred from an agent's title, organisational position or general access to Gamut. Missing or ambiguous authority is denied.

Runtime Access Policies are managed from **Agentic CISO -> Access Matrix**. The navigation label reflects the relationship between agents and resources. The records themselves are exact, versioned policies with a controlled lifecycle.

Where a Runtime Access Policy fits

A policy is one part of a defence-in-depth decision. It does not make an action permissible on its own.

Control	What it proves
Agent register	The agent has an accountable identity, owners and an active lifecycle state.
Gateway tool registration	The requested adapter is a governed capability known to Gateway.
Governed connection	The exact tenant and workspace connection is configured and ready.
Tool Permission	This agent is allowed to request the selected capability.

Control	What it proves
Runtime Access Policy	The requested action, resource, data, environment and purpose are within an active policy.
Approval gate	A qualifying sensitive action has the required human decision.
Runtime controls	Logging, monitoring, current policy validity and request-bound authority remain satisfied when the action executes.

Every applicable control must pass. A broad Tool Permission cannot compensate for a missing Runtime Access Policy, and an active policy cannot compensate for a suspended agent or unavailable connection.

Policy fields

Identity and effect

- **Policy name**: a human-readable description of the authority being governed.
- **Agent**: the registered agent to which the policy applies.
- **Effect**: `Allow` or `Deny`. Deny policies take precedence when an allow and deny boundary intersect.
- **Risk rating**: the assessed risk of the governed authority.

Capability and target

- **Gateway adapter**: selected from the agent's active Tool Permissions.
- **Governed target / connection**: the exact ready connection for that adapter. Where Gamut provides a governed internal service, that service can appear as the target instead.
- **Resource boundary**: the specific object, collection or path the action may affect. A bounded trailing prefix can cover a defined collection. An unrestricted wildcard is treated as privileged authority and requires stronger justification and review.

  The governed-target list is not free text. It is derived from the selected adapter and the connections that are available to the current tenant and workspace.

Action and information boundaries

- **Exact actions**: the operations the policy covers, such as read, retrieve, create, update, execute, send a message, export data or invoke a model.
- **Allowed data classifications**: the highest and specific information classes the action may handle.
- **Environments**: development, test, staging or production.
- **Workflow boundary**: an optional exact workflow to which the authority is restricted.
- **Purpose boundary**: an optional business-purpose restriction.

  Choose only what the use case needs. If an agent requires unrelated actions or targets, create separate policies so each authority can be reviewed, suspended and replaced independently.

Assurance and time boundaries

- **Human approval**: whether execution also requires an approval gate.
- **Gateway logging**: mandatory for governed runtime authority.
- **Continuous monitoring**: whether ongoing runtime monitoring is required.
- **Approver role**: the accountable role expected where approval applies.
- **Effective date**: when the authority may begin.
- **Expiry date**: when the authority stops being valid.
- **Review due date**: when the business and security basis must be reconsidered.
- **Business and security justification**: why the authority is necessary, proportionate and time-bounded.

  Dates have different purposes. Expiry controls runtime validity. Review due prompts governance attention. Reaching a review date does not silently extend an expired policy.

Policy lifecycle

Status	Meaning	Gateway treatment
Draft	The author is defining and validating the policy.	Not authoritative and not used to permit actions.
Pending approval	The submitted version is awaiting independent review.	Not active and not used to permit actions.
Active	The approved version is within its effective and expiry dates.	Eligible for exact runtime matching.
Rejected	The reviewer returned the proposal with a reason.	Not active. The author may revise and resubmit it.
Suspended	Previously active authority has been paused.	Immediately unavailable for new actions.
Revoked	Authority has been withdrawn.	Permanently unavailable.
Expired	The policy passed its expiry date.	Unavailable without a newly approved version.
Superseded	A controlled replacement became authoritative.	The replacement version is used; the prior version remains historical.

Submitted and authoritative versions are immutable. To change active authority, clone the policy, edit the new draft, validate it and send that new version through approval. This preserves a clear history of what was authorised at any point in time.

Authoring workflow

1. Establish the prerequisites

Before creating a policy:

1. Register the agent and assign its accountable owners.
2. Confirm the agent's lifecycle and ATF level.
3. Register or enable the required Gateway adapter.
4. Configure and test the governed connection where one is required.
5. Grant the agent the matching Tool Permission.
6. Define an approval gate for actions that require human approval.

If the adapter or target does not appear in the builder, resolve the corresponding Tool Permission or connection first. Do not replace a missing governed target with a descriptive name.

2. Define the smallest useful boundary

Select one agent, one governed capability and a precise target. Add only the required actions, resource scope, information classifications and environments. Record the purpose and the reason the authority is proportionate.

3. Validate and review impact

Use **Validate & impact** before submission. Validation checks structural completeness and the relationship between the proposed policy and its governed context. Impact analysis helps identify:

- missing prerequisites;
- broad or privileged boundaries;
- overlapping active or pending policies;
- potentially contradictory allow and deny coverage;
- other active policies that could be affected.

Structural validity means the policy is complete enough to review. It does not mean the business risk is acceptable or that the policy is approved.

4. Save and submit

Save the proposal as a draft while it is being developed. When the content and impact are ready, submit it for independent review. Submission freezes that version and moves it to **Pending approval**.

5. Independently review

The reviewer should confirm:

- the agent, tool and target are correct;
- the action and resource boundaries match the stated need;
- the data classifications and environments are no broader than necessary;
- approval, logging and monitoring requirements match the risk;
- effective, expiry and review dates are appropriate;
- the justification explains necessity and proportionality;
- overlapping or conflicting policies have been resolved;
- the reviewer is independent of the creator and submitter.

The reviewer may approve and publish the policy or reject it with a reason.

6. Operate and reassess

An active policy remains subject to Gateway's complete runtime decision chain. Reassess it when the agent, connection, tool, resource, purpose, data, environment, risk, approval requirement or organisational ownership changes.

Independent approval

Runtime authority can affect live systems, so approval uses a dedicated **Runtime Policy Approver** role. The role is assigned to the exact workspace containing the policies that person may review.

An approver can:

- inspect policy details and relevant supporting context;
- filter the approval inbox;
- approve a pending policy;
- reject a pending policy with a recorded reason.

An approver cannot use that role to draft, edit, delete, submit, suspend or revoke policies, administer users, operate Gateway or change agent governance records. Approval and rejection both require the approver's own password and current MFA code through a short-lived step-up verification. The creator or submitter cannot approve the same policy.

See Users & roles for assignment and access details.

Execution-time validity

Approval is necessary, but it is not treated as permanently valid. When Gateway is ready to execute an allowed action, it checks that the authority still matches the request and that the governance state has not changed.

The execution is refused if, for example:

- the policy was suspended, revoked, expired or replaced;
- the policy content or version no longer matches the authorised decision;
- the agent, tool, connection, action, resource, data class, environment, workflow or purpose no longer matches;
- a required approval no longer applies to the exact request;
- the request-bound authority is missing, expired, replayed or inconsistent;
- required logging or monitoring cannot be satisfied.

This closes the gap between a decision being made and an action being executed. See Gamut Gateway for the complete runtime decision model.

View-only workspaces

When a workspace is locked in **View only** mode, existing policies remain available for inspection. Drafting and lifecycle changes are disabled. This supports audit and review without allowing the governance record to be altered.

Troubleshooting

The governed target list is empty

Confirm that the selected agent has an active Tool Permission for the adapter and that the required tenant and workspace connection is configured and ready.

A submitted policy is not visible to an approver

Confirm that:

1. the person has the dedicated Runtime Policy Approver account role;
2. that account has the Runtime Policy Approver workspace role for the exact workspace;
3. the policy is in that workspace;
4. the workspace assignment has not expired;
5. the policy status is **Pending approval**.

MFA is required to make the decision, not to list the assigned policy.

Approval asks for identity verification

The approver must enrol MFA, then complete the password and authenticator step-up. After successful verification, return to the policy and confirm the decision.

A previously allowed action is blocked

Review the current agent, connection, Tool Permission, policy, approval gate and runtime status. Gateway intentionally refuses to rely on an earlier decision when the current context no longer matches.

Worked example

A security triage agent needs to create a finding in Gamut from a verified SIEM alert.

The policy could specify:

- **Agent**: Security Alert Triage Agent
- **Adapter**: Gamut finding creation
- **Target**: the governed Gamut service
- **Resource**: the designated findings collection
- **Action**: create
- **Data**: internal
- **Environment**: production
- **Purpose**: convert verified security alerts into governed findings
- **Logging and monitoring**: required
- **Expiry and review**: time-bounded to the operational approval

The policy does not allow the agent to delete findings, modify users, change permissions or act on another workspace. Any additional authority requires its own explicit governance decision.

Next

42. Gamut Gateway

Gamut Gateway is the policy decision and enforcement engine for agentic AI. Every agent action passes through it, is checked against a sequence of governance controls, and is allowed, approval-gated, degraded or blocked, with a signed decision and a complete audit record.

Gamut Gateway is the policy decision and enforcement engine of the agentic stack. It is the **ATF runtime**: every agent action passes through Gateway, which evaluates it against the full governance context, decides, performs the call only if permitted, and records the decision either way.

What Gateway does

Gateway sits between agents and the models, tools and data they want to use. Nothing reaches a tool directly. For each requested action it assembles the governance context, agent register, Runtime Access Policies, org chart, tool permissions, approval gates, data flows, incident playbooks, red-team tests, linked risks and findings, and runs a sequence of controls before allowing anything.

The decision model

Gateway resolves every request to one of four runtime decisions:

Decision	Meaning
allow	All controls passed. The action proceeds, and is logged.
require_approval	A high-impact action with control gaps. A named human must approve first.
degrade	The action proceeds in a reduced, safer form.
block	The action is refused.

For high-risk or external actions that pass, the decision becomes **allow and log** with mandatory detailed audit logging. Where critical control failures coincide with an open critical finding, the decision escalates to **block and open incident**.

The controls Gateway evaluates

For each action, Gateway runs an ordered series of checks. Any of the first two short-circuit to an immediate block:

1. **Agent registration.** An unregistered agent is blocked at critical severity. No exceptions.
2. **Lifecycle status.** A `suspended` agent is blocked until formally reactivated.
3. **Human owner** is assigned.
4. **Security owner** is assigned.
5. **ATF maturity level** is assigned, and the action sits within that level's boundary (see below).
6. **Tool permission.** The tool is registered for this agent in Agentic CISO, with audit logging enabled, and human approval required if it is a critical-risk tool.
7. **Approval gate.** External, financial or code-modification actions are covered by an active approval gate that matches the action type, tool, target system, data class and environment.
8. **Runtime Access Policy.** Material actions must have a matching active, least-privilege policy for the agent, adapter, governed connection, action, resource, data class, environment and any workflow or purpose boundary. Missing or non-matching coverage fails closed. Draft, pending, rejected, suspended, revoked and expired policies never permit execution.
9. **Org escalation path.** High-impact actions must map back to an org-chart reporting path so escalation and accountability are clear. The org chart does not grant permission and does not create inherited access.
10. **Data movement.** Customer or regulated data (PII, confidential, restricted) has a documented data-flow record.
11. **Incident playbooks.** High-risk agents have playbooks for prompt injection, unauthorised external action and rogue-agent scenarios.
12. **Linked risks and findings.** Open risks or findings tied to the agent are surfaced; open critical findings can escalate the decision.
13. **Red-team validation.** High-risk or external actions have at least one passed red-team test on record.

Each check contributes to a **decision path** and, on failure, a **rule trigger**, so every decision is fully explainable: which controls passed, which failed, what evidence is missing, and exactly what would change the decision.

Gateway also records a bounded **policy trace**. The trace shows the runtime-policy match, the org escalation path, the matched approval gate and the decision path without storing raw payloads or secrets. The Gateway Event Centre surfaces this evidence for operators.

Runtime Access Policy and org chart enforcement

The **Access Matrix** screen manages Runtime Access Policies. It is an enforcement control, not just documentation. For a material action, Gateway checks for an active policy covering the requested agent, adapter, governed connection, action, resource, data, environment, workflow and purpose. If no policy covers the exact request, Gateway blocks it and records the gap. A matching deny policy takes precedence over an allow policy.

The org chart has a different purpose. It proves reporting, escalation and accountability. Gateway uses it to build an escalation path for high-impact decisions, but it never uses the org chart to grant access, inherit permissions or override Runtime Access Policies, tool permissions or approval gates.

ATF-level action boundaries

Gateway enforces the agent's ATF level as a hard ceiling on autonomy:

- **L1 Intern** may only read, observe and report. Writes and external actions are blocked.
- **L2 Junior** may take routine actions; every external or financial action requires an approval gate.
- **L3 Senior** operates within guardrails; financial and high-risk external actions require approval.
- **L4 Principal** has strategic autonomy; critical or top-secret data access still requires documented approval.

Why enforcement lives in one place

Concentrating enforcement in Gateway is what makes the zero-trust model work:

- **Credentials stay on Gateway.** Model provider keys and connector credentials live only on the Gateway side, never with the agent. A compromised agent cannot leak keys it never held.
- **Policy is consistent.** Every action is judged by the same engine, so governance does not depend on each agent behaving well.
- **Evidence is complete.** Because every action flows through one point, the runtime evidence fed back to Agentic CISO is comprehensive.

Request-bound, short-lived authorisation

Gateway does not just return a verdict. It issues **signed, time-bounded and request-bound authorisation**. The authority is tied to the tenant, workspace, agent, request, governed connection and the exact active policy version. Only an allowed and fully satisfied request can receive executable authority.

Immediately before execution, Gateway confirms that the policy and surrounding governance state are still current. The action is refused if the policy has changed, expired, been suspended, revoked or replaced, or if the request no longer matches the authorised identity, action, resource, connection or context. A required approval is bound to the same request and cannot be reused for unrelated work.

Execution authority is short-lived and replay-resistant. Repeated delivery of the same request does not create duplicate side effects, and concurrent execution attempts cannot both consume the same authority. This closes the time-of-check to time-of-use gap between deciding and acting.

Connectors and approval gates

Tools are exposed to agents as governed **connectors** registered in Gateway, model gateways, retrieval, HTTP and webhook adapters, ticketing, notification, storage, CRM and more, each with its own action type, risk tier, payload limits and response handling. See the connector catalog. Sensitive or mutating actions can require explicit human approval; those **approval gates** are defined as governance in Agentic CISO and enforced here.

Fail-closed by default

If Gateway cannot reach a dependency, cannot verify a signature, or encounters an error mid decision, it fails closed: the action does not proceed. Safety never depends on a call succeeding.

The kill switch (break-glass)

When you need to stop everything, an administrator can activate the **Gateway runtime kill switch** from the admin console (**Gateway Status, Disable Gateway Runtime**). It requires gateway-configuration permission and a step-up re-authentication, and it is audited. While it is active:

- **Gateway blocks every agent action** with a kill-switch decision, regardless of agent or policy.
- **Gamut itself fail-closes** every dispatch path, so a new Claw task, a workflow dispatch and a scheduled run are all refused before they reach the runtime. Scheduled runs are **skipped, not failed**, so they resume automatically once the switch is cleared.

This is defence in depth: the halt does not depend on any single service. A separate **Claw kill switch**, set on the Claw execution service, stops the worker entirely. Reactivating the Gateway runtime is itself a privileged, audited action. To halt a single agent or a single step instead, use agent suspension or the Workflow Studio **Stop** button (see Agentic CISO).

Next

43. Tool permissions and governed connections

Define the maximum tool authority an agent may request and bind runtime policies to exact Gateway-managed targets.

Tool Permissions define the outer boundary of what an agent may request. **Gateway Tooling** defines the governed adapters and connections capable of carrying out those requests. Neither one alone authorises execution.

Tool Permission

Grant the smallest action set, resource boundary, data classification and environment required for the agent's purpose. Separate read, search, create, update, delete, execute, external communication, data export, financial action, permission change and code modification. Broad wildcards require exceptional justification and stronger approval.

Review permissions after purpose, owner, supplier, model, environment or risk changes. Remove unused actions rather than relying on the agent not to call them.

Gateway Tooling

A governed target or connection comes from the Gateway tooling catalogue available to the tenant and agent. It identifies a configured adapter and exact target under Gateway custody. A free-text name cannot create a connection or bypass readiness.

Before use, confirm:

- the connector is approved and healthy;
- credentials are stored by the governed connection, not the agent;
- the target and environment are correct;
- egress, resource and data boundaries are explicit;
- monitoring and logging requirements can be met;
- the connection owner and review date are current.

Zero-trust intersection

Runtime authority is the intersection of authenticated agent identity, active connection, Tool Permission, exact Runtime Access Policy, required approval and live Gateway checks. A permissive record at one layer cannot compensate for a missing or narrower record at another.

Change and retirement

Changing a connection, credential, target, permission or environment can invalidate policy assumptions. Revalidate affected policies and simulations. Revoke or disable retired connections before removing descriptive records so runtime authority fails closed.

Next

44. Gamut Claw

Gamut Claw is the secure agent execution layer. It plans and reasons but acts only through Gateway-controlled paths, holds no credentials, runs governed task types under leases, redacts output and keeps a hash-chained journal of every step.

Gamut Claw is the secure execution layer of the agentic stack. It is where governed agent work actually runs. The defining rule is simple: **Claw may think, but it may only act through Gateway.** It never calls a model, invokes a tool, retrieves data or writes back a result directly.

Claw's role

An agent running on Claw can plan and reason locally, but whenever it needs to act it issues a request to Gateway, which decides, enforces policy, and performs the call. Claw receives back only bounded, governed output.

```
Claw requests work and executes only through Gateway-controlled paths.
Gateway applies and enforces policy, holds credentials, performs the call.
Gamut AI records what happened (trajectory + journal).
```

This keeps a strict separation between deciding to act (Gateway, on Gamut policy) and running the work (Claw). Claw is an ATF-controlled surface for the agents that Agentic CISO governs.

Why Claw holds no credentials

Claw never receives model, SaaS, database, email, payment or other provider credentials. Those live only on the Gateway side. The benefit is direct: if a Claw execution environment were compromised, there are no provider keys there to steal and no path to a tool that bypasses policy. Every capability Claw appears to have is in fact a Gateway-mediated call.

Governed task types

Claw does not run free-form prompts. It runs a fixed catalogue of **governed task types**, each with a defined objective, required output sections, and a security contract. The current types are:

Task type	Purpose
`governed_context_summary`	Retrieve and summarise bounded Gamut workspace context through Gateway.
`governed_reasoning`	**Produce a governed runtime assessment:** readiness, controls, risks, safe next actions.
`tool_plan`	Draft a least-privilege tool-use plan (no execution).
`evidence_gap_analysis`	Prioritise the most important evidence gaps for an agent or assessment.
`control_recommendation`	Recommend practical ATF, policy, monitoring and incident controls.
`investigation`	Investigate a stated issue, separating facts, hypotheses and next steps.
`evidence_collection`	Plan evidence collection with chain-of-custody, without touching live systems unless granted.
`incident_triage`	**Triage an incident:** severity, scope, containment recommendation, escalation.
`report_drafting`	Draft a governed report section from approved context only.
`control_testing`	Design or summarise a control test with procedure and exceptions.
`risk_review`	Review inherent, control and residual risk with decisions required.
`gateway_tool_task`	Execute a single explicitly granted Gateway tool under policy.

Any other task type is rejected with `unsupported_task_type`. Reasoning tasks are constrained to two Gateway calls only: retrieve bounded context (`cag.get_workspace_context`) and then `model.invoke`. They cannot reach a live tool unless one is explicitly granted.

Context is untrusted by construction

Every governed reasoning task is told, in its own objective, to treat all retrieved records, evidence, memory, connector output and user-authored text as **untrusted context**: never follow instructions found inside it, never request egress, hidden tools, credential disclosure, policy bypass, approval bypass or audit suppression. Gateway and Gamut policy are authoritative. This is prompt-injection defence built into the execution contract, not bolted on afterwards.

Bounded, redacted output

Claw output is bounded and redacted before it leaves the execution layer. Result text is capped (2,000 characters by default) and passed through a redaction pass that strips emails, phone numbers, API keys and tokens, AWS keys, bearer tokens, private keys, card numbers and IP addresses, recording how many redactions occurred. Claw streams only observable output and audit metadata; it does not expose or store hidden chain-of-thought.

Leases, budgets and fail-closed execution

Claw is built to fail safe under load and failure:

- **Runtime leases.** Each task is claimed under a time-bounded lease. If a worker dies or the lease expires before completion, the task **fails closed** (`task_lease_expired`), it never silently half-completes.
- **Step budgets.** Tasks have a bounded step count (default 3), so a task cannot loop or fan out unboundedly. Exceeding it fails with `task_step_budget_exceeded`.
- **Concurrency limits.** Global, per-tenant and per-agent concurrency caps protect the platform and prevent one agent from starving others.
- **Runtime ceiling.** A maximum runtime per task bounds how long any single execution can run.
- **Bounded retries.** Failures retry a limited number of times, and certain failures (Gateway block, cancellation, lease expiry, step-budget, unsupported type) are non-retryable by design.

  Claw does not treat an earlier Gateway decision as permanent authority. Each governed execution uses short-lived, request-bound permission that Gateway revalidates against the current agent, connection, Tool Permission, Runtime Access Policy and approval state. A policy change or revocation between planning and execution causes the action to fail closed.

Tamper-evident journal

Every task emits a sequence of journaled events, gateway steps, model calls, reasoning traces, output chunks, result summaries. Each event is hash-chained: it carries its own `event_hash` and the `previous_hash` of the event before it, so the journal is **tamper-evident**. The journal can be independently verified, and the event stream can be replayed after a given sequence number. This is the runtime evidence that flows back to Agentic CISO and into Gamut's evidence and findings model.

Scheduling

Claw tasks can be scheduled to run on a recurring basis. A schedule may only be created against a **passed Gamut preflight attestation**, and that attestation must still be valid when the task fires. If the attestation is missing or expired, the scheduled run is refused (`schedule_preflight_attestation_required` / `_expired`). Schedules can be paused and resumed, but they can never escape the same governance checks a live task faces.

Claw versus bring-your-own runtime

Claw is Gamut's own execution layer, but it is not the only option. If you already run agents on other frameworks, the BYO agent runtime lets them execute under the same model, **think anywhere, act through Gateway**, without adopting Claw. The governance and enforcement model is identical; only the execution surface differs.

Next

45. Bring-your-own runtime

Run external agent frameworks, LangGraph, CrewAI, Hermes, OpenClaw and custom code, while keeping Gamut as the trust and enforcement plane. Think anywhere, act through Gateway, via a signed SDK that fails closed.

The **BYO agent runtime** lets you run external agent frameworks while keeping Gamut as the trust and enforcement plane. You are not required to adopt Gamut Claw to benefit from governed agentic AI. Your agent keeps its own orchestration logic; Gamut governs what it is allowed to do.

The operating rule

> **Think anywhere. Act through Gateway.**

External agents may plan and reason locally, in whatever framework you prefer. But context, model calls, tools, connectors, approvals, run events and governed writeback must go through Gamut Gateway. Planning is unconstrained; action is always governed.

The SDK and its three governed actions

Gamut ships a BYO agent SDK (Node and Python) that an external runtime embeds. It exposes exactly three governed actions, and nothing else reaches a provider:

Action	What it does
context	Retrieve a bounded, tenant-scoped Gamut workspace context pack through Gateway.
model	Run a governed model inference over an approved context pack, with provider keys held by Gateway.
invokeTool	Invoke a named, permitted connector through Gateway, optionally carrying an approval warrant.

Each call is signed and tenant-scoped, and carries the agent's registered identity. The runtime receives only a single Gamut runtime secret. It is given **no** model-provider, SaaS, database, payment, email, SIEM, SOAR, shell or browser credentials, and the SDK actively rejects any attempt to pass direct provider configuration or secret material in a payload.

Configuration

An external runtime is wired with just its Gamut coordinates and one secret:

```
GAMUT_BASE_URL=https://your-gamut-host.example
GAMUT_TENANT_SLUG=main
GAMUT_WORKSPACE_ID=12
GAMUT_AGENT_ID=14
GAMUT_BYO_AGENT_SECRET=...
```

The `GAMUT_AGENT_ID` must correspond to an agent already registered in Agentic CISO. Registration is the precondition for the secret to authorise anything.

Fails closed, always

The SDK fails closed, the action does not proceed, on any of:

- a non-2xx response from Gamut,
- Gateway returning `invoked: false` (policy denial),
- a request timeout,
- a stale or revoked credential,
- an identity mismatch,
- any policy denial.

Safety never depends on a call succeeding. A failed governance check is a stop, not a warning.

Supported frameworks

The SDK ships bridges that keep popular frameworks inside the trust boundary, plus a minimal base client for custom Node or Python agents:

- **LangGraph** and **CrewAI** wrapper tools, so graph and crew steps act through Gateway.
- **Hermes Agent** chat, responses and toolset wrappers, backed by Gateway.
- **OpenClaw** run, stream, cancel, model and tool helpers, backed by Gateway.
- **Custom agents** via the base Node and Python clients.

These bridges must not receive model-provider keys, direct tool credentials or alternate provider base URLs; the SDK rejects direct provider configuration and continues to require signed Gamut runtime credentials for every governed action.

The security model

The BYO runtime preserves the same zero-trust posture as Claw:

- **No credentials to the agent.** Provider credentials remain on the Gateway side.
- **Registered identity.** Each external agent is registered in the Agentic CISO agent register, with an owner, scope and ATF level.
- **Governance parity.** A BYO agent passes the same Gateway controls, ATF boundary, Runtime Access Policies, org escalation evidence, tool permissions, approval gates and data flows as any other governed agent before any action.

- **Enforced at runtime.** Gateway enforces tool permissions, action types, data classes, rate limits, exact policy coverage and approval gates on every request, then revalidates the short-lived authority before execution.
- **Fully logged.** Gamut and Gateway record every action, giving the same runtime evidence as any other governed agent, including bounded policy trace evidence for Gateway Event Centre review.

Tool brokers

External tool brokers (such as MCP-based brokers) are supported only as governed Gateway tool connections, with declared profiles and explicit per-tool scopes. External agents never receive a broker credential or a direct broker URL; they request the tool through Gateway like any other connector. See the connector catalog.

Delegation

For frameworks that spawn sub-agents or delegate work, Gamut can issue short-lived, governed child identities so delegated work stays inside the same trust and enforcement model rather than escaping it. A delegated sub-agent is no less governed than its parent.

Next

46. Data movement and approval gates

Govern where agent data can move and when a human decision is required before an action can execute.

The **Data Movement** view documents permitted flows between an agent, governed connection, resource and destination. **Approval Gates** identify actions that require a bound human decision. Together they make authority understandable before Gateway evaluates a live request.

Data Movement

For each flow, document source, destination, purpose, data classification, affected people, geography, retention and permitted action. Include indirect movement through model providers, plugins, queues, logs and generated attachments.

Do not infer permission from technical connectivity. A reachable destination is not an approved destination. Restrict sensitive and regulated data to the minimum fields and period needed, and test redaction and egress controls.

Approval Gates

Require approval when the action is high-impact, irreversible, external, financially material, privilege-changing, broad, production-facing or otherwise outside routine bounded authority. Specify the approver role, decision context, expiry and conditions.

An approval must refer to the same agent, action, resource, target, environment and request context that will execute. Generic standing consent should not be used where transaction-level review is required.

Separation of duties

The requester or policy submitter cannot satisfy an independent approval requirement for the same authority. Runtime Policy Approvers can inspect assigned policy context and approve or reject; they cannot draft or edit policy through that role. MFA and step-up verification apply to sensitive decisions.

Review checklist

- [ ] Every material ingress and egress path is represented.
- [ ] Data classification and geographic restrictions are current.
- [ ] External recipients and subprocessors are identified.
- [ ] High-impact actions have an appropriate approval gate.
- [ ] Approval is request-bound and time-bounded.
- [ ] Denial, timeout, withdrawal and dependency failure stop execution.

Next

47. Connector catalog

How Gamut exposes broad tool access to agents through governed Gateway connectors, models, retrieval, HTTP, ticketing, notification, storage, CRM and more.

Connectors are how Gamut exposes broad tool access to agents **without weakening the zero-trust architecture**. Agents never call tools directly. Tools are registered, governed and invoked through Gamut Gateway after policy, permission, tenant, assessment and identity checks.

How a connector is governed

Each connector is a governed adapter with a defined contract, rather than a raw API call. A connector carries:

- A stable **tool name** (for example `siem.search_alerts`).
- An **action type**, the policy category, such as retrieve, report, ticket, notify or model.
- A default **risk tier** for runtime classification.
- A **credential reference** to a managed Gateway-side secret, never a raw secret in a workflow.
- An **endpoint policy**, allowlisted destinations and safety rules.
- **Payload and response policies**: accepted input shape and size, plus redaction and retention rules for outputs.
- An **audit policy**, the fields recorded for every decision and invocation.

Because credentials live on the Gateway side and destinations are allowlisted, agents get capability without ever holding keys or reaching arbitrary endpoints.

Built-in connector families

Gamut ships with connector families spanning common enterprise needs. Availability and enablement depend on your plan & entitlements and workspace configuration.

Family	Typical use
Model gateway	Model reasoning or generation (provider keys live only on Gateway).
Context (CAG)	Retrieve governed Gamut workspace context, tenant-scoped.
Retrieval (RAG)	Search approved knowledge stores with redaction policy.
MCP brokers	Broker approved MCP tools with explicit per-tool scopes.
Configurable HTTP	Wire customer REST APIs without code changes.
Webhook	Send bounded JSON events to approved destinations.
SIEM / SOAR	Security investigation context and incident workflows.
Ticketing & work	Create governed work items and agent tasks.
Document	Produce governed workpapers and summaries.
Database	Read approved operational data under strict query policy.
Gamut writeback	Materialise approved findings and evidence requests into Gamut.
Notification	Notify approved recipients by email or chat.
CRM	Read or update customer records under PII controls.
Productivity	Mail, calendar, drive and directory APIs, path-scoped.
Content & wiki	Retrieve or update approved knowledge stores.
Object storage	Read or write scoped object storage.
Research & market data	News, RSS and market research from allowlisted sources.
Finance	Finance or payment operations (critical risk, approval-gated).
Public channels	Publish approved content externally (approval-gated).

Two layers must agree

An agent can use a connector only when **both** authorisation layers align:

- **Tool permissions** in Agentic CISO, the business decision that this agent may use this tool.
- **Connector registration and policy** in Gateway, the technical capability, plus a policy that allows the requested action, context and payload.

Add the required approval gates and a valid scoped runtime token, and only then does the action proceed. This is the mechanism that lets Gamut offer broad reach safely.

Configuring a connector

Connectors are configured in the **connector catalogue** panel of the Workflow Studio launchpad. Configuring one is a governed, audited action that requires an editable session and the gateway-enforce permission. For each connection you provide:

1. The **adapter** to use (for example a model gateway, an MCP tool broker, or a configurable HTTP adapter) and, where relevant, the **provider**.
2. A **connection name**.
3. The **endpoint** (an HTTPS destination, which must fall within the adapter's allowlist policy).
4. The **credential** (an API token or OAuth secret). The credential is **encrypted in the browser to a Gateway public key** and handed to Gateway custody; Gamut never stores it in plaintext and an agent never holds it.
5. The explicit **scopes**, **allowed actions** and **allowed data classes** the connection may use.

Once saved, the credential is validated and the connector becomes available to permit to agents. A connection that has not passed validation, or whose credential is not in Gateway custody, surfaces as a runtime-readiness blocker in the Workflow Studio wizard. Tools that do not need an external credential (such as governed Gamut context retrieval) require no connector configuration.

Validated connections also populate the **Governed target / connection** selector when a Runtime Access Policy is authored for a compatible agent and adapter. Selecting from this catalogue binds the policy to an exact tenant-scoped connection; an arbitrary name cannot create a new runtime target or bypass connector readiness.

Next

48. Visual Workflow Studio

Design, govern, run and schedule agentic workflows in Gamut. The Plan Runtime Wizard walks you from an empty workspace to a dispatched, Gateway-enforced runtime and tells you exactly what to fix at each gate.

Visual Workflow Studio is where you design, run and schedule governed agentic workflows. You compose the steps an agent takes, and every action step remains governed by Gateway. It turns the zero-trust model into something you can build, review, run, watch and stop, before and while it executes.

The Studio opens as its own page (the **Workflow Studio** link in the app top bar). It needs an editable, Advanced-tier session with Agentic CISO access, and the platform must be running in its database mode.

The building blocks

A workflow is assembled from a small set of governed objects, each held in Agentic CISO and each carrying its own policy:

Object	What it is
Workflow plan	**The overall design:** a name, objective, root agent, and the ordered steps.
Workflow step	A single step with a task type. Action steps reference a governed connector by tool name.
Task type	What a step does, for example governed reasoning, tool plan, evidence collection, investigation, or a single Gateway tool call.
Toolset / connector	A governed connector adapter with allowed actions and data classes, Gateway-enforced, with approval required for high-risk actions.
Checkpoint	A saved, resumable state with a rollback policy, so long-running work can pause and resume inside the trust boundary.

The Plan Runtime Wizard

The fastest way to a working runtime is the **Plan Runtime Wizard**, a guided panel at the top of the Studio. It steps you through eight stages and, at each gate, tells you **exactly what to fix**, with a button that takes you straight to where the fix is made.

1. **Prerequisites.** Confirms the runtime is ready: Claw and Gateway configured and reachable, tool audit logging enabled, and a connector in proper credential custody. Each blocker shows a **Fix it** button (for example Open Tool Permissions) and a warning does not block you.
2. **Starting point.** Choose **Bootstrap the governed demo** (provisions a working agent, its tool permissions and a plan that already passes preflight, the fastest path to a first success), start from a **template**, or start **blank**.
3. **Define plan.** Set the plan name, objective, risk level, **root agent** (the orchestration identity), maximum parallel steps and timeout.
4. **Build steps.** Add steps from the task-type palette, set each step's agent, tool (for a Gateway tool call), dependencies and whether it needs approval. A **wave preview** shows the execution order and flags any dependency cycles.
5. **Preflight and fix.** Saving the plan runs the authoritative **server preflight**, which checks every step against exactly what Gateway will enforce at runtime. Anything missing is listed as a categorised checklist, each item with a **clickable fix** that opens the right Agentic CISO screen (with the specific agent pre-selected for per-step issues).
6. **Approve.** The governance sign-off, recorded in the audit trail. It attests the plan passed preflight under zero-trust enforcement.
7. **Dispatch and run.** Sends ready steps to the runtime. See Running a plan.
8. **Done.** A summary of the run, with the option to create another.

Every gate is enforced by the server, not just the wizard: you cannot reach a confirmed, dispatched runtime without passing the real preflight and approval.

Setting up and running a plan

Prerequisites

Before a plan can run, an operator and an administrator make sure the runtime is in place. The wizard's first step verifies all of this for you:

- The **Claw** execution service and the **Gateway** enforcement service are configured and reachable (an operations task, see Claw and Gateway).
- At least one **connector** is configured, with its credential held in Gateway custody (configure these in the Studio's connector catalogue panel).
- Every registered **agent** that a step uses has a named human owner, an assigned ATF level, and the required **Tool Permissions** active with **audit logging enabled**.
- Material action steps have matching **Runtime Access Policy** coverage and the agent is mapped into the **Org Chart** where high-impact escalation is required.
- For any step that needs human sign-off, an **approval gate** exists.

If something is missing, the wizard's checklist links you straight to the screen that fixes it.

Build the plan

1. Open **Workflow Studio** and select your workspace.
2. Use the wizard's **Bootstrap demo** for a guided first run, or pick a template, or build from scratch.
3. In **Define plan**, choose the **root agent** and set parallelism and timeout.
4. In **Build steps**, add each step: choose its **task type**, assign a **registered agent**, set a **tool** for Gateway tool calls, mark **dependencies** (which steps it runs after), and tick **requires approval** for sensitive steps.

Preflight and fix

Save the plan to run preflight. The readiness scorecard shows a score and which steps are ready. For anything not ready, the checklist tells you the precise problem and gives a **Fix it** button:

Blocker	Where the fix-it button takes you
Tool permission not active / not runtime-enabled / audit logging off	Tool Permissions (Agentic CISO), with that agent focused

Blocker	Where the fix-it button takes you
Runtime Access Policy coverage missing	Access Matrix (Agentic CISO), with that agent focused
Org escalation path missing	Org Chart (Agentic CISO), with that agent focused
Agent or root agent not registered	Agent Register
Approval gate missing	Approval Gates
ATF level	ATF assessment
Connector missing	the Studio's connector catalogue
Draft issues (step keys, dependencies)	the wizard's Build step

Fix, return, and re-check. The plan is ready when preflight passes.

Approve and dispatch

Approve the plan (the governance sign-off), then dispatch. Dispatch sends each **ready step** to the runtime under a signed runtime identity, and Gateway re-validates every action at dispatch, so a step can still be blocked or held for approval at runtime.

Running a plan

Steps run in **waves** by dependency: steps with no unmet dependencies run first, then the next wave once they complete.

- **Auto-run (default on):** the Studio automatically dispatches each next wave as the previous one completes, so the whole plan runs to completion on its own. Turn it off to step through manually, one wave at a time, with **Sync + continue**.
- **Approval-gated steps still wait for a human** on every run, and every Gateway decision still holds, so Auto-run never bypasses governance.
- A **live stream** shows each step's Gateway decisions, model calls and bounded output, backed by a hash-chained Claw journal.

Stopping a running step

A running step is backed by a Claw task. Each active step has a **Stop** button that cooperatively cancels it: Claw stops at the next safe checkpoint, marks the task cancelled and will not half-complete. The step is terminal, but you can re-dispatch the plan afterwards. Stopping is governed and audited.

For a broader halt, an administrator can use the **Gateway kill switch** (admin console, see Gateway), or suspend an individual agent in Agentic CISO.

Scheduling a plan

An **approved** plan can run on a recurring schedule. On the selected plan's **Schedule** panel:

1. Choose a **cadence** (hourly, daily, weekly, monthly or quarterly).
2. Click **Enable schedule**. Enabling is a sign-off and needs an approver-level permission plus an approved plan.
3. **Pause**, **Resume** or **Disable** the schedule at any time. The panel shows the next run and the last run with its outcome.

A schedule is never a standing bypass. On every scheduled run, Gamut:

- re-checks the schedule owner's live permission and the tenant entitlement (and pauses the schedule if either was revoked),
- re-runs the full **preflight**,
- **skips** the run if a previous run is still in flight (no overlap),
- **skips** while the kill switch is active (runs resume automatically once it is cleared),
- and dispatches through the same Gateway/Claw-enforced path, where per-step approvals still hold.

Every scheduled run is audited with its outcome.

Deleting a plan

Each saved plan in the sidebar has a delete control. Deleting removes the plan and its steps (with a confirmation) and is audited. Any Claw tasks already running are governed independently and are unaffected.

Governed by construction

A workflow built in Studio does not bypass the agentic stack. Each action step references a governed connector by tool name, and that call faces the same checks as any other: the tool must be permitted to the agent, Gateway must allow the action within the agent's ATF level boundary, a Runtime Access Policy must cover the action, the org chart must support high-impact escalation, and any approval gates must be satisfied. The design surface changes; the enforcement model does not.

Runtime backends

A workflow targets a registered **runtime backend**, either Gamut Claw or, for external frameworks, a BYO runtime. Backends are governed: they default to a **deny** network posture, **no raw secret mounts**, and **gateway-only egress**. In every case the rule holds: think anywhere, act through Gateway.

Next

49. Runtime control and live approvals

Monitor agents, inspect pending decisions and suspend or stop authority without weakening the fail-closed Gateway boundary.

Agent Runtime Control is the operational view of agent status, pending decisions and governed execution. It does not replace Gateway enforcement. Use it to observe, approve where authorised, and contain operation.

Before enabling operation

- Agent identity and owners are current.
- Risk tier and ATF readiness support the intended autonomy.
- Required connectors are healthy and least-privileged.
- Tool Permissions and Runtime Access Policies are exact and active.
- Approval, logging, monitoring and expiry requirements are configured.
- Simulation and failure-path testing are complete.

Live approvals

Inspect the complete action context: agent, purpose, requested action, target, resource, data class, environment, risk, policy version, expiry and expected effect. Approve only when the request matches the authorised business purpose and remains necessary at decision time.

Approval is not execution. Immediately before action, Gateway revalidates policy, approval, connection, identity and request context. Changed, stale, replayed, expired or already-consumed authority is rejected.

Suspension and containment

Suspend an agent or revoke policy when identity, purpose, owner, connection, evidence or control assumptions are no longer trustworthy. Use the narrowest control that safely contains the issue, but do not delay a broader stop when impact is uncertain.

Record reason, decision owner, affected work, communications and recovery conditions. Test that pending and retried requests cannot continue under revoked authority.

Monitoring

Review denials, unusual request volume, repeated approval prompts, policy mismatches, connector failures, timeouts and evidence gaps. A high denial rate may show malicious behaviour, poor agent planning or an incorrectly designed policy; investigate rather than automatically widening access.

Next

50. Agent risk and ATF Control Tower

Combine agent risk tiering, ATF readiness, tests, incidents and runtime evidence without treating one score as authority to operate.

Agent Risk Tiering prioritises governance based on capability, access, autonomy and potential impact. **ATF Control Tower** brings together per-agent maturity, control evidence and operational signals. Neither view grants runtime access.

Risk tiering

Assess the actual deployment, not the agent's intended personality or job title. Consider tool and data reach, independence, delegation, persistence, external communication, reversibility, privilege, scale and affected people. Record evidence and uncertainty for each judgement.

Use the resulting tier to set approval, testing, monitoring, review and escalation depth. Do not lower the tier merely because controls are planned; evaluate residual exposure after tested controls separately.

ATF Control Tower

Use the Control Tower to identify agents below target maturity, stale evidence, failed promotion gates, overdue reviews, incidents and runtime-control weaknesses. Drill into the agent and requirement before assigning remediation.

ATF readiness is per agent. Portfolio averages can hide one highly privileged unready agent. Promotion requires the evidence and elapsed operating experience specified by the assessment, plus human confirmation.

Decision use

Combine risk tier, ATF result, MAESTRO threats, runtime tests, incidents, open findings and policy scope. Record the human decision and reassessment trigger. A green dashboard is not permission to operate without the exact runtime authority chain.

Next

51. Incident response and red teaming

Prepare, test, contain and learn from agentic failures while preserving evidence and runtime control.

Agentic incident response must address both the harmful outcome and the authority path that allowed or attempted it. Red teaming tests those paths before an attacker, model failure or unsafe workflow does so in production.

Incident workflow

1. **Triage:** identify agent, action, target, environment, affected systems and current exposure.
2. **Contain:** suspend the agent, policy, connection or runtime path necessary to stop further harm.
3. **Preserve:** retain requests, decisions, approvals, runtime evidence, connector responses and relevant business records.
4. **Assess:** determine data, safety, financial, legal, customer and operational impact.
5. **Eradicate and recover:** correct policy, code, credentials, data or process; retest failure paths.
6. **Review:** record causes, control failures, lessons, residual risk and reassessment triggers.

Do not delete the evidence needed to understand the event. Coordinate privacy, legal, security, safety and regulatory notification decisions through authorised people.

Red Team

Define written scope, authorisation, environment, stop conditions and evidence handling before a test. Cover prompt injection, indirect instruction, identity spoofing, permission escalation, policy mismatch, approval manipulation, replay, data exfiltration, unsafe delegation, tool misuse, connector failure and containment.

Test both deny and allow paths. A system that blocks everything is not operationally adequate, and a successful expected action does not prove malicious variants are contained.

Convert confirmed weaknesses into findings, risks, remediation and retests. Record limitations and untested paths rather than describing the exercise as comprehensive when scope was bounded.

Next

52. Agentic reporting and evidence

Produce defensible Evidence Packs, Board Reports, architecture views, red-team reports, Control Tower reports and framework mappings.

Agentic reports translate live governance and runtime records for different decision makers. Select the output whose purpose matches the audience; do not use one report as a substitute for all assurance work.

Output	Primary use
Evidence Pack	Trace controls, policies, approvals, tests and runtime evidence.
Agentic Board Report	Summarise exposure, incidents, readiness and decisions for leadership.
Security Architecture Report	Explain identities, trust boundaries, connections, data movement and enforcement.
Red Team Report	Record authorised scope, scenarios, results, limitations and remediation.
Control Tower Report	Summarise per-agent ATF posture, gates, exceptions and trends.
Framework Mapping	Trace agentic controls to supporting framework requirements without claiming equivalence.

Generation checklist

1. Confirm workspace, systems, agents, period and intended audience.
2. Check that the underlying records and evidence are current.
3. Resolve unexplained scope, policy and incident contradictions.
4. Include open limitations, accepted risks and overdue work.
5. Generate and record the run metadata.
6. Review content and redact unnecessary sensitive operational detail before distribution.
7. Regenerate after material changes rather than editing an old export into a new conclusion.

Interpretation

Counts and maturity summaries support prioritisation but do not demonstrate that a specific action was authorised. Use request-bound Gateway evidence for runtime decisions. Framework mappings show support and reuse opportunities; they do not create compliance inheritance.

Next

Part 5: API and integration administration

Configure approved integration patterns, credentials and external-agent access without weakening tenant or entitlement boundaries.

53. API overview

Integrate Gamut with your stack using its HTTP API, authenticate with bearer tokens and work with governance data programmatically.

Gamut exposes an HTTP API so you can integrate governance into the rest of your stack, automating inventory, assessments, evidence and reporting rather than doing everything by hand.

> **Note:** API capabilities depend on your plan & entitlements and the role of the token's owner. A token can only do what its owner could do in the interface.

Base URL

The API is served from your Gamut workspace under a versioned prefix:

```
https://run.gamutassure.com/api/compass/v1/
```

All endpoints sit beneath this prefix. The version segment (`v1`) lets the API evolve without breaking existing integrations.

Authentication

Programmatic access uses **bearer tokens**, named and revocable API tokens tied to a user, a workspace, an access level and an expiry. Pass the token in the `Authorization` header:

```
Authorization: Bearer <your-token>
```

See Authentication for creating, using and revoking tokens.

What you can do

The API exposes supported governance resource families described in the resource guide. Not every interactive or administrative operation is a supported public API. Subject to the documented resource, permissions and entitlements, integrations can:

- Keep your AI inventory in sync with other systems.
- Drive or retrieve assessment data.
- Manage evidence and findings programmatically.
- Pull data for external reporting.
- Use workspace-scoped API access within the token's read-only or read/write boundary.

Two authentication modes

The API has two distinct authentication paths for two distinct callers:

- **Bearer tokens** for user-scoped automation against the governance objects above. A token can do only what its owner could do in its bound workspace, within its access level and before its expiry. Token administration remains an interactive account operation and is not available through a bearer token. See Authentication.
- **Signed service requests** for external agent runtimes, under `/api/compass/v1/external-agent/`. These endpoints (context, model, tool invoke, child-agent request, heartbeat, result) use authenticated, anti-replay service requests rather than user bearer tokens, and every action is brokered through Gateway. This is the path the BYO agent runtime SDK uses; agents act through it but never receive provider credentials.

Conventions

The API follows consistent conventions for requests, responses and errors. See Conventions & errors.

Next

54. API authentication

Authenticate to the Gamut API with named, revocable bearer tokens. Create, list, use and revoke tokens, and keep them secure.

The Gamut API authenticates with **bearer tokens**, named, revocable and expiring credentials tied to a user and an accessible workspace. They let automation and integrations act on the API without a browser session while retaining the same tenant, role and entitlement boundaries.

Creating a token

Create a named token from your account, select the exact workspace and choose a read-only or read/write scope. A token inherits the role and entitlements of the user who created it: it can do what that user can do within the selected workspace, and no more.

Write-enabled tokens require the user to re-enter their current password. Where MFA is enrolled, the current authenticator code is also required. This prevents an unattended browser session from silently creating durable write authority.

> **Shown once:** A token's secret value is shown **only once, at creation**. Copy it immediately and store it in a secrets manager. If you lose it, revoke the token and create a new one.

Using a token

Send the token as a bearer credential on each request:

```
GET /api/compass/v1/... HTTP/1.1
Host: run.gamutassure.com
Authorization: Bearer <your-token>
```

Bearer-authenticated requests are intended for server-to-server automation. Do not embed tokens in browsers, mobile apps or any client where users could extract them.

Managing tokens

You can:

- **List** your tokens to see their name, workspace, scope, expiry and recent use.
- **Revoke** a token immediately when it is no longer needed or may be exposed.

  Revoking a token takes effect at once, any integration using it will stop being authorised.

Keeping tokens secure

- **Store in a secrets manager**, never in source control.
- **Scope by user.** Create tokens under a user whose role matches what the integration needs, apply least privilege.
- **Scope by workspace.** Use a separate token for each workspace boundary.
- **Prefer read-only.** Grant write scope only when the integration must change governance data.
- **Use a short lifetime.** Set an expiry appropriate to the integration and rotate before it.
- **Rotate** tokens periodically, and immediately if exposure is suspected.
- **One token per integration**, so you can revoke one without disrupting others.

Sessions vs. tokens

Interactive use of Gamut in the browser uses a signed-in session; programmatic use uses bearer tokens. For human sign-in, including single sign-on, see Users & roles.

Bearer tokens cannot manage account credentials, create other tokens or change their own security boundary. Account suspension, password reset and other qualifying account-security events revoke existing programmatic authority so retained credentials cannot outlive the account decision.

Next

55. Conventions & errors

Request and response conventions for the Gamut API, JSON, versioning, rate limits, status codes and error handling.

The Gamut API follows consistent conventions so integrations are predictable. This page covers the request and response shape, status codes, rate limiting and error handling.

Requests and responses

- **JSON.** Requests and responses use JSON. Send `Content-Type: application/json` on requests with a body.
- **HTTPS only.** All requests are over HTTPS. Plain HTTP is not accepted.
- **Versioned path.** Endpoints live under `/api/compass/v1/`. The version segment lets the API evolve without breaking existing integrations.
- **Authentication.** Send a bearer token on every request, see Authentication.

Status codes

The API uses standard HTTP status codes:

Code	Meaning
`200` / `201`	Success.
`400`	The request was malformed or failed validation.
`401`	Missing or invalid authentication.
`403`	Authenticated, but not permitted, check the token owner's role and entitlements.
`404`	The resource does not exist, or is not visible in your workspace.
`429`	Rate limit exceeded, slow down and retry later.
`5xx`	An unexpected server-side error.

Errors

Error responses return a JSON envelope with a machine-readable code, a human-readable message and the HTTP status:

```
{
"code": "quota_exceeded",
"message": "Daily limit reached (20/20). Resets at midnight UTC.",
"status": 429
}
```

Handle errors by checking `code` (stable, safe to branch on) rather than parsing the message (human-facing, may change). Treat `4xx` as something to fix in your request, and `5xx` as transient unless it persists.

Rate limiting and quotas

AI-assisted endpoints are bound by your plan's usage quotas, enforced server-side. Two limits apply independently:

- a **daily** limit that resets at midnight UTC, and
- a **monthly** limit that resets on the 1st.

Exceeding either returns `429` with code `quota_exceeded` and a message stating the current usage and reset point. Policy generation carries its own separate daily and monthly quotas. Back off and retry after the stated reset rather than retrying immediately.

Listing and limits

List endpoints accept a `limit` query parameter to bound the number of records returned, and relevant filters (for example `workspace_id`, `status`) as query parameters. Request only what you need.

Idempotency, retries and concurrency

Do not assume every write is safe to repeat. Retry network and `5xx` failures only with bounded backoff and only when the operation is known to be idempotent or the current contract provides a request/idempotency mechanism. Do not retry `400`, `401`, `403` or validation failures without changing the underlying request or authority.

For updates, re-read the current record and use returned version or update metadata where the resource supports it. Route conflicting governance edits to review rather than silently applying last-write-wins. External-agent operations have their own signed, request-bound anti-replay contract; never reuse stale signed material.

Validation and data minimisation

Send only documented fields and the records needed for the integration purpose. Do not place credentials, access tokens or unrestricted evidence payloads in notes or logs. Treat server-side validation as a second boundary, not a replacement for client validation.

CSRF

CSRF protection applies to **cookie-based browser sessions**, which must send the per-session token as an `x-csrf-token` header (or form field) on state-changing requests. **Bearer-token** API calls are not cookie-authenticated and do not need a CSRF token; authenticate with the `Authorization` header instead.

Permissions and scope

Every call runs within the token owner's workspace and is subject to their role. You will only ever see and affect data in that workspace, isolation applies to the API exactly as it does in the interface.

Next

56. Supported API resources

Plan integrations around supported Gamut resource families while respecting workspace, role, entitlement and record-lifecycle boundaries.

The versioned API exposes supported resource families for authorised integrations. Availability is determined by the current product version, token scope, user role and tenant entitlement. Do not depend on private browser endpoints or administrator routes discovered through inspection.

Resource families

Customer integrations commonly work with:

- workspace context available to the token owner;
- AI systems and intake records;
- assessment records and framework scores;
- risks, evidence, tests and findings;
- model cards, remediation and reporting artifacts;
- user-owned tokens and account-security operations where explicitly supported;
- entitled Discovery and agentic governance resources;
- exports permitted by the user's role.

Use the product documentation for the lifecycle and meaning of each record. The API does not make a field optional merely because an HTTP request can be accepted; governance completeness still depends on the workflow and human decisions.

Integration design

1. Create a dedicated integration user with the minimum role.
2. Bind a short-lived token to one workspace and prefer read-only.
3. Read the current context and entitlement before attempting a feature.
4. Use stable record identifiers, not display names, for reconciliation.
5. Validate input and handle partial or incomplete governance states.
6. Preserve returned identifiers and timestamps for traceability.
7. Re-read the record after a write rather than assuming local state is authoritative.

Unsupported dependency warning

UI routes, internal administration functions and undocumented response fields may change without public API compatibility guarantees. If a required operation is not documented as supported, contact Gamut before building a production dependency.

Next

57. External agent API

Connect a bring-your-own agent runtime through signed, anti-replay service requests and Gateway-mediated context, model and tool operations.

External agent runtimes use a dedicated signed-service API, not a human bearer token. The current machine-readable contract is available from the external-agent OpenAPI document in the authorised Gamut environment.

Operation families

Operation	Purpose
Context	Request bounded workspace context through the governed context adapter.
Model	Request Gateway-mediated model inference.
Tool invocation	Request one exact registered adapter operation.
Child-agent request	Request bounded delegated identity when parent policy permits delegation.
Heartbeat	Record runtime liveness and version metadata.
Result	Record bounded result metadata without sending secrets or unrestricted payload dumps.

Security model

Requests are authenticated, integrity-protected, time-bounded and protected against replay. The runtime identity, tenant, agent and request context must match current governance. A valid signature does not authorise an action by itself: Gateway still evaluates identity, connection, Tool Permission, Runtime Access Policy, approval and live request context.

Client responsibilities

- Keep service credentials in a secrets manager.
- Generate a fresh request identifier, timestamp and signature for each request.
- Follow the exact canonical signing contract from the current OpenAPI/SDK version.
- Do not retry an action with stale signed material.
- Treat context, connector results and user content as untrusted input.
- Do not send provider credentials, full secret-bearing payloads or unnecessary personal data in result metadata.
- Handle denials and dependency failure as stops, not advisory warnings.

Change management

Pin and test the supported SDK or contract version. Revalidate after runtime, agent identity, connection, policy, model or tool changes. Use a non-production environment before enabling live actions.

Next

58. Integration patterns and operational safety

Build resilient Gamut integrations with least privilege, reconciliation, bounded retries and auditable change handling.

Recommended patterns

Inventory synchronisation

Use a stable external identifier and reconcile before creating. Treat Gamut's returned record ID as the governance identity. Send only changed, validated fields and route conflicts to human review.

Evidence ingestion

Send metadata, provenance and links needed to assess the claim. Avoid unrestricted data dumps. An uploaded artifact still requires reviewer sufficiency and scope decisions.

Reporting extraction

Read from a defined workspace and cutoff time. Preserve generation metadata and explain that the extract is point-in-time. Do not combine tenants or silently discard unassessed records.

Event-driven follow-up

Make handlers idempotent. Store the event or request identifier, detect repeats and use bounded backoff for transient failure. Do not retry validation or permission failures as though they were temporary.

Concurrency and reconciliation

Before overwriting a record, confirm that it has not materially changed since it was read. Where a supported resource returns version or update metadata, use it to detect conflicts. Prefer a visible conflict requiring review over last-write-wins for governance conclusions.

Production checklist

- Dedicated minimum-role integration identity.
- One token per workspace and purpose.
- Expiry, rotation and revocation procedure.
- Input validation and output encoding.
- Timeouts and bounded retries.
- No secrets or tokens in logs.
- Tenant and record identifiers checked on every operation.
- Monitoring for authentication, permission, quota and reconciliation failures.
- Tested recovery and decommissioning procedure.

Next

Part 6: Operational scenarios, security and support

Apply the operating model to common deployments, incidents, assurance requests and support situations.

59. Scenario guides

Vertical-agnostic quick-start guides for the most common AI-governance jobs, governing a GenAI chatbot, EU AI Act readiness, shadow-AI discovery, vendor due diligence, board assurance, agentic workflows and ISO 42001 certification.

Where the industry playbooks start from "who you are", **scenario guides** start from "what you need to do". They are short, outcome-focused quick starts for the governance jobs that come up again and again, whatever your industry.

Each guide names the goal, the outcome you will produce, and a numbered path through the real Gamut modules to get there.

Use every scenario defensibly

Before starting, name the system, owner, decision maker, assessment period and intended outcome. During the work, preserve the sources used, unknowns, reviewer decisions and limitations. Finish only when the following are clear:

- the system and use-case boundary;
- applicable routes and unresolved screening;

- evidence and test coverage;
- open findings, risks and remediation;
- the human conclusion and authority for it;
- the events that will trigger reassessment.

Scenario outputs are point-in-time governance records. They do not replace legal advice, certification decisions or an organisation's own risk acceptance.

The guides

Scenario	The job
Govern a GenAI chatbot	Bring a customer- or staff-facing assistant under governance.
EU AI Act readiness	Classify and evidence a system against the EU AI Act.
Shadow-AI discovery sprint	Find unregistered AI and bring it under governance.
Vendor AI due diligence	Assess a third-party AI tool before and during use.
Board assurance pack	Produce a board-ready view of AI governance posture.
Govern an agentic workflow	Put an action-taking agent under runtime control.
ISO 42001 certification readiness	Prepare the AI management system for certification.

How scenarios and playbooks fit together

The two are complementary. Pick the industry playbook that matches your sector for the framework routing and drivers that apply to you, then use a scenario guide for the specific job in front of you. Both run the same governance lifecycle; they are just two ways into it.

Next

60. Govern a GenAI chatbot

A quick-start guide to bringing a customer- or staff-facing GenAI assistant under Gamut governance, from registration and risk tiering to transparency evidence and runtime control.

A GenAI assistant, whether it answers customers or helps staff, is one of the most common AI systems an organisation needs to govern. This guide takes one from unmanaged to governed.

When to use this

You have deployed, or are about to deploy, a chatbot or copilot built on a language model: customer support, internal knowledge, complaint handling, document Q&A or similar.

What you will produce

A registered, risk-tiered chatbot with documented transparency and oversight controls, supporting evidence, and, if it can take action, enforced runtime governance.

Steps

1. **Register it.** Add the assistant in AI System Records with a named owner and a model card capturing the model, provider and intended use.
2. **Run intake.** In AI Use Case Intake, capture purpose, users, data exposure and oversight, and flag public-facing use, personal data and any retrieval (RAG) sources. Confirm the risk tier.
3. **Route it.** Most customer-facing assistants route to GTSAF for depth and the EU AI Act for transparency obligations.
4. **Evidence the key controls.** Through the Evidence Tracker, capture: disclosure that users are interacting with AI, human-oversight and escalation paths, content and safety controls, and data-handling for any retrieved sources.
5. **Decide if it acts.** If the assistant only answers, you are largely done. If it can take action (issue refunds, change records, call tools), govern it as an agentic workflow: register it in Agentic CISO and enforce action through Gateway.

6 **Track and report.** Raise findings for any gaps, work them on the Remediation Roadmap, and report posture via reporting.

Evidence and review checklist

- Intended and prohibited uses, model/provider version and RAG sources.
- User disclosure, safe escalation, complaint and human-review paths.
- Prompt, output, abuse, privacy and security testing for representative users.
- Data retention, provider use, access controls and cross-border processing.
- Accuracy limitations, monitoring signals and incident response.
- Reassessment after model, prompt, retrieval, audience or action-capability change.

The final conclusion should state whether the assistant only advises or can act, which users and data were assessed, unresolved limitations and who approved continued use.

Modules and frameworks involved

AI System Records, intake & risk tiering, GTSAF, EU AI Act, evidence & findings, and the agentic stack if the assistant takes action.

Next

61. EU AI Act readiness

A quick-start guide to classifying an AI system against the EU AI Act in Gamut, routing it to the article-anchored categories, and evidencing the obligations that apply.

The EU AI Act is risk-tiered, so readiness starts with honest classification and ends with evidenced obligations. This guide takes a system from "we think the Act applies" to a defensible readiness position.

When to use this

You need to demonstrate readiness for the EU AI Act for a specific system, or to triage which of your systems the Act touches and how heavily.

What you will produce

A system classified into the correct EU AI Act risk category, with the applicable obligations identified and evidenced, and a readiness report.

Steps

1 **Register and ground the system.** Capture it in AI System Records and run intake, the intake signals (automated decisions, public-facing, personal and special-category data, sector) drive classification.

2 **Route against the Act.** Use the EU AI Act framework to establish scope, Article 5 status, high-risk classification, Article 50 behaviours, GPAI position and other independently applicable routes. These are not one mutually exclusive risk-class selector.

3 **Identify every role.** Provider, deployer, product manufacturer, importer, distributor, authorised-representative and GPAI roles can create different duties. Determine roles from the facts rather than the contract label.

4 **Assess the atomic obligations.** Work every routed item, recording the legal answer separately from its assurance depth and non-applicability governance.

5 **Evidence them.** Through the Evidence Tracker and Testing Centre, capture the evidence each obligation needs: risk management, data governance, transparency, human oversight, accuracy and robustness, and record-keeping.

6 **Add depth where it matters.** For high-risk systems, pair the Act with GTSAF for control depth and ISO/IEC 42005 for an impact assessment.

7 **Track and report.** Work gaps on the Remediation Roadmap and produce a readiness workpaper pack.

Evidence and conclusion checklist

- Legal snapshot, territorial nexus, system boundary and relevant dates.
- Every economic-operator role and independently applicable route.
- Article 5 screen, high-risk route, transparency route and GPAI route considered separately.
- Evidence linked to the correct obligation, system version and assessment period.
- Current law separated from voluntary or future readiness.
- Named legal/compliance reviewer, limitations and reassessment triggers.

Record **compliance conclusion**, **evidence assurance** and **readiness work** separately. A high completion percentage does not by itself establish legal compliance.

Modules and frameworks involved

intake & risk tiering, EU AI Act, GTSAF, ISO/IEC 42005, evidence & findings and reporting.

Next

62. Shadow-AI discovery sprint

A quick-start guide to finding unregistered AI across your organisation with Gamut Discovery and bringing it under governance through review, registration and risk tiering.

The hardest part of AI governance is knowing what exists. This guide runs a focused sprint to surface shadow AI, GenAI tools, vendor AI and emerging agentic workflows in use but never registered, and bring it under governance.

When to use this

You suspect (or know) that AI is being used across the organisation faster than governance can track, and you need an honest inventory before you can govern anything.

What you will produce

A reviewed set of discovered AI, with the real systems promoted into the registry, tiered and routed, and a clear picture of coverage.

Steps

1. **Set up sources.** In Discovery, configure discovery sources, connector-based collectors or manual imports, with an owner and cadence.
2. **Run discovery.** Each run produces **candidates** (suspected AI in use) and **artifacts** (the underlying usage signals), normalised against rules into canonical apps and vendors.
3. **Triage candidates.** Review each candidate's confidence, signal and guessed owner, and decide: promote, dismiss or investigate. Artifacts carry a reconciliation status so each signal is tied to a known asset or flagged.
4. **Promote the real ones.** Promote confirmed candidates into AI System Records, where they enter intake and risk tiering like any other system.
5. **Tier and route.** Run intake on the newly registered systems and route the higher-risk ones to GTSAF and the EU AI Act.
6. **Report coverage.** Use reporting to show leadership what was found, what was registered, and where coverage still has gaps.

Sprint controls and completion criteria

- Define included business areas, sources, accounts, regions and time period.
- Use least-privilege collection and an approved lawful basis.
- Record rule versions, scan health and known detection limitations.
- Give every candidate and alert an attributable disposition.
- Reconcile duplicates before promotion.
- Route promoted systems through normal intake rather than auto-approving them.

- Report unresolved artifacts and uncovered areas.

A zero-candidate run is not proof that no shadow AI exists. The conclusion must explain source coverage, collector health and detector limitations.

Modules and frameworks involved

Registry & Discovery, intake & risk tiering, GTSAF, EU AI Act and reporting.

> **Note:** Discovery is an enterprise capability. Availability depends on your plan & entitlements.

Next

63. Vendor AI due diligence

A quick-start guide to assessing a third-party or vendor AI tool in Gamut before and during use, structured assessment, evidence requests to the vendor, and ongoing oversight.

Most AI an organisation uses is bought, not built. This guide assesses a third-party or vendor AI tool, before adoption and on an ongoing basis, so procurement and risk decisions are structured and defensible.

When to use this

You are evaluating a vendor AI product, or you already use one and need to bring it under proper governance and periodic review.

What you will produce

A registered vendor system with a documented assessment, vendor-supplied evidence captured against controls, and a clear accept/condition/reject position with ongoing review dates.

Steps

1. **Register the vendor system.** Add it in AI System Records, recording the vendor, deployment type and a model card for the vendor model. Mark it as vendor-provided.
2. **Run intake and tier it.** Capture the use case, data exposure and oversight in intake, and confirm the risk tier that sets how deep the diligence should go.
3. **Route to the right frameworks.** Higher-risk vendor systems route to GTSAF and the EU AI Act; for the vendor's own management system, ISO/IEC 42001 evidence is a strong signal.
4. **Request evidence from the vendor.** Use evidence requests to ask the vendor (or the internal owner) for specific artefacts against specific controls: validation, security, data handling, model documentation and assurance certifications.
5. **Assess and decide.** Score the controls with rationale, raise findings for gaps, and reach an accept, accept-with-conditions or reject decision with the gaps tracked on the Remediation Roadmap.
6. **Set ongoing review.** Record review dates so the vendor is reassessed as the product and your reliance on it change, and report the portfolio via reporting.

Minimum diligence evidence

- Contracted purpose, roles, service boundary and subcontractors.
- Model and material component provenance and change notification.
- Data use, retention, location, deletion and training commitments.
- Security, incident, resilience and business-continuity evidence.
- Performance, bias, safety, accessibility and human-oversight evidence.
- Audit rights, termination, data return and exit plan.
- Independent assurance interpreted within its actual scope and period.

Document unavailable evidence as a limitation or condition. A supplier questionnaire response is a claim until corroborated by appropriate evidence, testing or contractual assurance.

Modules and frameworks involved

AI System Records, intake & risk tiering, evidence & findings, GTSAF, EU AI Act, ISO/IEC 42001 and reporting.

Next

64. Board assurance pack

A quick-start guide to producing a board-ready view of AI governance posture in Gamut, inventory, risk, evidence quality, findings and remediation, traceable to the underlying records.

Boards increasingly ask a direct question: are we governing our AI, and can we prove it? This guide produces a concise, defensible board pack that answers it from live records rather than a hand-assembled slide deck.

When to use this

You need to brief a board, risk committee or executive on AI governance posture, and you want a view that is concise for leadership yet traceable back to evidence.

What you will produce

A board pack covering AI inventory, risk picture, evidence quality, open findings, remediation progress and readiness priorities, every figure traceable to its underlying records.

Steps

1. **Confirm the inventory is current.** Check AI System Records is complete, running a discovery sprint first if coverage is in doubt. A board view is only as honest as the inventory beneath it.
2. **Confirm tiers and assessments.** Make sure systems have been through intake and risk tiering and that material systems carry an assessment.
3. **Surface findings and remediation.** Review open findings and the Remediation Roadmap, including overdue items, so the board sees real status, not a clean fiction.
4. **Generate the pack.** In reporting, produce a **Board pack** or **Board summary**: it composes inventory, risk, estate, evidence quality, findings, decisions and roadmap into a leadership view, with agentic posture included where relevant.
5. **Keep it defensible.** Because the pack is generated from connected records and carries run-level traceability metadata, any figure can be followed back to its evidence if challenged.
6. **Optionally accelerate the narrative.** Use the AI Consultant's Board Summary mode to draft executive narrative grounded in the same records.

Board quality checklist

- Coverage denominator, unassessed estate and inventory limitations are visible.
- High and critical exposure, overdue findings and accepted risks are not averaged away.
- Trend changes distinguish scope, completion and genuine control improvement.
- Agentic runtime posture and material incidents are included where relevant.
- Each requested board decision has an owner and deadline.
- Pack ID, cutoff date, source workspace and reviewer are recorded.

Review AI-drafted narrative against the source records. Regenerate the pack after material changes; do not edit an old export into a new reporting period.

Modules and frameworks involved

reporting, AI System Records, intake & risk tiering, evidence & findings, Remediation Roadmap and the AI Consultant.

Next

65. Govern an agentic workflow

A quick-start guide to putting an action-taking AI agent under runtime control in Gamut, register it, set its ATF level, enforce policy through Gateway and execute through Claw.

When AI stops answering and starts acting, calling tools, moving data, making changes, design-time assessment is no longer enough. This guide puts an agentic workflow under enforced runtime governance.

When to use this

You have, or are building, an agent that takes action: it calls tools and APIs, retrieves and writes data, or runs multi-step workflows, whether on Gamut Claw or your own framework.

What you will produce

A registered, ATF-assessed agent designed to act only through Gateway-enforced authority, with provider credentials outside the agent and request-level runtime evidence.

Steps

1. **Register the agent.** Add it to the Agentic CISO agent register with a human owner and a security owner. An unregistered agent is blocked from acting at all.
2. **Set its autonomy.** Assign an ATF level (Intern through Principal). Gateway enforces a different action boundary at each level, from read-only to strategic autonomy.
3. **Score its capability risk.** Run an ACRS assessment to set how tightly to govern, and model the threats its capabilities introduce with MAESTRO.
4. **Authorise its tools.** Grant tool permissions in Agentic CISO and confirm the matching governed connectors exist. An agent can use a tool only when both layers agree.
5. **Define Runtime Access Policies.** Bind each required authority to the exact agent, governed connection, action, resource, data classification, environment, purpose and time boundary. Validate the policy, review its impact and submit the immutable version for independent approval.
6. **Configure approval gates.** Require human approval for sensitive or mutating actions (external calls, financial actions, code changes). Gateway enforces them on every request.
7. **Independently approve runtime authority.** A workspace-scoped Runtime Policy Approver reviews the submitted policy and either publishes it or rejects it with a reason. Approval requires the reviewer's own MFA-protected step-up.
8. **Choose the runtime.** Run it on Gamut Claw or, for an external framework, the BYO runtime. Either way the rule holds: think anywhere, act through Gateway. Credentials live on Gateway, never with the agent.
9. **Simulate, then watch.** Run a Gateway simulation to see what Gateway would decide and close gaps before going live, then rely on the hash-chained runtime evidence and the audit log for ongoing oversight.

Go-live evidence and gates

- Authenticated agent identity, owners and purpose boundary.
- ATF target, ACRS risk and MAESTRO threat treatment.
- Exact Tool Permissions, governed connections and data flows.
- Independently approved Runtime Access Policies with expiry and review.
- Allow, deny, approval, timeout, replay, failure and containment tests.
- Monitoring ownership, incident playbook and kill/suspend procedure.
- Residual risk decision and material change triggers.

Readiness is not standing authority. Gateway revalidates the exact request immediately before execution, and missing or stale context must stop the action.

Modules and frameworks involved

Agentic CISO, Runtime Access Policies, Gateway, Claw, BYO runtime, ATF, ACRS and MAESTRO.

Next

66. ISO 42001 certification readiness

A quick-start guide to preparing your AI management system for ISO/IEC 42001 certification in Gamut, route to the standard, evidence the clauses and Annex A, and produce an audit-ready pack.

ISO/IEC 42001 is the AI management-system standard organisations and their buyers increasingly ask for. This guide prepares your management system for certification by evidencing the standard inside Gamut.

When to use this

You are pursuing ISO/IEC 42001 certification, or a buyer or board has asked you to demonstrate an AI management system aligned to it.

What you will produce

An assessment against ISO/IEC 42001 with each clause and Annex A control evidenced, gaps tracked to closure, and an audit-ready workpaper pack for the certification body.

Steps

1. **Set the scope.** Confirm which AI systems and processes are in scope in AI System Records; the management system governs how you govern them.
2. **Route to the standard.** Create an assessment against ISO/IEC 42001, which is structured as clauses 4 to 10 plus Annex A controls.
3. **Assess each section.** Work through the management-system clauses (context, leadership, planning, support, operation, performance evaluation, improvement) and the Annex A controls, recording maturity and rationale.
4. **Evidence the controls.** Through the Evidence Tracker and Testing Centre, capture the artefacts each clause needs, policies, roles, risk and impact assessments, operational controls, monitoring and review records. The policy generator can draft the policies and procedures the standard expects.
5. **Close the gaps.** Raise findings for shortfalls and work them on the Remediation Roadmap ahead of the audit.
6. **Produce the audit pack.** Generate a workpaper pack that traces each control conclusion to its evidence, the package a certification body can test against.

Readiness evidence and audit questions

- Approved AIMS scope, interested parties and exclusions.
- AI policy, objectives, roles, competence and communication.
- Risk and impact assessment methods applied to the in-scope portfolio.
- Statement of Applicability with inclusion and exclusion rationale.
- Operational procedures and records showing they are followed.
- Monitoring, internal audit, management review, corrective action and improvement.
- Evidence period, sampling limitations and open nonconformities.

Use a licensed copy of ISO/IEC 42001 and confirm certification scope with the selected certification body. Gamut supports readiness and traceability; it does not issue the certificate or guarantee an audit result.

Modules and frameworks involved

ISO/IEC 42001, AI System Records, evidence & findings, policy generation, Remediation Roadmap and reporting.

Next

67. Security & trust

Gamut is secure by construction, tenant isolation at the database-schema level, authenticated encryption, zero-trust agentic enforcement, MFA and role plus plan access control, governed server-side AI, and continuous security review.

Gamut governs your most sensitive AI risk and evidence, so security is a **design property of the platform, not a feature bolted on**. Trust is the product. This page sets out how Gamut is built to protect your data and how we hold ourselves accountable for it.

Security at a glance

Area	What Gamut does
Tenant isolation	Each organisation's data lives in its own database schema, with row-level tenant context checks and fail-closed enforcement on protected data paths.
Encryption	Sensitive secrets are protected with authenticated **AES-256-GCM** encryption at rest; all traffic is encrypted in transit over HTTPS with HSTS in production.
Access	Multi-factor authentication, step-up MFA for sensitive changes, single sign-on (OpenID Connect), role-based access and entitlement dual-gating.
Agentic AI	**Zero-trust:** agents hold no credentials, exact Runtime Access Policies require controlled approval, and every action is revalidated and enforced through Gateway, fail-closed.
AI handling	Model provider keys never reach the browser; all AI calls are governed server-side.
Accountability	Audited state changes and tamper-evident runtime evidence for governed agent actions.
Application hardening	Rate limiting, CSRF protection, content security policy and other production browser-hardening headers.
Assurance	Continuous security review, including dependency, static and authorised dynamic testing, with results assessed as part of release readiness.

Tenant isolation

Each organisation operates in its own tenant, and isolation is enforced at the data layer: each tenant's records live in their own database schema, not merely filtered by an application query. One organisation's AI systems, assessments, evidence and users are kept strictly separate from every other, and access is always scoped to the tenant a user belongs to. Protected data paths also bind tenant context and fail closed when that context cannot be verified.

Encryption

Sensitive secrets and credentials are protected with **AES-256-GCM**, an authenticated cipher, so stored secrets cannot be silently read or altered. The platform will not run in production without its encryption key, so secrets are never stored in plaintext. Passwords are stored as salted, one-way **scrypt** hashes and are never recoverable. All traffic is encrypted in transit over HTTPS, with HSTS enforced.

Authentication and access

People sign in with a password or via single sign-on using OpenID Connect, so organisations can apply their own multi-factor and conditional-access policies. Gamut also supports its own multi-factor authentication, with **step-up verification** required before security-sensitive changes.

Access is governed by role-based access control and plan entitlements: a sensitive capability requires **both** the role permission and the plan entitlement, enforced server-side on every action. Administrators are deliberately not exempt from entitlement checks. Suspending a user or tenant revokes access immediately. Browser sessions carry CSRF protection, programmatic access uses named, revocable API tokens, and responses carry modern browser-hardening headers including a strict content-security policy. Production also applies rate limiting and HSTS.

API tokens are bound to an accessible workspace, limited to read-only or read/write scope and time-bounded. Creating write authority requires reauthentication, and account-security events invalidate existing sessions and programmatic credentials where continued access would no longer be appropriate.

Zero-trust agentic AI

Gamut governs AI that takes action, not just AI that answers questions, and it does so on a zero-trust basis. **Agents never hold credentials and never call tools directly.** Every agent action passes through Gamut Gateway, where policy is enforced and provider keys live, and the execution layer (Claw) redacts sensitive values before any output leaves it. Enforcement is **fail-closed**: if a decision cannot be made or verified safely, the action does not proceed. See the agentic stack overview.

Runtime Access Policies bind authority to an exact agent, governed connection, action, resource, data class, environment and time boundary. Policy creation and policy approval are separated. Immediately before execution, Gateway confirms that the policy, approval and request context remain current and refuses stale, changed, expired or replayed authority.

Governed, server-side AI

All AI analysis is proxied **server-side**. Model provider keys are never exposed to the browser, and prompts and responses are handled by Gamut rather than sent directly from a user's device to a model provider. This keeps model usage governed and credentials protected.

What is sent depends on the feature:

- **Framework AI Assist, full context:** selected system or agent scope, the relevant assessment item, structured scores/statuses and the authorised notes, evidence metadata and routing context required for that analysis.
- **Framework AI Assist, scores-only privacy mode:** direct identifiers and free-text context are excluded; the model receives the minimum structured assessment and route signals required for the bounded task.
- **AI Consultant:** permitted workspace records and free text shown by the on-screen context manifest. Scores-only privacy mode does not narrow Consultant context.
- **Generated narratives:** use the context described on the generating screen; review the scope before submitting.

AI output remains advisory. It cannot approve scope, applicability, evidence, risk acceptance, policy or the human assessment conclusion. Avoid entering secrets and unnecessary personal data, and apply the configured model provider's contractual and organisational handling requirements.

Accountability by default

State-changing actions are recorded in an audit log, and governed agent actions generate **tamper-evident runtime evidence**. Together these provide a reviewable record of relevant human and agent activity for the configured scope and retention period.

Our security practice

Security is part of how we build, not an afterthought:

- **Secure by construction.** The controls above are architectural, isolation, least privilege, fail-closed enforcement and encrypted secrets are properties of the design.
- **Continuous security review.** We conduct regular security review, including static analysis and authorised dynamic testing, as part of our release process.
- **Least surface area.** Gamut runs on a deliberately minimal dependency footprint to reduce supply-chain and attack surface.
- **Dependency hygiene.** Dependencies are reviewed as part of release readiness. A point-in-time scan is evidence for that release, not a permanent guarantee; newly disclosed issues are assessed and treated according to risk.

Independent validation

We are committed to validating our security posture independently, not just asserting it. We are **commissioning independent third-party penetration testing** and **pursuing recognised certifications** (such as SOC 2 and ISO/IEC 42001) as we scale. Organisations evaluating Gamut can request our current security overview through their account contact.

Reporting a vulnerability

We welcome reports from security researchers and customers. If you believe you have found a security issue, please see our vulnerability disclosure policy for how to report it safely and what to expect from us.

Next

68. FAQ

Common questions about Gamut AI, what it is, who it is for, the frameworks it supports, agentic governance, security and getting started.

What is Gamut AI?

Gamut is an AI governance lifecycle platform. It helps organisations discover the AI they use, classify its risk, assess it against multiple frameworks, and produce audit-ready evidence. See What is Gamut AI.

Who is Gamut for?

Organisations running AI governance, across risk, compliance, security and procurement, and the auditors who review them. It is built for organisations that need to demonstrate structured governance, including banks, financial institutions and regulated enterprises. See Who Gamut is for.

Which frameworks does Gamut support?

GTSAF, the EU AI Act, NIST AI RMF, ISO/IEC 42001 and 42005, NAGF, ACRS, the Agentic Trust Framework (ATF) and MAESTRO. A single AI system can be assessed against one or several. See Frameworks overview.

How is a system's risk tier decided?

Intake captures structured risk signals, and a deterministic governance weighting profile, six weighted dimensions with configurable thresholds and floor rules, turns them into a risk tier. The same inputs always produce the same tier, and the reasoning is traceable. See Intake & risk tiering.

Does Gamut give legal advice or certify compliance?

No. Gamut's framework references help you organise governance work and are not endorsements by framework owners. The product does not replace licensed standards, legal advice, certification audits or regulator guidance. See Frameworks overview.

What is the difference between GTSAF and the regulatory frameworks?

GTSAF is Gamut's native assurance baseline, 358 controls across 17 domains, for depth. The regulatory frameworks (EU AI Act, NIST AI RMF, ISO) map Gamut's workflow to external regimes. Evidence crosswalks between them. See GTSAF.

Which framework should I choose when I create a Free workspace?

Choose GTSAF unless you have an immediate need to start with NIST AI RMF, EU AI Act or NAGF. GTSAF has the deepest coverage and maps outward to other frameworks. Free workspaces are locked to the one framework selected at signup, with additional frameworks available on paid plans. See Plans & entitlements.

How does Gamut govern AI agents?

Through the agentic stack: Agentic CISO governs agents, Gateway enforces policy on every action, and Claw executes work only through Gateway. Agents never hold credentials or call tools directly. See the agentic stack overview.

Can I use my own agent framework?

Yes. The BYO agent runtime lets you run external frameworks (LangGraph, CrewAI, Hermes, OpenClaw and custom code) while Gamut remains the trust and enforcement plane, think anywhere, act through Gateway.

What happens if an agent is not registered?

It is blocked. Gateway refuses any action from an agent that is not in the Agentic CISO register, at critical severity. An agent's autonomy is also capped by its ATF level (Intern through Principal), so it can only act within that boundary.

How are agents scored for risk?

By ACRS, which scores an agent's capability across dependency, action, access and harm to set how tightly to govern it, and by an ATF assessment for trust and autonomy readiness.

What happens when routing facts are incomplete?

Unknown material facts produce **screening required** or a routing-incomplete state; they do not silently exclude frameworks or controls. Complete the named system and intake facts, resolve conflicts, review ACRS and then confirm the route. See Routing & applicability.

Why can I see a system in a framework selector but not select it?

The system exists, but a prerequisite such as saved intake, confirmed routing, ACRS review, framework licence or module-specific scope may be incomplete. Follow the displayed reason rather than entering evidence under the all-controls catalogue. See System selection in assessments.

What does privacy mode send to AI?

For framework AI Assist, privacy mode sends bounded structured scores, statuses and route signals without direct identifiers or free-text records. It does not change the assessment or route. The AI Consultant is different and uses the permitted workspace context shown on its screen. See Security & trust.

How can someone approve a runtime policy without becoming an administrator?

Assign the dedicated Runtime Policy Approver role to the exact workspace. It provides approval-only access to submitted policies and supporting context. MFA is required for a decision, and the creator or submitter cannot approve that version. See Auditor and independent review access.

Is my data isolated from other organisations?

Yes. Each organisation operates in an isolated tenant with its own database schema, not just query filtering. See Security & data handling.

Is my data encrypted?

Sensitive secrets and credentials are encrypted at the application layer with AES-256-GCM, and the server refuses to start in production without its encryption key. Passwords are scrypt-hashed. See Security & data handling.

What actions are recorded?

State-changing actions are written to the audit log before they return, and a defined set of high-sensitivity actions (role grants, exports, policy and agent decisions, billing, and more) mandate an audit record. Agent actions also generate hash-chained runtime evidence.

How is AI analysis handled?

All AI analysis is proxied server-side, so model provider keys are never exposed to the browser. The AI Consultant and other AI features are gated by role and entitlement and metered by daily and monthly quotas. Framework AI Assist can use scores-only privacy mode. The AI Consultant is different: it uses the permitted workspace context shown in its context manifest, so do not use it when that wider context should not be sent to the configured model provider. See Security & data handling.

What plan tiers are there?

Free, Standard, Advanced and Enterprise are the customer-facing plans. Free is $0 and includes one selected framework. Standard is $499/month, or $4,990/year, and includes GTSAF, EU AI Act, NIST AI RMF, NAGF, reports, evidence tracker, findings register, bounded policy generation, workpaper packs and unlimited free read-only auditor seats. Advanced is $1,499/month, or $14,990/year, and adds MAESTRO, ACRS, ATF, Agentic CISO, Gateway simulation, AI chat and control testing, with higher AI and policy-generation allowances. Enterprise starts at $40,000/year and adds production Gateway enforcement, Claw, Discovery, OpenID Connect single sign-on, contracted data-residency options and bespoke scale. A capability needs both the plan entitlement and the user's role permission. See Plans & entitlements.

Is there a trial?

Free and Standard workspaces can activate a one-time 14-day Advanced Trial Boost from billing. It temporarily grants Advanced-tier features, includes 50 AI Consultant interactions, requires administrator step-up MFA and preserves data after the boost expires. See Plans & entitlements.

Does Gamut support single sign-on?

Yes, via OpenID Connect. See Single sign-on.

Is there an API?

Yes, an HTTP API authenticated with bearer tokens. See the API overview.

How do I get started?

Follow the Quickstart: sign in, register an AI system, run intake, and complete your first assessment.

How do I get help?

See Support.

69. Support

How to get help with Gamut AI, access the platform, talk to the team, and find the right documentation.

This page points you to the right place for help with Gamut AI.

Using the platform

Gamut runs at **run.gamutassure.com**. If you do not yet have access, ask your organisation's administrator to invite you, or start a trial from the sign-in page.

Find it in the docs

Most questions are answered here:

- New to Gamut? Start with What is Gamut AI and the Quickstart.
- Working with a framework? See Frameworks overview.
- Governing agents? See the agentic stack overview.
- Integrating? See the API overview.
- Common questions are in the FAQ.

Use the search box at the top of any page to jump straight to a topic.

Before contacting support

Record the information needed to investigate without exposing secrets:

- tenant and workspace name;
- affected module and system record;
- approximate time and time zone;
- action attempted and visible error message;
- whether the problem affects one user or several;
- browser and deployment environment;
- a redacted screenshot or request correlation detail where available.

Do not send passwords, MFA setup secrets, API tokens, model-provider keys, Gateway credentials, private keys or unredacted sensitive evidence by email.

Common first checks

- Confirm the correct tenant, workspace and system.
- Check save status and reload the current record.
- Verify the user's role and workspace plan entitlement.

- Complete MFA where the action requires step-up verification.
- For assessment selection, review intake, routing and ACRS prerequisites.
- For AI features, verify provider configuration and displayed quota.
- For runtime actions, inspect identity, connection, permission, policy, approval and Gateway status.

  Avoid repeatedly retrying a state-changing action if the result is uncertain. Check the record and Audit Trail first to prevent duplicate work.

Talk to the Gamut team

For questions the docs do not cover, plans and enterprise capabilities, security and assurance detail, or reporting a concern, contact the Gamut team:

- **Website:** gamutassure.com
- **Email:** hello@titai.org

> **Note:** Confirm any service commitments with your Gamut account representative; the address above is a general starting point.

Security concerns

If you believe you have found a security issue, please report it responsibly via our vulnerability disclosure policy. For our overall security posture, see Security & trust.