

ODA3 INSTITUTE

IMPLEMENTATION GUIDANCE

GAISSF™ v1.0 Reference Architecture

Reference designs, security zones, trust boundaries and deployment patterns

GAISSF-IMP-013 | Public

© ODA3 Pvt Ltd

Document Control

Document ID	GAISSF-IMP-013
Title	GAISSF™ v1.0 Reference Architecture
Doc Type	Implementation guidance — informative
Purpose	Reusable architectural blueprints and security patterns for GAISSF™ implementation
Primary Audience	Enterprise, security and AI architects; platform, MLOps and DevSecOps engineers; safety engineers; governance leads; assessors
Dependencies	GAISSF-NOR-001 to GAISSF-NOR-008; GAISSF-IMP-009 to GAISSF-IMP-012, where published and current
Precedence	GAISSF-NOR-001 (Framework Standard) and GAISSF-NOR-004 (Control Catalogue) prevail
Control Baseline	59 controls across nine domains
Status	Final Publication
Publish Date / Review Cycle	Issued 29 June 2026; updated 05 October 2026; review on material change to GAISSF-NOR-001 or GAISSF-NOR-004
Contact	contact@oda3.org
Classification	Public

GAISSF™ v1.0 Reference Architecture

Reference designs, security zones, trust boundaries and deployment patterns

Reliance notice: This document provides implementation-independent reference designs. It does not mandate a specific product, vendor, cloud, topology or technology stack. The designs are illustrative and must be adapted to system context, threat model, sector, jurisdiction and safety requirements. It does not replace normative control statements, applicable law, sector-specific engineering or professional judgement. Use of this architecture does not establish GAISSF™ conformance, legal compliance, system safety, or eligibility under any future GAISSF certification scheme; certification is not currently offered. This document does not define which controls belong to any conformance tier - see GAISSF-NOR-001.

How to Use This Document

Reader	Recommended use
Executive or board reader	Review purpose, decision rights, legal and regulatory integration, limitations and Notably Absent sections.
Program or control owner	Use workflows, templates, gates, evidence fields and escalation criteria as implementation aids.
Engineering or operations team	Translate guidance into system-specific procedures, configurations, runbooks and testable acceptance criteria.

Reader	Recommended use
Legal, privacy or compliance team	Validate jurisdiction-specific obligations, deadlines, retention, disclosure, privilege and regulator-facing requirements.
Independent assessor or reviewer	Use the document as informative context only; test claims against the authoritative normative sources and applicable assessment criteria.

Foreword

A reference architecture helps organizations implement GAISSE consistently without prescribing a single technology stack. The designs in this document organize control placement, trust boundaries, telemetry, evidence, human oversight and fail-safe behavior across common AI deployment models.

1. Purpose and Scope

This document defines reusable architectural patterns for enterprise AI services, generative-AI applications, retrieval-augmented generation, agentic systems, externally hosted model services, private-model platforms, edge deployments and physical-AI systems.

The patterns are implementation-independent. Organizations should select, combine and tailor them based on system criticality, autonomy, data sensitivity, supplier dependence, deployment model and applicable safety obligations.

2. Architecture Principles

Principle	Architectural implication
Explicit trust boundaries	Every transition between users, applications, models, tools, data, suppliers and physical actions is identified and controlled.
Least privilege by capability	Permissions are scoped to models, tools, actions, data, memory and environments—not only user identities.
Separation of control and data planes	Governance, policy and approval functions are separated from operational data and inference flows.
Evidence by design	Architecture generates attributable logs, decisions, tests and state records required for assurance.
Fail securely and safely	Failures produce contained, observable and recoverable states; physical systems enter a defined safe state.
Human authority for consequential actions	High-impact actions require approval, review, override or escalation appropriate to risk.
Versioned and reproducible components	Models, data, prompts, policies, tools and configurations are identifiable and recoverable.
Supplier boundaries remain visible	Inherited controls and external dependencies are represented explicitly rather than treated as internal.
Monitoring is independent where feasible	Critical telemetry and control-health verification are not solely dependent on the component being monitored.

3. Logical Reference Architecture

3.1 Layered architecture

Governance and Assurance Plane

Inventory • risk • policy • exceptions • approvals • evidence • audit • external claim records



Security and Policy Enforcement Plane

Identity • authorization • AI gateway • policy engine • content controls • transaction approval



Experience and Channel Layer

Web • mobile • API • chat • voice • operator console • embedded interface



Application and Orchestration Layer

Business logic • prompt orchestration • agents • memory • tool routing • workflow state



Model Service Layer

Model routing • inference endpoints • local models • hosted models • evaluation wrappers



Data and Knowledge Layer

Training data • retrieval stores • vector databases • feature stores • operational records



Platform and Infrastructure Layer

Cloud • containers • networks • secrets • endpoints • accelerators • edge compute



Physical Actuation and Environment Layer

Robotics • vehicles • machinery • sensors • actuators • human environment

3.2 Cross-cutting services

Service	Purpose	Evidence generated
Identity and access	Authenticate users, services, agents, tools and administrators	Access decisions, role changes, privileged sessions
Policy decision point	Evaluate contextual policy before model, data or tool use	Policy version and decision log
Telemetry pipeline	Collect security, behavior, safety and control-health events	Immutable or protected event records
Configuration registry	Maintain approved model, prompt, tool and policy versions	Release and change history
Evidence service	Index and protect artifacts linked to GAISSF controls	Manifest, hashes, signatures and custody records
Evaluation service	Run regression, adversarial, safety and performance tests	Evaluation reports and failed-result records
Incident integration	Create, route and track AI-related incidents and near misses	Tickets, timelines and corrective actions

4. Security Zones and Trust Boundaries

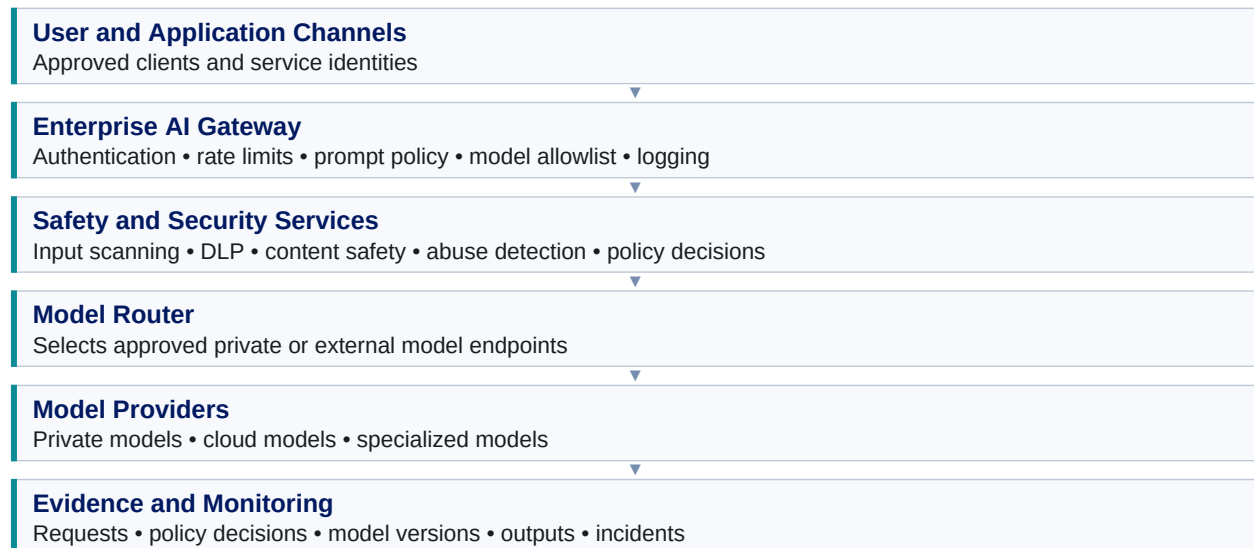
Zone	Typical components	Required boundary controls
Z0 — Public/untrusted	Internet users, public data, external prompts and files	Input validation, rate limiting, abuse detection and isolation
Z1 — User interaction	Authenticated users, channels and operator consoles	Identity, session control, disclosure and user-level policy
Z2 — Application	Business logic, APIs, orchestration and workflow state	Service identity, authorization, secure coding and transaction control
Z3 — AI mediation	AI gateway, prompt policy, model router and output controls	Policy enforcement, model allowlists, logging and content safeguards
Z4 — Model execution	Hosted or private inference endpoints	Model identity, version pinning, network restriction and monitoring
Z5 — Data and memory	Vector stores, datasets, memory, logs and feature stores	Data classification, access, lineage, poisoning controls and retention
Z6 — Tool/action	Plugins, enterprise APIs, code execution and external actions	Per-tool permissions, approval, sandboxing and rollback
Z7 — Assurance	Test harnesses, evidence repositories and monitoring systems	Independence, integrity, restricted write access and traceability
Z8 — Physical control	Sensors, controllers, actuators and safety systems	Command authentication, safety interlocks, fail-safe and override

5. Core Data Flows

Flow	Architecture controls
User input to application	Authentication, validation, malware scanning, rate limits and purpose constraints
Application to model	Prompt policy, context minimization, model routing, version pinning and request logging
Application to retrieval	Query authorization, tenant isolation, data classification and provenance filters
Model output to user	Output validation, content safety, disclosure, confidence/limitation handling and logging
Agent to tool	Capability authorization, parameter constraints, transaction approval and action recording
System to supplier model	Encrypted transport, data-use restrictions, provider controls and response validation
Telemetry to assurance plane	Protected transport, timestamping, integrity and access segregation
AI decision to physical actuator	Independent safety gate appropriate to system risk and sector obligations, command authentication, bounds checking and safe-state logic

6. Pattern A — Enterprise AI Gateway

6.1 Blueprint



6.2 Suitable use

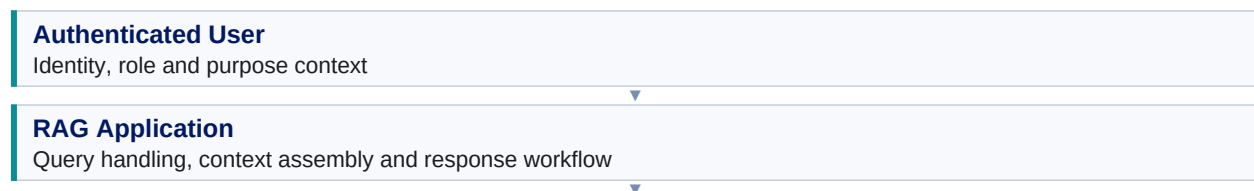
Centralized control of multiple AI applications and model providers, particularly where organizations need consistent policy, logging, supplier control and evidence.

6.3 Key risks and controls

Risk	Architectural control
Gateway becomes single point of failure	High availability, graceful degradation and bypass prevention
Sensitive context reaches external model	Data classification, redaction, routing rules and contractual restrictions
Model substitution	Provider and model allowlists with cryptographic or registry verification
Policy drift	Versioned policy deployment and independent conformance checks
Incomplete application coverage	Network and identity controls preventing direct unsanctioned model access

7. Pattern B — Retrieval-Augmented Generation

7.1 Blueprint



Retrieval Policy Layer

Authorization, tenant isolation, source filtering and relevance constraints

**Knowledge Stores**

Approved documents, vector indexes and source metadata

**Model Service**

Inference using bounded retrieved context

**Citation and Output Layer**

Source attribution, content controls and user disclosure

**Evidence Layer**

Query, sources, model version, output and user feedback

7.2 Key controls

Concern	Pattern control
Unauthorized retrieval	Document-level and attribute-based access controls before retrieval
Poisoned knowledge	Source approval, integrity checks, lineage and change review
Cross-tenant leakage	Separate indexes or enforced tenant filters with negative testing
Unsupported answer	Citation requirement, confidence handling and abstention behavior
Stale knowledge	Freshness metadata, expiry and re-index controls
Prompt injection in documents	Content scanning, instruction isolation and tool-policy enforcement

8. Pattern C — Agentic AI with Tool Use

8.1 Blueprint

User/Trigger

Authenticated request and approved objective

**Agent Controller**

Goal constraints • plan limits • termination • state

**Policy and Approval Gate**

Per-tool permission • transaction risk • human approval

**Tool Sandbox**

Isolated execution with allowlisted tools and parameters

**Enterprise Services**

APIs, databases, ticketing, code repositories and transactions

**Memory Service**

Scoped, retained and integrity-checked memory



Observation and Evidence

Plans • tool calls • approvals • outputs • state changes

**Containment**

Circuit breaker • kill switch • rollback • incident routing

8.2 Agent authorization model

Authorization dimension	Example constraint
Identity	Agent instance and sponsoring user/service are attributable
Objective	Approved goal class and prohibited objectives
Tool	Explicit allowlist and denied capabilities
Action	Read, draft, recommend, execute or administer
Resource	Permitted systems, accounts, files and datasets
Transaction	Value, frequency, destination and approval thresholds
Time	Session duration and credential expiry
Delegation	Whether sub-agents may be created and with what permissions

9. Pattern D — Private Model Platform

9.1 Blueprint

Model and Data Registry

Approved models, datasets, lineage, licences and signatures

**Build and Training Environment**

Isolated pipelines, reproducible builds and controlled compute

**Evaluation Environment**

Security, safety, robustness, privacy and performance testing

**Release Registry**

Signed model package, configuration and approval status

**Inference Platform**

Segregated endpoints, autoscaling, quotas and service identity

**Monitoring and Drift**

Behavior, data drift, abuse, failures and control health

**Evidence Repository**

Training, evaluation, release and operational artifacts

9.2 Platform separation

Environment	Permitted activities	Prohibited activities
Development	Experimentation with non-production data and isolated models	Direct production deployment
Training	Approved data processing and model creation	Unapproved external connectivity
Evaluation	Independent test execution and comparison	Modification of approved test results
Staging	Production-like integration and release validation	Use as hidden production
Production	Approved inference and monitoring	Interactive model modification

10. Pattern E — External Model Service

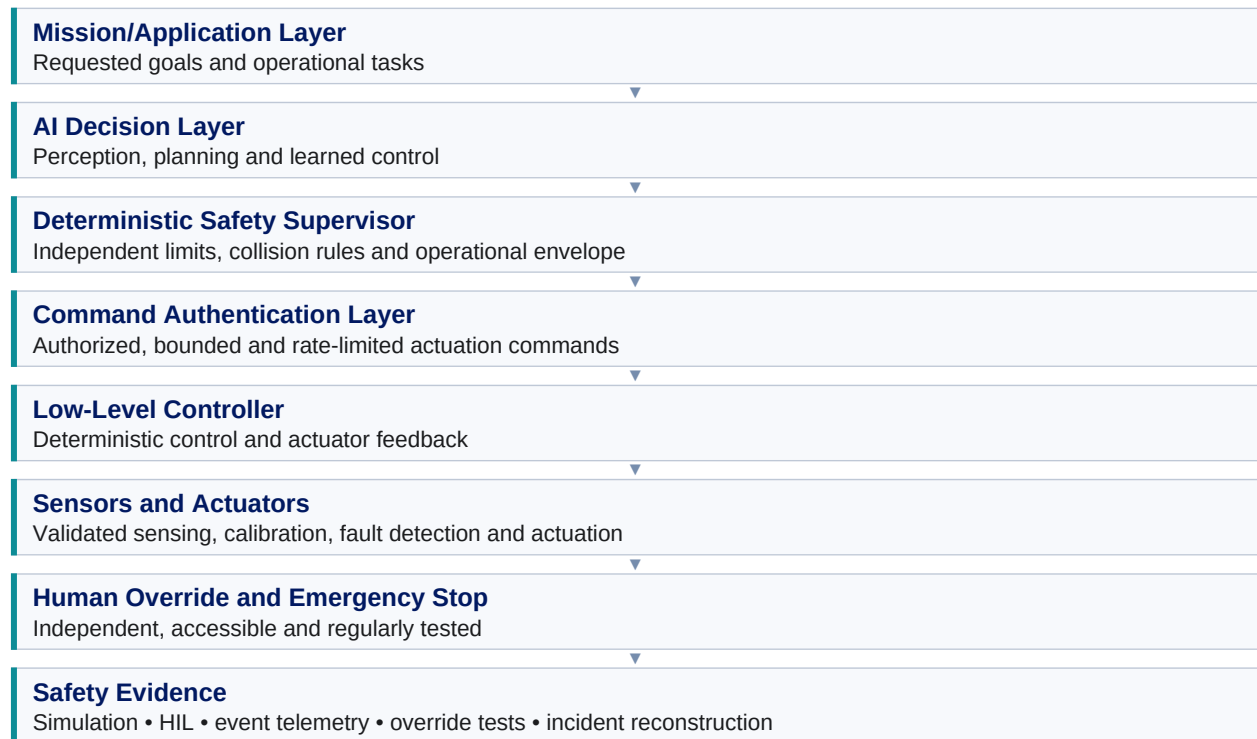
Boundary concern	Reference control pattern
Provider identity and service version	Approved provider registry, endpoint verification and contract record
Data disclosure	Pre-send classification, minimization, redaction and routing
Provider change	Contractual notification, behavioral monitoring and fallback
Availability	Multi-provider routing, queueing, degradation or approved outage behavior
Response integrity	Schema validation, content controls and anomaly detection
Evidence limitations	Record unavailable provider internals and inherited-control assumptions
Exit strategy	Portable prompts, test suites, data and alternate endpoint capability

11. Pattern F — Edge and Offline AI

Architecture area	Reference design
Model package	Signed, versioned and verified before activation
Local data	Encrypted storage, minimization and controlled synchronization
Policy	Locally enforceable controls when central services are unavailable
Telemetry	Buffered, integrity-protected events with later synchronization
Update	Authenticated, rollback-capable and staged software/model update
Degraded operation	Documented behavior during network, sensor or service loss
Physical access	Tamper resistance, secure boot and maintenance authentication

12. Pattern G — Physical AI Safety Architecture

12.1 Blueprint



12.2 Safety separation

Where practicable, critical safety constraints should be enforced by a deterministic or independently assured supervisor that does not rely solely on the same learned component whose behavior it constrains.

Failure scenario	Architectural response
Loss of model confidence	Reduce capability, request human control or enter safe state
Sensor inconsistency	Fault detection, sensor voting and restricted operation
Unsafe planned action	Independent safety supervisor blocks command
Communication loss	Local safe-state logic
Unauthorized command	Authentication failure and command rejection
Override failure	Redundant independent emergency-stop path

Relationship to PAI-SFTM. Where physical AI is in scope, GAISSTTM Domain 9 applies together with PAI-SFTM v1.0 (Physical AI Security Framework), which provides more granular physical-AI security controls. The controlled mapping is ODA3-2026-07-CSX-PAISF-004. The mapping shows lineage and correspondence; it does not establish equivalence, evidence sufficiency or automatic conformance.

13. Governance and Assurance Plane

Capability	Architecture component	Control outcome
Inventory	System/model/data/tool registry	Complete and current scope
Risk	Risk and impact register	Documented treatment and ownership
Policy	Versioned policy repository and decision engine	Consistent enforcement
Exceptions	Time-bound exception workflow	Visible residual risk
Evidence	Protected evidence repository	Traceable and reviewable operation
Testing	Independent evaluation using documented methods; VTS integration only where a released and applicable VTS asset exists	Repeatable, traceable assurance evidence
Change	Release and impact-management service	Controlled architecture evolution
External claims	Claim-management records (certification is not currently offered)	Accurate external representation

14. Observability and Evidence Architecture

Event category	Minimum attributes
Identity	Actor, agent, service, role and authentication context
Request	Time, system, purpose, channel and data classification
Policy decision	Policy version, decision, reason and obligations
Model invocation	Provider, model/version, configuration and latency
Retrieval	Sources, permissions, index version and result identifiers
Agent action	Plan step, tool, parameters, approval and outcome
Output	Response identifier, controls applied and delivery channel
Safety event	Hazard, sensor state, command, block, override and safe state
Change	Component, old/new version, approver and release
Evidence integrity	Hash, signature, timestamp and repository location

15. Resilience and Recovery Patterns

Failure	Resilience pattern
Model endpoint outage	Fallback model, queue, user notification or controlled service degradation
Policy service outage	Fail closed for high-risk actions; cached signed policy for approved low-risk use

Failure	Resilience pattern
Retrieval outage	Abstain or use clearly identified reduced-function mode
Monitoring outage	Pause expansion, restrict consequential actions or revert
Supplier behavior change	Canary comparison, anomaly detection and provider switch
Data corruption	Integrity verification, restore and re-index from approved sources
Agent runaway	Step limit, circuit breaker, credential revocation and state rollback
Physical control failure	Independent safe state, emergency stop and incident evidence

16. Architecture Control Catalogue

Control plane	Required capabilities	Failure considerations
Identity	User, service, agent and administrator identity; strong authentication	Credential theft, shared identity, orphaned access
Authorization	Contextual policy for model, data, tool and transaction	Privilege escalation, policy bypass
AI mediation	Input, model routing, output and usage policy	Direct endpoint bypass, policy drift
Data protection	Classification, lineage, access, retention and leakage prevention	Poisoning, unauthorized retrieval, cross-tenant leakage
Execution isolation	Sandboxing, network restrictions and resource limits	Escape, lateral movement, uncontrolled code
Safety	Independent constraints, override and safe state	Common-mode failure, unavailable override
Observability	Protected logs, traces, state and decisions	Blind spots, tampering, missing context
Resilience	Fallback, degradation, rollback and continuity	Unsafe fallback, stale recovery state

17. Deployment Topology Variants

Topology	Advantages	Risks and controls
Centralized gateway	Consistent policy and evidence	Concentration risk; require HA and bypass prevention
Federated gateways	Local autonomy and jurisdiction fit	Policy drift; require common standards and assurance
Private isolated platform	Data and model control	Operational complexity; require specialist competence
Managed external service	Speed and scalability	Supplier opacity; require contracts, monitoring and exit
Edge/embedded	Low latency and offline operation	Update, physical access and telemetry constraints

Topology	Advantages	Risks and controls
Hybrid physical AI	Integrated decision and actuation	Safety separation and independently assured supervisor appropriate to system risk and sector obligations

18. Threat Modelling

- Define assets, users, affected persons and unacceptable outcomes.
- Map components, data flows, trust boundaries and suppliers.
- Identify misuse, attack, failure and hazard scenarios.
- Place preventive, detective, corrective and recovery controls.
- Define evidence and monitoring for each material scenario.
- Review after material change and incident.

Threat modelling may use STRIDE, PASTA or another documented method appropriate to the system. The minimum record should include component, asset, threat, attack or failure path, consequence, control, evidence, owner and residual risk. Technology examples shall remain illustrative and non-endorsing.

19. Nonfunctional Requirements

Quality	Architecture requirement
Security	Least privilege, isolation, integrity and attack resistance
Safety	Bounded behavior, independent constraints and safe state
Privacy	Minimization, separation, retention and user rights
Reliability	Defined availability, error handling and recovery
Observability	Attributable and protected decision/event records
Portability	Controlled exit from suppliers and model substitution
Maintainability	Versioned components, automated tests and documented dependencies

20. Legal and Regulatory Integration

GAISSF provides an operational security, safety, governance and evidence framework. It does not determine which laws apply, establish statutory compliance, create a legal safe harbour, define a legal standard of care, replace competent legal advice, or guarantee that evidence collected for GAISSF purposes will satisfy a regulator, court or other authority. Organizations should maintain a jurisdiction-specific legal and regulatory obligations register and map applicable obligations to their GAISSF scope, controls, evidence, reporting processes and decision authorities.

20.1 Legal and regulatory interface requirements

- Identify applicable jurisdictions, regulators, sector rules, contractual duties and internal legal authorities before relying on this guide.
- Maintain a controlled obligations register, reporting-clock register, retention schedule, legal-hold process and conflicts-of-obligation procedure.
- Treat statutory, regulatory and contractual analysis as a parallel workstream to GAISSF implementation and assurance.
- Record the legal basis, responsible owner, scope, decision, evidence, deadline and review date for each material obligation.
- Do not represent GAISSF conformance, maturity, or any future GAISSF certification as legal approval, permission to operate, immunity from enforcement or proof that harm cannot occur.

20.2 Data residency and cross-border transfer control pattern

Control point	Architectural capability
Classification	Data category, subject, jurisdiction and restriction attributes.
Policy	Approved processing regions, transfer mechanisms and supplier locations.
Routing	Region-aware allow, deny, redact, tokenize or local-process decisions.
Key custody	Region-appropriate encryption and controlled key access.
Telemetry	Regional stores, minimized fields and controlled cross-region aggregation.
Support access	Time-bound, approved and logged cross-border administrative access.
Evidence	Transfer decision, policy version, destination, legal basis and exception record.

The architecture shall enforce the organization's approved transfer policy. It shall not assume that every transfer to a non-adequate jurisdiction is prohibited or that one jurisdiction's localization rule applies universally.

20.3 Auditable legal-policy gateway

Consequential agentic and physical actions should pass through an auditable policy decision point capable of deny, allow, require approval, restrict data, substitute a permitted route, log and escalate, or place the system in a safe state.

20.4 Decision-record and explanation telemetry

Record element	Content
Identity	System, model, policy, tool and transaction identifiers.
Context	Input, relevant retrieved sources and approved context.
Decision basis	Material rules, factors, reason codes or explanation artifact where appropriate.
Action	Output, tool call, physical command and downstream effect.
Human involvement	Review, override, approval and escalation.
Limitations	Confidence, unavailable telemetry, protected fields and withholding rationale.
Integrity	Timestamp, provenance, version linkage and tamper evidence.

21. Architecture Decision Records

Material architecture decisions should be documented using an Architecture Decision Record (ADR).

ADR field	Required content
Decision ID	Unique identifier
Context	System, risk and constraints
Decision	Selected architecture or pattern
Alternatives	Options considered
Rationale	Why the decision was selected
GAISST controls	Affected control IDs
Assumptions	Supplier, data, user and operating assumptions
Consequences	Benefits, limitations and new risks
Evidence	Design review and approval artifacts
Review trigger	Events requiring reconsideration

22. Architecture Review Method

1. Confirm system purpose, users, affected persons and consequence level.
2. Map components, data flows and trust boundaries.
3. Identify applicable GAISST controls and sector obligations.
4. Review identity, authorization, policy and human-oversight placement.
5. Review model, data, agent, supplier and physical-safety boundaries.
6. Evaluate failure, containment, rollback and safe-state behavior.
7. Confirm telemetry, evidence and incident reconstruction capability.
8. Record gaps, assumptions and approved decisions.

23. Sample Architecture Review Findings

Finding	Risk	Recommended treatment
Applications call provider directly	Policy and evidence bypass	Route through approved mediation or enforce endpoint policy
Agent uses shared admin token	Unbounded actions and poor attribution	Issue scoped per-agent credentials
RAG filter applied after retrieval	Unauthorized data enters context	Authorize before retrieval
Safety override depends on AI controller	Common-mode failure	Implement independent override path
Monitoring logs are editable by operators	Evidence tampering	Use segregated protected telemetry store

24. Reference Architecture Selection Guide

System characteristic	Recommended starting pattern
Many applications and providers	Enterprise AI Gateway
Knowledge-grounded assistant	Retrieval-Augmented Generation
Autonomous or multi-step tool use	Agentic AI with Tool Use
Organization trains or hosts models	Private Model Platform
Predominantly third-party model APIs	External Model Service
Intermittent connectivity or embedded device	Edge and Offline AI
Robotics, vehicles, machinery or actuators	Physical AI Safety Architecture

25. Architecture Anti-Patterns

Anti-pattern	Risk	Corrective pattern
Direct application-to-model access	Policy bypass and fragmented evidence	Mediated gateway or controlled service layer
Shared unrestricted agent credentials	Unbounded and unattributable actions	Per-agent identity and capability authorization
Retrieval authorization after search	Unauthorized content may enter context	Authorize before retrieval
Logging only user prompts	Insufficient reconstruction	Log policy, model, sources, tools, state and outcomes
Safety logic inside learned model only	Common-mode failure	Independent safety supervisor; deterministic where appropriate to system risk and sector requirements
Unversioned prompts and policies	Non-reproducible behavior	Controlled configuration registry
Provider treated as internal trust zone	Inherited risks become invisible	Explicit supplier boundary
Monitoring with unrestricted write access	Evidence tampering risk	Segregated assurance plane

26. Limitations

- The patterns do not represent every valid architecture.
- A reference pattern does not determine the applicability of individual controls.
- The diagrams are logical and do not define network topology or product configuration.
- Sector safety standards may require architectures not shown here.

- Architecture alone does not demonstrate operating effectiveness.

27. Notably Absent

- No mandated vendor, cloud provider, model or product.
- No assumption that centralized architecture is always preferable.
- No universal requirement to retain full prompt or output content where privacy or law prohibits it.
- No claim that an AI gateway alone establishes GAISSF conformance.
- No substitution for specialist safety architecture in physical AI.
- No guarantee that architectural controls eliminate emergent behavior or unknown threats.
- No universal prohibition on all cross-border data transfers.
- No requirement to expose raw model internals where a meaningful and lawful decision record can be provided by other means.
- No claim that a logical architecture or named technology establishes conformance.

Annex A — Architecture Inventory Template

Field	Entry
System name	
Purpose	
Owner	
Users	
Model services	
Data stores	
Agents and tools	
Suppliers	
Trust boundaries	
Physical interfaces	
Monitoring	
Evidence repository	
Applicable controls	

Annex B — Trust Boundary Record

Boundary ID	Source zone	Destination zone	Assets	Threats	Controls	Telemetry	Owner

Boundary ID	Source zone	Destination zone	Assets	Threats	Controls	Telemetry	Owner

Annex C — Architecture Decision Record Template

Field	Entry
Decision ID	
Title	
Context	
Decision	
Alternatives	
Rationale	
GAISSF controls	
Assumptions	
Consequences	
Evidence	
Approval	
Review trigger	

Annex D — Architecture Review Checklist

- ☐ Scope and component inventory complete
- ☐ Trust boundaries documented
- ☐ Model and provider versions controlled
- ☐ Data lineage and access enforced
- ☐ Agent and tool permissions bounded
- ☐ Human approval and override defined
- ☐ Failure and rollback paths tested
- ☐ Physical safe state defined where applicable
- ☐ Telemetry supports incident reconstruction
- ☐ Evidence repository protected
- ☐ Supplier boundaries and assumptions recorded
- ☐ Architecture decisions approved

Annex E — Source Traceability

Architecture subject	Controlled source
Normative requirements	GAISST-NOR-001
Control records	GAISST-NOR-004
Terminology	GAISST-NOR-005
References	GAISST-NOR-006
Release and decisions	GAISST-NOR-007 and GAISST-NOR-008
Adoption planning	GAISST-IMP-009, where published and current
Operational workflows and evidence	GAISST-IMP-010, where published and current
Deployment and rollback	GAISST-IMP-011, where published and current
Migration and crosswalk treatment	GAISST-IMP-012, where published and current

Related guidance	Document
Implementation workflows	GAISST-IMP-010, where published and current
Deployment	GAISST-IMP-011, where published and current
Evidence collection	GAISST-IMP-017, where published and current
Physical-AI controls	PAI-SF TM v1.0; ODA3-2026-07-CSX-PAISF-004

Annex F — Publication Record

Date	Change
29 June 2026	Initial reference architecture for GAISST TM v1.0.
05 October 2026	Integrated the practitioner and regulatory material into Sections 1-27 before Annexes A-F; removed the conflicting profile-scope notice; reconciled certification and public-claim language to the current position (certification is not currently offered); bounded VTS integration to released and applicable assets; added the PAI-SF TM / CSX-PAISF-004 relationship with the mapping-only boundary; made Physical-AI safety architecture proportional rather than universally deterministic; normalized independent-assessor terminology and implementation-guide dependency status. No GAISST TM control was added, removed, renamed or amended.

GAISSTTM is a framework of ODA3 Institute, referenced under the GAISST Ecosystem Licence (GEL) v1.0. | GAISST Ecosystem developed by ODA3 Institute, licensed under GEL v1.0. | © 2026 ODA3 Pvt Ltd