

TIO End-State Architecture

Architecture Decision Record Set

For architecture governance, design review, solution teams, and controlled change

Contents

01	Purpose and ADR method
02	Decision index
03	ADR-001 to ADR-004: system and foundations
04	ADR-005 to ADR-008: truth, authority, and action
05	ADR-009 to ADR-012: Delegated Autonomy, models, sovereignty, and validation
06	ADR-013 to ADR-016: threat, transition, people, and TIO domains
07	Change and review process

How to use this document

Use these records to preserve the context, options, decisions, consequences, evidence, and review triggers behind the reference architecture. Client-specific target architectures should adopt, adapt, replace, or add ADRs explicitly.

01

Purpose and ADR Method

Make architecture decisions visible and revisable

An Architecture Decision Record captures a decision that materially shapes the architecture. Each record states the context, options considered, decision, rationale, positive and negative consequences, evidence, owner, and review trigger. A decision remains active until superseded; new evidence changes the record rather than rewriting history silently.

ADR Record Structure

Field	Required content
Status	Proposed, Accepted, Superseded, or Retired.
Context	Problem, constraints, stakeholders, and assumptions.
Options	Credible alternatives, including “do nothing” where relevant.
Decision	The selected architecture position.
Rationale	Why the decision best fits the evidence and TIO principles.
Consequences	Benefits, costs, risks, responsibilities, and future constraints.
Evidence	Sources, tests, experience, and uncertainty.
Owner	Role accountable for maintaining the decision.
Review trigger	Event that requires reconsideration.
2026.08 ALIGNMENT New controlled decisions	
- Record the three enterprise outcomes and the distinction between Enterprise Autonomy and Delegated Autonomy. - Record the Work and Decision Map and Assurance and Operations lifecycle as shared cross-domain views. - Record the two primary transition routes and the rule to retire what is replaced while explicitly deciding integration, retention, coexistence, return, or stop.	

02

Decision Index

Version 1.2 accepted reference decisions

ADR	Decision	Status	Owner
ADR-001	Alpha is the integrated enterprise capability	Accepted	Leadership / Enterprise Architecture
ADR-002	Separate Intelligence Core from Enterprise Knowledge and State	Accepted	Data / Technology / Enterprise Architecture
ADR-003	Ground on governed knowledge rather than indiscriminate corpus training	Accepted	Data / Privacy / Technology
ADR-004	Use one logical Alpha with a portfolio of intelligence	Accepted	Technology / Architecture
ADR-005	Systems of record retain authoritative truth	Accepted	Business owners / Data / Technology
ADR-006	Deterministic services commit consequential transactions	Accepted	Technology / Process owner / Risk
ADR-007	Distinct identity for human and non-human actors	Accepted	Security / Identity / Architecture
ADR-008	Authority attaches to a defined unit of work	Accepted	Process owner / Accountable executive
ADR-009	Delegated Autonomy expands only through Evidence Gates	Accepted	Leadership / Factory / Risk
ADR-010	Control is the universal objective; ownership is contextual	Accepted	Leadership / CIO / Sourcing
ADR-011	Threats are architecture constraints	Accepted	Security / Architecture / Risk
ADR-012	Validation claims are gate-specific	Accepted	TSLA / Architecture / Final author
ADR-013	Enterprise Knowledge and State is realized progressively	Accepted	Data / Architecture / Transformation
ADR-014	Resilience and recovery are sovereignty controls	Accepted	Technology / Business continuity / Risk
ADR-015	People retain accountability, challenge, and appeal	Accepted	Leadership / People / Legal / Risk
ADR-016	Alpha is cross-domain, not a sixth TIO domain	Accepted	TIO system owner
ADR-017	Two primary transition routes with explicit legacy outcomes		

03

ADR-001 to ADR-004: System and Foundations

ADR-001 — Alpha is the integrated enterprise capability

Field	Record
Status	Accepted
Context	Should the enterprise architecture name one model, one platform, or the complete governed capability?
Options considered	One monolithic model; A loose collection of tools without a common operating identity; Alpha as the integrated system
Decision	Alpha is the complete Sovereign Enterprise Intelligence System, not one model or product.
Rationale	The architecture must connect intelligence, knowledge and state, identity, authority, action, assurance, edge experience, and people. Naming the integrated capability helps executives govern the whole without erasing component boundaries.
Consequences	Creates a coherent mandate and experience. Requires explicit explanation so the name does not carry the architecture by itself.
Owner	Leadership / Enterprise Architecture
Review trigger	Material change to system scope or client evidence showing the concept confuses rather than helps.

ADR-002 — Separate Intelligence Core from Enterprise Knowledge and State

Field	Record
Status	Accepted
Context	Should enterprise knowledge be embedded in model weights or governed as a separate foundation?
Options considered	Train or fine-tune the model on the full corpus; Treat all data as external tools without a coherent foundation; Separate, governed foundation with controlled retrieval and state services
Decision	Maintain two co-equal foundations inside Alpha.
Rationale	Separation preserves updateability, provenance, correction, deletion, field-level access, authoritative truth, and model replaceability.
Consequences	Adds integration and governance work. Prevents model output from silently becoming the record.
Owner	Data / Technology / Enterprise Architecture
Review trigger	New technology proves equivalent controls with materially lower complexity.

ADR-003 — Ground on governed knowledge rather than indiscriminate corpus training

Field	Record
Status	Accepted
Context	How should enterprise information shape model behaviour and answers?
Options considered	Broad training on enterprise records; Uncontrolled retrieval from all repositories; Purpose-bound retrieval with adaptation for behaviour, not wholesale memory
Decision	Use governed retrieval, current-state services, and approved adaptation.
Rationale	Current facts, records, permissions, retention, and correction are difficult to manage inside model weights.
Consequences	Requires knowledge contracts, retrieval operations, provenance, and current-state integrity.
Owner	Data / Privacy / Technology
Review trigger	New model or privacy technology materially changes correction, deletion, provenance, and access-control trade-offs.

ADR-004 — Use one logical Alpha with a portfolio of intelligence

Field	Record
Status	Accepted
Context	Should one frontier model perform all enterprise work?
Options considered	Single-model standard; Independent point solutions; Portfolio with routing across models, methods, tools, and people
Decision	Adopt a model and method portfolio behind one coherent experience.
Rationale	Different work has different capability, sensitivity, cost, latency, reliability, and portability needs.
Consequences	Requires routing, evaluations, provider management, and consistent contracts.
Owner	Technology / Architecture
Review trigger	Portfolio complexity exceeds benefit or a new standard changes the economics materially.

04

ADR-005 to ADR-008: Truth, Authority, and Action

ADR-005 — Systems of record retain authoritative truth

Field	Record
Status	Accepted
Context	Can model memory or generated output establish enterprise state?
Options considered	Model memory as the record; Duplicate AI-owned record; Authoritative business services and repositories remain the record
Decision	Systems of record and governed repositories establish truth.
Rationale	Financial, legal, identity, employment, inventory, safety, and other material state need deterministic ownership and durable records.
Consequences	AI must retrieve and cite; integration and reconciliation remain necessary.
Owner	Business owners / Data / Technology
Review trigger	A specific system is formally replaced through an approved target architecture and migration.

ADR-006 — Deterministic services commit consequential transactions

Field	Record
Status	Accepted
Context	May the Intelligence Core write enterprise state directly?
Options considered	Direct model writes; Agent writes using broad credentials; Validated tool and transaction services with explicit commit
Decision	AI interprets, plans, and prepares; authoritative services validate and commit.
Rationale	Separates probabilistic reasoning from business rules, segregation, transaction integrity, and durable records.
Consequences	Adds gateways, schemas, approvals, idempotency, compensation, and recovery design.
Owner	Technology / Process owner / Risk
Review trigger	A bounded workflow demonstrates an alternative with equal or stronger controls.

ADR-007 — Distinct identity for human and non-human actors

Field	Record
Status	Accepted
Context	Should an agent inherit the person's or service's authority automatically?
Options considered	Shared user identity; Standing service account; Distinct identities with explicit delegation and downstream validation
Decision	Identify people, agents, models, tools, services, and devices separately.
Rationale	Supports least privilege, purpose, expiry, revocation, audit, non-repudiation, and confused-deputy protection.
Consequences	Requires agent registry, workload identity, token and policy architecture, and user-visible delegation.
Owner	Security / Identity / Architecture
Review trigger	Finalized standards or proven patterns justify a different implementation while preserving outcomes.

ADR-008 — Authority attaches to a defined unit of work

Field	Record
Status	Accepted
Context	At what level is Delegated Autonomy granted?
Options considered	Grant authority to Alpha or a model; Grant authority to an agent product; Grant authority to a bounded action inside a workflow
Decision	Delegate at the unit-of-work level through an Autonomy Envelope.
Rationale	One workflow can contain low- and high-consequence steps requiring different authority.
Consequences	More granular design and monitoring; clearer accountability and safer scaling.
Owner	Process owner / Accountable executive
Review trigger	Evidence shows another granularity improves control without hiding consequence.

05

ADR-009 to ADR-012: Delegated Autonomy, Models, Sovereignty, and Validation

ADR-009 — Delegated Autonomy expands only through Evidence Gates

Field	Record
Status	Accepted
Context	How does machine authority increase?
Options considered	Executive enthusiasm; Technical readiness alone; Measured performance, control, adoption, incident, and recovery evidence
Decision	Use the TIO Evidence Gates to expand, continue, redesign, reduce, or stop Delegated Autonomy.
Rationale	Authority should reflect demonstrated operating evidence and residual risk acceptance.
Consequences	Requires baseline, thresholds, monitoring, review rhythm, and willingness to stop weak work.
Owner	Leadership / Factory / Risk
Review trigger	Client evidence supports changes to Gate sequence or criteria.

ADR-010 — Control is the universal objective; ownership is contextual

Field	Record
Status	Accepted
Context	Must every organization own models and infrastructure to be sovereign?
Options considered	Complete insourcing; Complete vendor dependence; Choose ownership and sourcing according to required control outcomes
Decision	Define sovereignty through control of mandate, knowledge, authority, evidence, operations, continuity, portability, and exit.
Rationale	Scale, risk, economics, jurisdiction, and capability differ; ownership can strengthen control but also adds operator duties.
Consequences	Requires explicit control tests and honest dependency mapping.
Owner	Leadership / CIO / Sourcing
Review trigger	Material market, law, or client evidence changes sourcing trade-offs.

ADR-011 — Threats are architecture constraints

Field	Record
Status	Accepted
Context	Should agentic threats be managed only after design?
Options considered	Security review after build; General risk list; Threat-to-control architecture with prevention, detection, response, and recovery
Decision	Integrate threat modelling into the reference architecture and workflow design.
Rationale	Prompt, context, tool, identity, memory, transaction, and supply-chain attacks exploit architecture boundaries.
Consequences	Adds control design, tests, monitoring, incident and recovery duties.
Owner	Security / Architecture / Risk
Review trigger	Threat landscape or standards change materially.

ADR-012 — Validation claims are gate-specific

Field	Record
Status	Accepted
Context	Can document QA be represented as architecture validation?
Options considered	One generic validation claim; No validation language; Separate document, design, and operational gates
Decision	Use three named validation gates and state exactly which passed.
Rationale	Prevents clean publication files from being mistaken for field-proven architecture.
Consequences	May reduce marketing force; increases credibility and creates a real evidence roadmap.
Owner	TSLA / Architecture / Final author
Review trigger	New evidence permits a stronger claim within a defined scope.

06

ADR-013 to ADR-016: Threat, Transition, People, and TIO Domains

ADR-013 — Enterprise Knowledge and State is realized progressively

Field	Record
Status	Accepted
Context	Must all enterprise information be unified before Alpha creates value?
Options considered	Enterprise-wide data program first; Uncontrolled point use; Priority workflow slice, domain foundation, shared patterns, current-state integration
Decision	Use a partial-credit, federated realization path.
Rationale	Avoids waiting for perfection while preserving governance and reusable capability.
Consequences	Requires discipline to prevent each slice becoming a new silo.
Owner	Data / Architecture / Transformation
Review trigger	Client evidence shows a different sequence is safer or faster.

ADR-014 — Resilience and recovery are sovereignty controls

Field	Record
Status	Accepted
Context	Is normal-operation performance sufficient evidence of control?
Options considered	Availability only; Provider SLA only; Anticipate, withstand, recover, adapt, and preserve manual or alternate paths
Decision	Treat continuity, degraded operation, rollback, compensation, reconciliation, and failover as core architecture.
Rationale	An organization cannot control a capability it cannot stop, recover, or replace.
Consequences	Adds cost and operational rehearsal; may reduce short-term efficiency.
Owner	Technology / Business continuity / Risk
Review trigger	Mission or service-level requirements change.

ADR-015 — People retain accountability, challenge, and appeal

Field	Record
Status	Accepted
Context	Can automation remove human responsibility or affected-person recourse?
Options considered	AI accountability language; Human approval everywhere; Named accountability with meaningful oversight, challenge, appeal, and role design
Decision	Preserve human authority and rights according to consequence, law, trust, and legitimacy.
Rationale	Technical capability does not create legitimate authority or accountability.
Consequences	Requires Workforce Charter, training, human support, and careful monitoring design.
Owner	Leadership / People / Legal / Risk
Review trigger	Law, workforce evidence, or client decisions change the boundary.

ADR-016 — Alpha is cross-domain, not a sixth TIO domain

Field	Record
Status	Accepted
Context	Should the architecture become a new TIO domain?
Options considered	Create AI/Alpha domain; Place Alpha entirely in Technology; Build Alpha through the existing five domains
Decision	Leadership directs. Data informs. Technology enables. Factory produces. People shape and sustain.
Rationale	TIO capabilities remain enterprise capabilities for the age of AI; the architecture is an outcome of their coordination.
Consequences	Cross-domain ownership and dependencies must be explicit.
Owner	TIO system owner
Review trigger	Approved change to the TIO Capability Model.

07

Change and Review Process

How decisions evolve without losing traceability

1. Open a new decision or change proposal with context and evidence.
2. Identify affected documents, contracts, controls, workflows, providers, and stakeholders.
3. Review credible alternatives and consequences rather than editing the decision statement only.
4. Obtain the relevant architecture, business, data, security, legal, risk, people, and executive decisions.
5. Mark the existing ADR Accepted, Superseded, or Retired; do not delete it.
6. Update dependent artifacts, evaluations, controls, and transition plans.
7. Revalidate the affected architecture and document gates.
8. Record the decision and review trigger in the manifest and changelog.