TIRESIAS-ZT + ASPHODEL · COMPLIANCE / GRC

Auditable by construction, not by assurance.

Regulators no longer accept policy documents as proof. They ask who did what, under which authority, with what data, and can you show the signed record. This system answers that at the call, every call.

Asphodel + Tiresias-ZT · ACME Corporation

THE STAKES

What a regulator asks for is evidence, and the burden is now yours.

The EU AI Act, sectoral data-residency rules, and every internal audit share one demand: reconstruct any AI action after the fact, with attribution and data lineage. Three obligations you carry:

WHO	**Per-call attribution to a real principal** Which identity acted, from which department segment, under which capability — not "a service account."	`325,154 principals`
WHERE	**Data isolation you can demonstrate, not assert** Every agent's memory provably reachable only by that agent — enforced in the database, not the app.	`RLS forced`
WHAT	**A tamper-evident record of every governed action** Cryptographically chained, verifiable independently, durable across restart.	`Ed25519 chain`

> Evidence, not promises. The rest of this deck is where the evidence lives.

WHAT IS UNDER GOVERNANCE

367 AI agents doing real work — every one a governed tenant.

This is not a pilot with a wrapper. It is a live workforce mapped like-for-like onto the real ACME org, running under deny-by-default zero-trust policy on GCP.

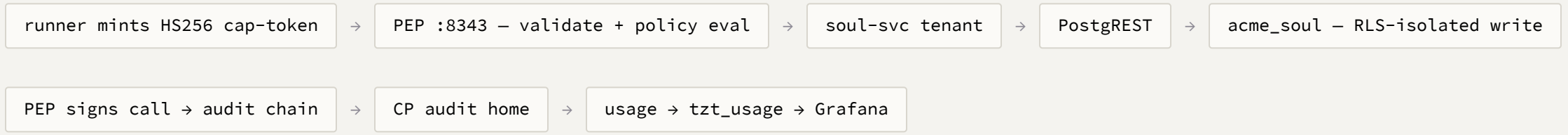
367	360	7	0
GOVERNED AGENTS	FLEET · ALL 28 DEPTS	SHOWCASE HAND-OFF DAG	UNGOVERNED PATHS

Source: registry (367 agent tenants, postgres-backed) · fleet_runner.py, 360 agents mapped dept × role-family across all 28 departments · personas.json 7-agent 14-edge DAG.

THE ARCHITECTURE SLICE THAT MATTERS TO YOU

Every action passes one enforcement point — and signs itself on the way through.

There is no side door. A governed job cannot reach data without minting a capability, being evaluated deny-by-default, and being written into the signed audit chain. The path is the control.



Source: governed request flow (one job) — X-Tiresias-Api-Key + X-Capability-Token required; policy deny-by-default; SoulKey Ed25519 audit chain to CP.

LIVE PROOF · NOT A SLIDE CLAIM

The controls are running right now. These are their real coordinates.

Nothing here is illustrative. Each item is a live endpoint, store, or key you can inspect during the walkthrough.

AUDIT	Signed chain + durable write-ahead log SoulKey Ed25519 signatures; WAL at /data/audit-wal; ships to CP audit home.	/cp/audit
ISOLATION	acme_soul with RLS forced _memories and _soul_api_keys carry FORCE RLS; cross-tenant needs SET ROLE service_role.	Cloud SQL :5432
IDENTITY	Real directory synced to policy store Samba-AD → sync_ldap.py → tzt_principal, segment:<dept> → policy.	ldaps://…:636
REVIEW	Analyst console: audit/verify, soulwatch, quarantine PEP /admin/*, role-token gated, at pdp.tiresias.watch.	PEP :8343

Source: Plane 3 (PEP), Plane 4 (data), Plane 1 (identity), Review/analyst console — ARCHITECTURE ground truth.

WALKTHROUGH 1 · THE SIGNED AUDIT CHAIN

Tamper-evident, per-call attribution — verifiable without trusting us.

The PEP signs every governed call into a hash-linked chain using an Ed25519 SoulKey. Any alteration breaks the chain, and the review console verifies it independently. This is what turns "we log things" into evidence.

WHAT GETS SIGNED

Each call: the acting agent tenant, the api-key + capability-token presented, the policy decision, and the forwarded action — chained to the prior entry so the sequence itself is attested.

WHERE IT SURVIVES

Durable write-ahead log at /data/audit-wal on the PEP, shipped to the CP audit home (postgres-backed, survives restart). Verification runs from the analyst console at /admin — audit/verify.

Source: Plane 3 — signed audit chain (SoulKey Ed25519), WAL /data/audit-wal, ships to CP; Review console /admin/* audit/verify.

WALKTHROUGH 2 · FORCED DATA ISOLATION

Isolation the database enforces — not the application asking nicely.

Every agent is its own soul-svc tenant. In acme_soul, Row-Level Security is FORCED: even the postgres role is subject to it. An agent physically cannot read another agent's memory, regardless of application bugs.

TENANT	One tenant per agent _soul_tenants + _soul_api_keys, key_hash = sha512(bearer). 367 isolated tenants.	per-agent
FORCE RLS	_memories and _soul_api_keys enforce RLS unconditionally postgres on Cloud SQL is subject to FORCE RLS — no superuser bypass.	acme_soul
ESCALATE	Cross-tenant reads are an explicit, auditable act Requires SET ROLE service_role — deliberate, not incidental.	service_role

> "Trust the app to filter by tenant" is a promise. FORCE RLS is a control you can demonstrate to an auditor.

Source: Plane 4 — per-agent isolation, FORCE RLS on _memories/_soul_api_keys, SET ROLE service_role for cross-tenant ops.

WALKTHROUGH 3 · IDENTITY LINEAGE → POLICY

Every agent traces to a real principal in a real directory.

Policy is not attached to anonymous service accounts. The real ACME directory is synced into the control-plane identity store, and each principal's department becomes a policy segment. Authority has provenance.

325,154	120,120	205,034	28 / 626
PRINCIPALS	HUMANS	NHI / SERVICE IDENTITIES	DEPARTMENTS / ROLES

`Samba-AD directory` → `sync_ldap.py` → `tzt_principal (segments)` → `segment:<dept> → policy`

Source: Plane 1 — 325,154 principals (120,120 humans + 205,034 NHI), 28 depts / 10 regions / 626 roles; sync_ldap.py → tzt_principal; segment:<dept> → policy.

WALKTHROUGH 4 · CONTENT PROVENANCE AT WRITE

Nothing enters memory without declaring its origin.

ContentInspection runs in enforce mode on the data plane. A write is rejected unless it carries both source_type and authored_by — so every stored fact answers "where did this come from" and "who authored it" before it is ever retrieved or acted on.

REQUIRED ON EVERY WRITE

source_type — the provenance class of the content.
authored_by — the attributable author.
Missing either → the write does not land.

WHY GRC CARES

It closes the provenance gap that makes AI outputs un-auditable. Downstream, every memory carries its lineage, so a regulator's "where did the model get this" has a stored answer.

Source: Plane 4 — ContentInspection (enforce) requires source_type + authored_by on writes to acme_soul.

GOVERNANCE · CONTINUOUS, NOT PERIODIC

Drift detection, quarantine, and SIEM — the controls watch themselves.

Attribution is the record; SoulWatch is the live guard. Behavioral drift triggers automatic quarantine, and every event streams to your SIEM in standard syslog — so oversight is continuous and exportable, not a quarterly review.

DRIFT	SoulWatch behavioral drift → auto-quarantine Shared postgres quarantine store; status + release from the analyst console.	auto
SIEM	RFC5424 syslog → Promtail → Loki Standard format into your existing security tooling.	:1514 → :3100
POLICY	Deny-by-default; allow forwards verbatim No implicit permissions; the default answer is no.	PEP :8343
CHANGE	ITSM change control on record Contract-B change-svc for governed change management.	:8085

Source: Plane 3 (SoulWatch, deny-by-default, RFC5424 syslog → Promtail:1514 → Loki:3100) · Plane 5 · acme-change-svc :8085.

THE AUDIT SURFACE IS ALSO THE COST SURFACE

The same governed record proves spend per model, per call.

Because every call is metered through the PEP into tzt_usage, the evidence trail doubles as verifiable economics — the self-host-versus-API answer, observed, not estimated.

MODEL TIER	OBSERVED $/CALL
opus-4.8 (paid)	$0.19
sonnet-4.5 (paid)	$0.03
deepseek-v4-pro	$0.002
gpt-oss:120b	$0.0015
Llama-70B (HF)	$0.0012
gemma:free	$0

Source: ROI dashboard (uid acme-roi-economics) — observed per-model $/call from tzt_usage; Grafana :3000.

CLOSE · COMPLIANCE / GRC

Evidence, not promises.

A signed, tamper-evident chain for every call. Data isolation the database forces. Every action traced to a real principal across 325,154 identities. Provenance required before a fact is stored. When the regulator asks, the answer is already written down — and signed.

- 367 governed agents · deny-by-default · auditable by construction · ACME Corporation