AGENT OS ◆ · FUTURE ARCHITECTURE

Two paths from *#0 demo* to a fully scalable Agentic platform.

Agent OS demonstrably runs the full AGT-04 cash-app and AGT-08 HITL flow against a Sim of RecVue. The Memory Layer is live. The Domain Event Store ships this week. The Tools Registry is schema-ready. The Connectors are in active implementation.

What remains is a target architecture that takes Agent OS from *demo-quality #0* to a production platform integrated with the entire RecVue customer base. Two distinct shapes are viable.

This document draws both — at three points (today, Architecture A, Architecture B) — using the same 6-band stack diagram. Same shape. Different tech in each box.

HOW TO READ THIS DOCUMENT

Three lenses on the same architecture.

The 5-minute take

Read pages 1–4 (cover · this page · principles · today's architecture) and pages 6 + 9 (Architecture A and B diagrams). The two architecture diagrams are visually identical in shape; only the per-module tech labels differ.

The 30-minute deep dive

Read it cover-to-cover.

- **Page 3** documents the architectural principles that hold across both architectures.
- **Page 4** shows where Agent OS is today.
- **Pages 5–7** walk through Architecture A — Agent OS extended.
- **Pages 8–10** walk through Architecture B — cloud-native multi-tenant.
- **Page 11** is the strangler-fig argument: A is on the path to B; the path is composed of port-by-port adapter swaps.

The lens convention

Pages with a **LENS: A** badge show Architecture A's tech labels. Pages with a **LENS: B** badge show Architecture B's. The diagram *shape* is identical between lenses; only the per-module labels differ.

ARCHITECTURAL PRINCIPLES

Eight cross-cutting traits hold across *both* architectures.

These are invariants. Architecture A and B differ only in adapter implementation — not in principles.

FCIS · Functional Core / Imperative Shell

Pure functional core (workflows, operations, types) with side-effecting shell (routes, repositories, adapters). Every module is built this way. Architectural law.

Event Shell · StreamAdapter · Domain Event Store

Triple substrate — Event Shell dispatches workflows, StreamAdapter captures every step, Domain Event Store translates raw events into domain events with projections, triggers, and audit log.

OTel Observability · Traceability

Every workflow run carries OTel correlation + causation IDs. Spans hit Tempo; metrics hit Prometheus; logs hit Loki. Same in A and B; B adds multi-region aggregation.

Cost governance · RBAC + audit

Per-tenant cost guardrails, RBAC enforcement, immutable audit logs across every action. A uses Postgres + Merkle proof chain; B uses QLDB.

Strangler-fig swappability

Every module owns one or more *ports*. A and B differ only in which *adapter* is plugged into each port. Migration is per-port adapter swap, not big-bang rewrite.

TODAY'S ARCHITECTURE

Where Agent OS is right now.

6-band stack with the Domain Event Store as a backbone band. Status dots: ● live · ◐ partial · ◐ in-progress · ○ designed · ○ missing.

SOURCES

- ● Integration Connectors · Connector-per-provider scaffold

MEDIATION HUB

- ◐ Connector Mesh · Markdown DropZone (only path live)
- ◐ Document Understanding · StubDocUnderstandingAdapter (canned payloads keyed by attachmentRef)
- ○ Integration Engine · Designed but deferred (per Connectors spec's Out-of-Scope)

DOMAIN EVENT STORE

- ◐ Domain Event Store · Translators + projections + triggers designed
- ● StreamAdapter · Live in production

INTEL & MEMORY

- ● Knowledge Graph · Drizzle + Postgres + JSONB
- ● Context Graph · workflow_events + LangGraph PostgresSaver checkpointer + WS relay
- ● Vector & Semantic Store · pgvector + HNSW (1024-dim Bedrock Titan v2)
- ○ Episodic Memory · Designed but not built (deferred from Memory Layer v1)
- ● Evaluation Framework · Eval harness in production

AGENT RUNTIME

- ● Orchestrator (AGT-01) · Event Shell dispatcher + LangGraph runtime + cost guardrails
- ◐ 8 Specialists (AGT-02..09) · AGT-04 Cash App + AGT-08 HITL shipped via #0 demo
- ○ Internal Tools Registry · Schema designed (tools, workflow_version_tools, credentials, mcp_servers)

EXPERIENCE & TRUST

- ● HITL Cockpit · React + TanStack Query + Zustand
- ○ CS & Finance Copilot UI · retrieve_kg_context tool registered
- ● Reasoning Graph editor · React Flow custom node registry
- ◐ Audit & Explainability · Workflow run viewer + KG slice replay
- ● Public APIs · Webhooks · OpenAPI 3.1 spec

JAVA TOOLS LAYER

- ● *Java Tools Layer* · *Not started*

CROSS-CUTTING: Functional Core / Imperative Shell · Event Shell · StreamAdapter · OTel Observability · Traceability (correlation + causation) · Cost governance · RBAC + audit · Strangler-fig swappability

● live ◐ partial ◐ in-progress ○ designed ○ missing

AGENT OS ◆

Architecture A.
Agent OS *extended*.

ARCHITECTURE A · DIAGRAM

Agent OS extended; single-instance.

LENS: A

Same band layout. Per-module tech labels show Architecture A's stack — the current Agent OS infrastructure plus the missing capabilities.

SOURCES

- Integration Connectors · Same scaffold + 4-6 production providers (HubSpot, Stripe, SAP, etc.)

MEDIATION HUB

- Connector Mesh · Per-connector route handlers
- Document Understanding · BedrockDocUnderstandingAdapter (Claude multimodal)
- Integration Engine · PgBoss-orchestrated sync + LangGraph mapping workflow + bulk Java API delivery

DOMAIN EVENT STORE

- Domain Event Store · Same as today (Postgres-backed event store + PgBoss projections)
- StreamAdapter · Same as today + Domain Event Store integration (translator hook)

INTEL & MEMORY

- Knowledge Graph · Same as today
- Context Graph · Same as today
- Vector & Semantic Store · Same as today (pgvector + HNSW)
- Episodic Memory · Postgres episodic table
- Evaluation Framework · Same as today + per-tenant eval suites + drift dashboards

AGENT RUNTIME

- Orchestrator (AGT-01) · Same as today + typed planner.run / policy.check / route.dispatch contract enforced
- 8 Specialists (AGT-02..09) · All 8 specialists live or in HITL-assistive mode
- Internal Tools Registry · Migrated tables

EXPERIENCE & TRUST

- HITL Cockpit · Same as today + multi-team queues + assignment routing
- CS & Finance Copilot UI · React chat UI + reply.draft / cite.ground tools + KG slice citations
- Reasoning Graph editor · Same editor + workflow-version diff view + reasoning replay
- Audit & Explainability · Above + Merkle-tree-on-Postgres proof chain + immutable WORM logs
- Public APIs · Webhooks · Same + webhook emission for case-resolved / extraction-complete events

JAVA TOOLS LAYER

- *Java Tools Layer · Spring Boot sidecar per RecVue microservice (11 sidecars)*

CROSS-CUTTING: Functional Core / Imperative Shell · Event Shell · StreamAdapter · OTel Observability · Traceability (correlation + causation) · Cost governance · RBAC + audit · Strangler-fig swappability

WHAT A SOLVES

The full Agent OS capability map at the same scale tier as today. Agents work end-to-end against real RecVue data via Java Tools sidecars. CSRs have a working Cockpit. New customer integrations use the Integration Engine. Audit chain via Merkle-tree-on-Postgres.

WHAT A DOES NOT SOLVE

Multi-tenant runtime (still requires per-tenant deployment), geographic scale (single region), compliance ceiling beyond Merkle-on-Postgres, and the 1.2M-LOC RecVue platform's own remediation.

ARCHITECTURE A · DELTA

What changes from today to A.

20-row table; one row per module. Effort is a T-shirt size for the work to ship A's additions on top of today's state.

MODULE	TODAY	ARCHITECTURE A	EFFORT
Integration Connectors	Connector-per-provider scaffold; Salesforce + NetSuite designed	Same scaffold + 4-6 production providers (HubSpot, Stripe, SAP, etc.) + Production-ready auth flows · Webhook ingestion endpoints · Provider-specific schema discovery	L
Connector Mesh	Markdown DropZone (only path live); CSV pipeline as workflow nodes	Per-connector route handlers; webhook intake; Kafka-bridged buffering + Webhook signature validation · Per-source rate limiting · Dead-letter queues	M
Document Understanding	StubDocUnderstandingAdapter (canned payloads keyed by attachmentRef)	BedrockDocUnderstandingAdapter (Claude multimodal) + Vision-OCR for PDF receipts · Email/Slack/EDI extraction	M
Integration Engine	Designed but deferred (per Connectors spec's Out-of-Scope)	PgBoss-orchestrated sync + LangGraph mapping workflow + bulk Java API delivery + AI mapping assistant · Transform DSL · Sync run audit	XL
Domain Event Store	Translators + projections + triggers designed; ships this week	Same as today (Postgres-backed event store + PgBoss projections) + Per-projection retry policy tuning · Phase-2 temporal-window triggers · Phase-3 saga choreography	M
StreamAdapter	Live in production; wraps every LangGraph workflow run	Same as today + Domain Event Store integration (translator hook) + Translator registry hooks · Per-event causationId chaining	S
Knowledge Graph	Drizzle + Postgres + JSONB; React Flow editor; pgvector for similarity within KG	Same as today; expanded to Phase-2 entity types (Customer, Contract, Product) + Phase-2 entity types · Slice tokenization tuning · Multi-hop expansion	M
Context Graph	workflow_events + LangGraph PostgresSaver checkpointer + WS relay	Same as today; formalized workflow_snapshots view (per Domain Event Store design) + workflow_snapshots view · Cross-run state diffing	S
Vector & Semantic Store	pgvector + HNSW (1024-dim Bedrock Titan v2)	Same as today (pgvector + HNSW); no scale swap + Per-tenant index partitioning · Re-embed batch cron	S
Episodic Memory	Designed but not built (deferred from Memory Layer v1)	Postgres episodic table; per-customer/per-agent rolling windows; PgBoss-driven decay + Episodic schema · Per-tenant window sizing · Overrides → confidence floor adjustment	M
Evaluation Framework	Eval harness in production; per-graph eval runs	Same as today + per-tenant eval suites + drift dashboards + Per-tenant eval suites · Drift alerting	S
Orchestrator (AGT-01)	Event Shell dispatcher + LangGraph runtime + cost guardrails	Same as today + typed planner.run / policy.check / route.dispatch contract enforced + Typed contract enforcement · Per-customer policy bundles	M
8 Specialists (AGT-02..09)	AGT-04 Cash App + AGT-08 HITL shipped via #0 demo; others PDF-spec only	All 8 specialists live or in HITL-assistive mode; LangGraph + Bedrock + Build out AGT-02/03/05/06/07/09 · Per-agent eval suites	XL
Internal Tools Registry	Schema designed (tools, workflow_version_tools, credentials, mcp_servers); db tables not migrated yet	Migrated tables; full polymorphic registry with credential encryption (AES-256-GCM) + Tool versioning UI · MCP sync cron · Per-tenant credential scoping	M
HITL Cockpit	React + TanStack Query + Zustand; shipped via #0 demo	Same as today + multi-team queues + assignment routing + Multi-team queues · Skill-based routing · SLA dashboards	M
CS & Finance Copilot UI	retrieve_kg_context tool registered; UI not built	React chat UI + reply.draft / cite.ground tools + KG slice citations + Chat UI · Citation viewer · Per-tenant copilot personas	L
Reasoning Graph editor	React Flow custom node registry; live in agentic-os	Same editor + workflow-version diff view + reasoning replay + Version diff view · Replay scrubber	M
Audit & Explainability	Workflow run viewer + KG slice replay; no proof chain yet	Above + Merkle-tree-on-Postgres proof chain + immutable WORM logs + Merkle-tree append-only ledger · Per-action proof verification UI	L
Public APIs · Webhooks	OpenAPI 3.1 spec; /api/v1/* routes for KG; webhooks not yet emitted	Same + webhook emission for case-resolved / extraction-complete events + Webhook emitter · HMAC signing · Retry/dead-letter	M
Java Tools Layer	Not started; design only	Spring Boot sidecar per RecVue microservice (11 sidecars); MCP server + REST + 11 sidecars · Per-microservice tool registration · RecVue API auth bridging	L

AGENT OS ◆

Architecture B.
Cloud-native, *multi-tenant*.

ARCHITECTURE B · DIAGRAM

Cloud-native, multi-tenant from day one.

LENS: B

Identical band layout to Architecture A. Per-module tech labels swapped — Weaviate, Neo4j, Confluent Kafka, K8s multi-region, Okta JWT + RLS, QLDB.

SOURCES

- Integration Connectors · Same scaffold

MEDIATION HUB

- Connector Mesh · K8s connector mesh with sidecar buffering
- Document Understanding · Same Bedrock adapter + Lambda-based pre-processing pipeline
- Integration Engine · Kafka-orchestrated sync + Flink streaming transforms + per-tenant K8s engine instances

DOMAIN EVENT STORE

- Domain Event Store · EventStoreDB (or Postgres-per-service partition) + Confluent Kafka backbone + Flink projections
- StreamAdapter · Multi-region StreamAdapter with Kafka triple-output (DB + Kafka + OTel)

INTEL & MEMORY

- Knowledge Graph · Neo4j (graph) + Apache Atlas (ontology versioning) + GraphQL federation gateway
- Context Graph · EventStore + Temporal for workflow state
- Vector & Semantic Store · Weaviate (managed
- Episodic Memory · DynamoDB for low-latency lookups + Postgres for analytics
- Evaluation Framework · Same harness running on K8s eval cluster

AGENT RUNTIME

- Orchestrator (AGT-01) · K8s Orchestrator pods with sticky-tenant routing
- 8 Specialists (AGT-02..09) · Same agents on K8s agent runtime + multi-region failover
- Internal Tools Registry · Same registry + service-mesh tool gateway (Istio/Envoy) for MCP traffic

EXPERIENCE & TRUST

- HITL Cockpit · Same React app + Okta SSO + per-tenant theming
- CS & Finance Copilot UI · Same UI + per-tenant Bedrock models + Pinecone or Weaviate-backed retrieval
- Reasoning Graph editor · Same editor + collaborative editing (Yjs/Liveblocks)
- Audit & Explainability · Above + QLDB (immutable ledger) + Merkle proof chain + SEC 17a-4 retention
- Public APIs · Webhooks · Same + Kong API Gateway + per-tenant rate limits + GraphQL federation

JAVA TOOLS LAYER

- *Java Tools Layer · Same sidecars deployed in K8s with parent microservice*

CROSS-CUTTING: Functional Core / Imperative Shell · Event Shell · StreamAdapter · OTel Observability · Traceability (correlation + causation) · Cost governance · RBAC + audit · Strangler-fig swappability

WHAT B UNLOCKS

Multi-tenant runtime; one platform serves all customers. Geographic scale; multi-region active-active. Compliance ceiling; QLDB-backed cryptographic proof chain meets SEC 17a-4. Cost elasticity; K8s autoscale + managed services scale per-tenant load. Independent module velocity; domain services on K8s ship independently.

RISKS

Vendor lock-in (Weaviate, Neo4j, Confluent are commercial managed services). Operational complexity. Migration cost (every adapter rewritten, though FCIS core stays). RecVue's own platform must reach a state where multi-tenant integration is meaningful.

ARCHITECTURE B · DELTA

What changes from A to B.

20-row table; one row per module. Effort is the swap cost — every adapter rewritten, but the FCIS core stays.

MODULE	ARCHITECTURE A	ARCHITECTURE B	EFFORT
Integration Connectors	Same scaffold + 4-6 production providers (HubSpot, Stripe, SAP, etc.)	Same scaffold; runs in K8s connector pool with horizontal autoscaling + Per-tenant connector instances · Kafka-bridged delivery	M
Connector Mesh	Per-connector route handlers; webhook intake; Kafka-bridged buffering	K8s connector mesh with sidecar buffering; Kafka topics per-tenant + Multi-region buffering · Tenant-aware routing	M
Document Understanding	BedrockDocUnderstandingAdapter (Claude multimodal)	Same Bedrock adapter + Lambda-based pre-processing pipeline; per-tenant model selection + Per-tenant fine-tuned models (optional)	S
Integration Engine	PgBoss-orchestrated sync + LangGraph mapping workflow + bulk Java API delivery	Kafka-orchestrated sync + Flink streaming transforms + per-tenant K8s engine instances + Streaming transforms · Backpressure handling	L
Domain Event Store	Same as today (Postgres-backed event store + PgBoss projections)	EventStoreDB (or Postgres-per-service partition) + Confluent Kafka backbone + Flink projections + Multi-region event replication · Per-tenant event partitions	L
StreamAdapter	Same as today + Domain Event Store integration (translator hook)	Multi-region StreamAdapter with Kafka triple-output (DB + Kafka + OTel) + Cross-region replication · Schema-Registry-backed event payloads	M
Knowledge Graph	Same as today; expanded to Phase-2 entity types (Customer, Contract, Product)	Neo4j (graph) + Apache Atlas (ontology versioning) + GraphQL federation gateway + Multi-region replication · Per-tenant ontology versioning	L
Context Graph	Same as today; formalized workflow_snapshots view (per Domain Event Store design)	EventStore + Temporal for workflow state; Postgres-per-service partitions + Multi-region workflow execution · Workflow versioning	L
Vector & Semantic Store	Same as today (pgvector + HNSW); no scale swap	Weaviate (managed; native hybrid keyword+vector search) + Per-tenant Weaviate classes · Multi-region replication	L
Episodic Memory	Postgres episodic table; per-customer/per-agent rolling windows; PgBoss-driven decay	DynamoDB for low-latency lookups + Postgres for analytics + Per-tenant DynamoDB tables · Cross-region replication	M
Evaluation Framework	Same as today + per-tenant eval suites + drift dashboards	Same harness running on K8s eval cluster; Iceberg-backed eval result store + Multi-region eval execution · Long-term drift analytics	M
Orchestrator (AGT-01)	Same as today + typed planner.run / policy.check / route.dispatch contract enforced	K8s Orchestrator pods with sticky-tenant routing + Per-tenant orchestrator pools · Tenant-aware rate limits	M
8 Specialists (AGT-02..09)	All 8 specialists live or in HITL-assistive mode; LangGraph + Bedrock	Same agents on K8s agent runtime + multi-region failover + Per-tenant agent pools · Tenant-aware model routing	L
Internal Tools Registry	Migrated tables; full polymorphic registry with credential encryption (AES-256-GCM)	Same registry + service-mesh tool gateway (Istio/Envoy) for MCP traffic + Service-mesh-routed MCP · Per-tenant tool quotas	M
HITL Cockpit	Same as today + multi-team queues + assignment routing	Same React app + Okta SSO + per-tenant theming + Per-tenant theming · SSO hardening	S
CS & Finance Copilot UI	React chat UI + reply.draft / cite.ground tools + KG slice citations	Same UI + per-tenant Bedrock models + Pinecone or Weaviate-backed retrieval + Per-tenant model fine-tuning hooks · Streaming response	M
Reasoning Graph editor	Same editor + workflow-version diff view + reasoning replay	Same editor + collaborative editing (Yjs/Liveblocks) + Multi-user collaboration · Tenant-aware sharing	M
Audit & Explainability	Above + Merkle-tree-on-Postgres proof chain + immutable WORM logs	Above + QLDB (immutable ledger) + Merkle proof chain + SEC 17a-4 retention + QLDB integration · Per-tenant retention policy	L
Public APIs · Webhooks	Same + webhook emission for case-resolved / extraction-complete events	Same + Kong API Gateway + per-tenant rate limits + GraphQL federation + Kong gateway · Federated GraphQL · Per-tenant SLAs	M
Java Tools Layer	Spring Boot sidecar per RecVue microservice (11 sidecars); MCP server + REST	Same sidecars deployed in K8s with parent microservice + K8s pod co-location · Service-mesh-routed MCP traffic	S

THE STRANGLER-FIG PATH · A → B

A is on the path to B.

Both architectures share the same FCIS core. They differ only in which adapter is plugged into each port. Migration from A to B is per-port adapter swap, not big-bang rewrite.

PORT	ARCHITECTURE A ADAPTER	ARCHITECTURE B ADAPTER
`GraphStore`	DrizzlePostgresGraphStore	Neo4jGraphStore
`EmbeddingProvider`	BedrockTitanEmbedder + pgvector	BedrockTitanEmbedder + Weaviate
`EventStream`	PgBossEventStream	KafkaConfluentEventStream
`EventStore`	PostgresEventStore	EventStoreDB / Postgres-per-svc partition
`Cache`	(none — UNLOGGED tables)	RedisAdapter
`AuditLedger`	MerkleTreePostgresLedger	QLDBLedger
`MultiTenancy`	TenantHeaderMiddleware + AsyncLocalStorage	OktaJwtClaimMiddleware + RLS
`LakehouseTier`	(none — warm-tier in Postgres)	IcebergLakehouseAdapter

THE ARGUMENT

A team that completes Architecture A is a team that has built every Architecture B port; it has just not swapped the adapter yet. A and B are not competing designs — they are *snapshots* of the same architecture at two different rungs of the scale-evolution ladder.

CLOSING

What happens next.

The conversation tree

1. You read this document.
2. You decide whether Architecture A is the right next step (or whether to skip directly to B).
3. If A: per-module specs are brainstormed for the missing capabilities (Java Tools Layer, Connector Mesh expansion, full agent fleet).
4. If B: the strangler-fig migration plan is brainstormed; per-port adapter-swap specs follow.

Either path inherits the same FCIS core, the same Domain Event Store backbone, the same observability discipline.

Reference material

The exec briefing covering RecVue's audit findings and the 8 sub-project portfolio: `docs/architecture/exec-briefing/briefing.pdf`

The Agentic AI soundness audit (the current-state document this future architecture extends): `docs/architecture/agentic-ai-soundness/`

This document's design spec: `docs/superpowers/specs/2026-04-27-future-architecture-design.md`

Module catalog (the JSON source of truth): `docs/architecture/future-architecture/module-catalog.json`

Efsora Labs · efsora.com · info@efsora.com