TECHNICAL BLUEPRINT · FLAGSHIP REPORT

The Mid-Market *Connected Planning* Blueprint

How organisations with 100 to 1,500 employees eliminate the manual latency between the ERP general ledger and the operational drivers that actually run the business — and the reference architecture that makes it possible on one platform. This blueprint describes EAConnect Planning as it is documented and code-reviewed: tenancy, master data, the model fabric, the governed write path, the dual-engine query runtime, processes and workflow, the AI layer, and the lifecycle that moves an application from sandbox to production. It closes with the architectural decisions an implementation team must make and the constraints it should know about.

AUDIENCE	SCOPE	BASIS	READING TIME
Leaders and architects at mid-size enterprises	One platform: integration · planning · analytics · AI	25 canonical domain documents, code-reviewed Jun 2026	23 min read

Series Hours, Not Months · Flagship

Standard architecture as documented; performance figures only where measured; specified-but-unshipped items labelled

00 Executive summary

Connected planning is a data-flow property, not a feature. It means that when the ledger closes, the plan knows; when a driver changes, the statement moves; and nobody carried a file between the two.

Mid-size companies typically run planning on three products they did not intend to buy as a system: an ERP that holds the truth, a planning tool that holds the plan, and a BI tool that holds the charts — joined by exports, a shared drive and one person's Tuesday. The latency between the ledger and the drivers is measured in days; the reconciliation between the planning numbers and the reporting numbers is a standing argument; and every change to the organisation is a change to three tools.

The architecture in this blueprint removes the seams by putting the four functions on one substrate. Its load-bearing properties, each documented in the sections that follow:

§02–03

Hard tenancy, one workspace per application

Organisation → environment (isolated schema and storage) → application (a business grouping with exactly one analytics workspace). Sandboxes are parallel stacks; promotion moves structure by code.

§04–05

Shared master data, several models, one fabric

Environment-shared dimensions with versioned hierarchies; models at their own grain publishing into each other; planning tables for drivers and rates; views as the single read/write contract.

§06–07

One governed write path

Every fact write — from a form, a process, a workflow, an agent or an API — passes validation, data-access control, slice locks and cell-level audit through one service, with an actor and a correlation id.

§08–09

Integration inside; two engines underneath

Connectors, flows and schedules are part of the product. PostgreSQL holds the transactional core; DuckDB executes analytics in-process per workspace over Parquet; a bridge exposes planning objects to both.

§10–11

Processes, workflow, three layers of validation

A process framework with a finance-operations library; BPMN flows and schedules; human-in-the-loop workflow apps; validation that blocks bad writes, checks readiness before gates, and manages exceptions.

§12

AI as a governed caller, not a bypass

MCP tools exposed opt-in and audited; agents that run as named users and can only file findings; a configuration agent that builds applications from a spec through the same services, never around them.

The blueprint is written for two readers at once. Leaders will find, in each section, what the property buys the business. Architects will find the mechanism, the object model and the decisions to make. Section 15 collects the decisions; section 16 the constraints.

01 The latency problem

The general ledger closes on working day five. The headcount plan was updated on the twelfth. The board pack was built on the twentieth from a version of both that was already stale. That gap is the cost this blueprint exists to remove.

The distance between the ledger and the drivers has three components, and each is a place where a mid-size company loses either time or trust.

COMPONENT 1

Load latency

Actuals reach the planning tool by export and import. Weekly at best; monthly in practice. Every day between the close and the load is a day the plan is compared against numbers that are no longer current.

COMPONENT 2

Structural latency

A new cost centre, a merged entity, a renamed account. The ERP knows on day one; the planning tool learns when a load fails or a report shows an unmapped line; the BI tool learns last.

COMPONENT 3

Reconciliation latency

"Planning revenue" and "reporting revenue" are computed by two products with two copies of the data. Explaining the difference is a recurring task with no owner and no end.

Enterprise platforms solve these with scale — a data warehouse, a master-data management product, an integration platform and a team to run them. The mid-market cannot carry that overhead and should not need to. The architectural answer is not to shrink the enterprise stack; it is to remove the seams, so that a load is a scheduled process into the same store the plan reads, a structural change is a versioned hierarchy the ledger feed maintains, and reconciliation is a single number because there is a single copy.

02 Reference architecture

Six layers and one cross-cutting governance plane. Every section that follows

is a layer of this diagram.

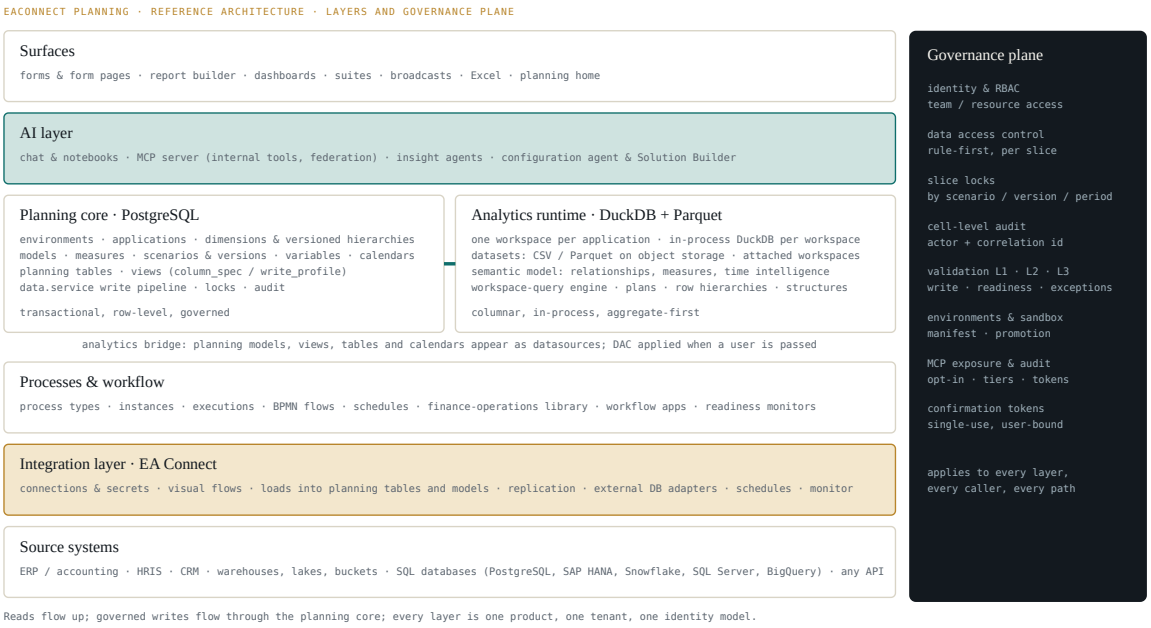

Figure 1. The reference architecture. The two engines in the middle are the subject of section 09; the governance plane on the right is the subject of sections 06, 07 and 12, and it is deliberately drawn spanning every layer.

LAYER	WHAT IT HOLDS	STORE	CANONICAL DOMAINS
Surfaces	Forms, reports, dashboards, suites, broadcasts, Excel	Metadata in PostgreSQL	forms and form pages · report builder · saved reports and dashboards · suites · charts and exports
AI	Chat sessions, MCP registry and audit, agents, insights, spec documents	PostgreSQL; spec bundles on object storage	MCP, chat, agents · configuration agent
Planning core	Environments, applications, dimensions, hierarchies, models, measures, scenarios, versions, variables, tables, views, fact data, locks, audit	PostgreSQL, one schema per environment	environment and application context · dimensional master data · models, scenarios, variables · measures · write pipeline · views · planning tables
Analytics runtime	Workspaces, datasets, semantic model, query execution	PostgreSQL metadata; Parquet on object storage; DuckDB per workspace	analytics workspaces and datasets · analytics bridge · report builder and query semantics · hierarchies and time intelligence
Processes	Process types, instances, executions, flows, schedules, workflow apps, readiness checks	PostgreSQL	built-in process framework · process builder · workflow apps · validation and readiness
Integration	Connections, flows, monitors, replication, external DB connections	Secrets store; PostgreSQL	EA Connect integrations (see the iPaaS article)

Twenty-five canonical domain documents describe the platform; the third column names the store each layer uses, which is the fact an architect most needs and the one most often missing from vendor diagrams.

03 Tenancy: organisation, environment, application

Isolation is decided at the environment level and is physical: a schema and a storage path. Everything a business unit builds lives in an application inside it.

Figure 2. Tenancy. The environment is the unit of isolation and of master data; the application is the unit of build, access and lifecycle; the workspace is the application's analytics runtime, one to one.

Three consequences follow for a mid-size company. A group with several business units can run one environment with several applications sharing Entity and Account, or separate environments where regulatory separation demands it; the platform supports both and the choice is a design decision. A sandbox is cheap to provision and safe to break, which is what makes the four-week implementation and every subsequent change low-risk. And because every call carries its environment and application, an object cannot leak across the boundary by accident — the application manifest (section 13) is computed from that context.

04 Master data and time

Dimensions, members, hierarchies, attributes and calendars are the vocabulary every model shares. They are governed through one write service

and versioned where the business changes.

Dimensions and members

Environment-shared by default — Entity, Account, Cost Centre, Product — or application-private where a business unit needs its own. Members carry attributes; reference dimensions model FROM/TO pairs for intercompany and transfer flows. Planner-extensible dimensions allow controlled inline member creation from a form, for cases such as a new project code, under a governance policy.

Hierarchies, versioned

A hierarchy has versions with validity dates. A reorganisation is a new version from an effective date; earlier periods keep the structure in force at the time. Time-dependent hierarchies and as-of reads are the mechanism that lets a report be run "as the organisation was" or "as it is now" — the platform's answer to the structural-latency problem in section 01.

One canonical write service

All dimension writes — from an admin screen, a CSV, an integration flow or the configuration agent — pass through a single canonical service that enforces codes, parent references, rollup and leaf flags, and data-access rules on member edits. There is no second way to create a member.

Calendars and time intelligence

A calendar is a configured object: fiscal or calendar type, granularity, start and end, week start, fiscal-year start, naming conventions and levels. The time dimension and a system planning view are generated from it. Several calendars can coexist. Time intelligence — period-to-date, prior period, rolling — resolves against the calendar in both planning and analytics.

> **For the leader**
>
> The question to ask of any platform is what happens on the morning after a reorganisation. Here the answer is a new hierarchy version dated to the reorg, loaded by the same feed that loads the ledger, with every historical report still reproducible. Nothing is restated by hand.

05

The model fabric

A real enterprise plan is not one cube. Rates, headcount and assumptions have their own grain and would be noise inside the P&L. They are their own models, publishing into the P&L through processes — clean, independently maintained, driver-linked.

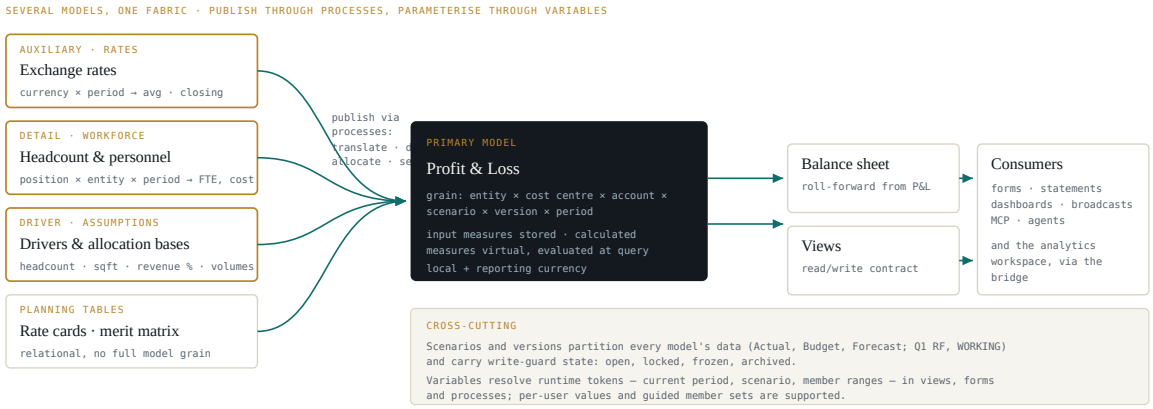

Figure 3. The model fabric. The P&L is the primary model; rates, workforce and drivers are auxiliary models at their own grain; planning tables hold relational reference data. Processes publish between them; scenarios, versions and variables cut across all of them.

Objects and their roles

OBJECT	DEFINITION	DESIGN NOTE
Planning model	A multi-dimensional fact schema: a set of measures at a dimension grain	Structural edits follow a draft → impact preview → commit lifecycle; the impact of a grain change is shown before it is applied
Measure	`input` (stored) or `calculated` (a formula evaluated at query time, never persisted)	Ratios and derived values without stale aggregates; calculated measures are never a write target
Semi-additive policy	Per-measure aggregation for non-summable quantities (headcount, balances) when a dimension is collapsed	Requires an environment flag that defaults off in production — see section 16
Scenario / version	Partitions of fact data; global across models in an environment	Versions carry lock state; a locked version is the artefact a board pack is built from
Planning variable	A runtime token resolved into views, forms and processes: periods, current scenario, member sets	Kinds include per-user values and guided member sets; exposed read-only to MCP by specification
Planning table	A relational table alongside models for drivers, rates, staging and governance data	Shares the governed write pipeline; column-level validation rules; CSV create and import
Planning view	A SQL-backed logical interface over fact data and tables; `column_spec` defines read identity, `write_profile` defines write authority	The single contract forms, processes and exports use; pre-joined in PostgreSQL for form performance

The view is the object to understand. Everything that reads or writes fact data does so through a view's declared columns and write map, which is why a form, a process and an export cannot disagree about what a column means.

06 The governed write path

There is one way to change a number. Every caller — a planner on a form, a nightly load, an allocation process, a workflow approval, a configuration agent, an API script — passes through the same ordered gates.

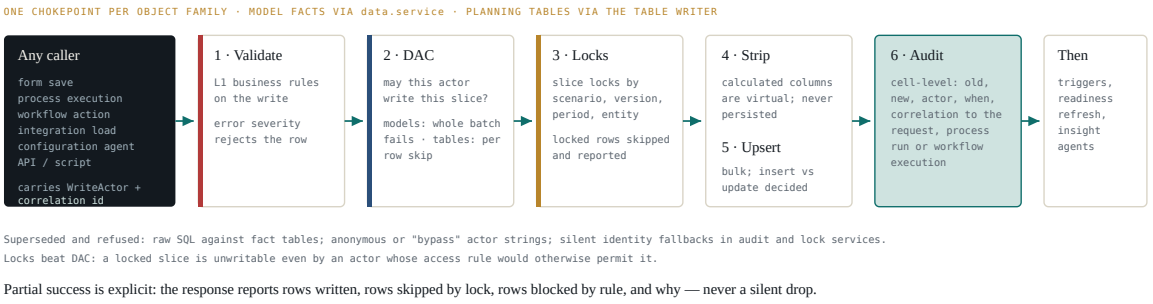

Figure 4. The write pipeline. Because every process, workflow and agent writes through it, each inherits validation, access control, locks and audit without implementing any of them — the property that lets a mid-size company add automation without adding risk.

Two details deserve a leader's attention. First, the *actor*: every write is attributed to a person or to a schedule owner, and the correlation id ties a changed cell back to the HTTP request, process run or workflow execution that changed it. An auditor asking "who changed this number and as part of what" gets an answer from one query. Second, *partial success*: a load that hits a locked version does not fail silently or half-apply; it reports which rows were skipped and why, which is what makes the nightly cycle in section 14 safe to run unattended.

07 Access: three layers with distinct jobs

"Can this person open this form" and "which rows may they write on it" and "is this version closed" are three different questions, answered by three

mechanisms that stack.

Figure 5. The access stack. Layer 1 is about artefacts, layer 2 about rows, layer 3 about time. An EMEA planner can open the sales plan (1), write EMEA rows only (2), and not at all once Q1 is locked (3).

The practical effect for a mid-size company with ten planners and fifteen viewers is that access is configured once, by rule and role, and reused. "EMEA planners write EMEA rows on the sales plan" is a rule bound to a role assigned to a team; a new hire joins the team and inherits it. Locks are the close: when the CFO approves Q1, the version is locked by slice and nothing — not a form, not a process, not an agent — can write to it without the lock being lifted, and the lift is itself audited.

08 Integration layer

The connector is not the interesting part. The interesting part is that the target of every pipeline is a typed planning table or model inside the same product, and that the process engine running the load is the one that runs the budget cycle.

EA Connect — the integration layer — lives inside the planning product as an Integrations area with Build (connections, tasks, flows, a visual flow builder), Manage (schedules, logs, files) and Monitor. Connections are named credentials in a secrets store. Flows are visual graphs of connector tasks. Processes wrap flows with validation monitors, gates and load steps. Every execution is recorded with node-level status and a timeline. The companion article on the in-built iPaaS walks through a NetSuite pipeline end to end; this section records the architectural points.

Connector catalogue

ERP and accounting (NetSuite, SAP HANA, Xero, Sage, Acumatica, Business Central, Restaurant365, QuickBooks Online via HTTP), HRIS (Workday, BambooHR, UKG, HiBob, Namely), data platforms (Snowflake, BigQuery, Redshift and other databases; lakes; GCS, Azure Blob, S3, SFTP), applications (Salesforce, Anaplan, Postgres), and a generic HTTP connector for any REST API. A snapshot; the list grows.

External database adapters

Separately from flows, external SQL databases can be registered as first-class datasources through an adapter interface — PostgreSQL, SAP HANA, Snowflake, SQL Server and BigQuery adapters are documented — so a warehouse table appears in the report builder's datasource tree without being copied.

Typed targets, loud failures

A load writes to a planning table or model through the governed write pipeline of section 06, so it either conforms to the target's definition or fails visibly at a named node with a log. Validation monitors (section 10) run readiness checks before a load is accepted.

Replication and change capture

A data-replication capability keeps a full copy of a source table in sync as an alternative to extract-and-load. Incremental loads by watermark and full reloads are the practitioner's choices for detail and reference data respectively; the platform mechanisms and the practice are set out in the iPaaS article.

09 The dual-engine query runtime

Two engines, deliberately. PostgreSQL is the transactional core where governed writes land row by row. DuckDB is the in-process analytical engine

that aggregates over columnar Parquet. A bridge lets each see what it needs of the other.

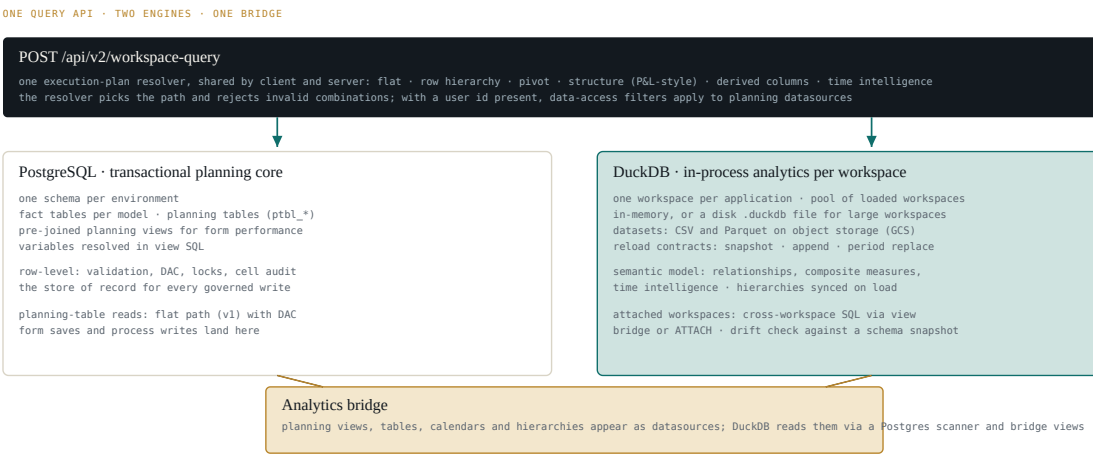

Figure 6. The dual-engine runtime. Writes and forms live on the left; aggregates, dashboards and MCP queries on the right; the bridge means a report can join a planning view to a warehouse dataset without either being copied.

Why two engines

A planning platform has two workloads with opposite shapes. Form saves and process writes are many small, governed, row-level transactions — a relational database's job, and PostgreSQL does it with locks, constraints and an audit table. Statements, dashboards and ad-hoc analysis are wide aggregations over millions of rows — a columnar engine's job, and DuckDB does it in-process, without a separate server, over Parquet files that object storage holds cheaply. Running both on one engine forces a compromise on one workload; running them on two and bridging them lets each be what it is.

Query semantics that finance needs

CAPABILITY	WHAT IT DOES	WHY IT MATTERS IN A STATEMENT
Row hierarchy	Drill by hierarchy levels (rollup) or parent-child (recursive), with an as-of reference date for planning hierarchies; empty nodes optionally hidden	A P&L that expands from Total Revenue to Service Revenue to Maintenance & Support, using the hierarchy version in force for the period
Structures	P&L-style presentation: ordered lines, subtotals, memo lines, sign handling	Gross margin, operating income and EBITDA as statement lines, not spreadsheet formulas
Subtotal strategy	Measure context that defines how a ratio behaves at a subtotal row	Margin % at the total is total margin over total revenue, not the average of the children
Semi-additive aggregation	Per-measure policy for collapsed dimensions (last, first, average)	Headcount and balances do not sum across months
Time intelligence	Period-to-date, prior period, rolling windows, resolved against the fiscal calendar	YTD actual versus YTD budget without hand-built columns
Client post-aggregation	Server returns aggregated SQL plus a row probe; the client pivots, applies structures and derived columns within configured thresholds	Complex layouts stay responsive; export from the browser respects structure

The same semantics serve the report builder, the saved-report runner, exports, dashboards and the MCP query tools, within the runner's limits. A number in a dashboard and the same number in a broadcast PDF are computed by one engine with one definition.

10

Processes, flows, workflow and validation

Automation is where planning platforms either stay governed or quietly stop being. Here every process writes through the pipeline of section 06, every flow

is a graph with gates, and validation is three layers rather than one rule table.

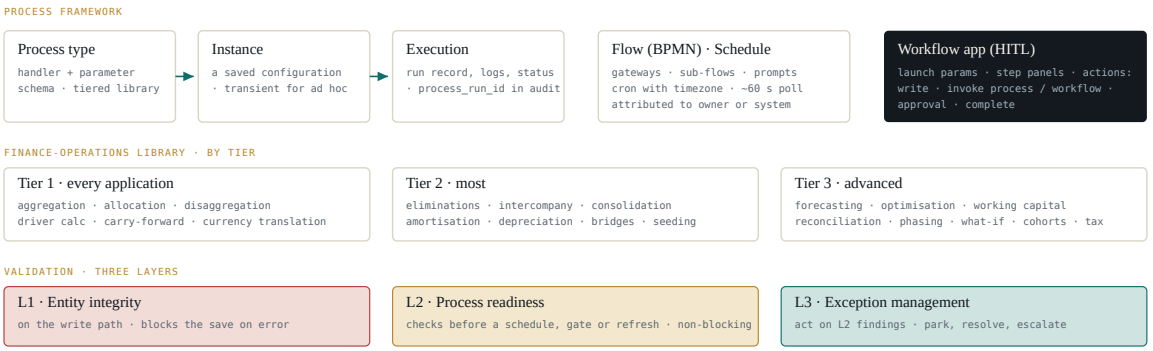

Figure 7. Processes, workflow and validation. The library is an architectural classification of process primitives by commonality; which handlers are shipped versus specified is a question to put to the platform's status tracker for any given tier-2 or tier-3 item.

Three design properties carry the governance. Handlers must write through the data service — no raw fact SQL — so every process inherits validation, access control, locks and audit. Executions are attributed: a scheduled run is credited to the schedule's owner or to the system, and its run id threads into every cell it changed. And validation is layered so that bad data is refused at write time (L1), a process does not start against an unready state (L2), and the findings from readiness checks are worked as exceptions with a lifecycle rather than lost in a log (L3).

Workflow apps are the human-in-the-loop layer: a guided, multi-step data operation with launch parameters, step panels and typed actions, including an approval step. They are composable — a workflow can invoke a process or a child workflow — exportable as a definition file for promotion, and exposable to MCP with confirmation tokens on mutations. The budget-submission and approval cycle in the four-week playbook is a workflow app.

11 Surfaces and distribution

Everything a planner or a board member touches is a surface over the same

model, and the application's suite is the front door that tells the process story.

Forms and form pages

Entry grids over planning views or tables, with the same write contract; multi-page layouts and sub-forms; a form workbench with an intelligence rail that surfaces insights and assistance in the page the planner is on.

Reports and dashboards

A report builder over the datasource catalogue — planning views, tables, workspace datasets, external databases — with the semantics of section 09; saved reports; dashboards of widgets; a chart library; drill-down.

Suites and home

A suite is an application-scoped navigation shell that orders dashboards, reports, forms and workflow apps into a story — assumptions → seeding → entry → review → approval → reporting — with a shared context (subsidiary, version, period) that flows into each item. A personal home holds per-user pins.

Broadcasts

Scheduled email delivery of a report, filtered per recipient from a mapping, optionally through an Excel template that preserves the recipient's layout and formulas; each run logged with sent and failed counts.

Excel

Structure-aware export from the report builder and the saved-report runner; an Excel add-in and cloud export service for teams who live in workbooks; the Solution Builder's spec round-trips to Excel for bulk editing.

Sharing and notifications

A view can be shared as a captured snapshot with an in-app notification and email; notifications are a single inbox across processes, workflows, shares and agent findings.

12 The AI layer

Three modes — ask, watch, build — with one governance principle: an AI

caller is a user, with a user's permissions, through the same services, audited.

MODE	MECHANISM	GOVERNANCE	COMPANION ARTICLE
Ask	In-product chat and notebooks; a form-page assistant; an embedded MCP server exposing about thirty-two internal tools in packs (meta, query, reports, planning, variables, workflow) to chat and to external hosts such as Claude Desktop and Cursor; external MCP federation for third-party tools	Every object opts in separately (`mcp_exposed`); calls run as the authenticated user; reads are tier A/B, side-effects tier C with a single-use confirmation token; an append-only audit log per call; a loop cap per chat turn	MCP in corporate finance
Watch	Insight agents: plain-language instructions, a schedule, a team, a run-as user; findings filed to an inbox with severity, tags and a resolve/park/dismiss lifecycle; pattern memory that fingerprints successful runs	Agents read their team's datasources and can only submit or resolve insights; the `insights.*` tools are invisible to chat and external hosts; run-as identity bounds visibility	Governed AI agents
Build	The configuration agent and Solution Builder: a spec with one row per object; intake from CSVs, documents, mockups, templates, interview or a live snapshot; match → validate → diff → approve → apply → verify; drift detection; capability gaps	Never bypasses RBAC, DAC, environment guards or locks; never edits platform code; replace and remove need a token; production writes need explicit allowance; unmet requirements become structured gaps	The AI design assistant

The three modes share the same MCP tool registry underneath — the configuration agent's tools are a namespace in the same registry as the query tools — which is why one audit log covers all of them.

The architectural point is that AI did not add a new path to the data. Chat reads through the query API; agents read through MCP and write only findings; the configuration agent applies through the same services as an administrator. The governance plane in Figure 1 did not need an extension for AI because AI was designed as a caller of what already existed.

13 Lifecycle and promotion

An application is a computed set of objects, and every lifecycle operation — copy, delete, export, import, promote — consumes the same set. Nothing is orphaned and nothing leaks.

The *application manifest* is the transitive closure of everything that belongs to an application: seeded from direct ownership, expanded through view SQL lineage, form-to-view chains, sub-forms, process references and workflow walkers, iterated to a fixpoint, and including the access rules that target manifest objects. Forms are classified as in-app when their whole closure is inside the manifest and cross-app otherwise. Environment-shared dimensions are allowed in the closure and referenced, never copied; federated views are excluded.

Copy

Clone in-app objects into a new application with ids remapped; provision its workspace; wire team access. The way a proven application becomes a template for another business unit.

Delete

The closure defines the delete set; the operation blocks if a write-back view has an external SQL dependency; cascades otherwise.

Export / import

Portable bundles with code-based dependency manifests. Objects resolve by stable code on import, which is why codes are never renamed through the spec.

Promote

Dev → test → prod. Structure only by default; data explicit and opt-in; secrets and connections never travel and are re-entered in the target.

For a mid-size company the operational meaning is simple: build and certify in a sandbox, promote by manifest on a Friday, and never re-create a form by hand in production. The moment someone does, the environments have diverged, and the configuration agent's drift detection (section 12) is the mechanism that finds it.

14 The connected cycle, end to end

Everything above, as one night and one month. This is what "no manual

latency" looks like as a sequence.

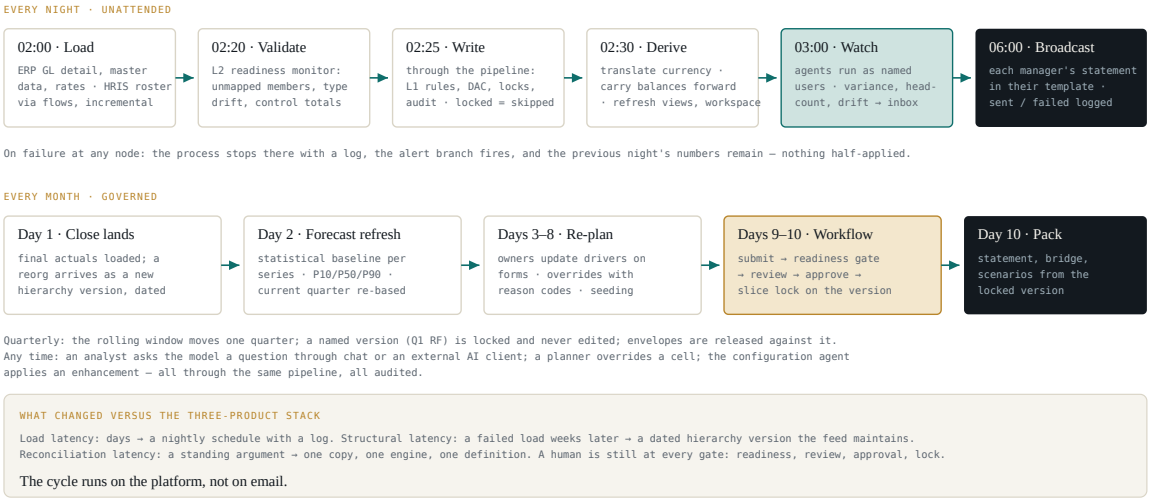

Figure 8. The connected cycle. The nightly row is unattended and safe to be; the monthly row is governed at every gate. The same tables carry through both.

15 Decisions for the architect

The platform's design-guidance questionnaire holds seventy-nine decisions with defaults. These are the ones that shape the architecture of an

implementation, with the practitioner's recommendation for a 100–1,500-employee company.

DECISION	OPTIONS	RECOMMENDATION AND RATIONALE
Environments	One environment, several applications · one per business unit · one per regulatory boundary	One environment with shared Entity, Account and Calendar unless regulation forces separation. Shared master data is most of the value of connected planning.
Model fabric	One model · several models publishing into a primary	Several. P&L, workforce and rates at their own grains. The questionnaire's blocking question "do all measure groups share one grain" is answered "no" for almost every real company.
Planning grain	Entity × cost centre × account × month, ± product, ± project	Start with the four; add product or project as a second wave when a driver actually needs it. Grain is expensive to change; the impact preview exists, but the conversations do not get cheaper.
Time	Fiscal or calendar · monthly, weekly, mixed · one or several calendars	Monthly on the fiscal calendar; add a second calendar only for a unit that genuinely closes on a different one.
Currency	Single · multi-currency with reporting currency	Single at first go-live even for multi-entity groups; translation as wave two, with the rates model and process in place from the blueprint.
Load strategy	Full reload · incremental by watermark · replication	Full reload for master data and small reference tables; incremental for GL detail with a periodic full reconciliation; replication where a source exposes change cleanly.
Access model	Team access only · DAC rules by entity · by entity and cost centre	DAC by entity from day one; add cost centre when departmental entry begins. Rules are cheap to add and expensive to retrofit into habits.

Locks	Version-level · slice-level by entity and period	Slice-level. A group closes entities on different days; slice locks let each close independently.
Analytics storage	In-memory workspace · disk-backed workspace	In-memory for a single application at mid-market volumes; disk-backed when the workspace holds multi-year detail or attached peers.
Semi-additive measures	Off (default) · on via environment flag	On, if headcount or balances appear in any report — and test the known partial blocks with parent-child hierarchies and structures before relying on it.
AI exposure	None · one model, one team · broad	One model and one report exposed to one team for a month; read the audit log; widen from evidence.
Build method	Admin screens · Solution Builder spec · both	Spec-first, always, so that the design document and the application never diverge; admin screens for fine-tuning, followed by drift → adopt.

Recommendations are practitioner opinion. Every row is a documented capability; the rationale is ours.

16 Constraints and honest notes

A blueprint that omits the platform's own recorded constraints is a brochure. These are drawn from the canonical documents' "known constraints" sections,

as of the June 2026 code review, and from the status of specifications this report has relied on.

Tenancy

Isolation is schema-per-environment; database-per-environment is deferred. Multi-tenant SaaS routing across organisations is out of scope of the documented design. One current-period marker per calendar per environment.

Access control

DAC v3 has no hierarchical role inheritance and no time-bounded assignments; capabilities apply uniformly to all rules on a role; rule export/import bundles are not yet available. Scenario is not a DAC target — use locks. Row-level read filtering on planning tables when access dimensions are set is deferred (capability check only today).

Query runtime

Semi-additive aggregation requires an environment flag that defaults off in production and is partially blocked with parent-child hierarchies, structures and client post-aggregation. The MCP report runner does not expose full structure, time-intelligence or hierarchy parity. Reports run without a user id are unfiltered (analytics behaviour). Some operational workspace routes rely on deployment-level rather than route-level auth.

Processes and workflow

The finance-operations library is an architectural classification; which tier-2 and tier-3 handlers are shipped is a status question. MCP workflow tools do not expose save/resume checkpoints (UI and REST only). Planning-table post-write triggers are partial.

Configuration agent

Version 1 excludes calendars, DAC roles and rules, validation rules, write triggers and schedules — these are reported as capability gaps when a spec needs them. Report and dashboard slugs are unique per instance. The blueprint engine that generates a full application from a questionnaire is a specified design, not shipped.

Specified, not yet shipped

Currency Translation Wizard; hierarchy versioning round-trip and as-of reads as specified in BUILD-SPEC-077 (the versioned hierarchy object and version selector are shipped and shown in the companion articles; the full as-of read surface is the specified part); variable MCP tool packs; the implementation-program plan tooling; agent pattern crystallisation into workflows (draft).

> **How to read this section**
>
> None of these constraints prevents the four-week go-live for the single-application scope this blueprint targets. Several of them matter for wave two — multi-entity with eliminations, multi-currency, broad AI exposure — and should be checked against the platform's current status before those waves are scoped.

17 In one page

What connected planning is

A data-flow property: the ledger loads on schedule into the store the plan reads; structural change arrives as a dated hierarchy version; there is one copy of the numbers, computed by one engine, under one identity model. No file is carried.

What makes it possible here

Hard tenancy with shared master data; several models on one fabric through views; one governed write path; two engines with a bridge; a process framework that inherits governance; three-layer validation; an AI layer that is a caller, not a bypass; a manifest-driven lifecycle.

What it asks of the organisation

Nine blocking design decisions answered in week one; a finance owner for four weeks; an ERP administrator who issues credentials on time; the discipline to reconcile to the trial balance before go-live and to promote rather than rebuild afterward.

What it does not claim

That multi-entity eliminations, multi-currency or a warehouse migration fit in the first four weeks; that the finance-operations library is uniformly shipped; that any figure in this report is a measured latency. Each of those is stated where it arises.

EAConnect Planning · Research library · Hours, Not Months · Flagship

Basis: the platform's canonical domain documentation (25 domains, code-reviewed against source on 2026-06-01), user guides v2.0 (Jun 2026), the design-guidance questionnaire, and the companion articles in this library. Recommendations in sections 15 and 17 are practitioner opinion. Items marked specified are design documents, not shipped features.

5,374 words · 23 min read · — pages