REPRESENTATIVE FINAL REPORT

Source-of-Truth & Data Ownership Report

A plain-language decision document showing which system owns each business fact, how records connect, and what happens when sources disagree.

Prepared for	**Representative Construction Client**
Prepared by	**StructuredLayer**
Document	**Version 1.0 | 19 July 2026**
Status	**Illustrative client-ready sample**

Illustrative records and values are used throughout. A client-issued report replaces them with approved systems, owners, identifiers, rules, evidence, and decisions.

01 / START HERE

What this report means

This report answers one deceptively simple question: **when two tools disagree, which one should the business trust?** It names the approved owner for each fact and defines how that fact may move into workflows, dashboards, reports, and AI-assisted work.

Plain-language rule

One business fact should have one approved authority. Other systems may display or use that fact, but they should not silently redefine it.

Authority

The system or role allowed to establish and correct a fact.

Stable identity

The identifier that keeps the same project, document, cost, or approval connected across tools.

Controlled use

The read, write, approval, retention, and exception rules applied to that fact.

The practical outcome

Teams stop arguing about which spreadsheet is current. Reports can trace values back to approved sources. Integrations know where corrections belong. AI can retrieve governed context without becoming the source of truth.

01	02	03	04	05
Identify	**Assign**	**Connect**	**Control**	**Validate**
List systems and facts	Name the authority	Add stable IDs	Define update rules	Test conflicts and access

Scope note

This is not a recommendation to force every record into one application. A connected operating layer can preserve several authoritative systems while making their relationships and update boundaries explicit.

02 / EXECUTIVE SUMMARY

The decision in one page

The representative environment uses specialist systems for project delivery, documents, finance, estimating, workflow coordination, and reporting. The recommended model keeps those systems in place, connects them through stable identifiers, and prevents downstream tools from silently becoming transactional authorities.

Business area	Approved authority	Connected use	Primary decision
Projects	Project-management system	Workflow, reporting, AI retrieval	Project status changes remain in the project system
Issued documents	Document-control platform	Revision alerts and project context	Current issue and superseded links remain traceable
Actual cost	Accounting system	WIP and management reporting	Corrections occur in accounting before refresh
Estimate baseline	Estimating system	Handoff and variance review	Award version is locked; later changes append
Cross-system workflow	Operating database	Queues, approvals, exceptions	Only approved states and roles may write
Issued reporting	Governed reporting layer	Leadership review	Reports read authorities but do not overwrite them

Recommended

Approve a federated source-of-truth model with stable cross-system identifiers.

Do not allow

Silent last-write-wins updates, unowned corrections, or AI-generated source values.

Validate next

Identifiers, permissions, conflict queues, write-back fields, and report lineage.

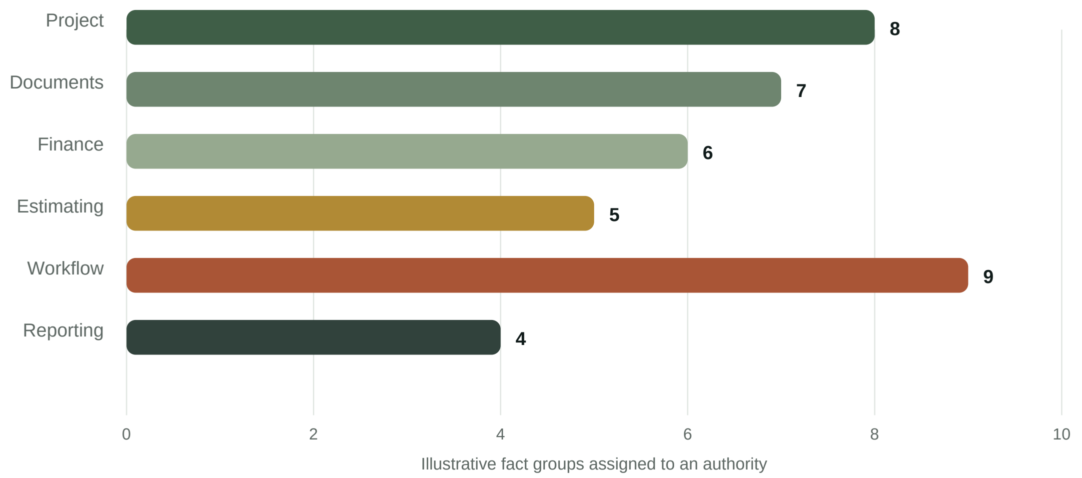
Executive interpretation

- No single application needs to own every business fact.
- Every authoritative fact needs a named system, responsible role, and correction path.
- The operating database owns relationships and workflow state, not the underlying transactions owned elsewhere.
- Dashboards and approved AI tools consume governed context; they do not become accidental systems of record.

03 / CURRENT LANDSCAPE

Several systems can be correct at the same time

The objective is not to choose one winner. It is to state which system is correct for which fact. The example below shows an illustrative allocation of 39 fact groups across six operating authorities.



Illustrative allocation only. Counts show how an ownership review can be summarised; they are not performance measures.

Source systems

Create and correct approved business transactions, documents, and baselines.

Operating layer

Connects identities, states, owners, approvals, and exceptions across source systems.

Consumption layers

Dashboards and AI use governed facts with source links and permission checks.

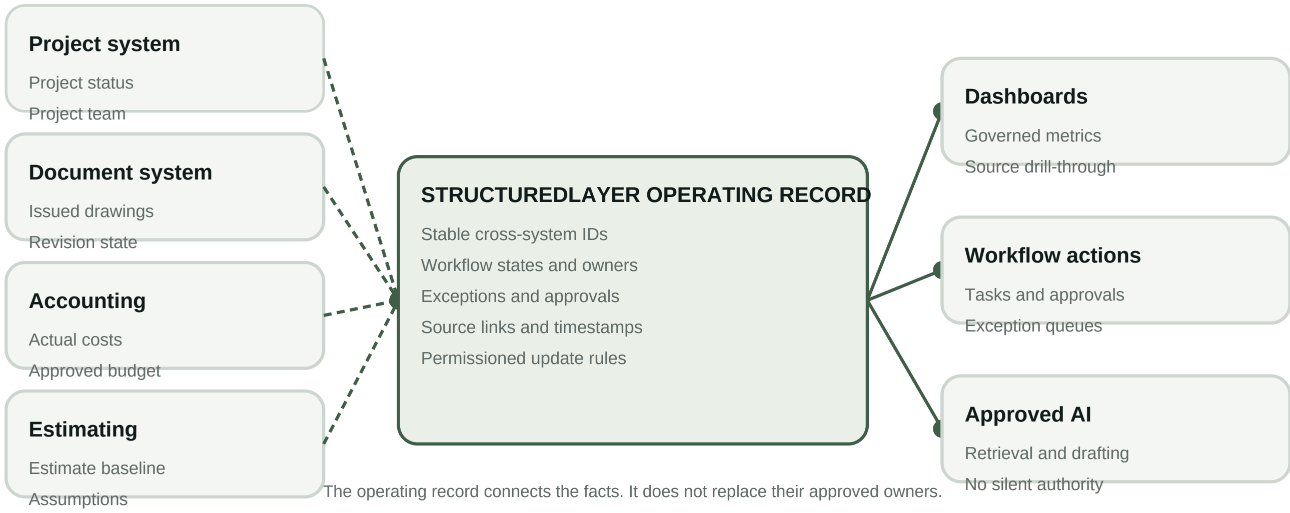
Typical warning signs

- The same project has different names or codes across tools.
- Reports overwrite a source value because the correction path is unclear.
- A drawing is stored in several folders without a reliable current-issue state.
- AI answers from copied files without knowing their authority, revision, or access boundary.
- Former staff, shared accounts, or undocumented integrations can still change records.

04 / OPERATING ARCHITECTURE

Keep authority where it belongs, connect context in the middle

Source applications remain responsible for the facts they are designed to govern. StructuredLayer connects those facts into a client-owned operating record that supports reliable workflows, reports, and bounded AI assistance.



The operating record connects the facts. It does not replace their approved owners.

What the middle layer owns

Relationships

Which company, opportunity, project, document, estimate, cost event, and approval belong together.

Operational state

Current owner, next action, due date, exception, approval gate, and handoff status.

Evidence

Source URL, source timestamp, retrieval time, version, decision, and audit history.

Important boundary

The operating record may coordinate an approved write-back, but it should not change a source-owned fact unless the field, identity, role, and update rule have been explicitly approved and tested.

05 / OWNERSHIP MATRIX

The operating contract for each source

This matrix is the central client decision. It names the authoritative facts, stable identifier, update boundary, and review owner for every connected system.

Source	Authoritative facts	Stable identifier	Controlled update rule	Review owner
Project system	Project ID, status, team, location	project_system_id	Import approved changes. Reports cannot overwrite source-owned fields.	Project controls
Document system	Issued files, revision, issue state	document_urn + revision	Append revisions. Preserve superseded links and source URL.	Document control
Accounting	Posted actuals, approved budget, billing	job_id + cost_code	Read posted values. Correct in accounting before refresh.	Finance
Estimating	Approved baseline, assumptions, alternates	estimate_id + version	Lock award version. Append post-award changes separately.	Estimating
CRM	Company, contact, relationship, opportunity	company_id + contact_id	Match identities before creation. Route uncertain matches to review.	Commercial owner
Operating database	Cross-system IDs, workflow state, approvals, exceptions	workflow_record_id	Role-permissioned, audited state transitions only.	Workflow owner
Reporting layer	Approved calculations and issued views	metric_id + period	Read governed sources. Never become authority for transactions.	Reporting owner

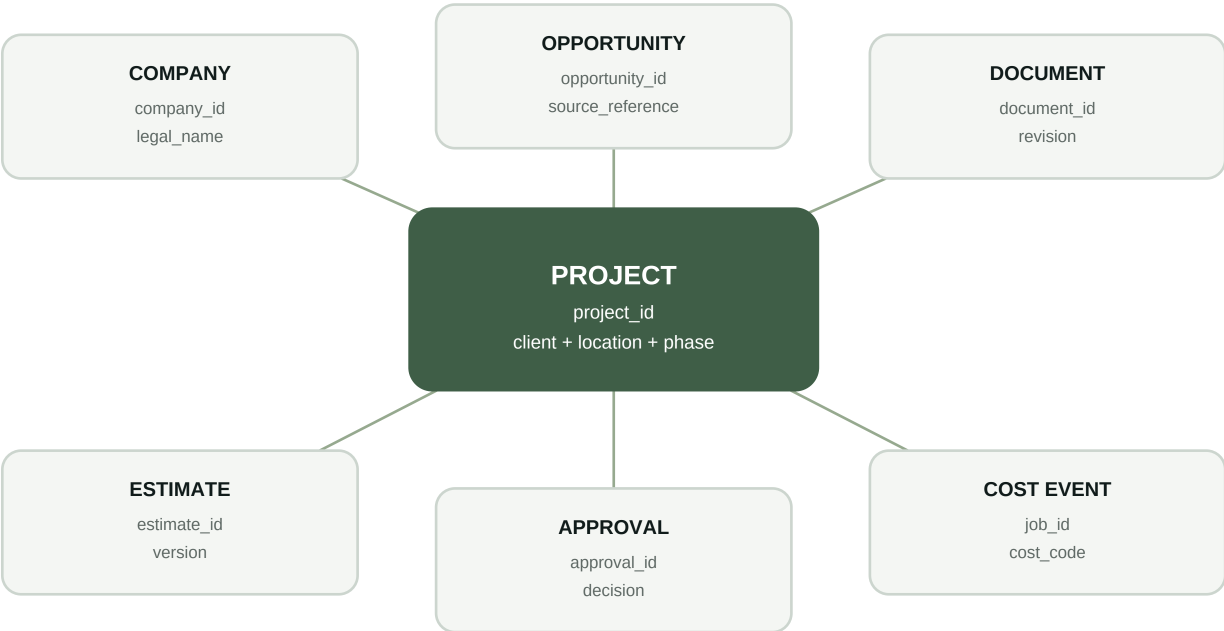
Client decisions required

- Approve or revise the authority assigned to every fact group.
- Name a business owner and a technical owner for every connection.
- Confirm which fields may be written back and which remain read-only.
- Confirm retention, deletion, export, and audit requirements.

06 / CONNECTED IDENTIFIERS

Stable IDs keep the same business object connected

Names change. Files move. People use abbreviations. Stable identifiers allow the system to recognise the same project, company, document, estimate, cost event, and approval across different tools.



Minimum identifier rules

Immutable

A stable ID is not reused or rewritten when a display name changes.

Source-aware

External IDs retain the originating system and retrieval evidence.

Reviewable

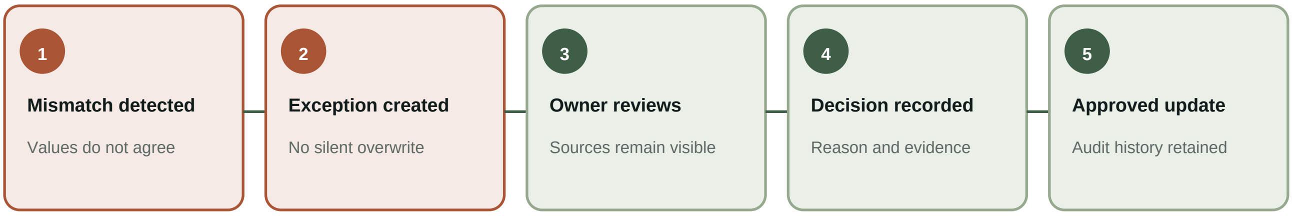
Ambiguous or duplicate matches enter an exception queue instead of merging silently.

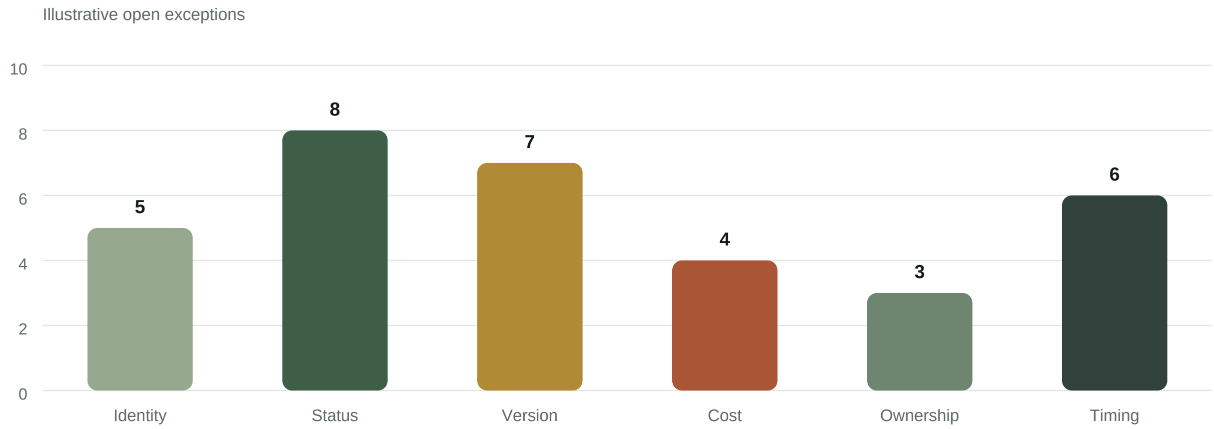
Relationship	Required link	Example validation
Opportunity to project	opportunity_id -> project_id	Award or authorised handoff must exist
Project to document	project_id -> document_id	Document source and revision are required
Project to estimate	project_id -> estimate_id + version	Approved baseline version is locked
Project to cost	project_id -> job_id + cost_code	Accounting mapping is reconciled
Action to approval	workflow_record_id -> approval_id	Named role and decision evidence required

07 / CONFLICTS & EXCEPTIONS

When sources disagree, the system should stop guessing

A mismatch is operational information. It should create a visible exception with an owner, source evidence, review time, release condition, and resolution history.





Illustrative open exceptions only. A client report uses actual categories, ages, owners, and agreed thresholds.

Exception	Example cause	Owner	Release condition
Identity conflict	Two possible matches for the same company	Data owner	Approve match, separation, or merge
Status conflict	Project system and report show different phase	Project controls	Confirm authority and effective date
Revision conflict	Two files appear current	Document control	Establish current and superseded links
Cost conflict	Imported actual differs from accounting	Finance	Correct or approve in accounting
Permission conflict	Unauthorised role attempts a transition	Security owner	Block action and review event

08 / PERMISSIONS & AI BOUNDARY

Access follows business authority, not tool convenience

The report separates the ability to view, propose, approve, and write. AI may retrieve or draft within an approved boundary, but it does not gain authority merely because it can technically perform an action.

Role	View	Propose	Approve	Write source	Close exception
Workflow operator	Approved scope	Yes	No	Approved fields only	Routine only
Business owner	Business scope	Yes	Yes	Through approved process	Yes
Technical owner	System scope	Technical only	No business approval	Integration controls	Technical faults
Reporting user	Approved views	Commentary	Issue report	No	No
AI assistant	Permission-filtered	Candidate values and drafts	No	No direct authority	No

Human approval remains mandatory for

Consequential actions

External releases, contractual commitments, payments, safety, compliance, and access decisions.

Authority changes

Changing the system of record, write-back permission, retention rule, or record owner.

Uncertain evidence

Conflicting, missing, stale, inaccessible, or low-confidence source information.

AI operating rule

AI-extracted values remain candidates until accepted by the authorised role. Approved AI receives only the current, relevant, permissioned context required for the task, with source links retained where practical.

09 / IMPLEMENTATION PLAN

Turn the ownership decisions into a working operating layer

Implementation should begin with one bounded workflow and approved historical examples before introducing live write-back or consequential automation.

01	02	03	04	05
Confirm	**Model**	**Connect**	**Validate**	**Launch**
Systems, owners, facts	IDs and relationships	Read paths and queues	Historical test cases	Training and stabilisation

Phase	StructuredLayer activity	Client participation	Exit evidence
Confirm	Interview owners, inventory systems, define authority	Provide process owners, samples, access constraints	Approved ownership register
Model	Create records, fields, IDs, permissions, conflict rules	Review meanings and decision rights	Approved dictionary and relationship map
Connect	Configure ingestion, matching, queues, source links	Authorise integrations and test identities	Working controlled data path
Validate	Run happy paths, failures, permissions, reconciliation	Perform acceptance scenarios	Signed acceptance record
Launch	Train roles, transfer access, stabilise defects	Operate workflow and resolve routine exceptions	Client-owned handover

What we need from the client

- A named sponsor and an accountable owner for each source system.
- Approved examples of records, reports, exceptions, and historical decisions.
- Authorised access or supervised demonstrations of relevant systems and workflows.
- Timely decisions on ownership, write boundaries, permissions, and acceptance criteria.

10 / ACCEPTANCE & MEASURES

Prove that the ownership model works under pressure

A clean diagram is not enough. The operating model must survive duplicate records, conflicting values, changed revisions, unavailable sources, unauthorised actions, and recovery after failure.

Traceability

A user can move from a dashboard value to the governing source and effective date.

Conflict control

A mismatch creates a visible exception and never silently overwrites the authority.

Client ownership

Named client roles can operate, export, recover, and disable the system using the handover pack.

Acceptance test	Expected evidence	Decision
Rename a project in one source	Stable ID preserves relationships; display name refreshes according to rule	Pass / Fail / Conditional
Import a conflicting project status	Exception is created; authority remains unchanged	Pass / Fail / Conditional
Receive a new drawing revision	Revision appends; current and superseded states are traceable	Pass / Fail / Conditional
Attempt unauthorised write-back	Action is blocked and security event retained	Pass / Fail / Conditional
Disconnect a source temporarily	Retry, alert, stale-state indicator, and recovery path work	Pass / Fail / Conditional
Ask approved AI for a project answer	Response respects permissions and exposes sources or uncertainty	Pass / Fail / Conditional
Reconstruct a reporting value	Source, calculation version, period, and issue history are traceable	Pass / Fail / Conditional

Candidate operating measures

- Percentage of governed fact groups with an approved authority and owner.
- Percentage of connected records carrying valid stable identifiers and source links.
- Age and volume of unresolved ownership, identity, version, and permission exceptions.
- Percentage of issued reports with source drill-through and calculation version.
- Number of unapproved write attempts, stale sources, and failed refreshes requiring review.

11 / CLIENT DECISION

What the client is approving

Approval of this report confirms the intended ownership model and authorises the next implementation stage. It does not approve systems, integrations, costs, or scope that are not explicitly named in the signed proposal.

Decision area	Client decision	Owner	Date
Authoritative systems and fact groups	Approve / Revise / Reject		
Stable identifiers and relationship model	Approve / Revise / Reject		
Read, write, and approval boundaries	Approve / Revise / Reject		
Conflict and exception rules	Approve / Revise / Reject		
Permissions, retention, and AI boundaries	Approve / Revise / Reject		
Implementation and acceptance plan	Approve / Revise / Reject		

Final handover package

Operating records

Ownership map, data dictionary, relationship model, integration register, and permissions matrix.

Control evidence

Exception catalogue, monitoring rules, acceptance record, issue log, and open-condition register.

Client operation

Runbook, training record, administrator transfer, backup and recovery guidance, and stabilisation handoff.

Confidentiality and retention

Client-issued reports, interviews, recordings, samples, and implementation records are handled under the signed engagement and applicable confidentiality terms. StructuredLayer does not use client project data to train public models. Temporary implementation access is removed during handover, and retention follows the approved client policy and contractual boundary.

StructuredLayer

Client-owned data and workflow systems for construction and property operations.
structuredlayer.com | **start.structuredlayer.com** | **start@structuredlayer.com**

Structure before automation.

Connected facts, controlled workflows, and approved AI assistance remain grounded in client-owned operating context.