SALESFUSION

The Consulting Portal as an Implementation Workspace

Technical architecture and implementation report

PUBLIC RELEASE · VERSION 1.0.0 · 2026-07-17

The Consulting Portal as an Implementation Workspace

Publication metadata

Field	Value
Author	SalesFusion Research
Publication date	2026-07-17
Version	1.0.0
Canonical slug	consulting-portal-implementation-workspace
Canonical path	/technical-papers/consulting-portal-implementation-workspace
Publication class	Technical architecture and implementation report
Approval state	Founder-approved public release

Revenue Cortex connection

The consulting portal is one workspace through which a Revenue Cortex can be installed and reviewed. It should preserve canonical inputs, client decisions, SalesFusion outputs, approvals, revisions, deliverables, and evidence without pretending that a portal alone is the revenue system.

Evidence-status box

> SalesFusion uses a governed workspace foundation for consulting implementation work. The foundation separates curriculum, canonical state, submissions, reviews, versions, decisions, deliverables, evidence, outcomes, tenant isolation, and AI draft boundaries. This paper describes that foundation and proposes the full workspace model. It does not claim measured improvements in client completion, review-cycle time, information retrieval, satisfaction, or project outcomes. Those claims require client usability, completion, review-cycle, and retrieval studies.

> The design uses governance context consistent with ISO/IEC 42001, resource-scoped access principles consistent with NIST SP 800-207, and HTTP interface semantics consistent with RFC 9110. These references support architecture framing only.

Abstract

Complex AI consulting is often run through calls, documents, chat threads, task lists, and shared folders. That structure is workable for small engagements but weak for governed implementation. It scatters state, loses review boundaries, mixes drafts with decisions, hides evidence, and makes it difficult to know which version of a deliverable is current. A chatbot transcript is not enough. A governed client implementation workspace needs canonical state, submissions, reviews, decisions, versions, deliverables, evidence, outcome records, and tenant isolation.

This paper defines the consulting portal as an implementation workspace rather than a content library or chat interface. The workspace organizes curriculum, client inputs, AI-generated drafts, consultant review, approved decisions, deliverables, and outcome records into typed state. AI can draft, summarize, identify missing evidence, and propose next steps. It cannot create final approval, silently modify canonical state, or treat a draft as a client decision. SalesFusion uses the workspace foundation; full usability and outcome evidence remain pending.

The problem

Consulting implementation work has an information-shape problem. A client answers questions in one place, uploads context in another, reviews a draft in a third, and approves a decision verbally or by message. Later, the team needs to know what the client actually approved, which evidence supported it, which deliverable version is final, which assumptions remain open, and whether the implementation produced the expected artifact. Without a governed workspace, those answers require manual reconstruction.

AI can make this worse. A model can summarize a transcript as if it were canonical state. It can turn a tentative answer into a final decision. It can merge old and new information without supersession. It can generate a polished deliverable that looks approved because it is well formatted. It can answer from another tenant's pattern if isolation is weak. In consulting work, the risk is not only data leakage. It is decision confusion: draft, review, approval, and delivery collapse into one persuasive document.

The portal should therefore be a stateful implementation environment. Its job is to help the client and operator move from curriculum and inputs to reviewed decisions and deliverables with explicit evidence and ownership.

Contributions

This paper contributes:

- A reference model for a governed consulting implementation workspace.
- Definitions for curriculum, canonical state, submissions, reviews, versions, decisions, deliverables, evidence, outcomes, and tenant isolation.
- A rule that AI draft is not final approval.
- A workflow model for submission, analysis, review, decision, deliverable generation, outcome recording, and version supersession.
- Implementation invariants for tenant scope, state ownership, review gates, and exportability.

- An evaluation design for usability, completion, review-cycle behavior, and retrieval accuracy.

Core definitions

Workspace is a tenant-scoped implementation environment where curriculum, client inputs, reviews, decisions, deliverables, and outcomes are structured records.

Curriculum is the ordered implementation content: modules, prompts, exercises, required inputs, acceptance criteria, and dependencies.

Canonical state is the current structured representation of the client's implementation context, built from approved submissions, reviewed evidence, and decisions. It is not a chat summary.

Submission is a client or operator input awaiting validation, review, or incorporation.

Review is a structured assessment of a submission, draft, deliverable, evidence packet, or decision request.

Version is an immutable or superseded record of a state object, deliverable, curriculum item, or decision packet.

Decision is an approved, rejected, deferred, or superseded choice with authority, evidence, scope, and effective time.

Deliverable is a versioned artifact created for client use, such as a plan, map, checklist, report, configuration, or implementation packet.

Evidence is a source, submission, artifact, review, or event that supports a state field, decision, or deliverable.

Outcome is a recorded implementation result, including completed, blocked, failed, no-result, superseded, or pending states.

Tenant isolation means each client workspace has scoped identity, storage, retrieval, permissions, exports, and audit boundaries.

AI draft is generated content requiring review. It is not canonical state, a final deliverable, or an approved decision until accepted through the appropriate gate.

Workspace architecture

The workspace has seven layers:

Layer	Purpose	Example records
Curriculum	Guides implementation sequence	module, task, prerequisite, acceptance criterion
State	Stores current approved context	canonical field, unknown, conflict, supersession
Submission	Captures human input	answer, upload, comment, correction
Review	Evaluates inputs and drafts	review note, issue, approval request
Decision	Records authority	decision packet, approval, rejection, deferral
Deliverable	Produces client artifacts	artifact version, export, final packet
Outcome	Tracks implementation result	completion, blocker, rollback, learning record

The workflow is not linear in practice, but state transitions should be explicit:

```
curriculum.assigned
submission.created
submission.validated
ai_draft.created
review.requested
review.completed
decision.requested
decision.approved | decision.rejected | decision.deferred
canonical_state.updated
deliverable.version_created
deliverable.approved
outcome.recorded
```

AI draft is deliberately before review and decision. A generated implementation plan can be useful, but it cannot be treated as final merely because it exists. A draft must reference source submissions and current canonical state. If the draft includes unsupported claims or fills unknowns, review should block it.

Data model and invariants

A submission record:

```
{
  "submission_id": "sub_001",
  "tenant_id": "tenant_ref",
  "curriculum_item_ref": "module_03_question_05",
  "submitted_by": "principal_ref",
  "submitted_at": "2026-07-16T14:00:00Z",
  "content_type": "structured_answer",
  "state": "awaiting_review",
  "evidence_refs": [],
  "supersedes": null
}
```

A decision record:

```
{
  "decision_id": "dec_001",
  "tenant_id": "tenant_ref",
  "decision_type": "implementation_scope",
  "subject_refs": ["state_field_12", "deliverable_v4"],
  "evidence_refs": ["sub_001", "review_002"],
  "decision": "approved",
  "approved_by_role": "client_owner",
  "effective_at": "2026-07-16T15:00:00Z",
  "expires_at": null,
  "conditions": ["applies_to_current_version_only"]
}
```

Core invariants:

Invariant	Purpose
A canonical state field must trace to approved evidence or an explicit unknown.	Guards against chat summaries becoming facts.
A submission cannot silently overwrite canonical state.	Preserves review and supersession.
An AI draft cannot be final approval.	Preserves human authority.
A deliverable version must identify source state, evidence, reviewer, and approval state.	Keeps version relationships reviewable.
Tenant identifier is required on every object and query.	Supports isolation and export.
Decisions are effective-time records and can be superseded, not erased.	Preserves auditability.

Implementation

SalesFusion uses the workspace foundation as a portal model with structured curriculum, submissions, reviews, versioned deliverables, evidence references, and state boundaries. The implementation pattern is intentionally different from a chat transcript. Chat or AI assistance may exist inside the workspace, but it writes proposals, drafts, or review aids. Canonical state changes require validation and approval events.

The interface should make state visible without turning the portal into administrative clutter. A user should see assigned curriculum, pending submissions, unresolved questions, drafts needing review, decisions awaiting approval, current deliverables, and blockers. The system should preserve prior versions and show what changed between versions.

Tenant isolation is enforced at the resource level. Each request is authenticated and tenant-scoped. Cross-tenant object identifiers should fail closed and create audit events. Exports should include only tenant-scoped, permitted, non-secret material with a manifest of included objects.

An implementation API can use resource boundaries such as:

```
POST /workspaces
GET  /workspaces/{id}/curriculum
POST /submissions
POST /submissions/{id}/review
POST /drafts
POST /decisions
POST /deliverables
POST /deliverables/{id}/versions
POST /outcomes
POST /exports
```

HTTP success indicates the request was accepted by the service, not that consulting work is complete. Completion requires the relevant review, decision, deliverable, and outcome state.

Evaluation

The evaluation should begin with workflow correctness and user comprehension. It should not start with claims about project outcomes. A realistic study can ask client-side and operator-side participants to complete tasks inside a controlled workspace.

Tasks:

- Find the source behind a canonical state field.
- Correct a stale submission.
- Distinguish AI draft from approved deliverable.
- Approve one decision and reject a broader one.
- Locate the current version of a deliverable.
- Identify unresolved blockers.
- Export workspace state.
- Reconstruct why a decision was made.

Metrics:

Metric	Definition
Task completion	Users can complete required workspace tasks.
Review-cycle clarity	Users can identify what needs review and why.
Draft/approval distinction	Users do not mistake AI drafts for final decisions.
Retrieval accuracy	Users find current state and evidence without wrong-version errors.
Correction success	Stale or incorrect submissions are corrected without overwriting history.
Tenant isolation tests	Cross-tenant access attempts fail closed.
Export completeness	Export contains expected permitted objects and excludes restricted material.

Later field studies can measure completion, review-cycle duration, support requests, and implementation outcomes, but only with visible method, sample, baseline, and limitations.

Risks and failure modes

The main failure mode is state confusion. A draft becomes treated as final. A submission becomes treated as approved. A prior deliverable version remains in circulation. A decision is remembered but not recorded. Another failure is tenant leakage, where retrieval, examples, indexes, exports, or operator views cross workspace boundaries.

AI-specific failures include unsupported completion of missing fields, overconfident summaries, accidental disclosure of restricted content, and prompt injection inside uploaded material. Controls include draft labeling, review gates, provenance, tenant-scoped indexes, source handling rules, export checks, and audit events.

There is also a product risk: too much governance can make the workspace feel heavy. The design should expose the state needed for action without forcing users to manage implementation metadata directly.

Limitations

The workspace model does not ensure successful consulting delivery. It cannot make poor curriculum good, replace expert review, or force client participation. Tenant isolation depends on correct implementation and testing. AI draft boundaries depend on product surfaces that make draft status unmistakable.

SalesFusion's current evidence supports an implemented workspace foundation. It does not support claims about usability improvement, faster completion, reduced project risk, or better outcomes.

Practical implications

Consulting portals should be designed around implementation state, not around content browsing. Every important client answer, review, decision, deliverable, and outcome should have a record, version, evidence link, and owner. AI should accelerate drafting and review preparation, but final authority should remain a separate state transition.

Teams should invest early in tenant isolation, versioning, export, and draft labeling. These controls are harder to retrofit after users rely on the workspace as the system of record.

Conclusion

A consulting portal for AI implementation should be a governed workspace. Curriculum, canonical state, submissions, reviews, versions, decisions, deliverables, evidence, outcomes, tenant isolation, and AI draft boundaries are the structure intended to keep implementation work reviewable rather than scattered across transcripts and ambiguous documents. SalesFusion uses the foundation of this model and proposes usability and workflow evaluations before making stronger claims.

References

- ISO, "ISO/IEC 42001:2023 - Artificial intelligence management systems." https://www.iso.org/standard/42001
- National Institute of Standards and Technology, "Zero Trust Architecture," NIST SP 800-207. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
- IETF, "RFC 9110: HTTP Semantics." https://www.rfc-editor.org/rfc/rfc9110.html

Changelog

- 2026-07-17 — Version 1.0.0: founder-approved first public release; publication metadata, evidence-status box, limitations, and verified references retained.