Field Survey Support Ticketing Platform

Project Vision & Architecture (v 1 – Constrained MVP)

---

1 . Executive Vision

1 . 1 Purpose

Build a multi-tenant, enterprise-capable ticketing platform specifically designed for Field Survey Suppo operations on construction projects.

The platform will: - Replace email, spreadsheets, and ad hoc communication - Provide structured intake and assignment - Create defensible audit trails - Deliver operational insight by project, ar and survey crew

This document defines a **constrained MVP (v 1)** to prevent scope creep and maintain architectural discipline.

All advanced customization and configuration capabilities are intentionally deferred.

---

2 . Product Philosophy (Guardrails)

1 . Workflows are opinionated and fixed in v 1 .
2 . Roles are predefined.
3 . State transitions are deterministic.
4 . Reporting is operational, not BI-grade.
5 . Multi-tenancy isolation is non-negotiable.
6 . Architecture must support enterprise evolution without premature complexity.

Anything outside these principles is locked behind "Future Configuration".

---

3 . Core Problem Statement

Construction projects generate high volumes of survey-related requests that must: - Be approved execution - Be assigned to field crews - Produce verifiable completion artifacts - Be traceable fo and legal defensibility

Existing systems (email, Excel, generic ticket tools) lack: - Domain alignment (areas, crafts, survey Proper audit history - Field-oriented assignment dashboards - Structured cancellation controls

This platform becomes the system-of-record for survey support activity.

4.v1 Scope Definition

4.1 Multi-Tenant Model

- Each Tenant may contain multiple Projects.
- Each Project contains tickets, users, areas, crafts.
- Tenants are strictly isolated.
- No cross-tenant visibility.

4.2 User Roles (Fixed in v1)

Tenant-Level

- Tenant Admin
- Billing/Admin Viewer

Project-Level

- Requester
- Approver
- Survey Lead
- Survey Crew (Party Chief / Instrument Man)
- Viewer

Roles are fixed and not configurable in v1.

5.v1 Workflow Variants (Strictly Limited)

Only two workflow variants exist in v1.

No dynamic workflow builder. No custom state creation. No per-project workflow editing.

5.1 Workflow Variant 1: Standard Approval

Purpose

Used for planned survey work requiring review before field execution.

State Machine

Draft → Submitted → Approved → Assigned → In Progress → Completed

Rejection Path: Submitted → Rejected

Cancellation Path: Assigned/In Progress → Cancel Requested → Cancel Approved/Cancel Rejected

Flow Description

1. Requester creates and submits ticket.
2. Approver reviews:
3. Approve → ticket moves to Approved
4. Reject → ticket moves to Rejected
5. Survey Lead assigns crew.
6. Assigned crew executes task.
7. Completion requires artifact upload (optional but encouraged).
8. Status moves to Completed.

Cancellation requires separate sign-off by Approver or Survey Lead.

5.2 Workflow Variant 2: Task Completion (Direct Assignment)

Purpose

Used for urgent or pre-authorized work.

State Machine

Created → Assigned → In Progress → Completed

Cancellation Path: Assigned/In Progress → Cancel Requested → Cancel Approved/Cancel Rejected

Flow Description

1. Request created (auto-approved).
2. Survey Lead assigns crew immediately.
3. Crew executes task.
4. Completion logged with optional attachments.

No approval gate exists in this variant.

---

6. Ticket Requirements (v1)

Each ticket must support: - Project - Area - Subarea - Craft - Requester - Assigned Survey C Status - Timestamps (Created, Approved, Assigned, Completed) - Attachments (.dwg, .dxf, .csv, photos - Comment thread - Immutable event history

---

7. Reporting (Operational Only)

v1 Reporting Includes: - Open vs Closed by Project - Requests by Area - Requests by Craft Crew workload - Daily summary report (per project)

v1 does NOT include: - Custom report builders - Cross-tenant analytics - Predictive modeling enforcement engine

---

8. Security & Isolation

Authentication

- Email/password (hashed securely)
- MFA optional
- SSO reserved for future phase

Authorization

- Role-Based Access Control (RBAC)
- All queries scoped by tenant_id

Data Protection

- TLS in transit

- Encrypted storage at rest
- Object storage for attachments
- Attachment size limits enforced

Audit

- Append-only event log per ticket
- Records state changes, approvals, assignments, cancellations

---

9. Modular Monolith Architecture

9.1 Architectural Principle

Single deployable application. Strict internal module boundaries. Clear separation of domain logic. distributed system complexity in v 1 .

9.2 High-Level Components

Frontend (React/Next.js)

API Layer (REST)

Application Core (Modular Monolith)

PostgreSQL (System of Record)

Object Storage (Attachments)

Background Worker (Async jobs)

Redis (Cache, optional in early v 1)

9.3 Internal Modules (Code-Level Boundaries)

1. Identity Module

- Users
- Authentication
- Password hashing
- Session management

2. Tenancy Module

- Tenant creation
- Project management
- Role assignments
- Membership validation

3. Ticket Module

- Ticket creation
- State machine enforcement
- Assignment logic
- Cancellation logic

4. Workflow Module

- Implements fixed state transitions
- Enforces workflow variant selection
- Validates legal transitions

5. Attachment Module

- File metadata
- Pre-signed upload URLs
- Access control validation

6. Notification Module

- Email notifications
- Async dispatch

7. Reporting Module

- Aggregated queries
- Daily report generation

8. Audit Module

- Append-only event recording
- State change tracking

Each module communicates via application service layer, not direct cross-table manipulation.

---

10. Data Model Overview (Simplified)

Core Tables: - tenants - projects - users - tenant_memberships - project_memberships - tickets ticket_events - approvals - attachments

All domain tables include tenant_id.

Indexes required for: - tenant_id + project_id + status - tenant_id + project_id + assigned_user - project_id + created_at

---

11. Background Processing

Async Jobs Include: - Email notifications - Daily report generation - Attachment validation

Jobs handled by worker process within same codebase.

---

12. Performance Constraints

Design target: - 1,000–50,000 tickets per project - Low-latency operational dashboards

Mitigation strategy: - Proper indexing - Paginated API responses - Pre-aggregated daily rollups (if needed)

---

13. Explicitly Deferred (Future Configuration)

The following are locked behind future phases:

- Custom workflow builder
- Configurable states
- Per-project workflow edits
- Custom role creation
- SCIM provisioning
- Enterprise SSO
- Cross-project analytics
- Advanced SLA engine
- In-browser DWG viewer
- External integrations (Procore, Autodesk, Teams)
- Multi-step conditional approval trees

---

14. Success Criteria for v1

1. Multi-tenant isolation is airtight.
2. Both workflow variants operate deterministically.
3. Audit trail is complete and immutable.
4. Operational reports are accurate and performant.
5. System handles 10,000+ tickets in a project without degradation.

---

15. Architectural North Star

Remain a disciplined modular monolith until: - Load demands separation - Team size requires extraction - Enterprise clients demand isolation boundaries

Premature microservices are explicitly rejected.

End of Document