

ST Engineering | Project Harmony

AI for Operations Demo Prototype Spec (Front-end Storyboard)

Version: v0.1 (draft)

Purpose: Specify a front-end only prototype that reuses interaction patterns from the supplied Palantir demo video, but re-framed to ST Engineering priority scenarios (OR + CA MRO).

Important note on scope and fidelity

This spec is written so a UI team can build a high-fidelity prototype with hardcoded or mocked data. It simulates actions, approvals, and ontology navigation using static JSON and deterministic rules.

1. Demo goals

Demonstrate how STE can run daily MRO and Operations Research decisions from one operational cockpit.

- Unified operational view. One place to see hangar slots, work packages, parts, and risks.
- Explainable recommendations. Users see why a risk exists and what options exist.
- Action with controls. Users execute actions with approvals and full audit (simulated in the prototype).

2. Personas and decision owners

- **Hangar Operations Planner (Site):** Owns slot plan, redelivery commitments, and buffer decisions.
- **Material Planner (Site):** Owns critical spares readiness and PO expedite/hold decisions.
- **Component Services Manager:** Owns rotatable pool availability, repair prioritisation, and transfers between pools.
- **Finance / Commercial (MBH):** Owns revenue assurance exceptions, accrual logic, and billing release.
- **Approver (Ops Lead):** Approves high-cost actions, slot swaps, contractor augmentation.

3. Scenarios

3.1 Scenario 1: C-check readiness and critical spares replanning (single site MVP)

Narrative: One hangar site runs multiple scheduled checks. Aircraft arrival times change. Critical parts and repairables must be available at need date. Planners need a single workbench to decide expedites, holds, transfers, and substitutions.

Primary decision products:

- Daily 'Red Parts' list by Work Package, with predicted delay to redelivery date.
- Planner action recommendations for each red part (expedite, transfer, swap, alternate, borrow).
- A one-click 'Action Pack' that creates the required transactions and routes approvals.

Key KPIs (prototype must display):

- On-time induction and on-time redelivery (by hangar, by fleet type).
- Critical part availability at need date (%).
- Number of expedites, number of PO holds, and net working capital impact.
- Mean time to resolve an alert (triage-to-action).

3.2 Scenario 2: Component Services + Maintenance-by-the-Hour revenue assurance

Narrative: Component Services supports airlines under flying-hours based contracts. Rules are managed in Excel. STE wants to move to governed logic with auditability, faster exception handling, and better component pool decisions.

Primary decision products:

- MBH revenue assurance dashboard (flying hours, accruals, exceptions, billing readiness).
- Component pool health view (availability, repair TAT risk, obsolescence, reposition recommendations).
- Exception-to-action flow for component shortages that may cause AOG or service level breach.

Key KPIs (prototype must display):

- Accrual variance to expected (by contract, by fleet).
- Revenue exception cycle time (detect to resolve).
- Rotable availability at outstations (coverage days).
- Repair TAT adherence and avoidance of AOG events (simulated).

4. Ontology model (front-end data model)

Prototype must expose an Ontology Explorer and consistent object cards. Objects are navigable with relationships that support drill-downs and graphs.

4.1 Core objects (examples)

- **Aircraft:** tailNumber, fleetType, customer, status, location
- **CheckEvent:** eventId, type (C/D), inductionDate, redeliveryDate, hangar, status
- **WorkPackage:** wpId, aircraft, checkEvent, scope, criticality, status
- **TaskCard:** taskId, wpId, requiredSkills, plannedHours, prerequisites, status
- **Part:** partNumber, description, category (consumable/repairable/rotable), leadTimeDays
- **PartDemand:** demandId, wpId/taskId, partNumber, qty, needDate, criticalFlag
- **InventoryPosition:** location, partNumber, onHand, reserved, available, safetyStock
- **RepairOrder:** roId, componentSerial, shop, promisedDate, status, TATDays
- **PurchaseOrder:** poId, supplier, partNumber, qty, status, promisedDate, expediteFlag
- **Shipment:** shipmentId, mode, origin, destination, eta, status
- **ComponentSerial:** serial, partNumber, condition, pool, lastOverhaulDate

- **ComponentPool:** poolId, coveragePolicy, locations, targetLevels
- **ContractMBH:** contractId, customer, fleetType, rateCard, ruleSetId
- **FlyingHoursReport:** reportId, customer, fleetType, period, flyingHours, source
- **BillingLine:** billingId, contractId, period, flyingHours, amount, status
- **Alert:** alertId, type, severity, owner, status, impactedObjects, createdAt
- **Action:** actionId, type, payload, status, approvalsRequired, createdAt
- **ApprovalTask:** taskId, actionId, approverRole, status, decisionAt
- **User:** userId, name, role, site

4.2 Relationships

- Aircraft 1..* CheckEvent
- CheckEvent 1..* WorkPackage
- WorkPackage 1..* TaskCard
- TaskCard 0..* PartDemand
- PartDemand -> Part
- PartDemand -> InventoryPosition (by partNumber and location)
- PartDemand -> PurchaseOrder / Shipment / RepairOrder (supply coverage)
- Alert -> impacted objects (WorkPackage, PartDemand, ComponentSerial, ContractMBH, BillingLine)
- Action -> linked Alert(s) and impacted objects
- Action -> ApprovalTask(s)

5. UI specification (screens and components)

Prototype follows the interaction style of the supplied demo: left navigation, central workspace, drillable object cards, scenario builder, and action flows.

5.1 Global layout

- Top bar: thin Accenture purple bar with 'ST Engineering | Project Harmony | AI for Operations Demo'. Right side shows 'Accenture | Palantir'.
- Left rail: module navigation (Control Tower, Alerts, Scenarios, Automations, Approvals).
- Global search: tail number, work package, part number, contract, alert ID.
- Breadcrumbs: e.g., Alerts > Alert A-1023 > WorkPackage WP-55.
- Right insights panel (inside workspace): KPI cards and context for selected object.

5.2 Page: Operations Control Tower

- Map/network view of locations (hangars, component shops, outstations).
- KPI tiles: active checks, red parts count, predicted delay risk, component pool coverage, MBH exceptions.
- Filters: site, fleet type, customer, event type, time window.
- Click location -> location drawer with active events + top alerts.
- Click KPI -> jump to filtered Alerts view.

5.3 Page: Alerts workbench

- Table columns: Alert ID, type, severity, owner, impacted location, expected impact, status.
- Facet filters: site, fleet, event type, alert type, status, severity.
- Bulk actions: assign owner, change status, create scenario.
- Row click opens Alert detail.

5.4 Page: Alert detail and investigation

- Header with alert summary and status pill.
- Tabs: Timeline, Explanation, Impacted Objects, Actions.
- Timeline chart: inventory vs safety stock vs projected demand coverage over time.
- Explanation: causal graph and key drivers.
- Impacted objects: WorkPackage, PartDemand, POs, RepairOrders, Shipments.
- Buttons: Acknowledge, Create scenario, Recommend actions.

5.5 Page: Scenario builder (what-if)

- Stepper: Begin -> Simulate -> Execute -> Automate.

- Simulate: 2–5 recommended options with compare metrics (days saved, cost, knock-on impact).
- Execute: build Action Pack with 1..n actions.
- Each action has an approval rule (threshold-based).

5.6 Page: Approvals and actions

- Approvals queue by role.
- Approval modal shows recommendation, evidence links, cost/time impact.
- Action status lifecycle: Draft -> Submitted -> Approved -> Executed.
- Audit trail panel: who approved what, when, why.

5.7 Page: Automations

- Automation library with toggles.
- Automation details: trigger condition, checks, action template, approval policy.
- Run log: inputs, proposed actions, approval outcome.
- Finalize automation simulates productionisation.

6. Mock API contracts

- GET /api/search?q=... -> ontology search results
- GET /api/kpis?site=... -> KPI tiles
- GET /api/alerts?filters=... -> alerts table rows
- GET /api/alerts/{id} -> alert detail + impacted objects
- POST /api/scenarios -> create scenario from alert
- POST /api/scenarios/{id}/simulate -> recommended options
- POST /api/actions -> create action pack
- GET /api/approvals -> approval tasks
- POST /api/approvals/{id} -> approve/reject
- GET /api/automations -> automation list
- POST /api/automations/{id}/run -> simulate automation run

7. Demo script (narration outline)

1. **Start in Control Tower:** Show global view. Filter to one site. Highlight KPIs and top exceptions.
2. **Ask the AIP agent:** Ask: 'What changed since yesterday for Hangar 3 and Component Pool A?' Agent summarises and points to alerts.
3. **Triage alerts:** Open Alerts view. Sort by severity and expected delay impact. Assign one alert to yourself.

4. Investigate one alert: Open alert detail. Show timeline and causal graph. Identify which demand and which supply object drives risk.

5. Simulate options: Create scenario. Compare transfer vs expedite vs repair reprioritisation. Show trade-offs.

6. Execute action pack: Submit chosen action with approvals. Show simulated SAP write-back and audit.

7. Repeat in MBH view: Switch to MBH dashboard. Show exception where flying hours do not match accrual. Create an action to resolve and release billing.

8. Branding and UX requirements

- Consistent Accenture purple top bar across all pages.
- No 'Supply Chain Demo' right panel and no Palantir marketing footer inside the app workspace.
- Use ST Engineering vocabulary in labels (Hangar, Work Package, Tail, Rotable, Repair Order, Flying Hours, MBH).
- All key objects are clickable and show an object card with key fields and related links.
- All charts and tables support filters and drill-down consistent with the original demo.