

Intelligent Space Platform

A Reference Architecture for an Open-Source
Cloud-Native Ground System

Open-Source • Cloud-Native • Self-Healing

chris.bomhardt@ra231.com • rightascension231.com

The Ground System Challenge

Vendor Lock-In

Proprietary systems create single vendor dependency

- Closed APIs limit integration
- Expensive licensing models
- Limited portability

High Operational Costs

Legacy infrastructure drives up sustaining costs

- Manual operations overhead
- Dedicated hardware expenses
- Limited resource sharing

Limited Scalability

Monolithic architectures constrain growth

- Single-mission silos
- Poor resource utilization
- Inflexible deployment

Complex Compliance

Federal security requirements are increasingly stringent

- Zero Trust mandates
- Continuous monitoring
- ATO documentation burden

Challenge: Organizations need flexible, cost-effective, compliant systems that scale

ISP Architecture: Three Main Layers

1

Mission Applications

Mission Operations Capabilities

Telemetry, commanding, mission planning, flight dynamics, and data analytics

2

Mission Services

Unified Data Architecture

Data Lakehouse, event streams, object store, distributed queries, and analytics

3

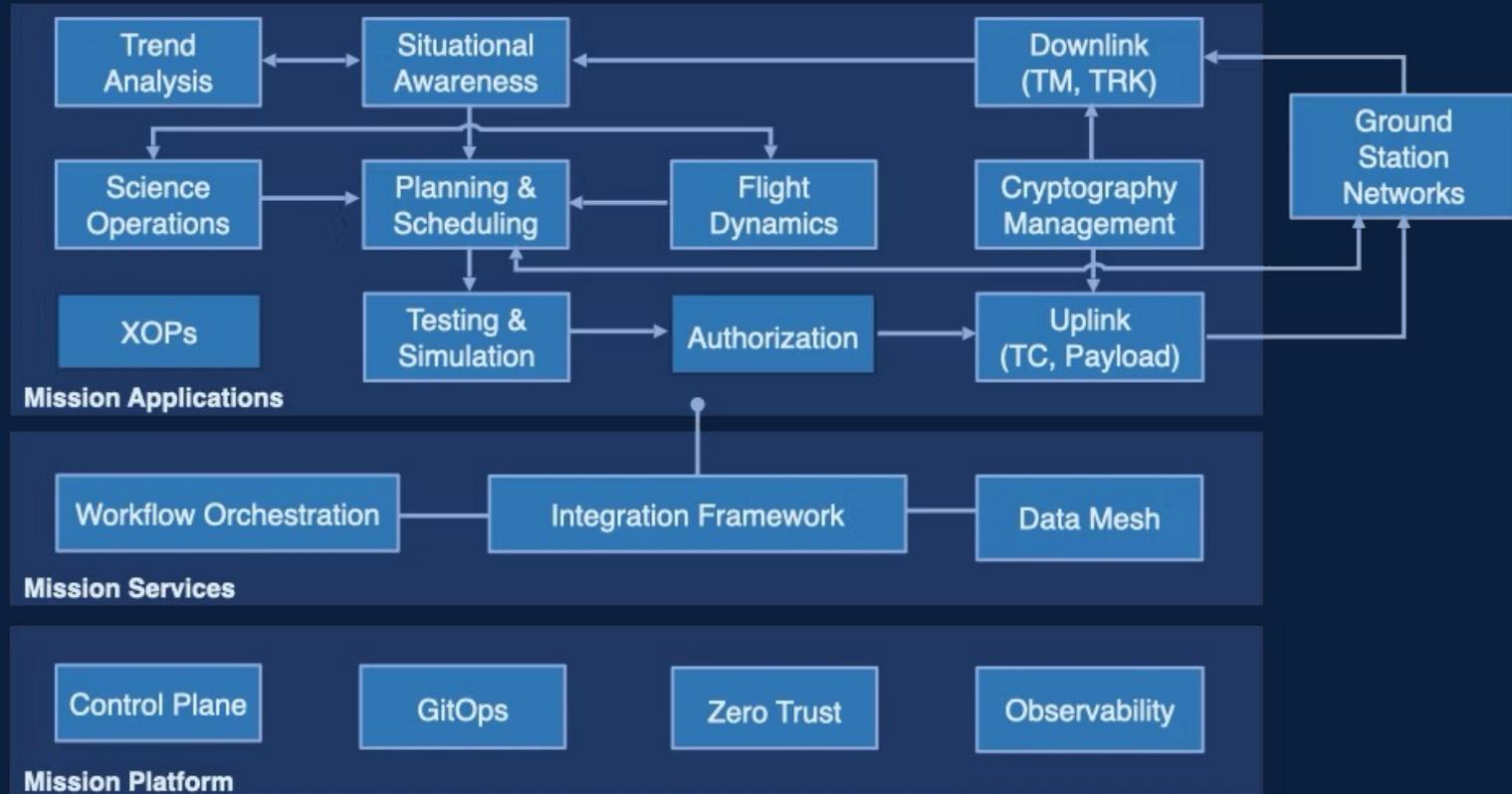
Mission Platform

Infrastructure Automation

GitOps workflows, policy enforcement, continuous delivery, and self-healing

Three architectural areas working together to provide comprehensive ground system capabilities

ISP Architecture: Three Main Areas



Application Layer: Mission Operations

Capabilities: Complete mission operations from telemetry to commanding

Mission Automation

- Observe (Tracking, Conjunction, Telemetry)
- Orient (Recognition, Conjunction Analysis, Trending)
- Decide (Planning & Scheduling, Simulation, Verification)
- Act (Commanding, Authorization, Procedures)
- Manage (Policies, Identity, Incident Reporting/Alerting)

Standards: OpenAPI (sync), AsyncAPI (async), JSON Schema, CCSDS Mission Ops ontology

Application layer provides complete OODA loop for mission operations with standards-based integration

Services: Unified Data Architecture

Capabilities: Real-time streams + Analytics = unified architecture

Integration Framework

- **Serverless Functions (Knative)** Scale to Zero adapters
- **Persistent Streams (Kafka)** Event backbone
- **Workflow (Synapse)** Orchestration Engine
- **Stream Processing (Flink)** Real-time analytics

Data Lakehouse

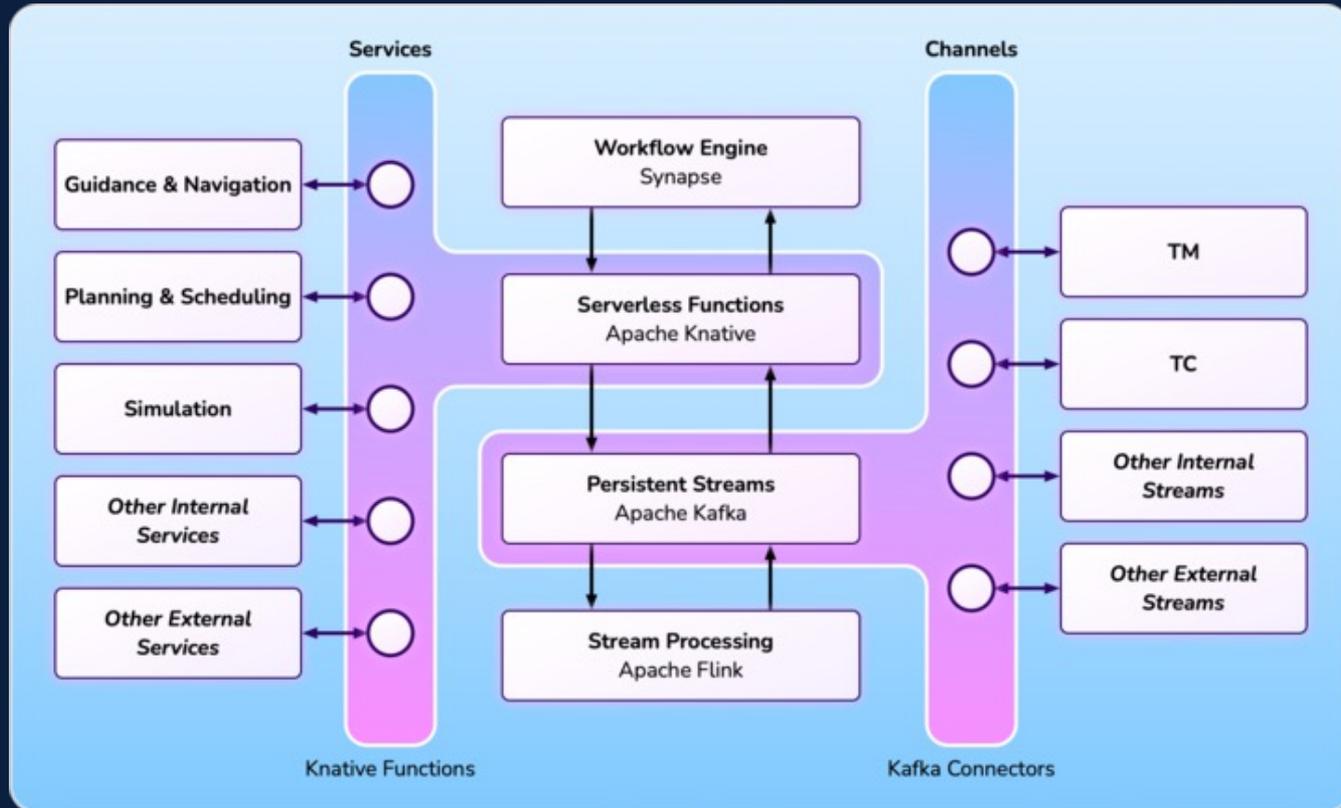
- **Data Catalog (Nessie)** Git-like versioning for data
- **Distributed Query (Trino)** Federated SQL across source
- **Tabular Data (Iceberg)** Time travel queries, schema evolution, it's a format
- **Object Data (S3 compatible)** Cost-creative storage

Stream Processing

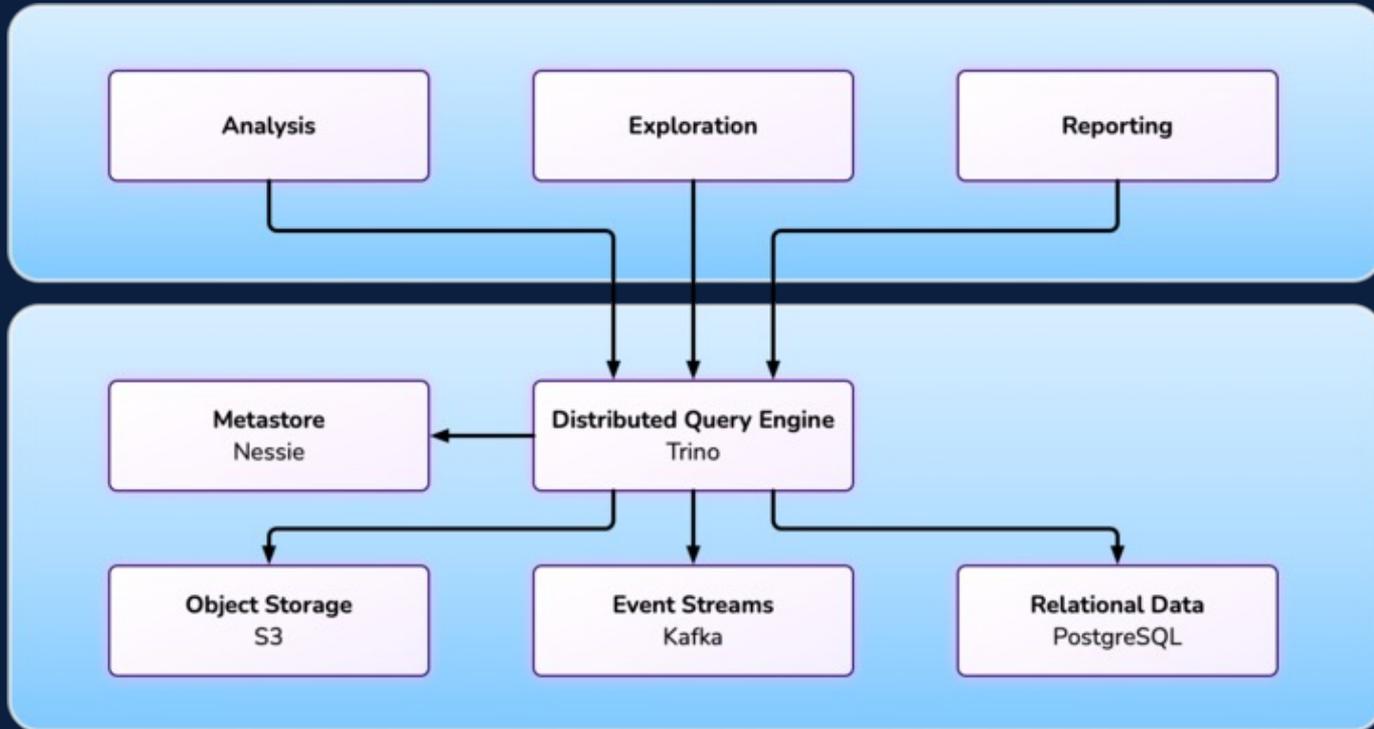
- **Persistent Streams (Kafka)** Event backbone with ordering guarantees
- **Stream Processing (Flink)** Stateful real-time analytics

Data mesh provides Lakehouse analytics + Hybrid SOA & Event Driven architecture

Services: Integration Framework



Services: Unified Data Architecture



Platform: Infrastructure Automation

Capabilities: Self-healing infrastructure with Kubernetes & GitOps

GitOps & Continuous Delivery

- **FluxCD** Declarative GitOps for Kubernetes
- **Crossplane** IaC across clouds
- **Harbor** Helm charts & container images
- **Bootstrap IaC** Initial cluster provisioning

Core Services

- **Service Mesh** Istio with Envoy proxy
- **Auto Scaling** Karpenter for node provisioning
- **Cluster DNS** CoreDNS/external DNS (Route53)
- **Observability** Grafana, Mimir, Loki, Tempo

Policy & Security

- **Policy as Code (Kyverno)** Declarative validation and mutation
- **Software Bill of Material** Automated TLS certificate management
- **Scanning** Static (Trivy) & Dynamic analysis (Falco) of workload behavior
- **IAM (Keycloak)** Cloud provider integration

Control plane provides zero-touch operations through GitOps reconciliation and policy enforcement

This is the way

Open Source

Zero lock-in

Cloud Native

Deploy anywhere

Self-Healing

GitOps automation

Auto-Scaling

Multi-mission ready

