Palantir

Foundry Technical Overview

End-to-End Data Operating System

Foundry is the operating system for the modern enterprise. Emphasis on "operating."

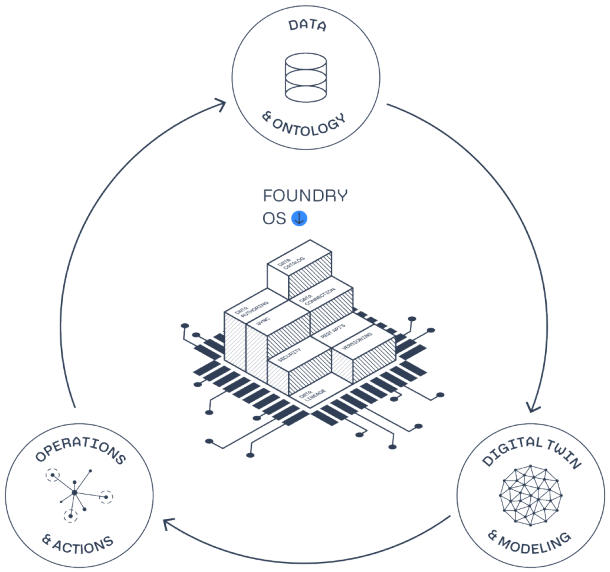
Data integration, visualization, and point analytics are helpful, but they don't provide a foundation to truly steer the organization and its operations. Our software creates software-defined feedback loops that span data, analytics, and business teams. These feedback loops are unique to Foundry and essential for local and global organizational learning.

Each layer of Foundry contributes to the goal of delivering outcomes for the business at increasing complexity and ambition.

Foundry in a Nutshell

Foundry is a highly available, continuously updated, fully managed SaaS platform that spans from cloud hosting and data integration to flexible analytics, visualization, model-building, operational decision-making, and decision capture.

Having all of these capabilities available as part of a unified platform protects against the friction and risks associated with siloed technologies, and ensures a seamless experience where data history, security, and privacy are protected and maintained.



Data-Driven Operations & Decision-Making

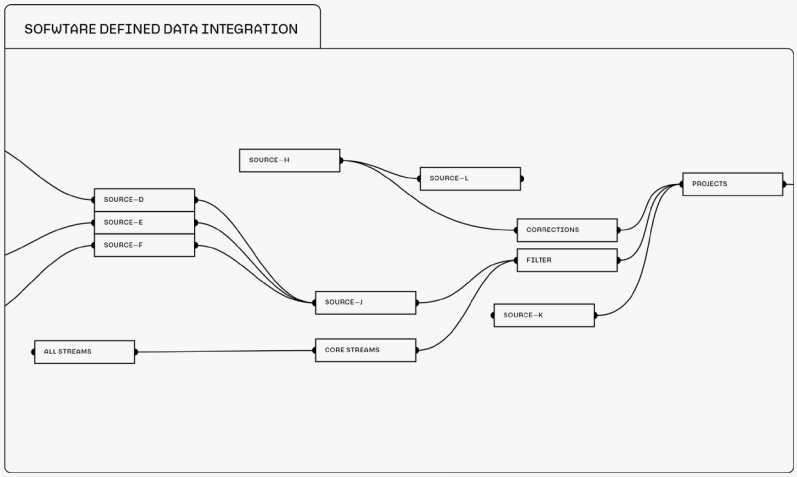
Complete awareness of every part of the company is delivered to all users, whether technically advanced or at the operational level, through intuitive and reusable tooling.

From zero to Connected Company in **less than a month**.

- Data integration is faster and cheaper over time
- Model quality is constantly tested by end users
- Custom applications can be built in days
- New strategies can be simulated, then assessed
- Decisions can be captured + synchronized

01 — Data Integration

Secure, scalable, and resilient integration of all data sources.



01. FOUNDRY DATA CONNECTION

- 200+ data connectors, leveraging an extensible plugin-based paradigm
- Flexible ingress topology, which can leverage agent-based, REST, JDBC, and other approaches
- Easy-to-configure schedules, success criteria, and permission models
- Multi-modal (structured, unstructured, streaming, IoT, geospatial, etc.)

02. DATA TRANSFORMATION

- Flexible architecture with bundled engines (Spark & Flink)
- Low-code / no-code transformation (Preparation, Contour)
- Treating Data like Code (versioning, branching, full change management)
- Full provenance through the Job Spec paradigm

03. PIPELINE ORCHESTRATION

- Build system that is engine-agnostic
- Intelligent refreshing / state-tracking across all pipelines
- Seamless integration with Foundry's health monitoring

04. SECURITY

- Role-, Classification-, and Purpose-based paradigms
- Integration with existing authorization models
- Propagation by default; extreme configurability

05. LINEAGE

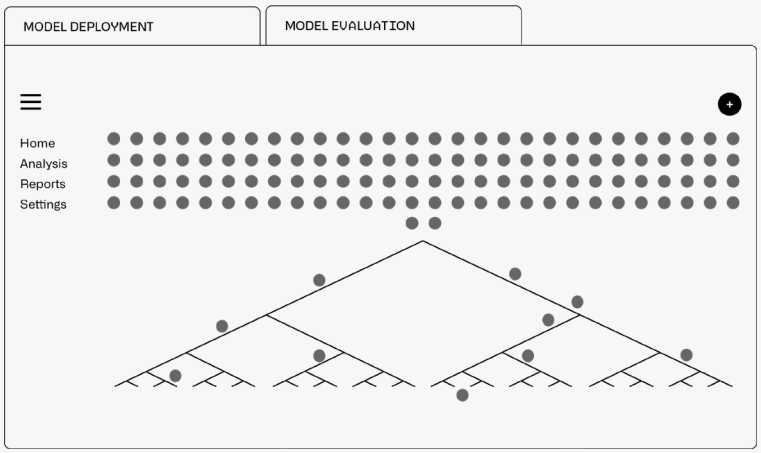
- Interwoven with security paradigm; provides immutable tracking
- Allows for impact analysis, granular usage analysis
- Rich APIs allow for navigation upstream and downstream, for a given resource

06. DATA HEALTH MONITORING

- Pre-built checks, and customizable checks
- Leverages Foundry's lineage system, for alerting and impact analysis
- Full triage & tracking through integration with Foundry Issues

02 — Model Integration

Flexible integration (or registration) of models and business logic.



01. CODE WORKBOOKS

- An integrated, end-to-end workbench for model construction (PySpark, R, SparkSQL)
- Native, secure data access for model builders (dataset and ontology paradigms)
- Integrated model training, health, and management services
- Flexible deployment options, for use in operations (batch and inference)

02. EXTERNAL MODEL INTEGRATION

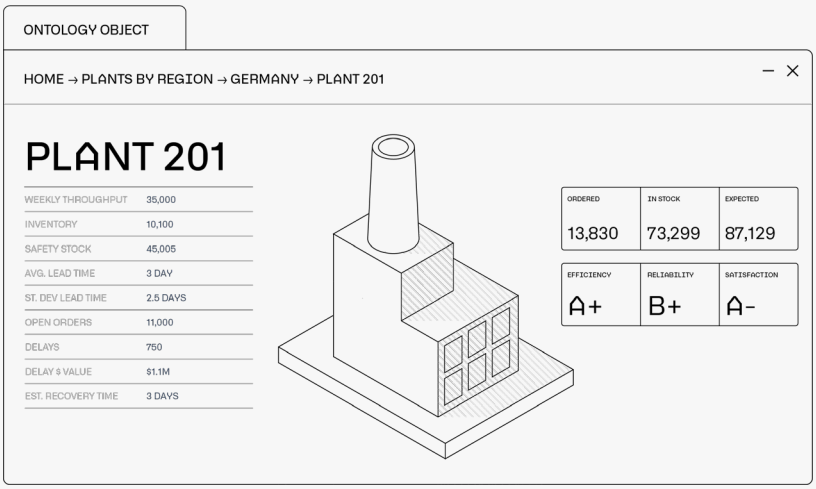
- Build and train your models in any industry-standard toolset
- API-driven connectivity to the Ontology from those external tools
- Promote into production through Foundry, when ready

03. MODEL OBJECTIVES

- “Mission Control” for models being used throughout Foundry workflows
- Rich, competitive evaluation of models; comparing performance
- Binding directly to the Ontology, which provides a “type system” for models - allowing them to be leveraged in myriad operational settings (without putting the onus on the application builder)

03 — Ontology

Connecting essential logic with data.



01. THE CORE ONTOLOGY

- Contains the key semantics of your world (objects and relations)
- Contains the key kinetics of your world (Functions, Actions)
- Integrated monitoring, and extensibility with external systems

02. DECISION CAPTURE / ENTERPRISE WRITEBACK

- Structured mechanisms for capturing data from end users, back into the ontology
- Native frameworks for propagating data capture to external systems
- External system responses can be woven into multi-step workflows
- Full provenance through the Job Spec paradigm

03. OUT-OF-THE-BOX OBJECT EXPLORATION

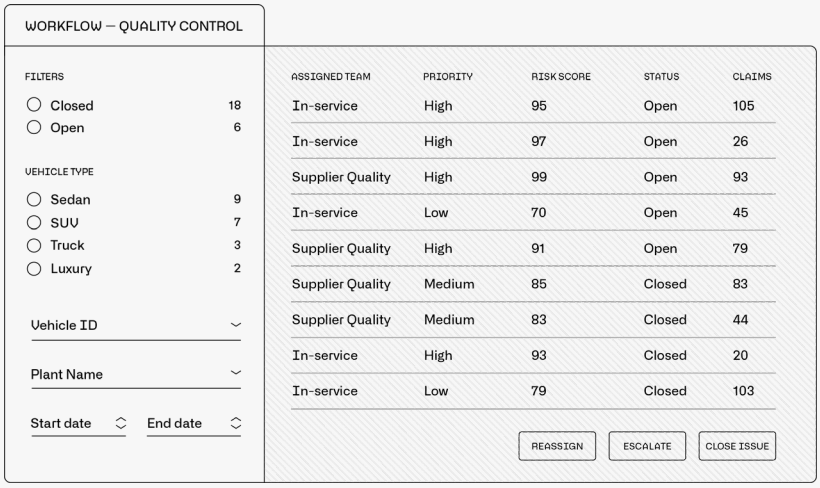
- Object Explorer provides a secure, scalable, point-and-click view into the ontology
- Quiver provides a chart-based paradigm, allowing for (among many other workflows) the navigation of multi-dimensional, real-time streaming data
- Map provides a geospatial canvas for exploring the ontology
- Each of the “base” applications is replete with a widget library that is continuously updated

04. THE FOUNDATION FOR OPERATIONS

- Workshop & Slate are end-to-end application builders that leverage the ontology
- The ontology is natively “scenario-aware”, allowing for what-if analysis and running compound simulations
- The Objects Gateway, among myriad other APIs, allows for the ontology to be leveraged in external toolsets and frameworks

04 — Workflows

Integrated exploration of possible actions.



01. EVOLVING ANALYTICS INTO WORKFLOWS

- Object Explorer is designed to incorporate Actions, and evolve exploration into workflows
- Quiver allows for the derivation of new object sets, and derived streams of real-time data
- Taurus provides a business-centric method of defining and managing rule sets

02. WORKSHOP

- Foundry's native no/low-code application builder, which operates atop the Ontology
- Manages the underlying storage, compute, ontological data and model bindings, and security paradigm - allowing application builders to focus on building
- Designed for a "low floor & high ceiling" in terms of user workflow complexity - ranging from spartan applications for a small group, to mission-critical applications at the heart of operations centers

03. SLATE

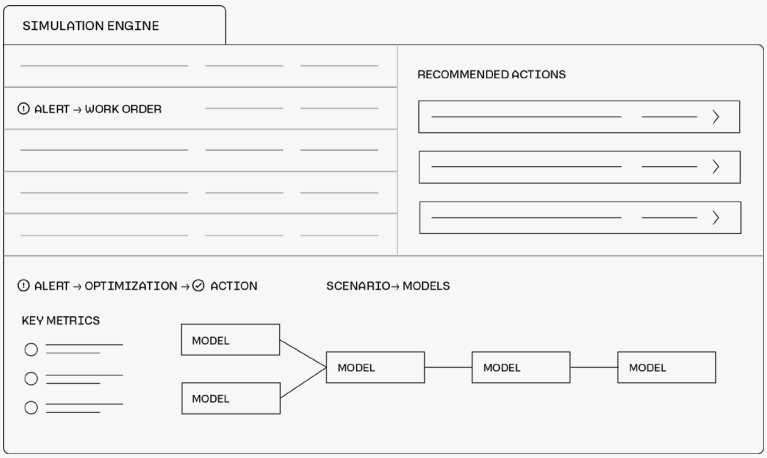
- Foundry's solution for WSIWYG, widget-driven development
- Leverages core JavaScript paradigms, intended to feel familiar to front-end developers
- Flexibility to interact with both ontological data, and tabular/SQL-shaped data

04. APIs / EXTENSIBILITY

- Objects Gateway provides secure, full-spectrum to the Ontology for builders
- Custom webhooks and writeback procedures can be authored directly in Foundry's applications
- Foundry's Third Party Authorization framework allows external client applications to be registered with the platform, and fully leverage Foundry's granular security paradigm

05 — Decision Orchestration

Synchronizing decisions
back to the source.



01. SCENARIOS & SIMULATIONS

- Treating Your Business Like Code; branch, simulate, and explore at full scale
- Simulations can leverage all types of models, including high-dimensional linear solvers
- Simulations can be tactical or long-lived, refreshing along with data and models

02. VERTEX

- Foundry's out-of-the-box application for graph/relational exploration of the ontology
- Allows for easy creation of new scenarios, and simulate "what-if" conditions
- Enables complex simulations that chain multiple models together

03. DEMOCRATIZING SCENARIOS & WHAT-IF ANALYSIS

- Scenarios defined in an analytical exploration (e.g., in Vertex) can be packaged, constrained, and surfaced to more operational users (e.g., using Workshop)
- All simulated data and scenario states can be accessed programmatically

04. SYNCHRONIZING DECISIONS BACK TO EXTERNAL SYSTEMS

- Foundry's "Actions" framework provides a structured, secure, and auditable mechanism for publishing decisions (and other data) to external systems
- Data egress leverages all of the capabilities of Foundry's Data Connection framework
- Full lineage is maintained from data to decision, allowing the organization to always ask "what was the state of the world?" when a particular piece of data or metadata was written externally