Palantir

↳ 2022

Connecting AI and Machine Learning with Operations

The Operational Connectivity Gap

Organizations are looking to leverage artificial intelligence (AI) and machine learning (ML) to accelerate and improve decision-making. But the reality of operationalizing AI/ML is complex, and the typical return on investment rarely lives up to expectations.

While organizations have embraced myriad technologies to build tailored AI/ML, the journey from development sandbox to impactful operational tool for users remains precarious – often replete with disconnected systems, data, and teams.

Many organizations lack the overarching framework to ensure that data, AI/ML models, and organizational capital can be captured and used across current and future use cases — resulting in fragmentation. One-off AI/ML models are developed to address a specific business need via a point-solution, but rarely operationalized for broader or more sustainable consumption across the organization and its processes. Operationalizing AI/ML is difficult without a solution that provides: a trustworthy data foundation, full-fidelity feedback loops between consumers and model builders, safe mechanisms for writing back to source systems, shared security and lineage frameworks spanning data, analytics, operational teams – and much more. These discrete capabilities additionally need to scale and interoperate with existing technology investments.

Foundry Enables Operational Connectivity for AI/ML

Foundry is a highly interoperable platform that connects the organization's data, analytics, and operations teams, along with their respective software systems. A key facilitator of this connectivity is the **Foundry Ontology** — a common foundation for data, logic, and models, specific to the organization.

The Ontology goes beyond business objects and relationships, capturing advanced data semantics (spatial, relational, temporal), "actions" (capturing complex chains of write-operations and system integrations), and granular permissions — all exposed via both UIs and secure APIs. An Ontology is quick to bootstrap, and typically grows over time with new use cases, capturing new data sources, objects, relationships, interactions, and processes.

For data science and AI/ML teams, this enables collaboration with business and operational teams on a shared substrate. Models — and their features — can be bound directly to the primitives and processes that drive the business; they can then be governed, released, and injected directly into core applications and systems — without additional adaptors or glue-code — and served in-platform (batch, streaming, or query-driven) or externally. As operators, business processes, and systems make decisions and take action, they write back into the Ontology, providing unprecedented feedback loops to model monitoring, evaluation, re-training, and MLOps.

For business partners, Foundry enables quick iteration towards outcomes. The Ontology and other best-in-class tooling make it easy to get started and deliver AI/ML-enabled operational outcomes — whether through new applications or augmentations to existing systems. Subsequent use cases can leverage an ever-increasing set of data, model assets, and inter-operations throughout the enterprise, decreasing time-to-value for new projects.

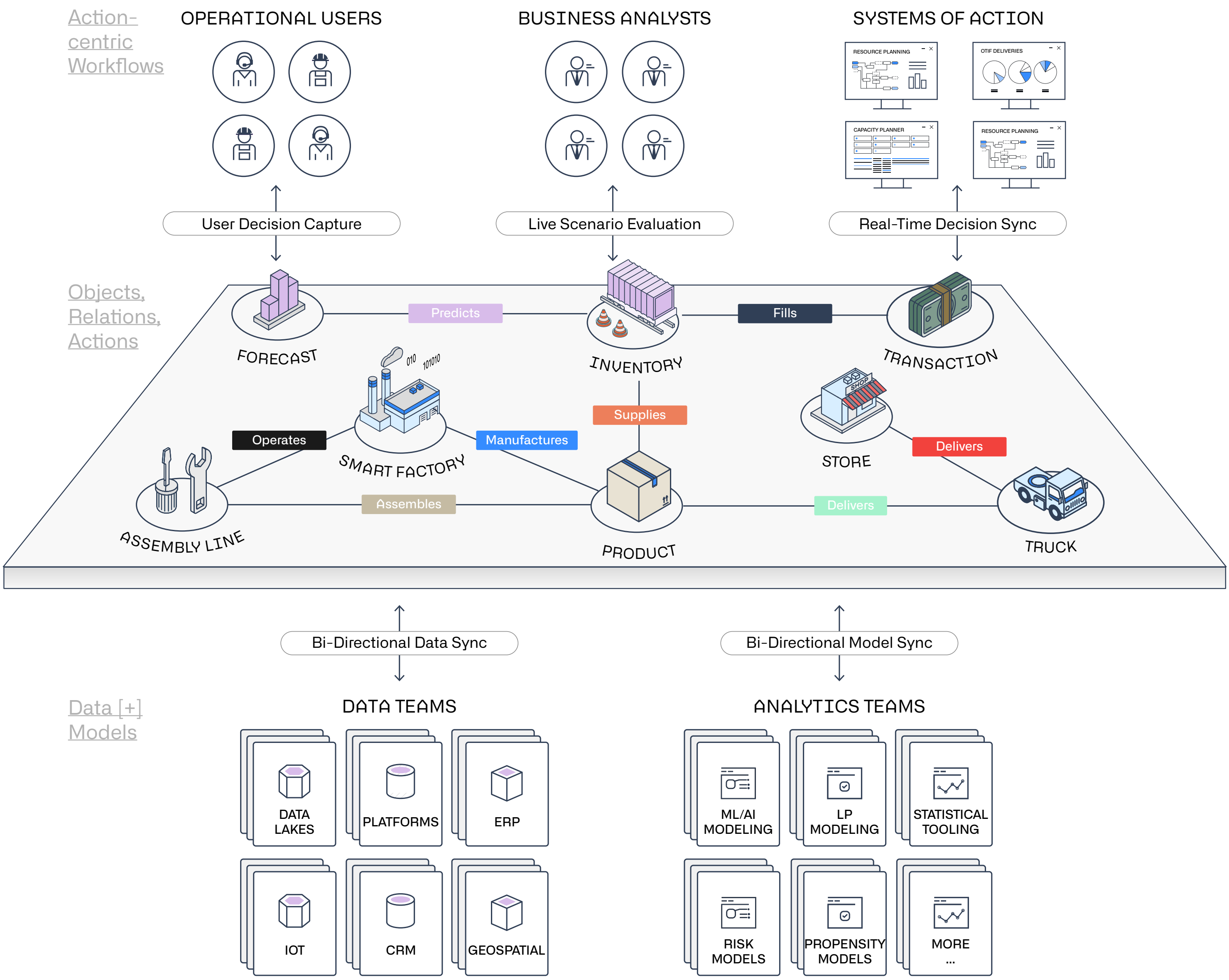


Fig. Foundry's closed-loop architecture connects data, analytics, and operations and powers enterprise-wide decision making

A Three-Step Framework for Connecting AI/ML to Operations with Palantir Foundry

01

Integrate your existing data assets and models with Foundry's tooling for building, evaluating, managing, and deploying models

Leverage automated software-defined integrations, with lineage from data to decision, to rapidly connect data and models outside of Foundry. Augmenting and amplifying existing data and model investments in hours, not months.

02

Bind models to the Ontology and deploy them to frontline users

Move your models out of the lab and onto the frontlines of the organization, enabling technical and non-technical users alike to interact with model levers, test "what-if" scenarios, run large scale simulations, and make decisions through entirely customizable user-facing applications.

03

Close the loop with operators and write back decisions to source systems with complete lineage and auditability → enable continuous learning

Continuously improve models, through the rich operational feedback generated by end user decisions and validation. Orchestrate granular feedback to existing enterprise systems, including source systems and systems of action.

The following sections use a notional enterprise ("Titan Logistics", operator of a multimodal logistic network) to illustrate how a team of data scientists can begin operationalizing their models through the Foundry platform. The preventative maintenance workflow described is a synthesis of many real-world implementations that Foundry has enabled – across shipping and logistics, energy, heavy manufacturing, aerospace, mining, medical manufacturing, defense, and telecommunications.

"Titan Logistics" → Case Study

Closing the Loop

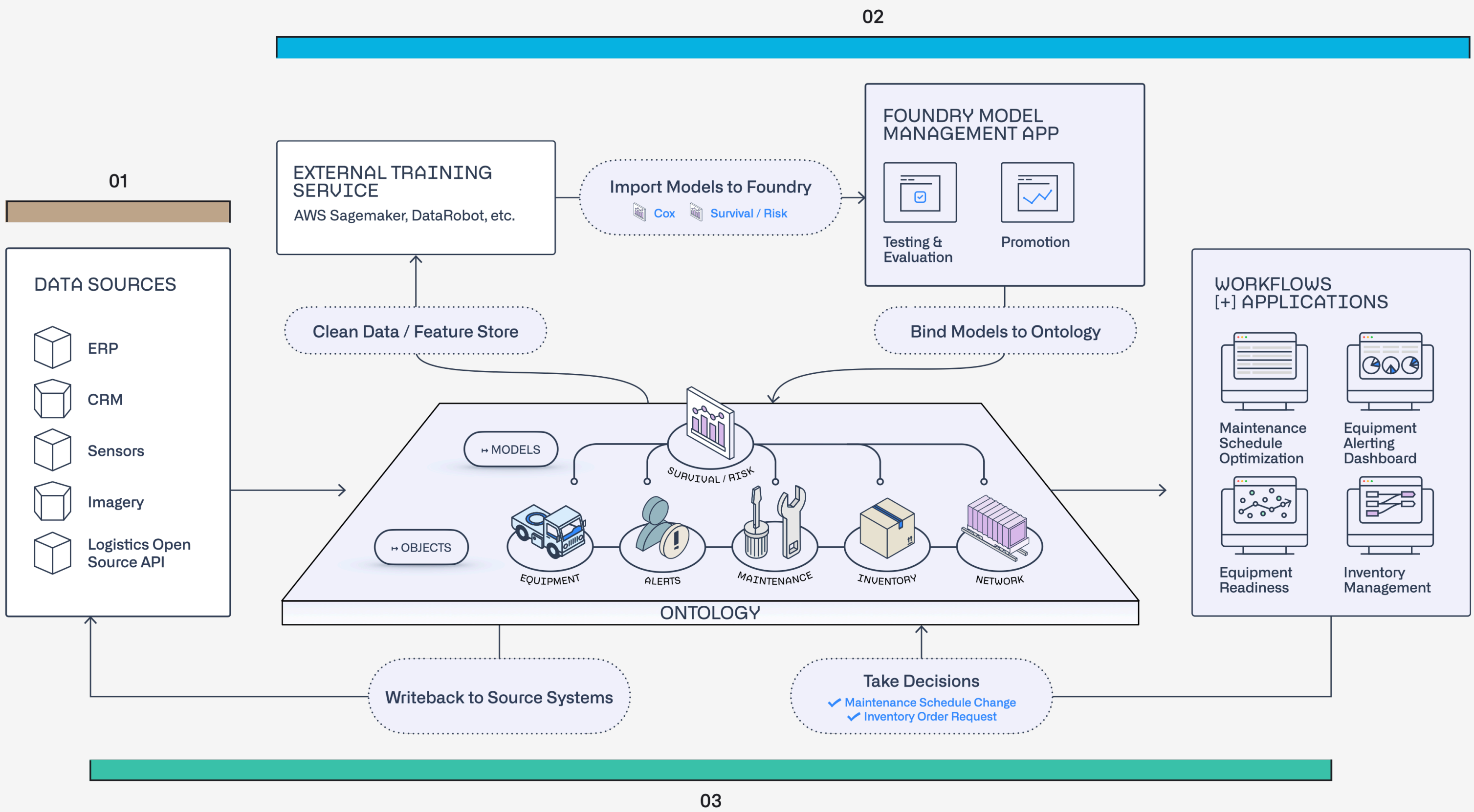
The data science team at Titan Logistics has historically used a collection of industry-standard tools (e.g., AWS SageMaker, DataRobot) for creating their AI/ML models. The team has built fault detection models that leverage the latest techniques from industry and academia. Each day, their business counterparts exported spreadsheets of model outputs for maintenance engineers. Engineers would then review recommendations in conjunction with context from other applications — a time-consuming and error-prone process — before committing decisions to a legacy scheduling system. On the frontlines of the business, the engineers then need to cross-check the modeling outputs with various maintenance schedules, and often perform ad-hoc analysis based on their real-world experience. Moreover, the models often bake in stale historical assumptions, requiring the engineers to manually correct the same discrepancies every time the data science team delivers a fresh set of outputs.

For the data science team, developing models is also a fairly manual process – involving the collation of insights from various data extracts and the need to perform ad-hoc experiments. While the team had been able to use raw datasets to create a performant model, they still faced friction and limitations around:

- Manual data curation and feature engineering
- Limited flexibility of models to handle new parts/equipment
- No connectivity between different analytics silos (model chaining, large scale simulation, etc.)
- Absence of user-facing application for operator interaction
- Lack of broader operational context to support accurate decision making at the operational level
- No write-back — not improving models from user insights

Fortunately, Titan has just deployed Foundry – unlocking the ability to effectively address these shortcomings and streamline the operationalization of the models. We'll examine each step on the journey, ultimately concluding with the impact on Titan Logistics' operations.

Fig. Titan's systems and interactions across the model lifecycle



→ External Data & Tooling

→ Foundry Tooling & Processes

01

Integrate existing data assets and models with Foundry's tooling for building, evaluating, managing, and deploying models

Data Integration

The team's first step is to connect Foundry to various data sources, building fully-automated data synchronization pipelines from thousands of sensors and legacy data systems (e.g., ERP, MES, geospatial, sensor data, imagery, semi-structured sources) into the Foundry Ontology. They set up granular permission structures for robust security control and data health checks to enforce data quality as part of this process.

Via a suite of native connectors and automation, Foundry enables bi-directional connection to operational and transactional systems (e.g. ERP, CRM, MES, Asset Config, Edge), allowing decisions made in Foundry to propagate holistically throughout the data landscape.

For preventative maintenance, Titan maintenance engineers need to inform decision-making within each team at the facility continuously. This cadence requires the integration of data points from siloed teams and systems (e.g., live triage status, equipment cost, current schedule status). By combining all of the relevant data sources into the Foundry Ontology, data scientists can craft models that reflect the complexity of the engineers' daily environment – and easily incorporate new data sources as they become available or salient.

Through version datasets and the Foundry Ontology (complete with security, lineage, and data health), the data science team now has a data management layer (i.e., feature store) for machine learning that allows discovery of features and efficient creation of machine learning pipelines.

Model Integration

As a next step, the Titan data scientists utilize Foundry's out-of-the-box integrations to quickly connect to AWS SageMaker and other ML-managed services. The Foundry Ontology provides a rich data integration layer for these model building services during model creation, allowing for a streamlined development lifecycle. Once models have been created and packaged (e.g., as live endpoints or as containers), Foundry provides a range of operational AI and ML capabilities intended to complement model building:

- Full versioning, branching, reproducibility, security, and lineage for integrated models
- Multiple applications for cross-functional collaboration with data and operations teams
- Foundry Model Management framework, which provides a gateway for context, relevant data sources, metadata, upfront and ongoing evaluation of model candidates, and connection to the Ontology
- Collective awareness of production environments, individual model health, and granular feedback-driven metrics – enabling rich and structured iteration

The extensive support for versioning and branching of data and models enabled their data scientists to quickly pivot from a data-centric (model fixed with iterations focused on improving data) to model-centric (data fixed with iterations focused on enhancing the model) throughout development. The ability to pivot in both dimensions quickly accelerates analytic improvements while maintaining engineering rigor throughout experimentation.

Additionally, the Titan team was able to build new analytics to complement models integrated from third-party sources through Foundry's end-to-end model development capabilities.

Foundry also offers a suite of interoperable connectors that let you bring your models into Foundry. The integration can be via native platform integrations (AWS Sagemaker ↗, Azure ML, DataRobot ↗, Databricks, etc.) or by importing your model artifact directly into Foundry (as code, libraries, or trained models). Support for all major open-source frameworks such as PyTorch, TensorFlow, and SKLearn and interoperability standards (ONNX) make this seamless.

Regardless of your model creation story, Foundry allows you to expose consistent semantics and standardized inference methods, making them easily consumable by other users, pipelines, and operational applications.

02

Bind models to the Ontology and deploy models to frontline users

Bind Models to the Ontology

With the Foundry Ontology now providing rich data to model builders and model iteration across various model-building tools, Titan's data scientists are ready to operationalize their models with frontline users. This is known as "Ontology binding" within Foundry. which links a given model — whether developed within Foundry or externally — to specific elements within the Ontology.

In this case, the preventative maintenance model is linked with various attributes on the Plant, Factory, Inventory, Supplier, Part, and Customer objects. Whenever the model is called from an operational application, it will feed inputs from these Ontology attributes. Whenever the model returns a set of values, they will be mapped accordingly back onto specified attributes within the Ontology. This creates a type system for models in the same way that type systems are instrumental to data systems and programming languages.

With the relevant models now bound to the Foundry Ontology, all teams within Titan Logistics — whether analytical, data-oriented, or business-oriented — can leverage models in a consistent, governed fashion. It's no longer the burden of the application builder to figure out how to stitch the model into user-facing workflows. Models can be invoked through their Ontology bindings; across hundreds of applications in a consistent fashion.

Use Ontology to Build Workflows and Operational Applications

Building on the Ontology means Titan Logistics has access to a rich set of building blocks for quickly assembling workflows and read-write operational applications (versus simple read-only dashboards). These include the semantic primitives (Objects, Relations) and kinetic primitives (Actions, Functions, Models) designed to be woven together for frontline users. In Titan's case, their IT teams developed a dynamic, real-time application within weeks – leveraging one of Foundry's no/low-code application building frameworks – and rolled it to maintenance engineers within days. This application baked in the data integrations set up in the first phase, along with models from the data science team - leveraging the fact that those models had already been "bound" to the Ontology – and therefore required no manual configuration to inject into application contexts.

Maintenance engineers (and other end users) can now access the full range of enterprise-governed data and models via intuitive applications, which require no data science or technical background. Critically, they can test different maintenance strategies in near real-time, e.g., including different usage assumptions for essential parts' failure risk (time on flight or rail, temperature, pressure) and explore full-fidelity "what-if" analyses dynamically through out-of-the-box Foundry capabilities – such as Scenarios and Vertex.

A **Scenario** enables the creation and comparison of "what-if" analyses. They're "forks" or "branches" of the data in your Ontology generated by applying one or more Actions and evaluating a set of models.

Vertex is a system modeling and simulation platform that provides a visual interface to investigate interactions between entities, processes, and the resulting flow of resources across the digital twin of your organization.

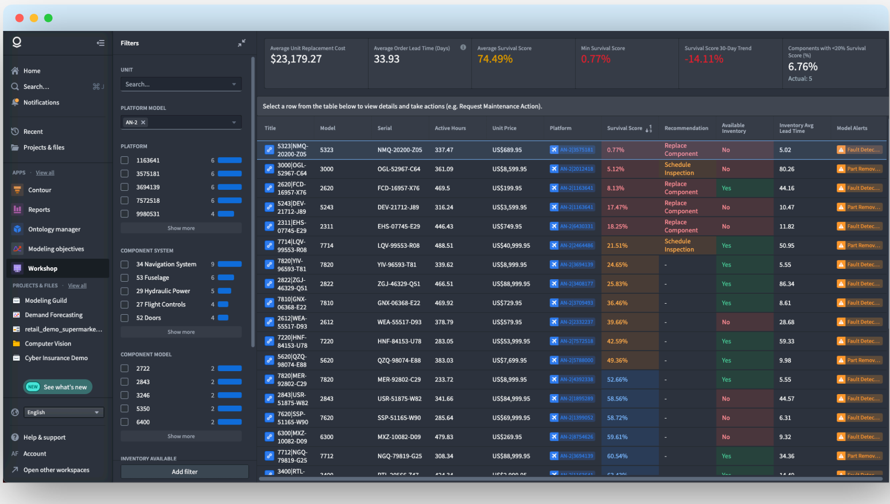


Fig. Example application interface that can be built with no code tooling on top of models

The maintenance engineers can combine subject matter expertise with contextual data (from the Ontology), model automation, and processing speed. This fusion of human and machine empowers them to make more informed trade-offs between preventative part change and failure risk by having all information pertaining to data and models within a single interface.

Titan's maintenance engineers also defined and implemented automatic alerting from live data to monitor failures across thousands of sensors. They used point-and-click tools to define business logic on top of data feeds — generating alerts when the risk became critical on specific parts. Foundry's tooling allows for patterns of interest to be visually identified, analyzed with low-code AutoML capabilities, and then exported to Jupyter-style notebooks for rich collaboration between the operational and data science teams.

03

Close the loop with operators and write back decisions to source systems with complete lineage and auditability → enable continuous learning

Capture Decisions

Titan's engineers are now wielding the Ontology through their intuitive applications to identify the optimal schedules for various types of maintenance. Closing the loop would not be possible without a shared foundation, that surfaced the most relevant, up-to-date, and contextualized information. Critically, as engineers make decisions, they are captured back in the Ontology — in accordance with governance paradigms — and with a full lineage trail that encompasses both data and model inputs.

For example, the team was alerted that a high-priority part's failure risk had just increased, based on recent sensor data and usage patterns, in conjunction with its age. They were able to pull the schedule for the plane quickly, and run different scheduling what-if analysis to confirm a recommended replacement for the plane while it undergoes maintenance — minimizing the effect of the change on Titan's network. The maintenance shop schedule was also amended, swapping out a lower-priority maintenance task to reduce the risk of downtime and delay.

This decision was captured back in the Ontology, and automatically pushed to their maintenance system of record, cueing the appropriate staff, equipment, and part transfer orders.

This extensible orchestration paradigm enables Foundry to serve as connective tissue, linking historically siloed, disconnected systems to power intelligent, more informed operations.

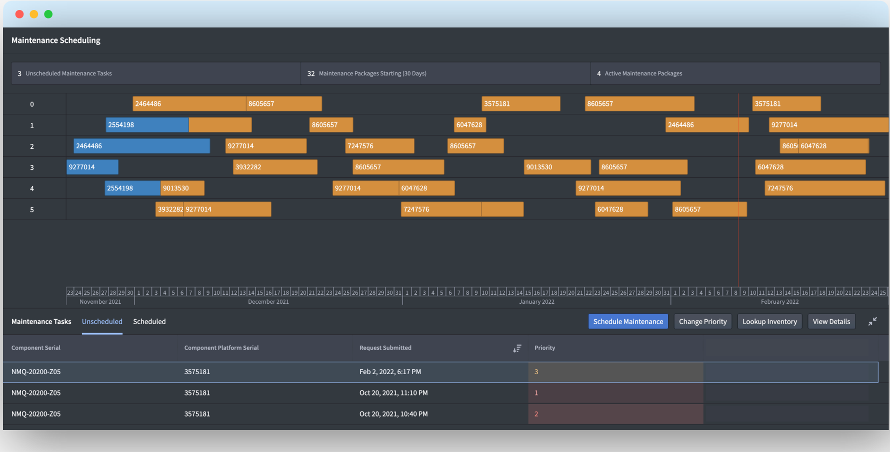


Fig. Demo Maintenance Scheduling application interface that can be built with no code tooling.

Powered behind the scenes by tightly-integrated anomaly and fault detection, forecasts, and optimization models.

Automatically Write-back to Source Systems

With Foundry's extensive suite of bidirectional connectors, all decisions made by maintenance engineers are recorded and written back to both the Ontology and to source systems.

A complete log of decisions and states is maintained in Foundry to ensure auditability and complete transparency. Decisions are always non-destructive since underlying data is brought in through system integrations and versioned.

As usage expands, Foundry acts as the connective tissue between disparate data and modeling tools — enabling seamless, collaborative decision-making across Titan Logistics. Decisions taken by the maintenance engineers can now be leveraged across various teams as novel data, fueling workflows in ways previously unimagined.

Use Model Feedback to Train and Improve Models

The gaps that prevented Titan Logistics from effectively leveraging models have been connected through Foundry. High-quality model outputs (e.g., recommendation, prediction, etc.) are available to engineers and other end-users to improve daily decision-making and outcomes, including monitoring risk on each part based on continuously-tuned survival/risk models. High-dimensionality scenarios (accelerated usage, higher temperature, etc.) can be run across single schedules or complex workflows.

As end user actions are taken in operational contexts, the decisions are captured back into the Ontology, and made seamlessly available for model monitoring, as well as to labeling and training environments. This feedback loop ensures that data scientists are able to quickly evolve models to meet ever-changing real-world conditions. Enterprise AI/ML efforts are now powering closed-loop operations across the enterprise at Titan Logistics.

Fig. **Foundry Model Management:** Enabling ongoing evaluation and monitoring of deployed and candidate models, as well as multi-stakeholder approval and release processes for deploying improvements into operations.

Getting in Touch

We’d love to discuss how Foundry can help your organization get critical AI/ML efforts connected into operations. Please reach out to business@palantir.com.