ModelOps Risk Industrialization RFP Requirements

This document is an example RFP for addressing ModelOps Risk Industrialization functional requirements. It is the result of interviews with several industry experts and analysts, as well as real-world customer experience.

Category	Requirement	Description
Model Inventory suggested weighting: 30%		
	Production Model Inventory	Provide a centralized store for viewing, managing, and maintaining all models (post-development) across the enterprise, regardless of the model type, framework, platform, environment. This should also support Vendor models.
	Model Information	A comprehensive set of information for models in-use, in-development or retired; details about model purpose, ownership, use, data source, methodology, ML/AI techniques and justification, input/output, assumptions, risk tier, critical dates, status, upstream/downstream models, validation/monitoring outcomes, and other descriptions.
	Model ID & Request	Ability to assign a unique Model ID and track all associated model information against the unique Model ID, inclusive of versions of the model.
	Model Related Changes and Policy Exceptions	Track model related changes, approvals, documentation and updates. Additionally, track associated data changes, model specification, usage and implementation changes. Finally, track policy exceptions.
	Model Status Tracking	Ability to systematically obtain the current stage of the model life cycle that a model is in, and the current status — including tracking model development, implementation, and production activities and status. Additionally, ability to track model use and status in production across systems. Finally, ability to track model decommissioning.
	Model Documentation Management Integration	Ability to persist model-specific documentation and/or link to existing content management systems that contain model documentation that covers development, validation, review, monitoring, changes, etc. Documentation should be snapshotted with all other model assets such that there is an audit trail to the specific version of the documentation for that version of the model.
	Model Asset Management	Collect and manage references to all model "artifacts" as part of the Model Information record for each model in the inventory. These artifacts could be coefficients, weights files, binary objects, etc. References to these model artifacts should be tracked for each version of the model.
	Dependency Management Integration	Manage the list of frameworks, libraries, and other dependencies that are required to execute a given model, such that the Model Risk team can rerun the model, as required. While the actual libraries are typically managed in an existing repository (e.g. Artifactory), the ModelOps capability needs to list the exact libraries/frameworks — and the version of the library — for each version of the model.
	Source Code Management Integration	Integrate with the enterprise's standard for source code management, such that the Model Risk team can rerun the model, as required. Associate details of the model's git-based assets with the Model Information for each model in the inventory.
	Test Results Management	Persist all instances of tests/metrics that have been executed on a per-model basis, regardless of the type of model or the platform upon which it executes. A "record" of a test/metrics job should be tied to an immutable snapshot of a model for history and auditability.
	Benchmark Model Management	Ability to register and manage Benchmark models in a central inventory, such that the Model Risk team is able to systematically or manually compare a proposed business model against a known "benchmark" model.
	Model Owner Attestation	Ability to capture model attestation from model owners for model inventory certification.
	Aggregated Model Risk	Support the implementation of the model risk aggregation framework.

Model Life Cycle
Automation & Management
suggested weighting: 40%

Model Life Cycle (MLC) Orchestration	Ability to define and orchestrate the entire life cycle of a model — from model request, to development, validation, implementation, production usage, to retirement — using a low-code workflow creation, execution, and management system.
Model Validation Submission Verification	For any model validation request, ability to automatically check that all required information and assets have been included in the model submission. If any items are missing, ability to reject the submission and automatically send a notification to the model owner to submit with the missing assets.
Playbook identification	Ability to systematically determine the appropriate "validation playbook" to run based on a configurable set of criteria about the model (e.g. model methodology, risk classification, etc.). The playbook entails running an automated workflow specifically for that criteria.
Orchestration: Automated Testing	Ability to automatically run a series of tests based on the identified playbook. The tests should leverage the managed data/other assets to automatically run the appropriate tests, track the test execution, persist the model test results, and provide visualizations for the results.
Orchestration:Automated Thres	Ability to automatically compare tests results against defined thresholds (which may be based on the "validation playbook") to determine potential issues/failures which may lead to findings. Any breaches in thresholds should be identified and lead into the appropriate notifications.
Orchestration: Automated Documentation Generation	Ability to automatically populate model information and test results into a validation report and/or other documentation, using a pre-existing template. The documentation should be persisted with the Model Information in the model inventory.
Orchestration: Automated Production Monitoring	Ability to automatically monitor models in production against the defined tests and comparing against thresholds. As required, automatically populate regular report documents and/or annual model review documentation based on the test results and comparison against thresholds. If any breaches occur, automatically raise the appropriate notifications/alerts.
Orchestration: Model identification and risk ranking	Workflow orchestration to drive the model identification and risk rating steps, which may include identifying models and data sources; conduct model/non-model assessment; risk-rank models based on materiality, complexity and impact; analyze model dependency.
Orchestration: Validation & Periodic Review	Workflow orchestrations to assist in Validation planning, scheduling, and task assignment, including approvals and escalations. This may involve integrating into existing systems, if they exist.
Orchestrate Model Execution	Ability to integrate into existing model execution systems to allow Model Risk teams to rerun model code and/or test the proposed implementation for the model. This should include out-of-the-box integrations with primary model execution systems.

Visualization, Reporting, and Issue Management suggested weighting: 20%		
	Risk Reporting Dashboard	Provide a user friendly view of various risk and compliance measures for all models across the enterprise. These measures include a view of (1) all "production" models used for business decisioning based on risk classification (2) where models are in their risk / compliance review cycle (3) any current risk / compliance issues occurring in production (4) upcoming risk / compliance reviews for all models across the enterprise.
	Findings/issues management	Issue findings and track action plan and remediation. Create findings reports to share with stakeholders.
	Alerts	Provide ability to automatically raise alerts and notifications based on model-execution errors, issues from model risk/other monitoring, or from specific actions/steps in the model life cycle orchestrations. Typically these alerts are handled by orchestrated remediation paths that are defined in a model life cycle (MLC), and would be tied to thresholds (see above) to allow for clear, but comprehensive, orchestration of when alerts are triggered. The ModelOps capability should be flexible to integrate with existing operational alerting and/or risk systems.
	BI tools integrations	Ability to integrate with BI tools to create custom reports to gain visibility of aggregated and detailed views on model performance against statistical, business, and risk thresholds.
Integrations, Architecture, & Security suggested weighting: 10%		
	API's	APIs that enable institutions to connect the system to existing IT, Data Science, Risk, and Security infrastructures and platforms, including integration with model development tools and production model execution platforms.
	Authentication/Authorization	Integration with the enterprise's oauth2 and LDAP/AD systems for authentication, leveraging existing oauth/ldap systems, processes, and AD/LDAP group structure to for authentication.
	Multi-Tenancy (Group-based isol	Group-based access control to isolate which groups/teams can see specific models, model assets, runtimes with specific deployed models, test results, integrating with the enterprise's oauth2 and LDAP/AD systems to obtain the associated group entity information.
	Product Architecture	ModelOps capability should be designed to be flexible and extensible, using a modern micro services architecture, to allow enterprises to customize and integrate the ModelOps services into existing systems, platforms, and processes.
	Infrastructure	Ability to run Software anywhere: on Cloud, on Prem, Hybrid.