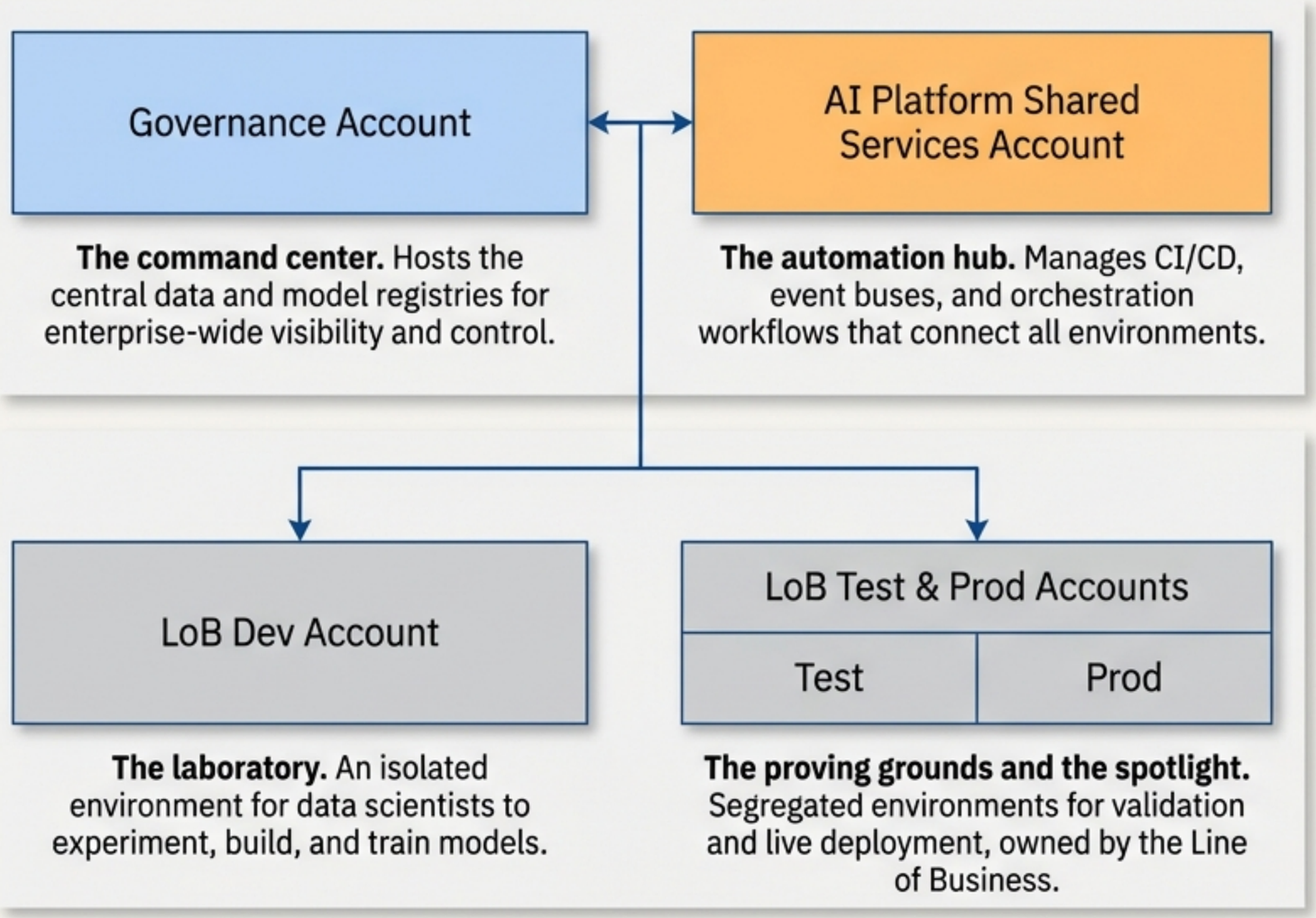
The Governed Path to Production

An AIOps Framework for Collaborative and Unified AI Development on AWS

A reference architecture for enabling systematic model progression across Dev, Test, and Prod environments through automated, event-driven pipelines.

The Foundation: Accounts, Personas, and Centralized Services

The Multi-Account Strategy



The command center. Hosts the central data and model registries for enterprise-wide visibility and control.

The automation hub. Manages CI/CD, event buses, and orchestration workflows that connect all environments.

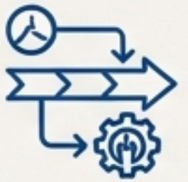
The laboratory. An isolated environment for data scientists to experiment, build, and train models.

The proving grounds and the spotlight. Segregated environments for validation and live deployment, owned by the Line of Business.

The Core Personas

Admin: Configures the SageMaker Unified Studio environment, domains, and project templates.

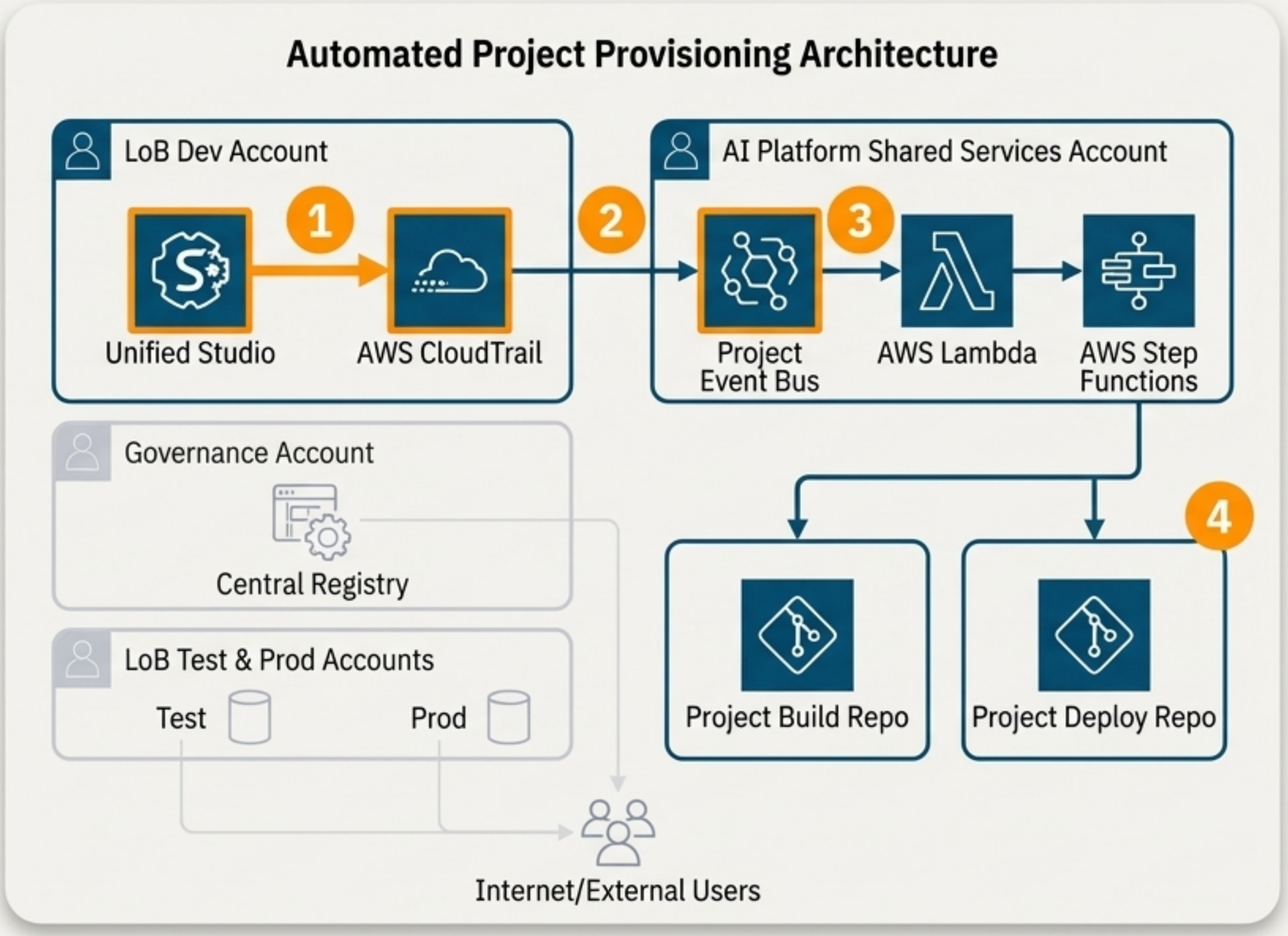
Data Scientist: Develops, trains, and evaluates models within the Dev environment. Initiates the first promotion approval.



AI Engineer: Defines deployment configurations, validates models, and promotes them from Test to Production.

Governance Engineer: Reviews compliance and performance results in the Central Registry, providing the final sign-off for Production.

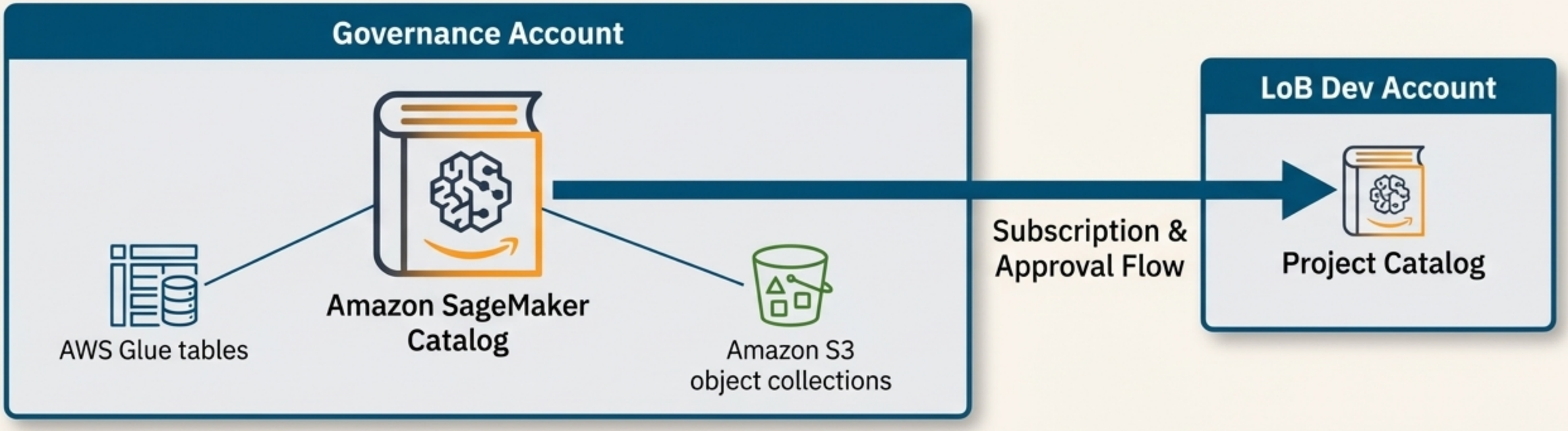
Phase 1: Automated Project Provisioning from a Central Blueprint



Workflow Steps

1. **Initiation (Admin/Data Scientist):** A Data Scientist creates a new project in Amazon SageMaker Unified Studio using pre-configured templates (e.g., Classical Regression, LLM fine-tuning, RAG) established by the Admin.
2. **Event Capture:** This action publishes a `CreateProject` event, captured by AWS CloudTrail and forwarded to a custom Amazon EventBridge bus in the Shared Services account.
3. **Automated Orchestration:** The event triggers an AWS Step Functions workflow.
4. **Repository Provisioning:** The workflow provisions dedicated Git repositories (e.g., `build`, `deploy`) for the new project, populating them with standardized pipeline code from the Use Case Templates. This ensures every project adheres to organizational best practices from day one.

Governed Data Access via the Enterprise SageMaker Catalog



The Amazon SageMaker Catalog in the Governance Account serves as a centralized data registry, enabling secure discovery and access to approved data assets across all accounts.

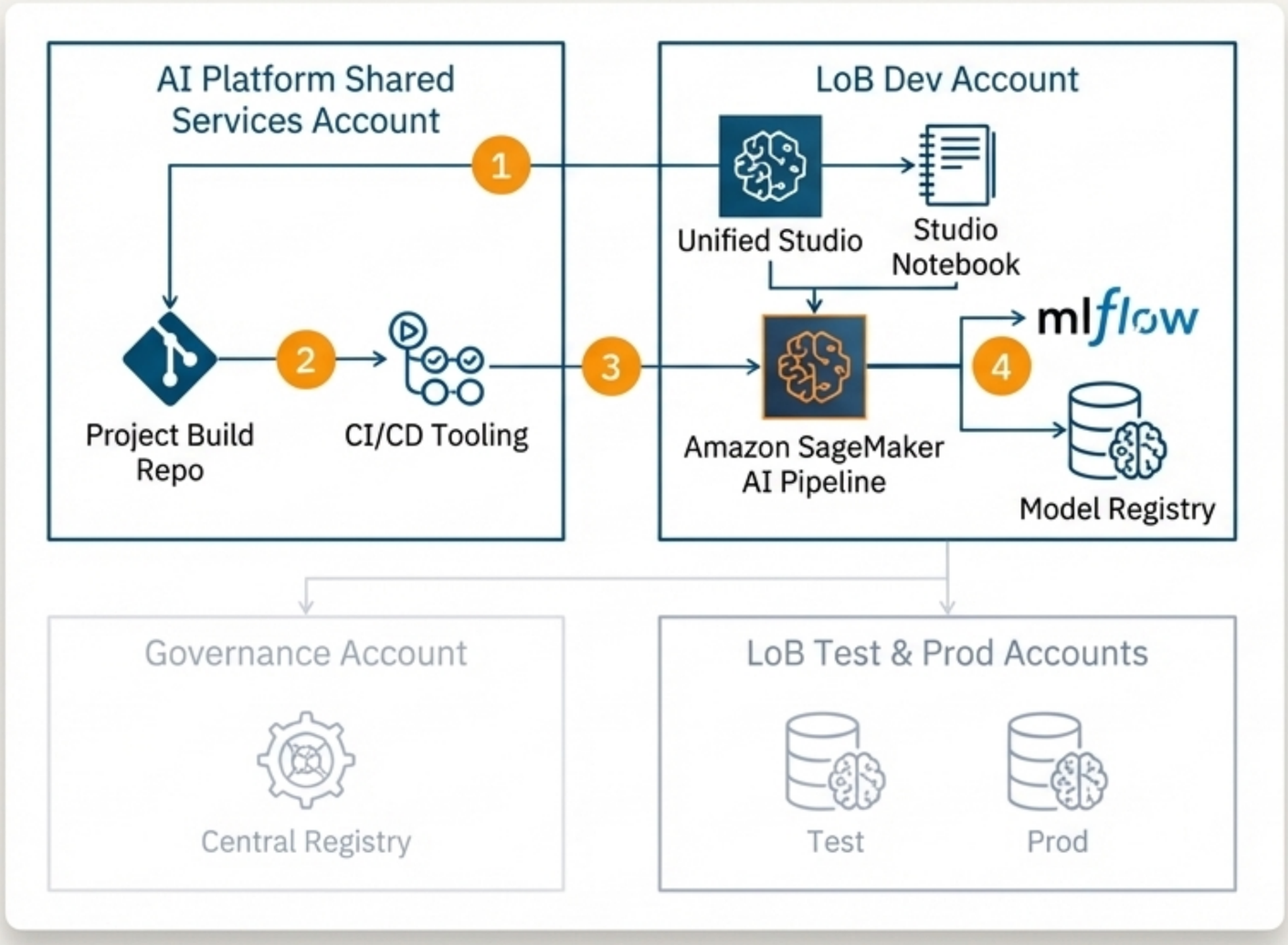
Key Features:

✅ ****Centralized Management****: Provides enterprise-wide management of data and model assets.

✅ ****Secure Subscription Flow****: Data Scientists subscribe to required datasets (e.g., AWS Glue tables, S3 object collections). Access is granted only after approval from the data producer.

✅ ****Fine-Grained Controls****: Implements access controls to ensure data scientists can only access authorized datasets for their specific project.

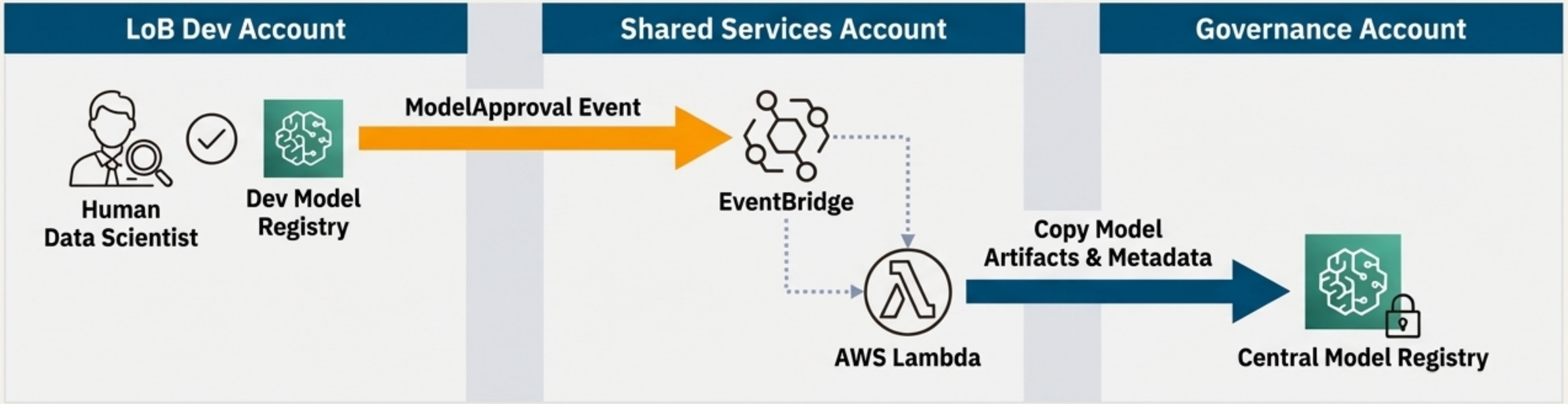
Phase 2: Building and Registering the Model in the Dev Environment



Workflow Steps

1. **Customize & Commit**: The Data Scientist customizes the build pipeline code in the Git repository (e.g., adding MLflow experiment tracking) and commits the changes.
2. **Trigger CI/CD**: The commit automatically triggers the CI/CD pipeline (e.g., GitHub Actions) based on path filter rules in the repo's `build` folder.
3. **Execute & Track**: The CI/CD pipeline runs the Amazon SageMaker AI pipeline in the Dev account to build, train, and evaluate the model. All metrics, parameters, and artifacts are tracked in MLflow.
4. **Register Locally**: Upon successful evaluation, the model is automatically registered in the **Amazon SageMaker Model Registry** within the **Project A DEV** account.

The First Governed Handoff: Promoting to the Central Registry



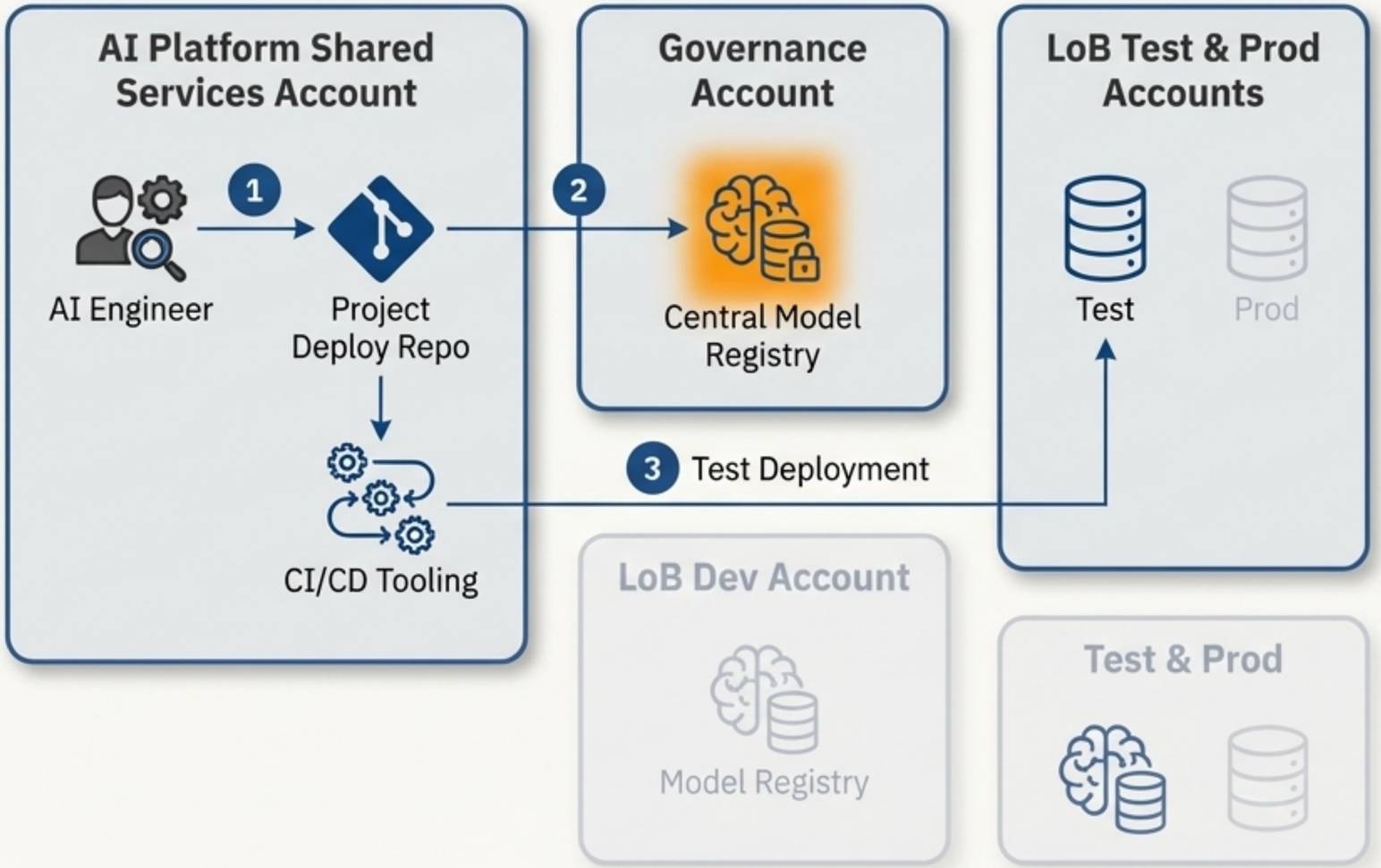
The Trigger Event

- **WHAT:** The Data Scientist approves the model in the Dev Model Registry (sets `stage = Dev`, `status = Approved`).
- **WHO:** Data Scientist.
- **WHY (The Insight):** This manual approval is a deliberate business decision that serves as a powerful trigger. It emits an event to the Shared Services EventBridge bus, initiating the promotion process.

The Automated Action

- An AWS Lambda function in the Shared Services account consumes the approval event.
- The function copies the model artifacts and metadata from the Dev registry into the **Central Model Registry** (in the Governance account).
- **Outcome:** The model is now visible to the wider organization, its lineage is preserved, and it is ready for the next stage of validation.

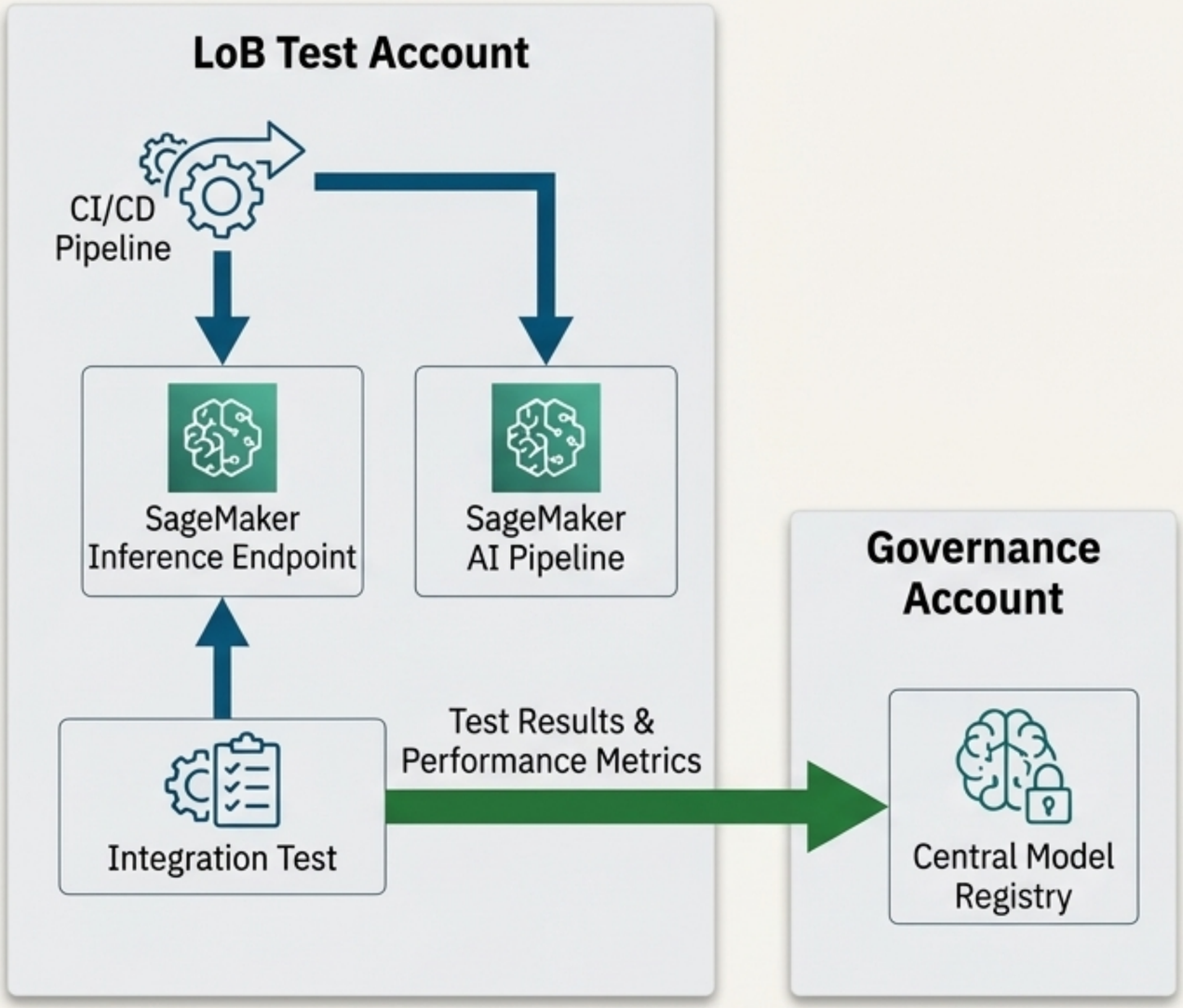
Phase 3: Validating and Triggering Deployment to the Test Environment



Workflow Steps

1. **Define Deployment (AI Engineer):** The AI Engineer validates the model and defines its deployment parameters (e.g., instance type, scaling configuration) by updating the project's `deploy` repo.
2. **Approve for Test:** The AI Engineer promotes the model in the **Central Model Registry** by setting its status to `stage = Test`, `status = Approved`.
3. **Trigger Test Deployment:** This approval acts as the second major event trigger. It is captured by EventBridge and initiates the CI/CD pipeline specifically configured for the Test environment.

The Gauntlet: Automated Integration Testing and Results Feedback



Deployment to Test

- The CI/CD pipeline deploys the model endpoint into the Project A **TEST** account.
- Optionally, the associated Amazon SageMaker AI pipeline can also be deployed for more complex validation scenarios.

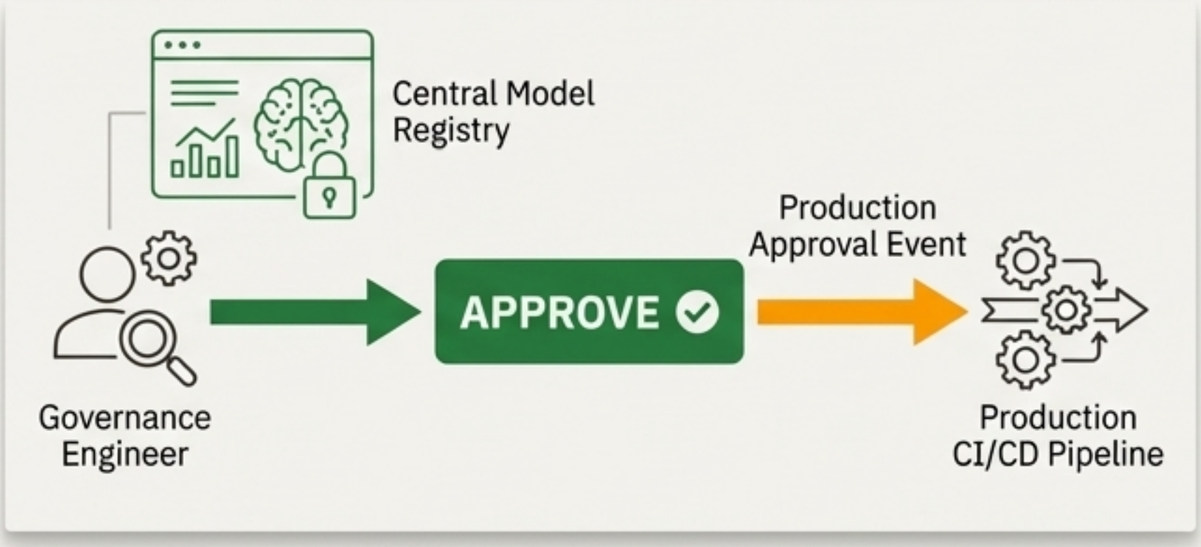
Automated Validation

- A suite of comprehensive integration tests is automatically performed against the new endpoint.

Closing the Loop

- The results of these tests (performance metrics, pass/fail status) are programmatically recorded back into the model's metadata in the **Central Model Registry**. This provides the Governance Engineer with all the necessary data to make an informed decision for production promotion.

Phase 4: Final Governance Review and Approval for Production



The Final Decision

WHO: Governance Engineer.

WHAT: Reviews the comprehensive integration test results and model metrics logged in the Central Registry.

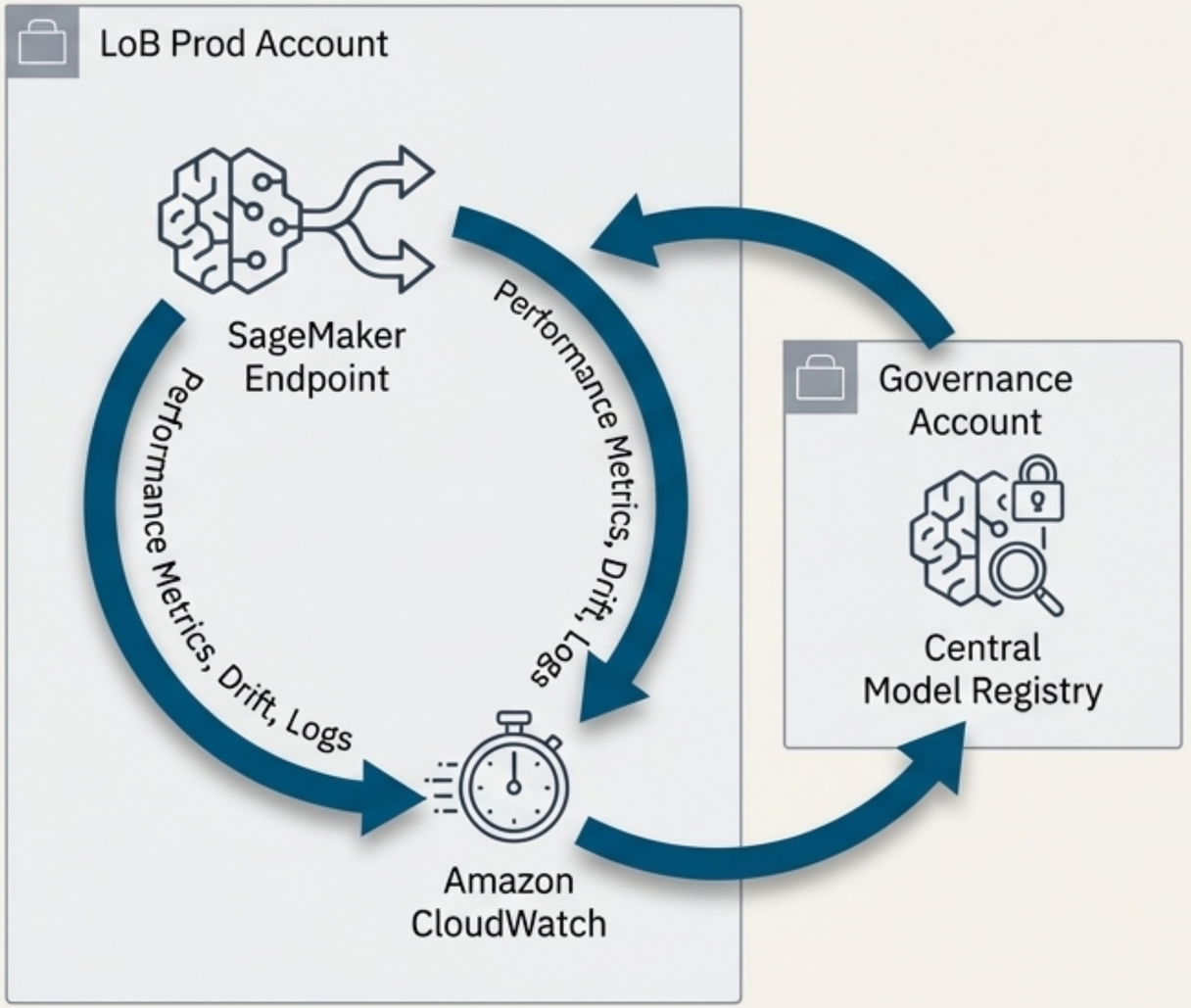
WHY: Upon meeting all compliance and performance requirements, the model is approved for production deployment (`stage = Prod`, `status = Approved`).

Triggering Production Deployment

This final approval emits the most critical event of the lifecycle.

The event triggers the production CI/CD pipeline, which builds the required artifacts and begins the deployment process into the Project A **PROD** account.

The Spotlight: Production Deployment with Continuous Monitoring



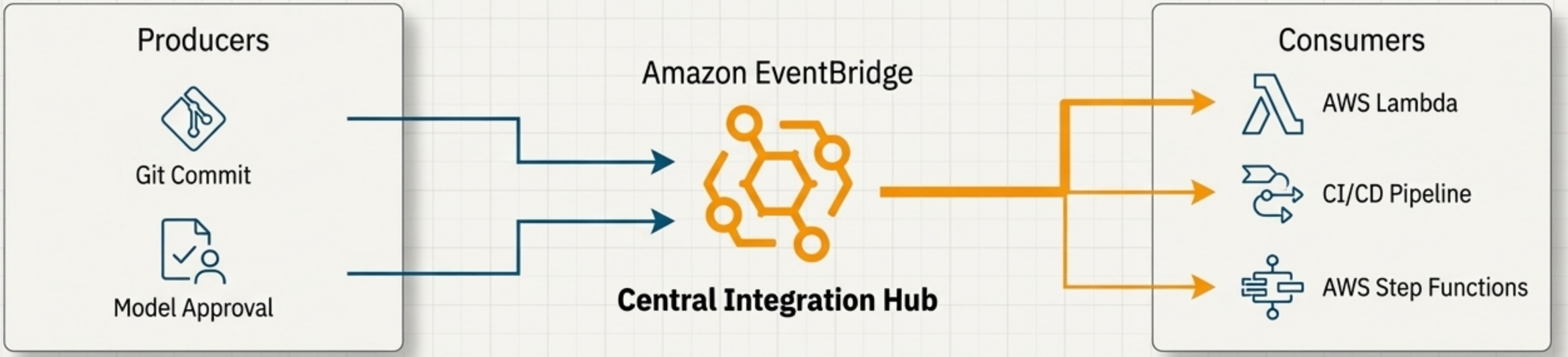
Live in Production

- The production CI/CD pipeline deploys the model endpoint into the PROD environment.
- The architecture supports advanced deployment strategies like A/B testing via Amazon SageMaker endpoints.

Continuous Monitoring & Governance

- Model performance (e.g., latency, error rates, data drift) is continuously monitored using Amazon SageMaker's monitoring capabilities.
- Results and logs are streamed to Amazon CloudWatch and persistently logged against the model version in the Central Model Registry.
- This provides end-to-end visibility and enables a continuous governance cycle, completing the governed path from development to operations.

The System’s Nervous System: A Loosely Coupled, Event-Driven Architecture



Amazon EventBridge serves as the central integration hub for all automation workflows across accounts.

How it Works
Producers: Actions like a `CreateProject` call or a `ModelApproval` status change emit events.
Routing: Resource-based policies on EventBridge buses in each account securely forward specific events to the central bus in the Shared Services account.
Consumers: Services like AWS Lambda or CI/CD tools subscribe to these events to trigger their respective workflows automatically.

Key Benefits
This approach creates a scalable and resilient system where components are loosely coupled, allowing teams to evolve their parts of the workflow independently without breaking the entire chain.

Security and Governance Controls are Integrated at Every Step

1. Secure CI/CD Authentication

Mechanism: OIDC JWT (OpenID Connect JSON Web Token).
Function: Provides secure, short-lived credentials for GitHub Actions workflows to access AWS resources without storing long-lived secrets like access keys.

2. Cross-Account Access Control

Mechanism: AWS IAM (Identity and Access Management) Roles.
Function: Enforces least-privilege permissions for all cross-account interactions, ensuring CI/CD pipelines and services only have access to the specific resources they need.

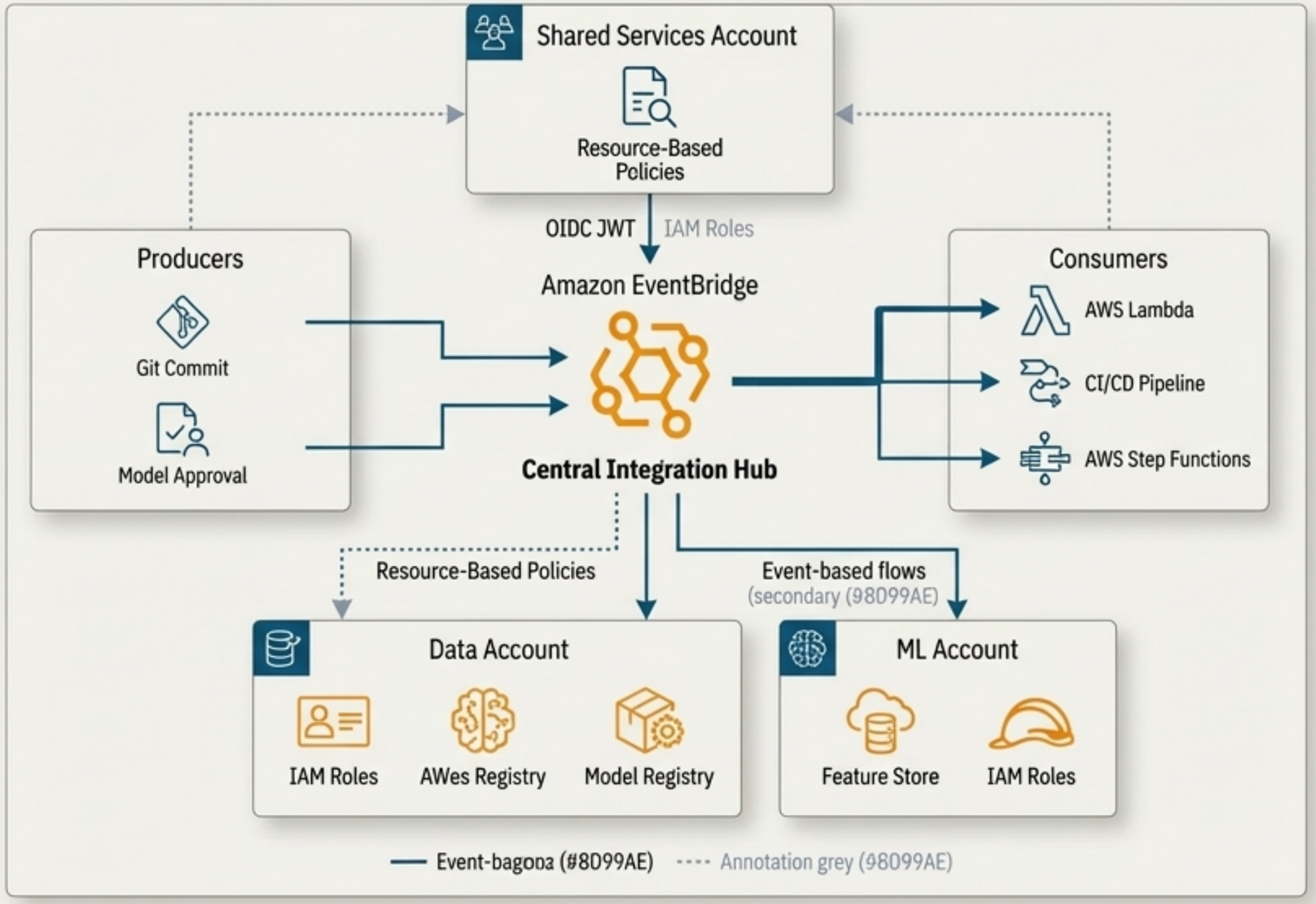
3. Governed Data Access

Mechanism: Amazon SageMaker Catalog.
Function: Implements fine-grained access controls to ensure data scientists can only subscribe to and use datasets they are explicitly authorized to access.

4. Secure Event Forwarding

Mechanism: Amazon EventBridge Resource-Based Policies.
Function: Securely manages how events are shared across organizational boundaries, ensuring only designated events are forwarded to the central bus.

The Result: A Unified Framework for Velocity with Governance



Key Outcomes

Accelerated Development: Standardized templates and automated pipelines reduce friction and allow teams to focus on building models, not infrastructure.

End-to-End Governance: A central source of truth for data and models, with automated gates and explicit approvals, ensures compliance and reduces risk.

Enhanced Collaboration: A clear, event-driven workflow defines the handoffs between Data Scientists, AI Engineers, and Governance teams.

Scalable and Reusable: The modular, loosely-coupled design allows the framework to scale across numerous projects and lines of business while promoting the reuse of components and patterns.