**Use Case & API Lifecycle Management**

**[Document Version: 1.0 | Date: March 6, 2025]**

*This document outlines the comprehensive implementation plan for our GenAI platform's lifecycle management capabilities. The platform will enable the creation, management, and monitoring of AI use cases and APIs across different environments (development, testing, and production) for various user personas including data scientists, developers, platform managers, and administrators.*

**User Personas & Journey Mapping**

We've identified four core user personas that will interact with the platform:

**Data Scientist**

**Primary Goals**: Experiment with models, create prototypes, analyze performance **Key Capabilities**: Create use cases, test models, analyze performance metrics **Environment Access**: Full access to Explore, limited access to Dev, read-only for UAT/Prod

**Developer**

**Primary Goals**: Implement and integrate AI services, optimize performance **Key Capabilities**: Configure APIs, implement integration code, test systems **Environment Access**: Full access to Explore/Dev, support access to UAT, limited access to Prod

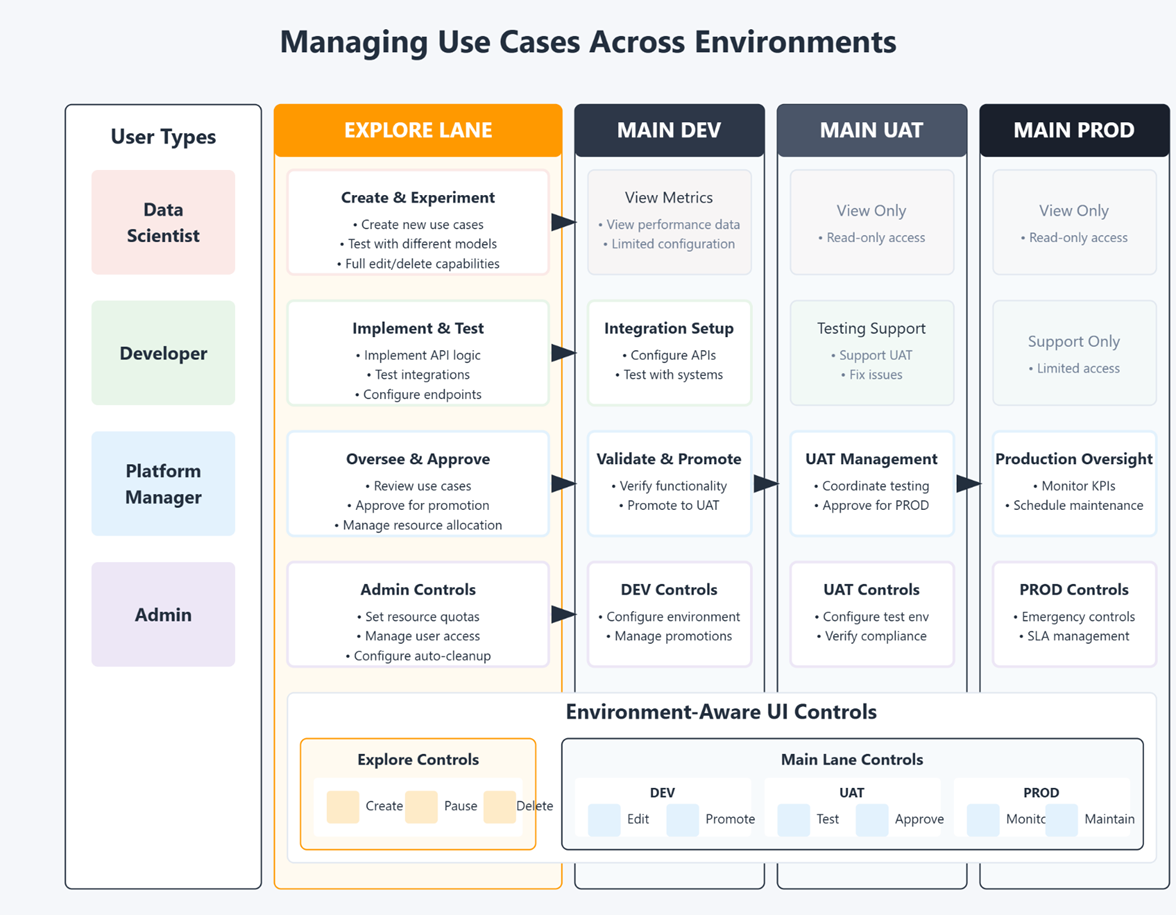
**Platform Manager**

**Primary Goals**: Oversee platform usage, approve promotions, manage resources **Key Capabilities**: Monitor usage, approve environment transitions, manage quotas **Environment Access**: Administrative access to all environments

**Super Admin**

**Primary Goals**: Manage platform configuration, users, and security **Key Capabilities**: Configure environments, manage users/teams, set global policies **Environment Access**: Full administrative access to all environments and settings

**User Journey Diagram**



**Use Case Lifecycle Management in Prototyping(BYA)**

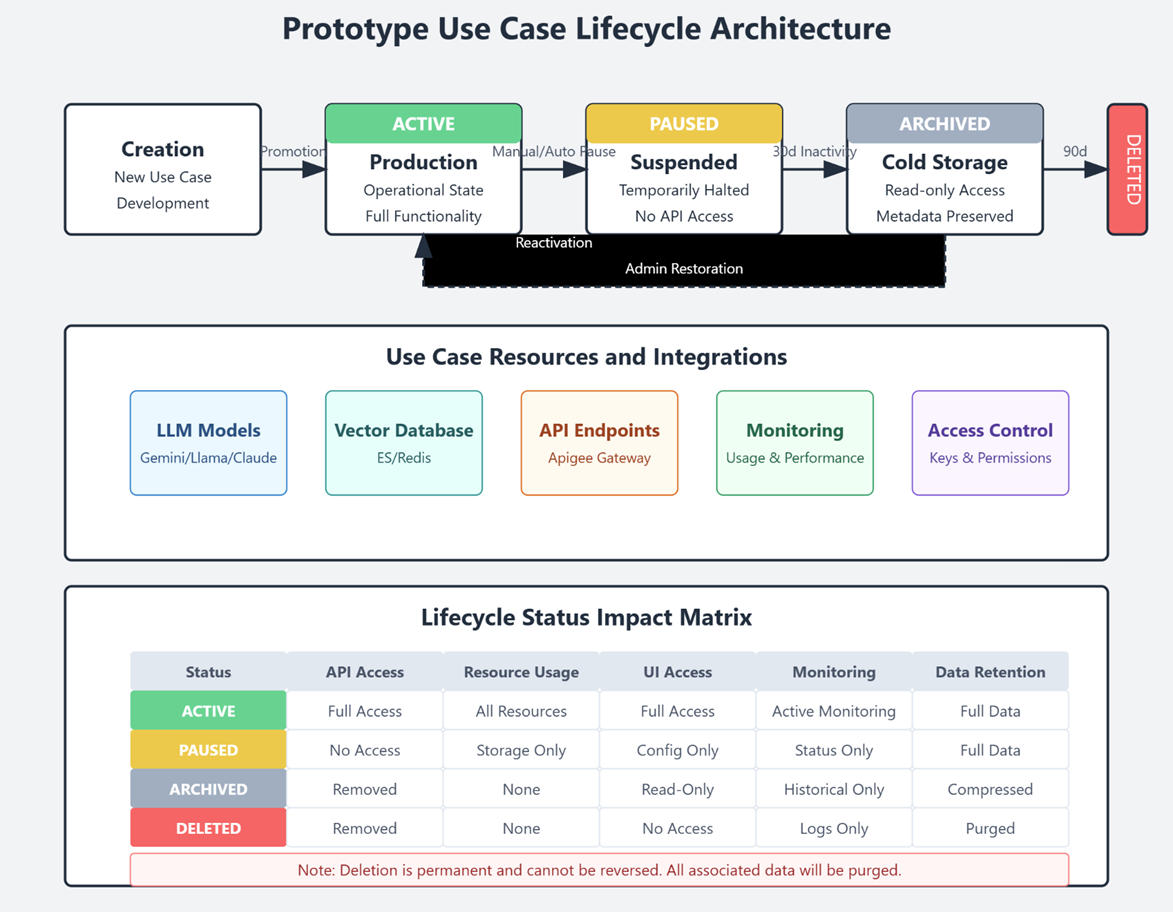
**Lifecycle States**

Use cases in the platform follow a defined lifecycle with the following states:

1. **Creation**: Initial development and configuration phase
2. **Active**: Fully operational state in a specific environment
3. **Paused**: Temporarily suspended state (no API access but configuration preserved)
4. **Archived**: Cold storage state with read-only access and minimal resource usage
5. **Deleted**: Permanently removed state (data purged according to retention policies)

**Lifecycle Transition Rules**

The platform will implement automated rules to govern state transitions:

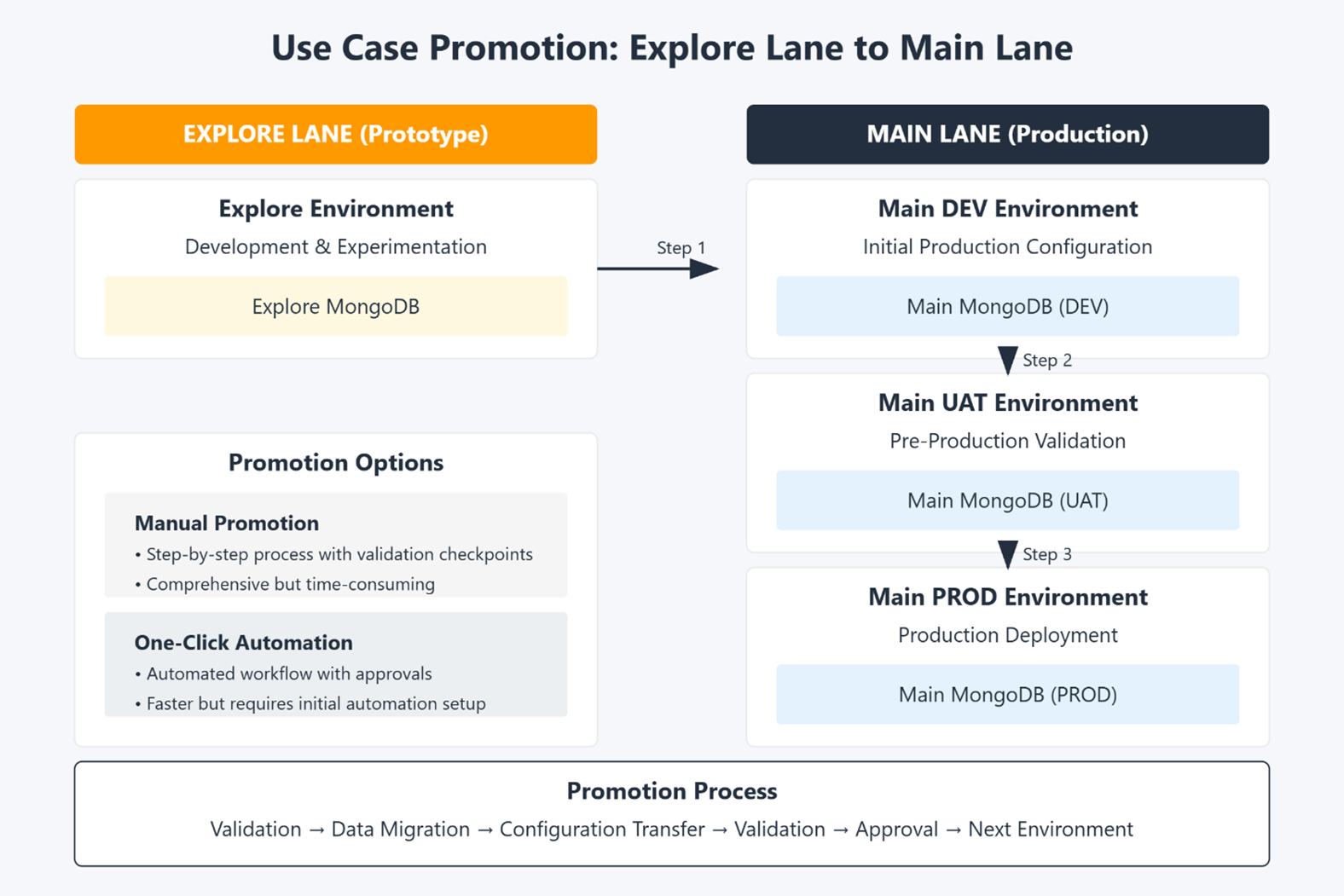


* **Auto-Pause**: Triggered after 14 days of inactivity (To be discussed on the inactivity period)
* **Auto-Archive**: Applied to use cases that remain paused for 30+ days
* **Auto-Delete**: Option for archived use cases after 90+ days (with approval)
* **Manual Controls**: Override options for platform managers and admins

**Decisioning Criteria**:



**Environment Promotion Workflow**

****

Use cases follow a structured promotion path across environments:

1. **Explore → Dev**: Initial promotion requiring basic validation
2. **Dev → UAT**: Promotion requiring thorough testing and approval
3. **UAT → Prod**: Final promotion requiring business approval and compliance validation

Each promotion includes configuration freezing, resource allocation validation, and appropriate approvals based on policy.

**Promotion Implementation in Architecture Alternatives**

**Single Codebase Approach**:

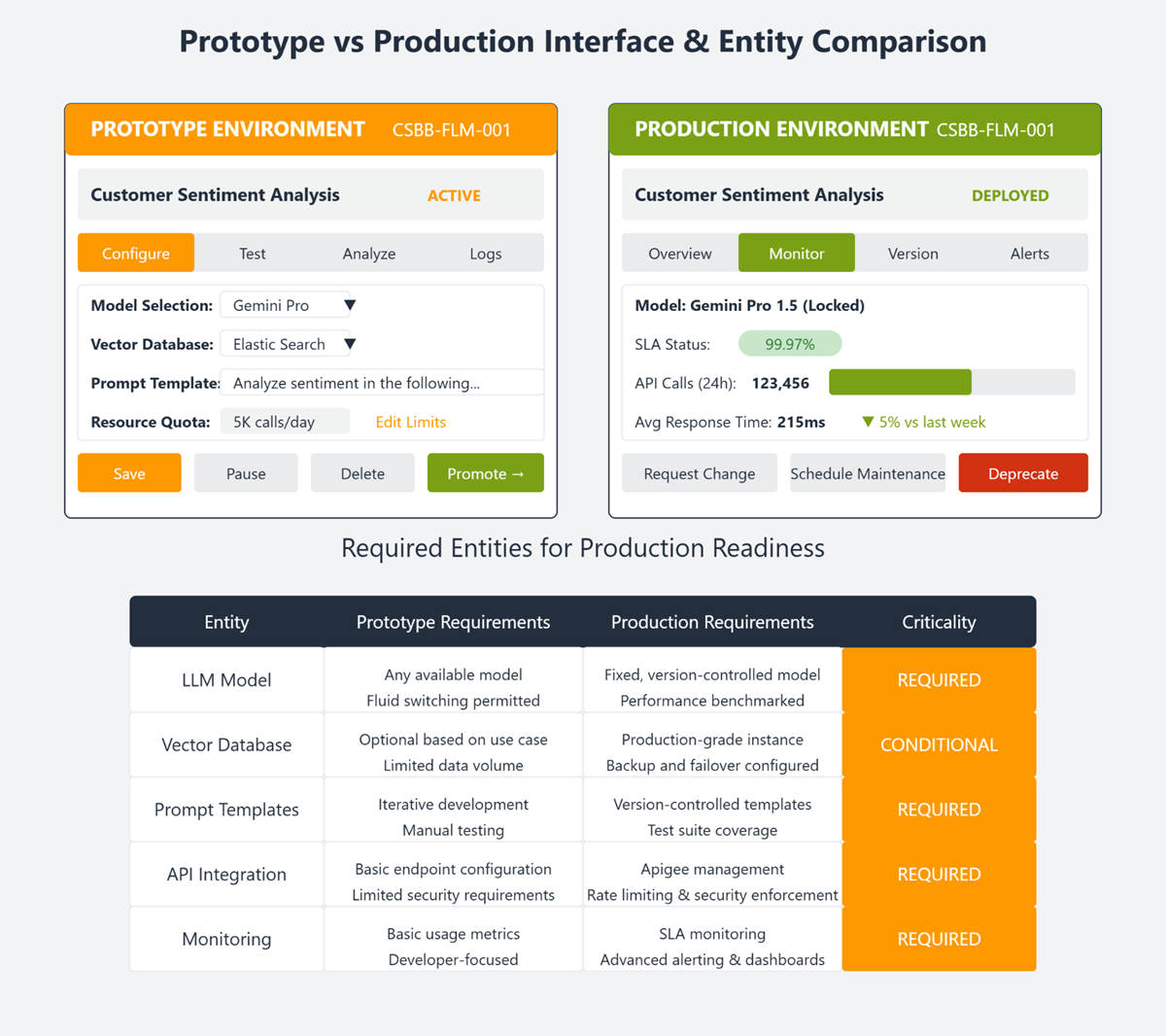
* Promotions primarily involve changing status flags and environment context
* Configuration data is copied between environment-specific data stores
* Access controls adjust automatically based on new environment

**Core Library Approach**:

* Promotions involve exporting configuration from Explore Application
* Importing and adapting configuration into Main Application
* Validating compatibility with production governance requirements

## **Production vs. Prototype Entity Comparison**

| **Entity** | **Prototype Flexibility** | **Production Requirements** |
| --- | --- | --- |
| LLM Model | Multiple models can be tested | Fixed model version with benchmarks |
| Vector DB | Optional or development instance | Production instance with backup strategy |
| Prompts | Iterative development | Version-controlled templates with tests |
| API | Basic endpoint configuration | Gateway-managed with security controls |
| Monitoring | Basic usage metrics | Full SLA monitoring and alerting |
| Access Controls | Developer-centric | Role-based with audit trails |



UI Interfaces:

API Access Management (Data science Persona):

API access for the existing usecases in Explore Lane:

Analytics: