# Technical Implementation Plan: Tellow AI Workflow Engine

**Version:** 1.1 (Updated Schema Nomenclature)

**Status:** Approved for Implementation

**Domain:** Workflow Orchestration / GenAI Infrastructure

## 1. Executive Summary

This document details the architectural specifications for the Tellow AI Workflow Engine. The system is designed to be a "World Class" node-based editor allowing users to construct complex, multi-model generative AI pipelines.

**Core Philosophy:**

1. **Strict Typing:** The graph is governed by strict "Socket Contracts" defined in the **AI Model Registry** to prevent invalid connections.
2. **Separation of Concerns:** The **Definition Layer** (Registry) is strictly separated from the **Instance Layer** (Canvas).
3. **Write-Time Compilation:** Complex graphs are "compiled" into flat manifests at save time to ensure O(1) read performance for end-users.
4. **Hybrid Input Handling:** Supports both dynamic (wired) inputs and static (hardcoded) assets transparently.

## 2. High-Level Architecture

The system operates on three distinct logical layers.

1. **The Registry (The Library):** The immutable catalog of available tools, AI models, and data types (stored in ai\_model\_\* tables).
2. **The Graph (The Canvas):** The user's specific arrangement of nodes and wires (stored in workflow\_\* tables).
3. **The Runtime (The Engine):** The execution logic that traverses the graph, validates constraints, and orchestrates API calls.

## 3. Database Schema Specification

The database is the single source of truth. We utilize **MySQL 8.0+** features (JSON Columns, Foreign Key Constraints).

### Domain A: The AI Registry (System Definitions)

These tables define *what is possible* in the system. They are generally read-only for the Workflow Builder but writable by the Admin/Dev team.

#### 1. ai\_model\_providers

**Purpose:** Stores the execution backends (e.g., OpenAI, Replicate, Fal.ai).

* auth\_config (JSON): Stores API key references or header templates.

#### 2. ai\_model\_socket\_types

**Purpose:** Defines the "Language" nodes speak. Prevents connecting an Image Output to a Text Input.

* data\_structure: ENUM('primitive', 'file', 'array', 'object').

#### 3. ai\_model\_registry

**Purpose:** The catalog of AI tools and models.

* parameter\_schema (JSON): Defines the "Settings" form (Strength, Seed, Steps) using JSON Schema standard.
* pricing\_config (JSON): Dynamic pricing rules (per second, per pixel, flat fee).

#### 4. ai\_model\_io\_definitions (The Contract)

**Purpose:** Defines the strict inputs and outputs for every model in the registry.

* **Crucial Feature:** constraints (JSON). This enforces model-specific limits (e.g., "Max Token Limit: 77").

#### 5. workflow\_system\_node\_definitions

**Purpose:** Blueprints for utility nodes (USER\_INPUT, STATIC\_ASSET) so the Frontend knows how to render them without hardcoding.

### Domain B: The Canvas (User Data)

These tables store the *instances* created by users.

#### 6. workflows

**Purpose:** The container for a project.

* input\_manifest\_summary (JSON): A cached summary of required inputs.

#### 7. workflow\_nodes

**Purpose:** An instance of a model or tool on the canvas.

* type: ENUM('AI\_MODEL', 'USER\_INPUT', 'STATIC\_ASSET', 'LOGIC\_GATE', 'OUTPUT').
* model\_id: Links to ai\_model\_registry if type is 'AI\_MODEL'.
* config\_values (JSON): Stores user-defined values (Strength, Labels, Asset URLs).

#### 8. workflow\_edges

**Purpose:** The wiring. Represents data flow from Source -> Target.

* **Constraint:** A UNIQUE constraint ensures two wires cannot feed the same input socket.

#### 9. workflow\_input\_manifests (The Optimization)

**Purpose:** A compiled, flat list of all requirements for the end-user.

* **Trigger:** Populated when the Creator clicks "Publish".
* **Benefit:** Allows the Frontend to render the end-user form in one simple query, avoiding expensive graph traversal during high traffic.

## 4. SQL Schema Definitions (Production Ready)

-- ==========================================  
-- DOMAIN A: THE AI REGISTRY  
-- ==========================================  
  
CREATE TABLE ai\_model\_providers (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(255) NOT NULL,  
 slug VARCHAR(50) NOT NULL UNIQUE, -- e.g., 'openai', 'fal-ai'  
 api\_base\_url VARCHAR(255),  
 auth\_config JSON,   
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE ai\_model\_categories (  
 id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(100) NOT NULL,  
 slug VARCHAR(100) NOT NULL UNIQUE, -- e.g., 'img-gen', 'video-gen'  
 color\_hex VARCHAR(7)  
);  
  
CREATE TABLE ai\_model\_registry (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 provider\_id BIGINT UNSIGNED NOT NULL,  
 category\_id INT UNSIGNED NOT NULL,  
   
 name VARCHAR(255) NOT NULL,  
 version VARCHAR(50) NOT NULL, -- e.g., "v1.0.2"  
 is\_active BOOLEAN DEFAULT TRUE,  
   
 -- CONFIGURATION & PRICING  
 parameter\_schema JSON NOT NULL,   
 pricing\_config JSON NOT NULL,  
   
 FOREIGN KEY (provider\_id) REFERENCES ai\_model\_providers(id),  
 FOREIGN KEY (category\_id) REFERENCES ai\_model\_categories(id),  
 UNIQUE KEY unique\_version (provider\_id, name, version)  
);  
  
CREATE TABLE ai\_model\_socket\_types (  
 id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(50) NOT NULL UNIQUE,  
 data\_structure ENUM('primitive', 'file', 'array', 'object') NOT NULL,  
 validation\_rules JSON  
);  
  
CREATE TABLE ai\_model\_io\_definitions (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 model\_id BIGINT UNSIGNED NOT NULL,  
 socket\_type\_id INT UNSIGNED NOT NULL,  
   
 direction ENUM('INPUT', 'OUTPUT') NOT NULL,  
 name VARCHAR(50) NOT NULL,  
 is\_required BOOLEAN DEFAULT TRUE,  
 is\_list BOOLEAN DEFAULT FALSE,  
   
 -- Model-specific constraints (e.g., max\_tokens: 77)  
 constraints JSON,   
   
 FOREIGN KEY (model\_id) REFERENCES ai\_model\_registry(id) ON DELETE CASCADE,  
 FOREIGN KEY (socket\_type\_id) REFERENCES ai\_model\_socket\_types(id)  
);  
  
CREATE TABLE workflow\_system\_node\_definitions (  
 id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 type\_slug VARCHAR(50) NOT NULL UNIQUE, -- 'USER\_INPUT', 'STATIC\_ASSET', 'IF\_ELSE'  
 name VARCHAR(100) NOT NULL,  
 config\_schema JSON NOT NULL  
);  
  
-- ==========================================  
-- DOMAIN B: THE WORKFLOW CANVAS  
-- ==========================================  
  
CREATE TABLE workflows (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 user\_id BIGINT UNSIGNED NOT NULL,  
 name VARCHAR(255) NOT NULL,  
 input\_manifest\_summary JSON,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE workflow\_nodes (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 workflow\_id BIGINT UNSIGNED NOT NULL,  
   
 type ENUM('AI\_MODEL', 'USER\_INPUT', 'STATIC\_ASSET', 'LOGIC\_GATE', 'OUTPUT') NOT NULL,  
   
 -- Link to the Registry (if applicable)  
 model\_id BIGINT UNSIGNED NULL,   
   
 ui\_metadata JSON,   
 config\_values JSON,   
   
 FOREIGN KEY (workflow\_id) REFERENCES workflows(id) ON DELETE CASCADE,  
 FOREIGN KEY (model\_id) REFERENCES ai\_model\_registry(id)  
);  
  
CREATE TABLE workflow\_edges (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 workflow\_id BIGINT UNSIGNED NOT NULL,  
   
 source\_node\_id BIGINT UNSIGNED NOT NULL,  
 source\_socket\_name VARCHAR(50) NOT NULL,   
   
 target\_node\_id BIGINT UNSIGNED NOT NULL,  
 target\_socket\_name VARCHAR(50) NOT NULL,   
   
 FOREIGN KEY (workflow\_id) REFERENCES workflows(id) ON DELETE CASCADE,  
 FOREIGN KEY (source\_node\_id) REFERENCES workflow\_nodes(id) ON DELETE CASCADE,  
 FOREIGN KEY (target\_node\_id) REFERENCES workflow\_nodes(id) ON DELETE CASCADE,  
   
 UNIQUE KEY unique\_wire (workflow\_id, target\_node\_id, target\_socket\_name)  
);  
  
CREATE TABLE workflow\_input\_manifests (  
 id BIGINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  
 workflow\_id BIGINT UNSIGNED NOT NULL,  
 node\_id BIGINT UNSIGNED NOT NULL,   
   
 variable\_key VARCHAR(50) NOT NULL,   
 label VARCHAR(255) NOT NULL,   
 input\_type ENUM('Image', 'Text', 'Video', 'Audio', 'Number', 'Boolean') NOT NULL,  
   
 is\_required BOOLEAN DEFAULT TRUE,  
 form\_order INT UNSIGNED NOT NULL,   
   
 validation\_rules JSON,   
 ui\_hints JSON,   
   
 FOREIGN KEY (workflow\_id) REFERENCES workflows(id) ON DELETE CASCADE,  
 INDEX (workflow\_id, form\_order)  
);