Software Requirements Specification (SRS): Teach by Doing (TbD) MVP - V2.0

Status: Approved for Build (Cloud MVP) **Author:** Greg Burns **Date:** November 22, 2025

---

1. Introduction

1.1 Purpose and Scope

The objective of V2.0 is to transition the **Teach by Doing (TbD)** engine from a local, conceptual prototype (V1) to a fully **scalable, cloud-native MVP** that can handle enterprise-grade video loads.

This release establishes the asynchronous, decoupled architecture required to be the ingestion front-end for the entire **Pathways as Data (PAD)** system. It focuses on stability and scalability for the core segmentation pipeline.

1.2 Core Architectural Shift (Decoupling)

This version addresses the critical constraint of video processing: **timeouts**. We shift from a synchronous API request (which would fail on Cloud Run) to an **asynchronous Fan-Out Architecture** using Google Cloud Pub/Sub .

---

2. Functional Requirements (FRs)

2.1 Asynchronous Ingestion & Tasking (The "Front Door")

- **FR-01 (Input Source):** The service **shall** accept a task from a **Google Cloud Pub/Sub message** containing a JSON payload.
- **FR-02 (Payload):** The payload **MUST** contain the fully qualified **Google Cloud Storage (GCS) URI** of the video file (`gs://bucket-name/video.mp4`), and no file stream data.
- **FR-03 (GCS Access):** The processing service **shall** use the Google Cloud Storage SDK to download the video file from the URI into its local, ephemeral disk for processing.

2.2 Core Processing Pipeline (V1 Logic)

- **FR-04:** The service **shall** execute the complete V1 segmentation logic (using `PySceneDetect` and `OpenCV`) to detect action events and segment the video.
- **FR-05:** The service **shall** perform visual data extraction (using `Tesseract` OCR) on segmented keyframes to capture `ui_element_text`.
- **FR-06:** The service **shall** generate a sequential list of `ActionNode` objects that conform to the established **PAD Schema v0.1**.

2.3 Output & Data Persistence

- **FR-07:** The service **shall** upload the final processed `Pathway.json` file to a separate, dedicated **GCS Results Bucket** (`gs://pad-results/`).
- **FR-08:** The service **shall** log the GCS URI of the final `Pathway.json` file to a defined logging service upon successful completion.

---

3. Non-Functional Requirements (NFRs)

3.1 Scalability and Stability (Cloud Run)

- **NFR-01 (Decoupling):** The application **MUST** be deployed as two separate Cloud Run services:
    1. **Dispatcher Service (API Gateway):** Receives the initial HTTP request, publishes to Pub/Sub, and returns a fast `202 Accepted`.
    2. **Worker Service (Processing):** Subscribes to the Pub/Sub topic and runs the heavy video processing.
- **NFR-02 (Autoscaling):** The Worker Service **shall** be configured for autoscaling to handle bursts of incoming video tasks concurrently.
- **NFR-03 (Statelessness):** The Worker Service **MUST** be completely stateless; all necessary data (video, schema, config) **MUST** be pulled from GCS at the start of a task.
- **NFR-04 (Cost Optimization):** The Worker Service **MUST** use an aggressive disk cleanup routine, deleting the local video file immediately upon job completion to preserve ephemeral disk space.

3.2 Security and Tech Stack

- **NFR-05 (Dependencies):** The Worker Service **MUST** retain the V1 "Bridge Stack" (`OpenCV`, `Tesseract`) and be packaged via a Docker container.
- **NFR-06 (IAM):** Service Accounts for the Worker **MUST** be scoped only for **GCS Object Viewer** (to download video) and **GCS Object Creator** (to upload the JSON result).

---

4. Interface Requirements (IRs)

4.1 Dispatcher (API) Interface

- **IR-01:** The dispatcher **shall** expose a RESTful `POST /submit` endpoint.
- **IR-02:** Upon receiving a request, the dispatcher **MUST** publish a message to the `tb-d-ingest-tasks` Pub/Sub topic and immediately return a **HTTP 202 Accepted** status code.

4.2 Worker (Pub/Sub) Interface

- **IR-03:** The Worker service **shall** be configured to be triggered directly by subscriptions to the `tb-d-ingest-tasks` Pub/Sub topic.
- **IR-04:** The Worker's output **shall** generate structured logs (Stackdriver compatible) for tracking task ID, processing time, and output GCS URI.