API Schema Definition: Teach by Doing (TbD) V6.0

Status: Production Ready **Version:** 6.0.0 **Scope:** Defines all external and internal interfaces for the TbD platform.

1. External API (The Front Door)

This is the public-facing API exposed by **Service A: Dispatcher**. Clients (like the Streamlit Frontend) use this to initiate processing jobs.

Endpoint: `POST /submit`

URL: `https://tbd-dispatcher-{ID}-uc.a.run.app/submit` **Content-Type:** `application/json`

Request Body (`TaskPayload`)

Field	Type	Required	Description
`task_id`	`string` (UUID)	**Yes**	A unique identifier for this specific processing job. Generated by the client.
`client_id`	`string`	**Yes**	Identifier for the calling application (e.g., "streamlit-console", "mobile-app").
`gcs_uri`	`string` (URI)	**Yes**	The Google Cloud Storage URI of the uploaded raw video file. Format: `gs://{bucket_name}/{blob_path}`

`output_bucket`	`string`	**Yes**	The GCS bucket where the final `Pathway.json` should be saved.
`config`	`object`	No	Optional configuration overrides (e.g., `{ "enable_stt": false }`).

Example Request:

```
{
  "task_id": "550e8400-e29b-41d4-a716-446655440000",
  "client_id": "streamlit-cloud-console",
  "gcs_uri": "gs://tbd-raw-video-tbd-v2/550e8400.../demo.mp4",
  "output_bucket": "tbd-results-tbd-v2",
  "config": {}
}
```

Response (Success)

- **Status Code:** `202 Accepted`

Body:

```
{
  "status": "Task accepted and queued",
  "task_id": "550e8400-e29b-41d4-a716-446655440000",
  "trace_id": "0af7651916cd43dd8448eb211c80319c" // OpenTelemetry Trace ID
}
```

-

Response (Error)

- **Status Code:** `400 Bad Request` (Invalid JSON) or `500 Internal Server Error` (Pub/Sub Failure).

2. Internal Service APIs (Microservices Mesh)

These APIs are **not** public. They are protected by IAM authentication and are only callable by **Service B: Worker** (or strictly authorized clients).

Service C: Temporal Encoder

Role: Generates semantic vectors from text sequences. **URL:** `https://tbd-temporal-encoder-{ID}-uc.a.run.app/encode_sequence` **Auth:** Required (OIDC Token)

Request Body (`SequenceInput`)

Field	Type	Required	Description
`sequence`	`List[str]`	**Yes**	An ordered list of text descriptions (one per step) to be encoded.

Example Request:

```
{
  "sequence": [
    "User clicks 'File' menu.",
    "User selects 'Save As'.",
    "User types filename."
  ]
}
```

Response (`VectorOutput`)

Field	Type	Description
`temporal_context_vector`	`List[float]`	A 512-dimensional float array representing the sequence context.
`temporal_encoding_method`	`string`	The algorithm version used (e.g., "LSTM_512_V6_REBUILD").

Service D: Object Detector

Role: Detects UI elements in a video frame. **URL:** `https://tbd-object-detector-{ID}-uc.a.run.app/detect_coordinates` **Auth:** Required (OIDC Token)

Request Body (`FramePayload`)

Field	Type	Required	Description
`frame_base64`	`string`	**Yes**	The raw image frame encoded as a Base64 string.
`target_text`	`string`	No	(Future Use) Text label to help guide detection. Default: "default".

Example Request:

```
{
  "frame_base64": "/9j/4AAQSkZJRgABAQEASABIAAD...",
  "target_text": "Save Button"
}
```

Response (`DetectionResult`)

Field	Type	Description
`ui_region`	`List[int]`	The bounding box coordinates `[x, y, w, h]` in absolute pixels.
`confidence`	`float`	The model's confidence score (0.0 - 1.0).

3. Asynchronous Messaging (Pub/Sub)

The system relies on Google Cloud Pub/Sub for decoupling.

Topic: `tb-d-ingest-tasks`

Publisher: Service A (Dispatcher) **Subscriber:** Service B (Worker) - *Push Subscription*

Message Attributes:

- `trace_id`: The OpenTelemetry trace ID for distributed tracing.

Message Data (JSON): Identical to the **`TaskPayload`** defined in Section 1.

Topic: `pad-agent-tasks`

Publisher: Service B (Worker) **Subscriber:** Downstream Risk/Simulation Agents

Message Data (String): The GCS URI of the final output file.

- **Example:** `gs://tbd-results-tbd-v2/550e8400.../pathway.json`

4. Final Output Schema (`Pathway.json`)

This is the structured asset generated by the system. It conforms to **PAD Schema v0.5**.


```
{
  "pathway_id": "550e8400-e29b-41d4-a716-446655440000",
  "title": "Native Insight: demo.mp4",
  "author_id": "tbd-v4-engine",
  "source_video": "demo.mp4",
  "created_at": "2025-11-25T10:30:00-0800",
  "total_duration_sec": 45.5,
  "metadata": {
    "target_vertical": "manufacturing",
    "compliance_tag": "AS9100",
    "license_tier": "royalty-pro",
    "pathway_version": "v1.0"
  },
  "nodes": [
    {
      "id": "node_1",
      "type": "action",
      "timestamp_start": 2.5,
      "timestamp_end": 3.5,
      "description": "User clicks the 'Settings' icon to open the configuration panel.",
```

```
      "semantic_description": "User clicks the 'Settings' icon to open the configuration panel.",
      "action_type": "click",
      "ui_element_text": "Settings",
      "ui_region": [100, 200, 50, 50],
      "confidence": 0.92,
      "active_region_confidence": 0.92,
      "temporal_context_vector": [0.01, -0.45, 0.22, ...], // 512 floats
      "telemetry_context": {
        "sensor_id": "DED-Robot-Arm-01",
        "machine_state": "ACTIVE_PRINTING",
        "ambient_temp_c": 24.6
      },
      "next_node_id": "node_2"
    },
    // ... more nodes
  ]
}
```