

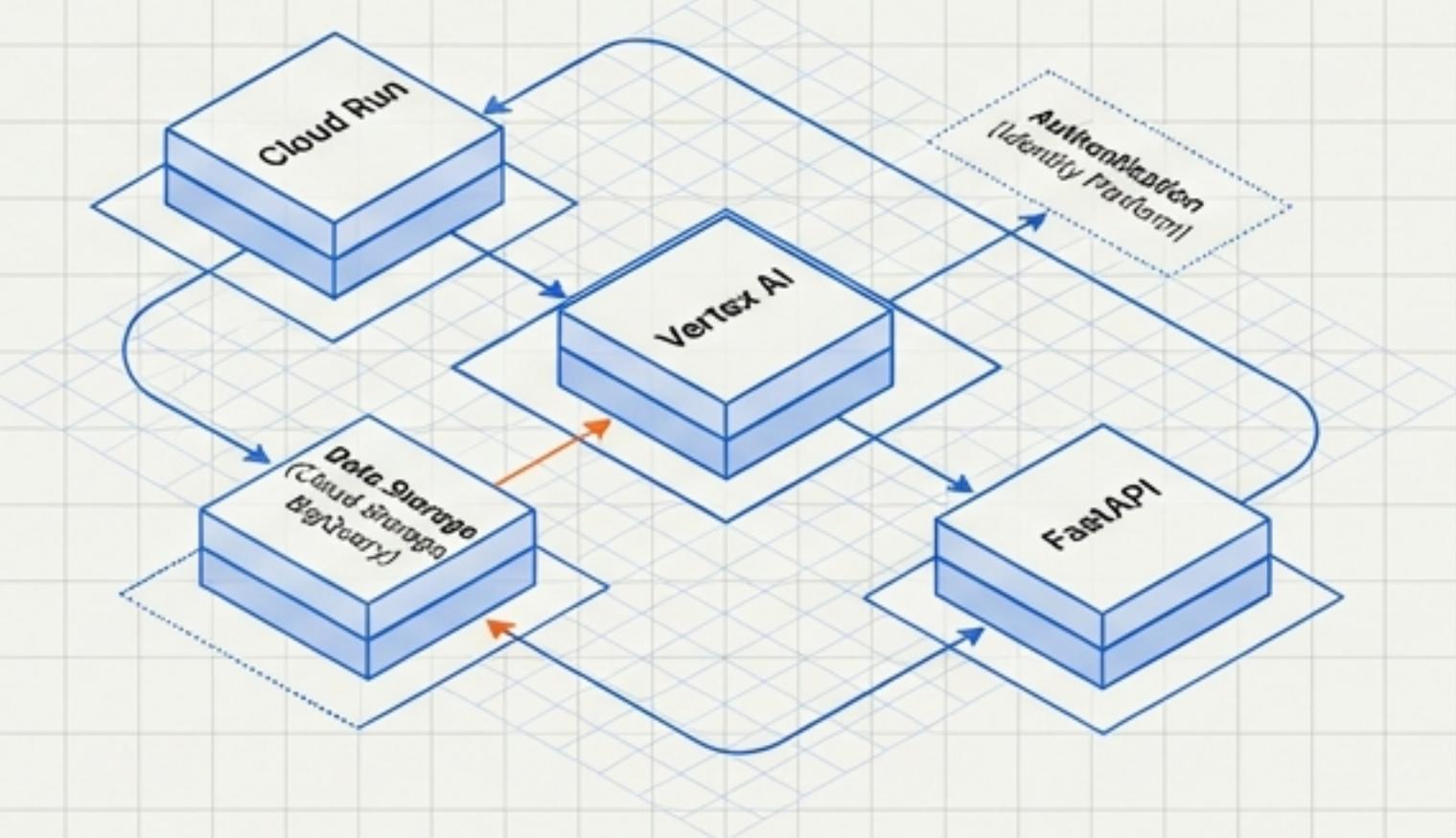
# The UVAI Blueprint: Architecting the Universal Video Intelligence Engine

Transitioning from Local Prototype to Enterprise SaaS using Google AIP Standards & Vertex AI

A terminal window titled "Terminal" displaying error logs. The logs indicate dependency conflicts between "ai-edge-torch" and "st-edge-torch", disk quota exceeded errors, and failed building wheels. It also shows attempts to install packages via pip and issues with environment variables and PATH. The process terminates with an exit code of 1.

```
Terminal
ERROR: ai-edge-torch dependency conflict
ERROR: st-edge-torch dependency conflict
*   * disk quota exceeded!
*   * env blobt
*   * env btoet
Tracing back (most recent call last)...
File "/Users/yossi/tide/ci/testeeceesovgaze.py", line 38, in <module>
    setst_a_senesetad()
File "/Users/yossi/tide/ci/testeeceesovgaze.py", line 9, in __temset__
    zetgoseseneat,jeoat()
File "/Users/yossi/tide/ci/testeeceesovgaze.py", line 280, in __easor_zeah_
    setaa ss.eas08.sseas.:lissi
Failed building wheel for ____snasit:
File "/Users/yossi/tide/ci/testeeceesovgaze.py", line 280, in __module__
    retrv. reecsettrobeet_oecchage_t_sercot()
File "/Users/yossi/tide/ci/testeeceesovgaze.py", line 56, in __soet__
    retrn aesaecoeatrtensrdesetst()
Could not install packages due to an EnvironmentError: Not ressofmasico at buildw
se-ce-edge-resmetestesayrcceccesbro,an-EnvitonmentErcoceca. Aasse.604290051:recoanBR
tret,erect,ceasctegtfiactngt
Startteosets:stortettettsesios.tsoe etake
    Could not install packages due to an EnvironmentError: seakded
WARNING: The scripts are not on PATH
Process terminated with exit code 1
```

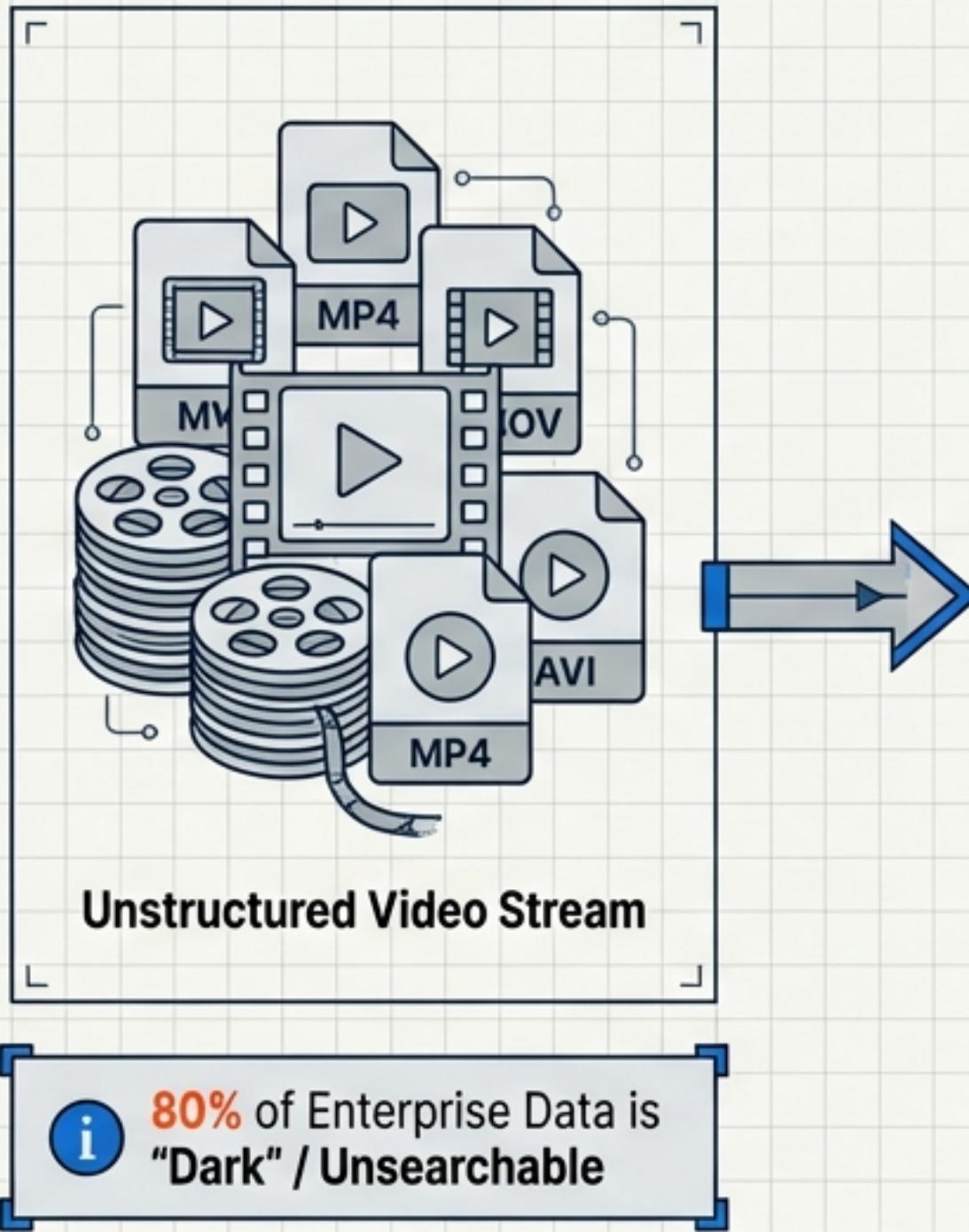
**CURRENT STATE:** EventRelay Prototype (Local/Repo-Bound)



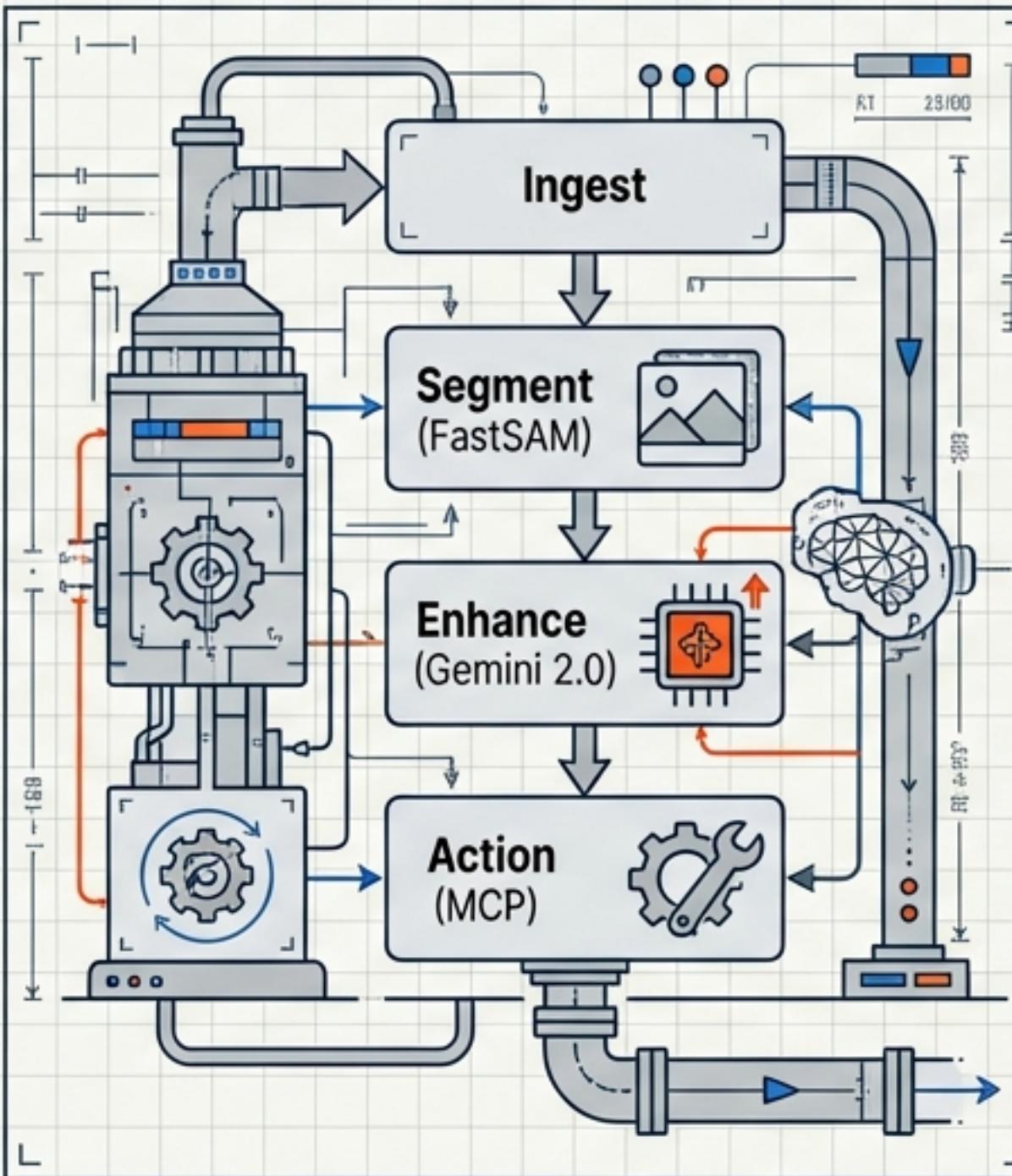
**TARGET STATE:** UVAI Enterprise Platform (Cloud-Native)

# The Digital Refinery: Converting Dark Data into Deployed Software

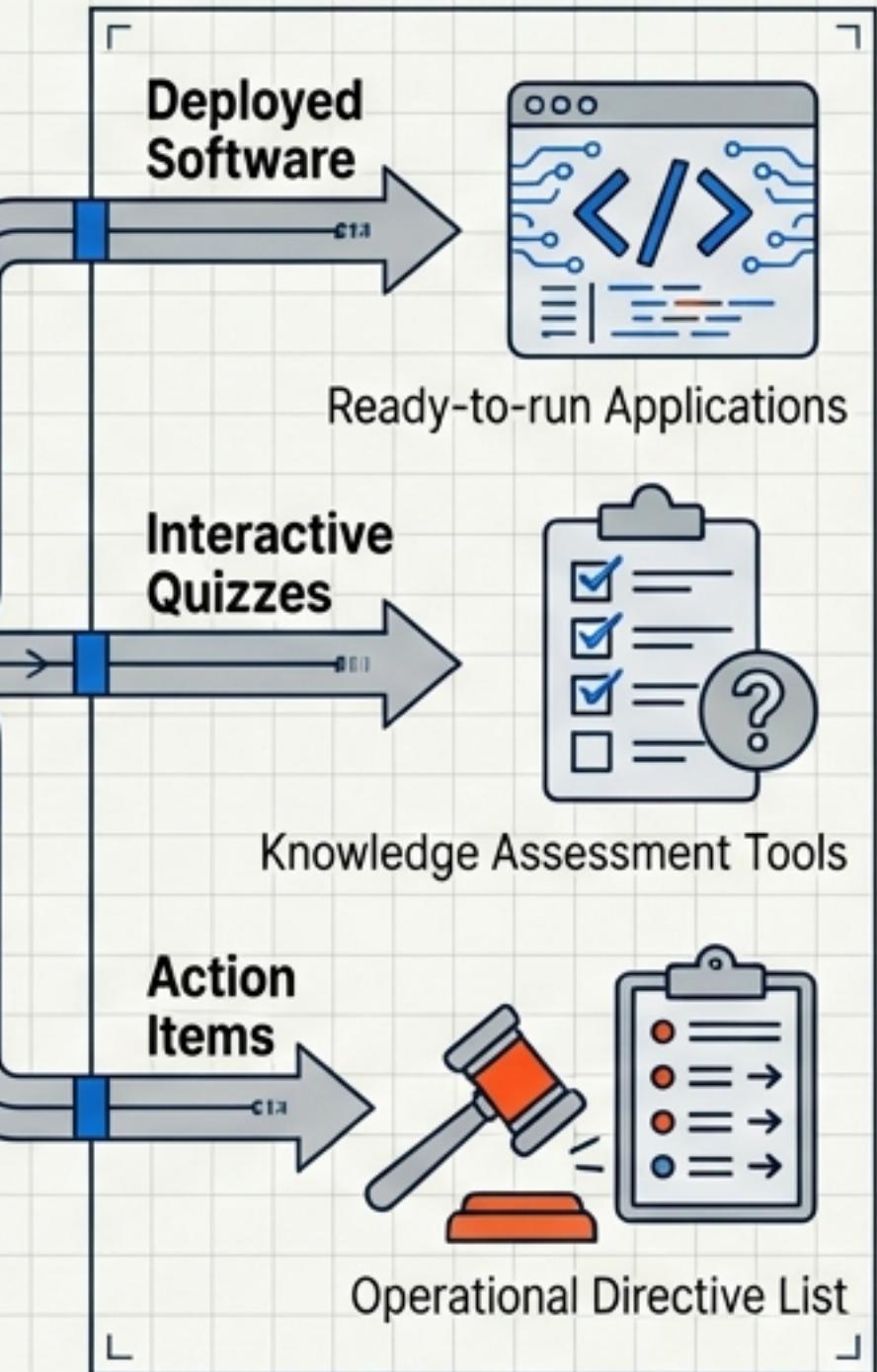
## Stage 1: Raw Material



## Stage 2: The Refinery (UVAI)



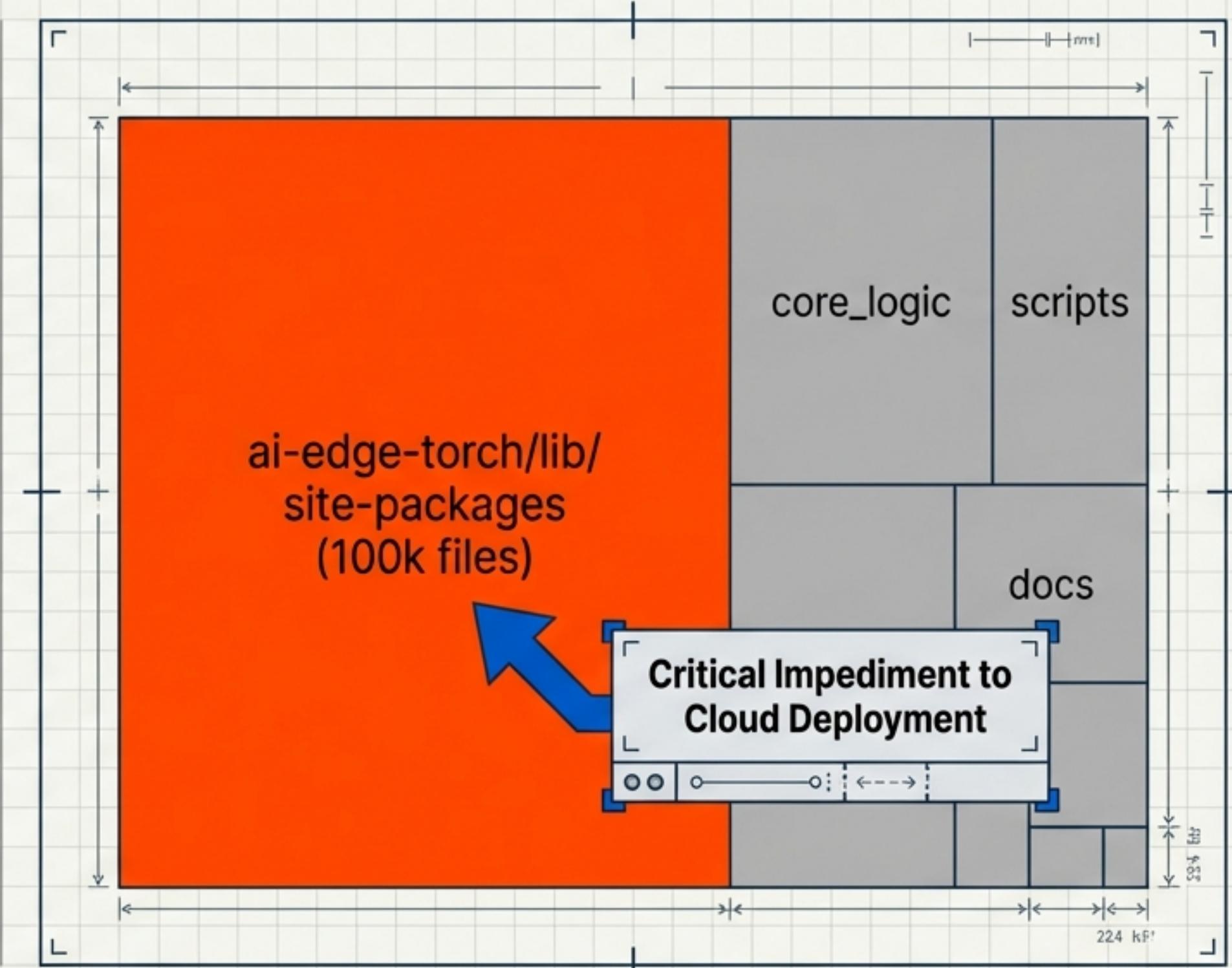
## Stage 3: Refined Products



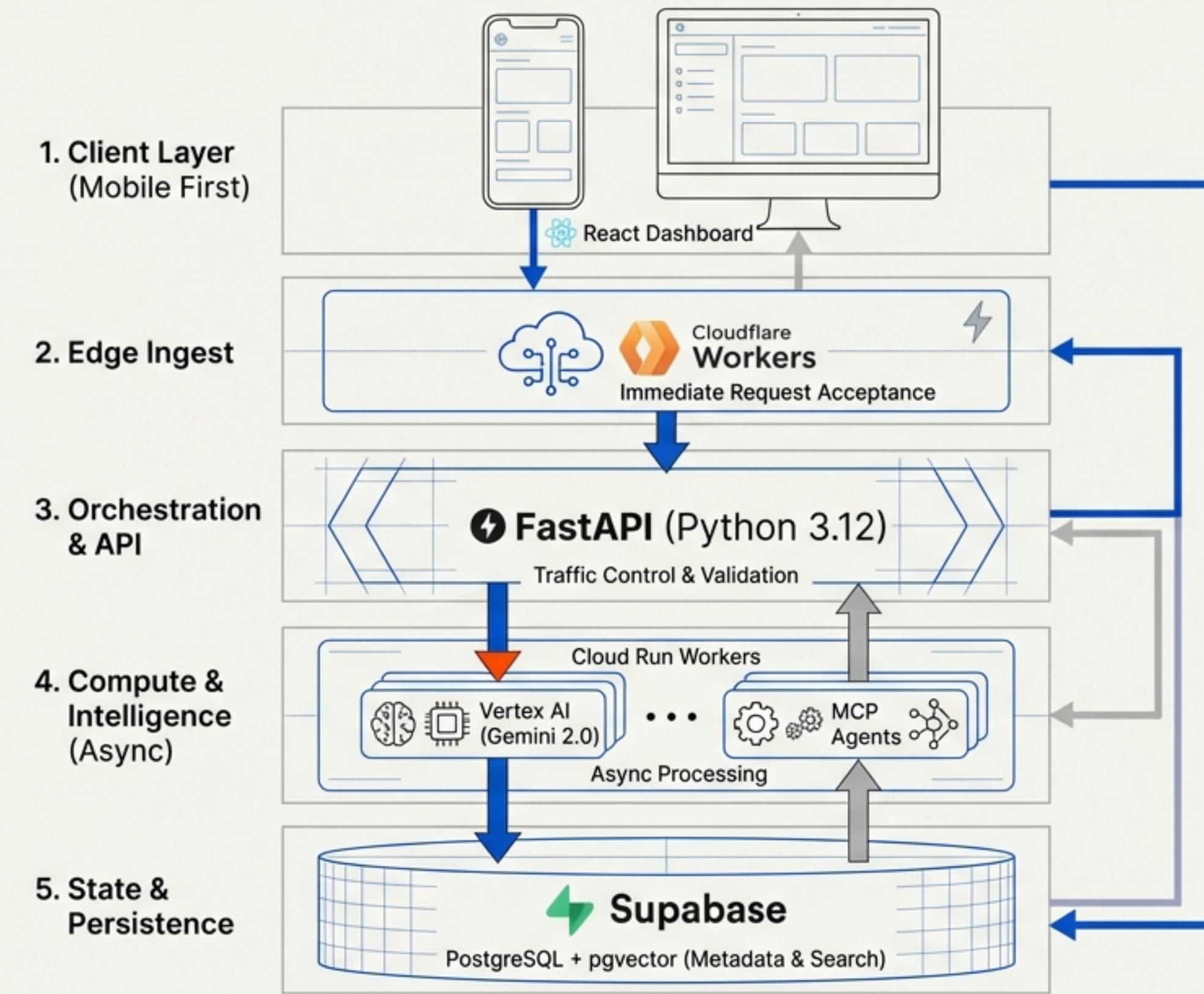
**Productivity Loss:** Knowledge workers lose **\$500B** annually retrieving manual insights. UVAI automates the **Video → Intelligence → Action** pipeline.

# The ‘Repo-Bound’ Bottleneck: EventRelay Technical Audit

- **Dependency Bloat:** 100,000+ files in vendored site-packages blocks cloud ingestion.
- **Local Fragility:** Execution tied to single-machine hardware; incompatible with mobile edge.
- **Static Failure Handling:** ‘failures/’ directory acts as a graveyard, not a retry queue.
- **Verdict:** High-value logic trapped in a non-scalable architecture.



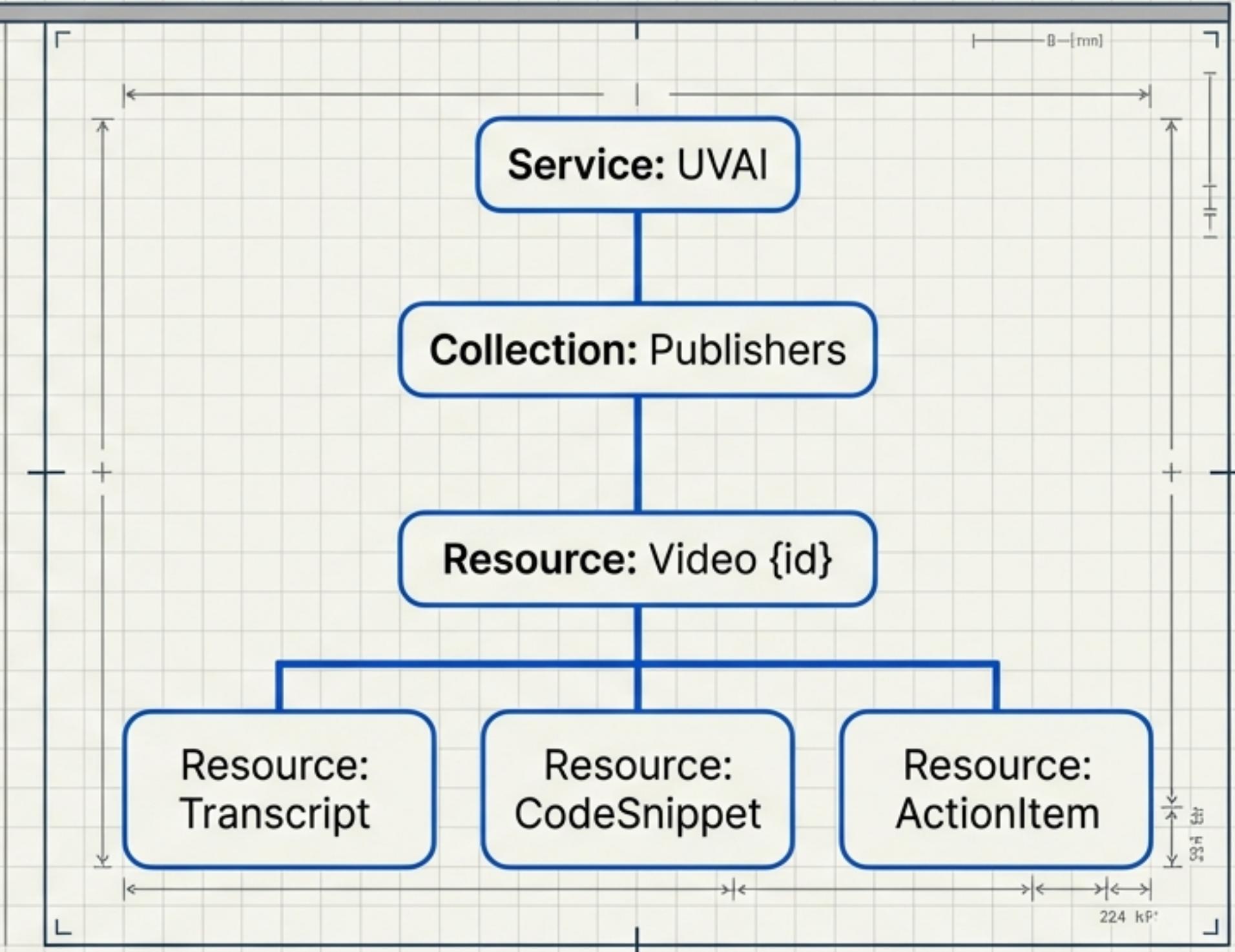
# Target Architecture: Decoupled, Containerized, and Cloud-Native



# Designing the API Surface (AIP-121)

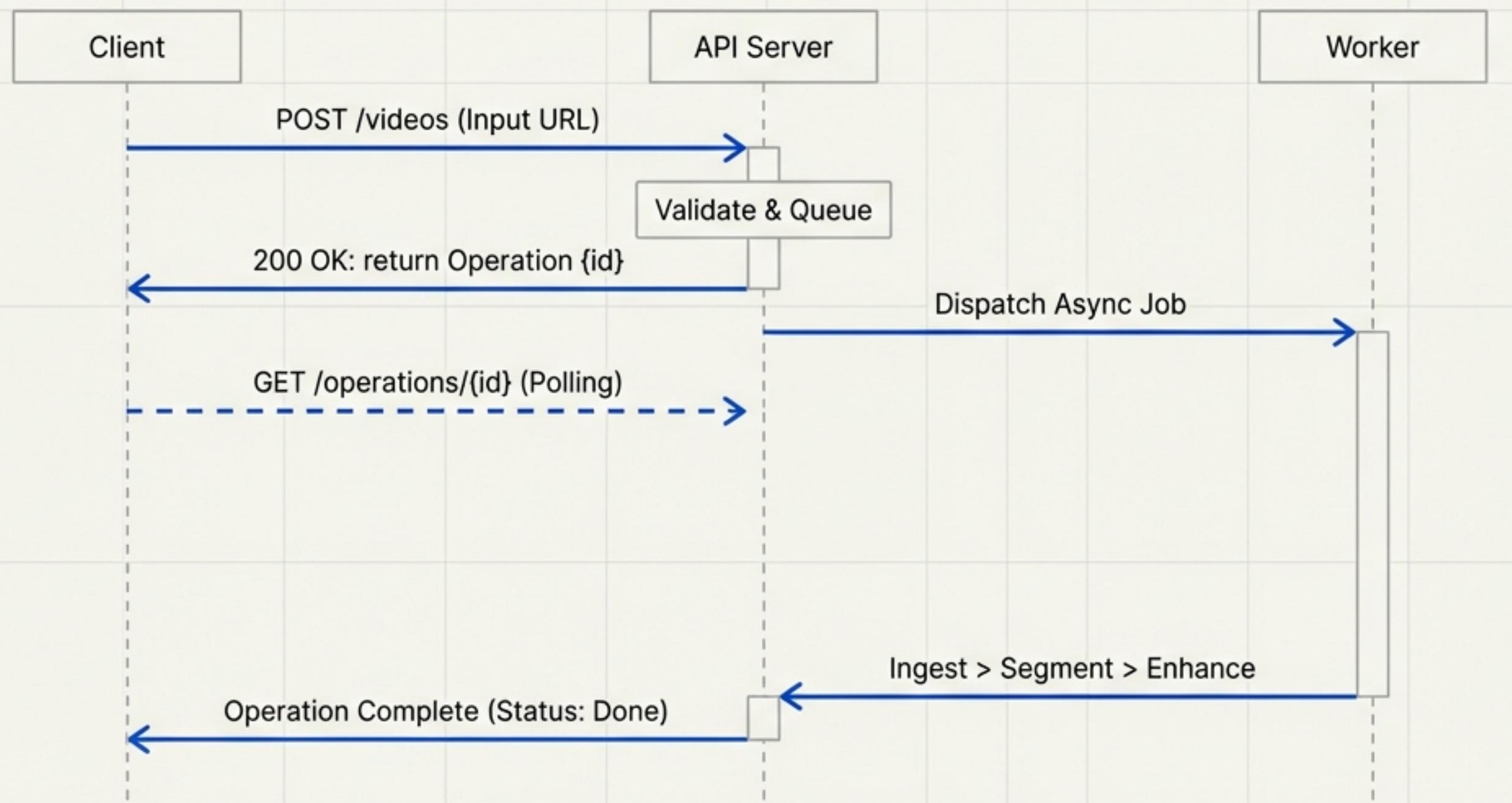
## Resource-Oriented Design Principles in Inter

- Resource-Oriented Design Principles.
- **Core Principle:** Shift from RPCs (Verbs) to Resources (Nouns).
- **Constraint:** Stateless protocol enables mobile/cloud decoupling.
- **Rule:** API Schema ≠ Database Schema.



# Handling Latency: AIP-151 Long-Running Operations

Solving the “2-Hour Video” Timeout Problem



**UX Impact:** Enables the “Visual Timeline” feature on the dashboard by providing granular status updates without blocking the UI.

# AIP-136: Custom Methods for Intelligent Actions

Mapping Complex Domain Logic to RESTful Endpoints

## Anti-Pattern / RPC Style

```
POST /runGrokAnalysis  
POST /extract_code_from_video
```

Messy, verb-heavy, inconsistent URL structure.

## AIP Standard

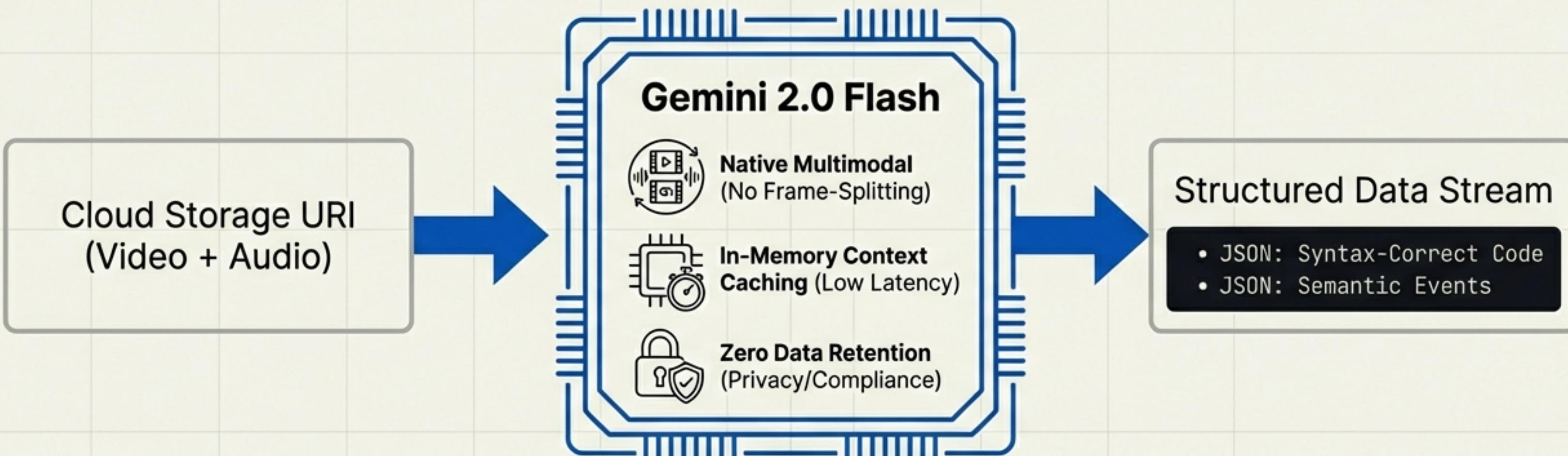
```
POST /videos/{id}:generateCode  
POST /videos/{id}:extractActionItems
```

Clean, resource-scoped, uses ':verb' syntax.

- **Concept:** Mapping complex domain logic (AI reasoning) to RESTful endpoints.
- **Constraint:** Actions with side effects must use POST.
- **Constraint:** Verbs must not contain prepositions.

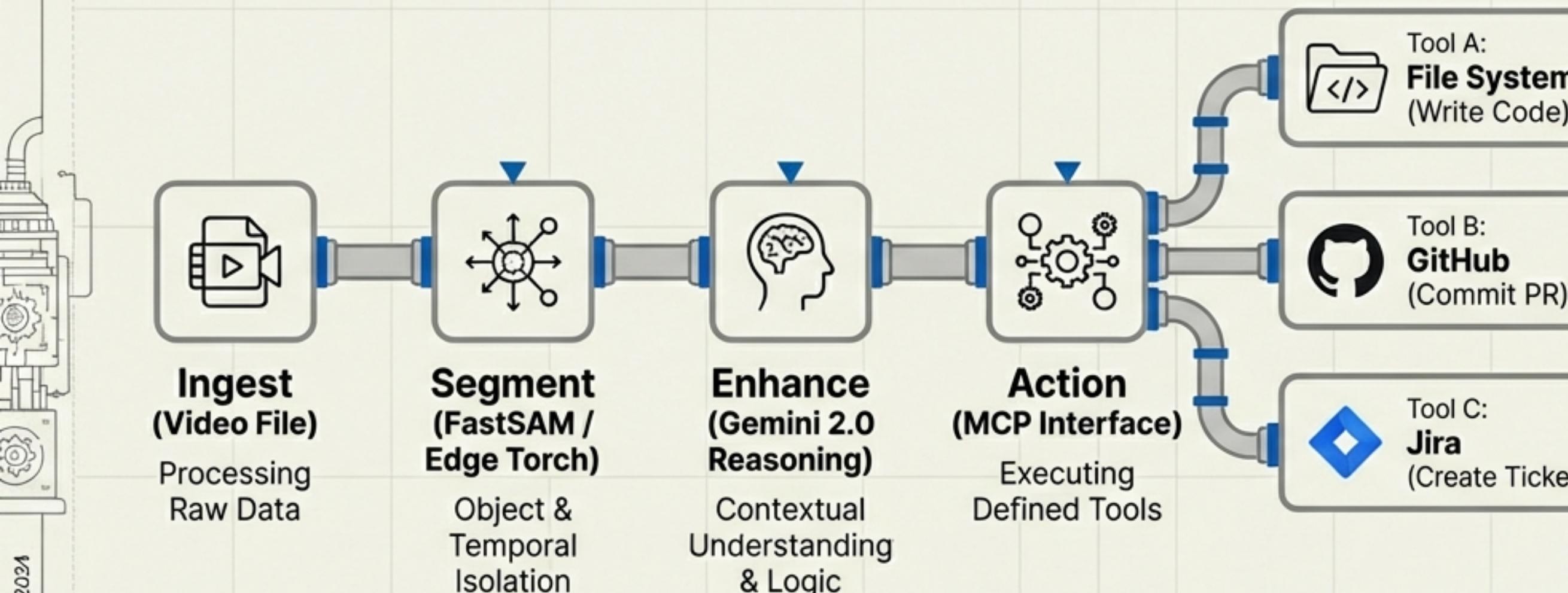
# The Intelligence Layer: Vertex AI & Gemini 2.0

Processing Multimodal Data with Contextual Grounding



- **Contextual Grounding:** The model 'watches' the code snippet while 'listening' to the explanation to ensure syntax accuracy.

# The Agentic Workflow: Model Context Protocol (MCP)



## The DevOps Agent

- Trigger:** "Clean up the repo" (**Alert Orange**)
- Action:** Agent scans site-packages, identifies imports, generates clean requirements.txt, deletes bloat.

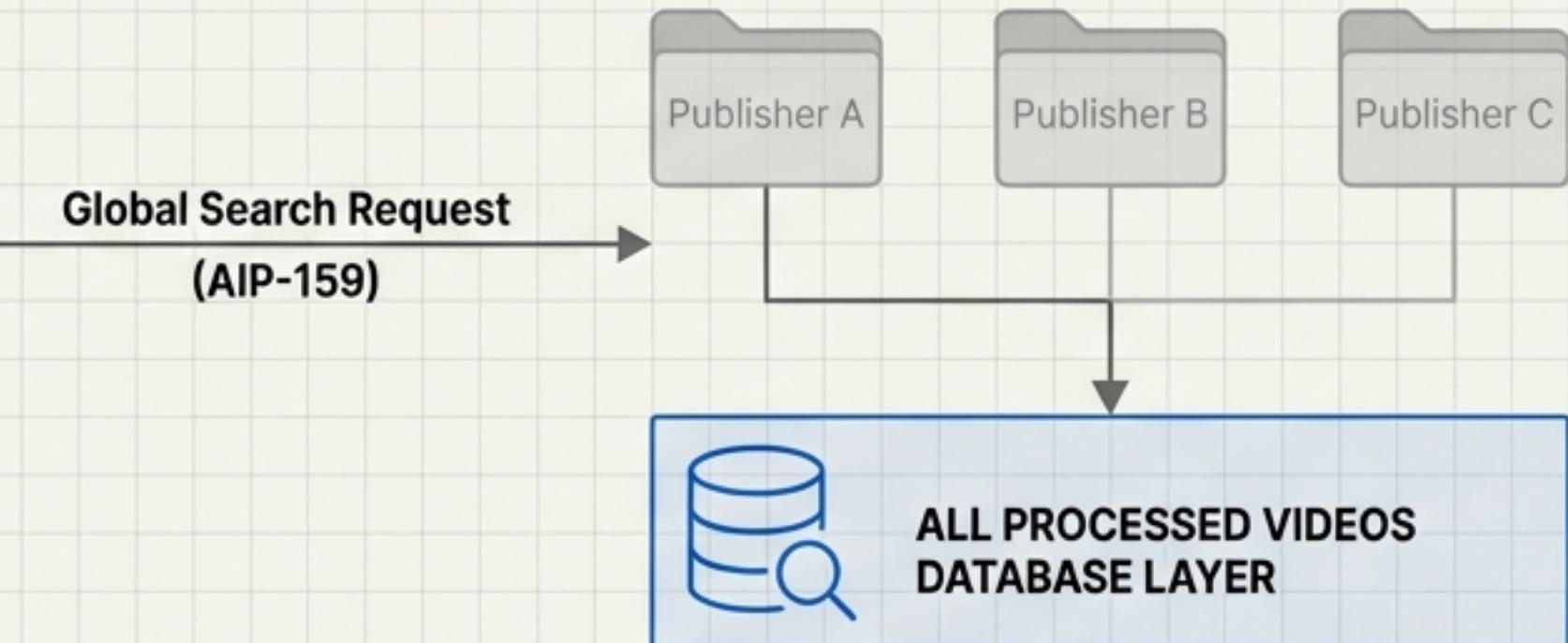
# Discovery & Standards: AIP-159 and AIP-210

## Reading Across Collections (AIP-159)

**API REQUEST**

```
GET /publishers/-/videos?filter=content:python
```

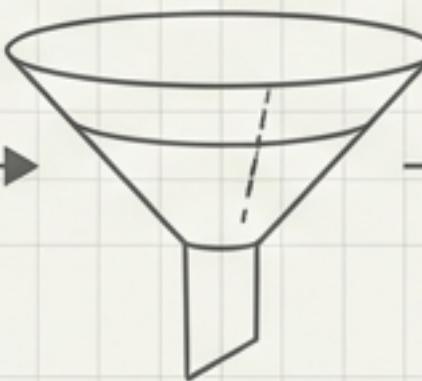
Enables global search for code snippets across all processed videos using the '-' wildcard.



## Unicode Compliance (AIP-210)

**INPUT:** Messy Input with Mixed Encodings

t€xt....\_ \ud83d\ude00

**NORMALIZATION FUNNEL (NFC)**

**OUTPUT:** Clean NFC Normalized Text

text....\_ 😊

Ensures consistent code generation and search across 50+ languages. Billing is based on Code Points, not Bytes.

OVE

# Developer Experience: Client Libraries (AIP-4210)

```
ooo  
from google.genai import Client  
  
# Initialize UVAI Client  
client = uvai.Client(project='my-video-app')  
  
# Declarative Pipeline Call  
video = client.videos.create(  
    uri='gs://bucket/tutorial.mp4',  
    actions=['generate_code', 'extract_quiz'])  
  
print(video.status) # 'PROCESSING'
```

**GAPICs:** Idiomatic libraries generated automatically from API definitions.

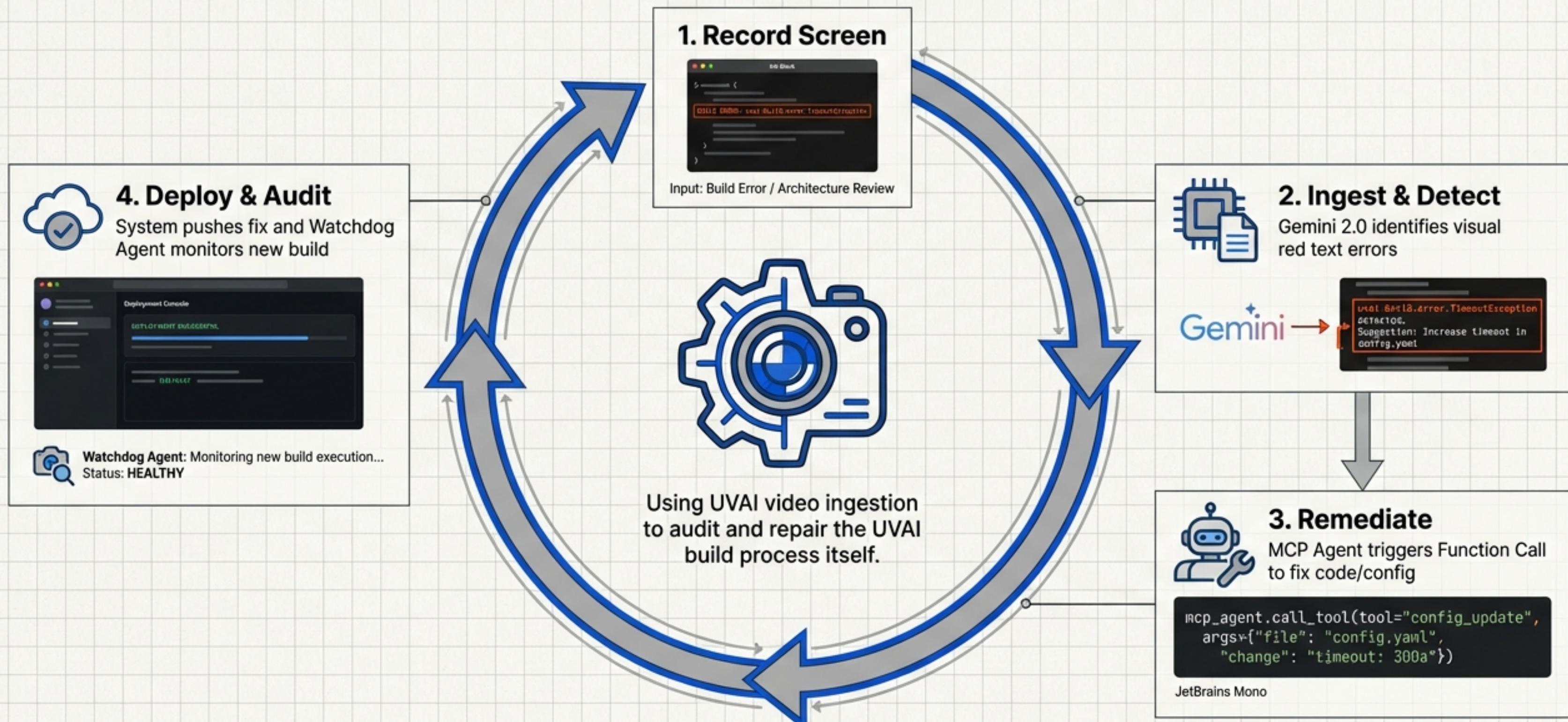


**Declarative-Friendly:** Supports 'Configuration as Code' for DevOps integration.



OVE

# The “Self-Repairing” Pipeline: A Dogfooding Strategy



# Execution Roadmap: From Cleanup to Cloud



## Phase 1: Remediation



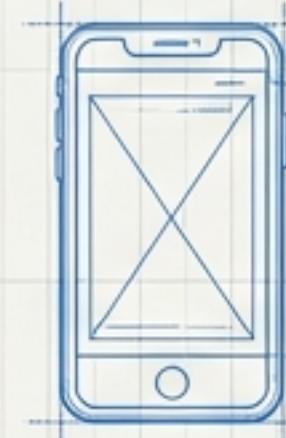
- **Agentic Pruning:** Eliminate 100k files in ai-edge-torch/site-packages.
- **Artifacts:** Generate clean requirements.txt & Production Dockerfile.

## Phase 2: Cloud Decoupling



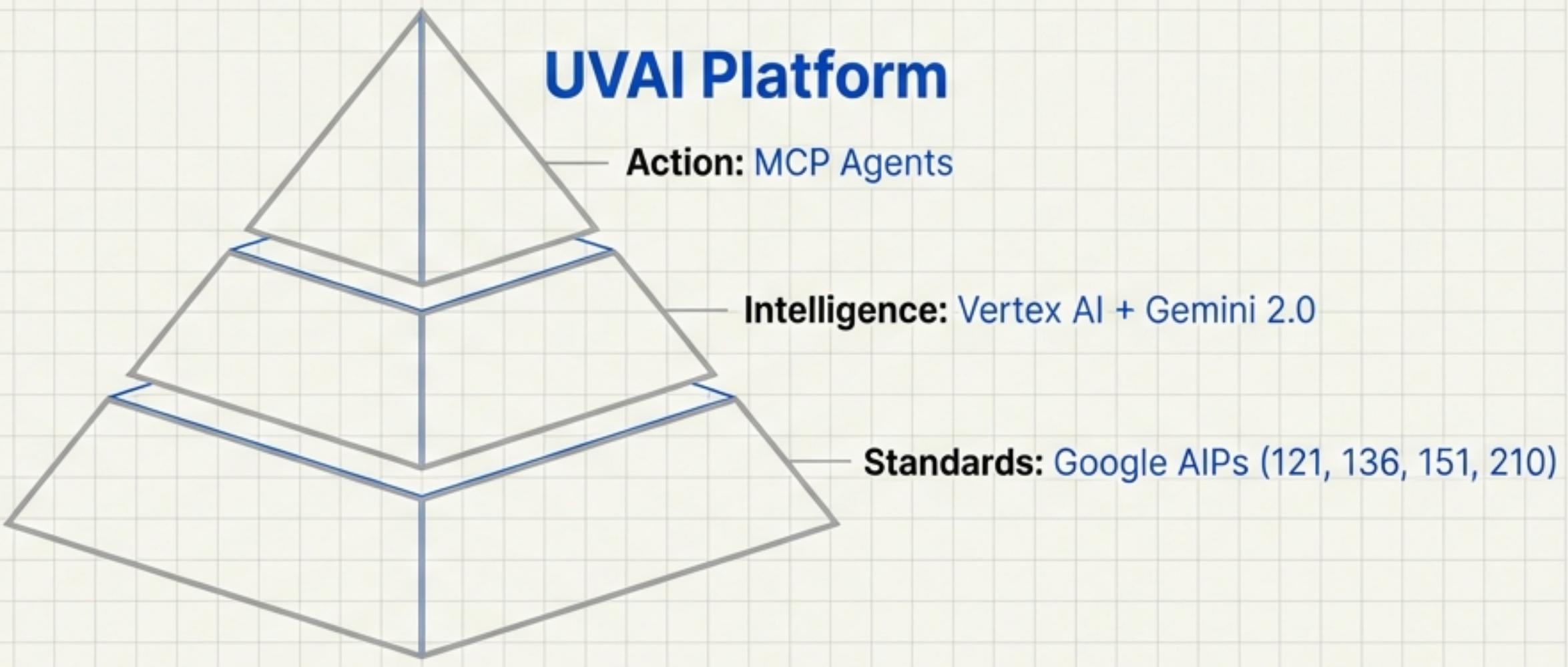
- **Backend:** Deploy FastAPI to Cloud Run.
- **Edge:** Implement Cloudflare Workers for instant ingestion.

## Phase 3: Experience



- **Frontend:** Refactor "state-fabric-concept" to React.
- **Launch:** Mobile-First Upload Flow & Visual Timeline.

# Summary: The Billion-Dollar Video Engine



- ▶ Scalability requires rigid standards (AIPs) to move beyond prototype.
- ▶ Value is created when video understanding (Gemini) drives software generation (MCP).
- ▶ The path forward: Execute the Architecture.

# **EXECUTE THE ARCHITECTURE**

**From Repository to Refinery.**