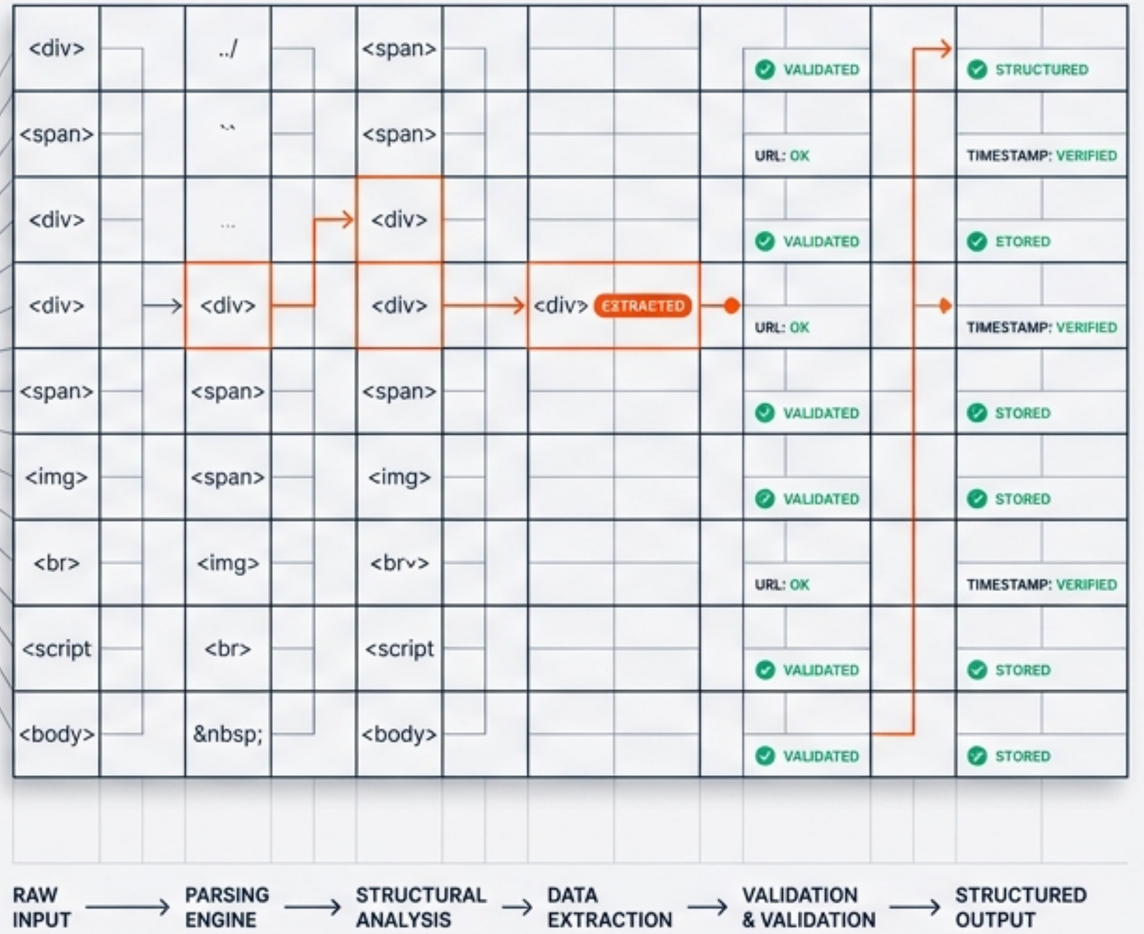
HTML Content Processing Pipeline

End-to-End Web Capture, Storage, and Intelligent Transformation



We have built a foundational engine for automated content processing at scale.

This is an end-to-end pipeline that captures web pages, secures them in a structured cache, and applies intelligent transformations to extract and filter content.

1. Capture

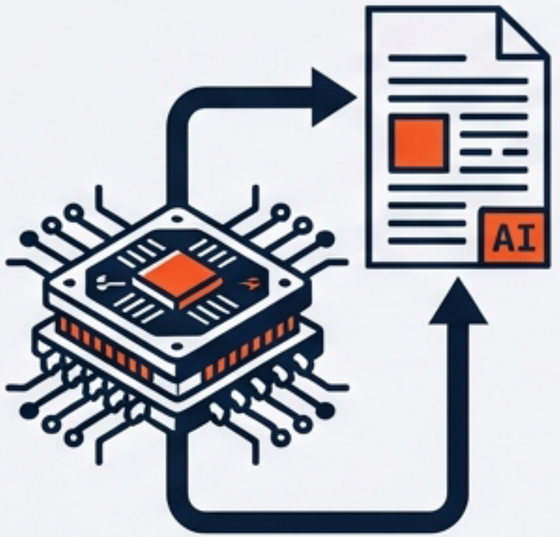
[redacted]

Fetching HTML from any public URL.

2. Refine

[redacted]

Structured storage via the LETS pipeline.



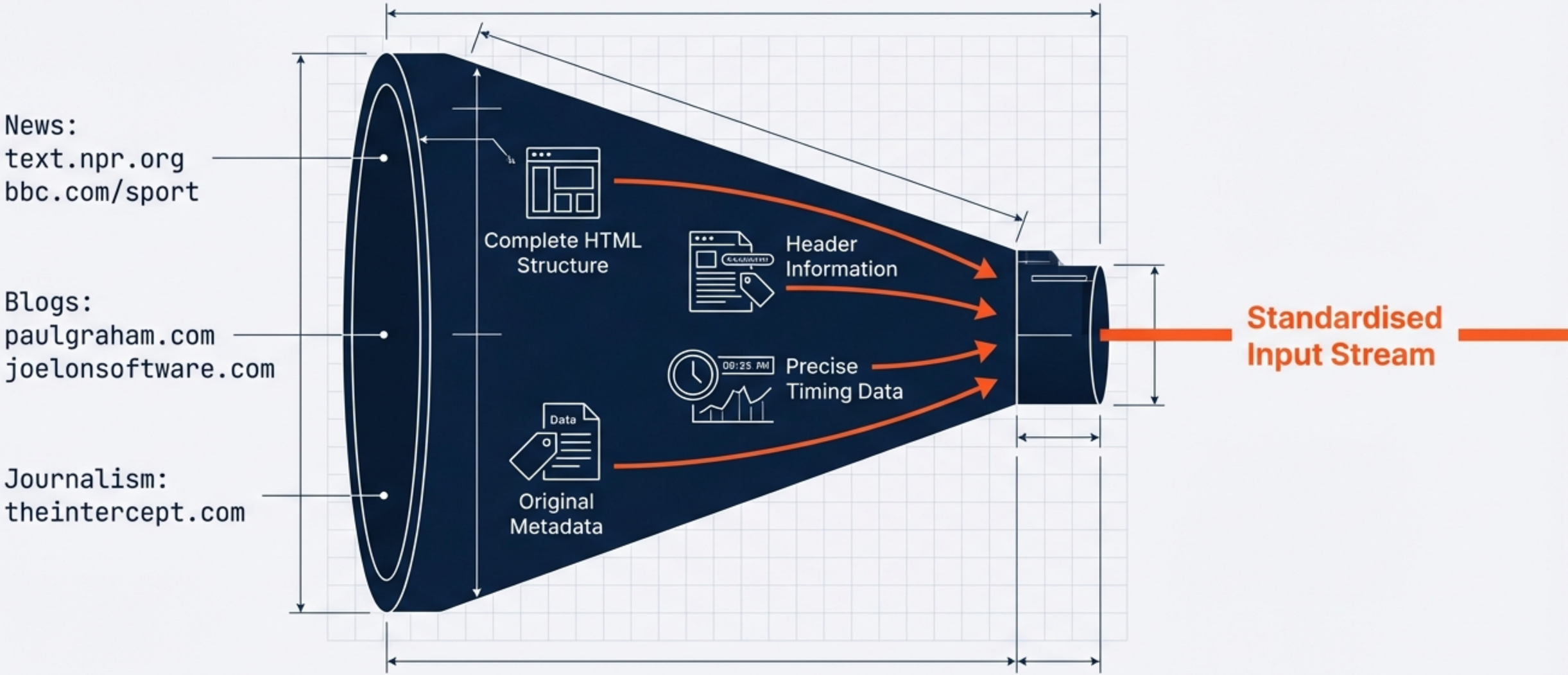
3. Utilise

[redacted]

Intelligent extraction for AI and analysis.

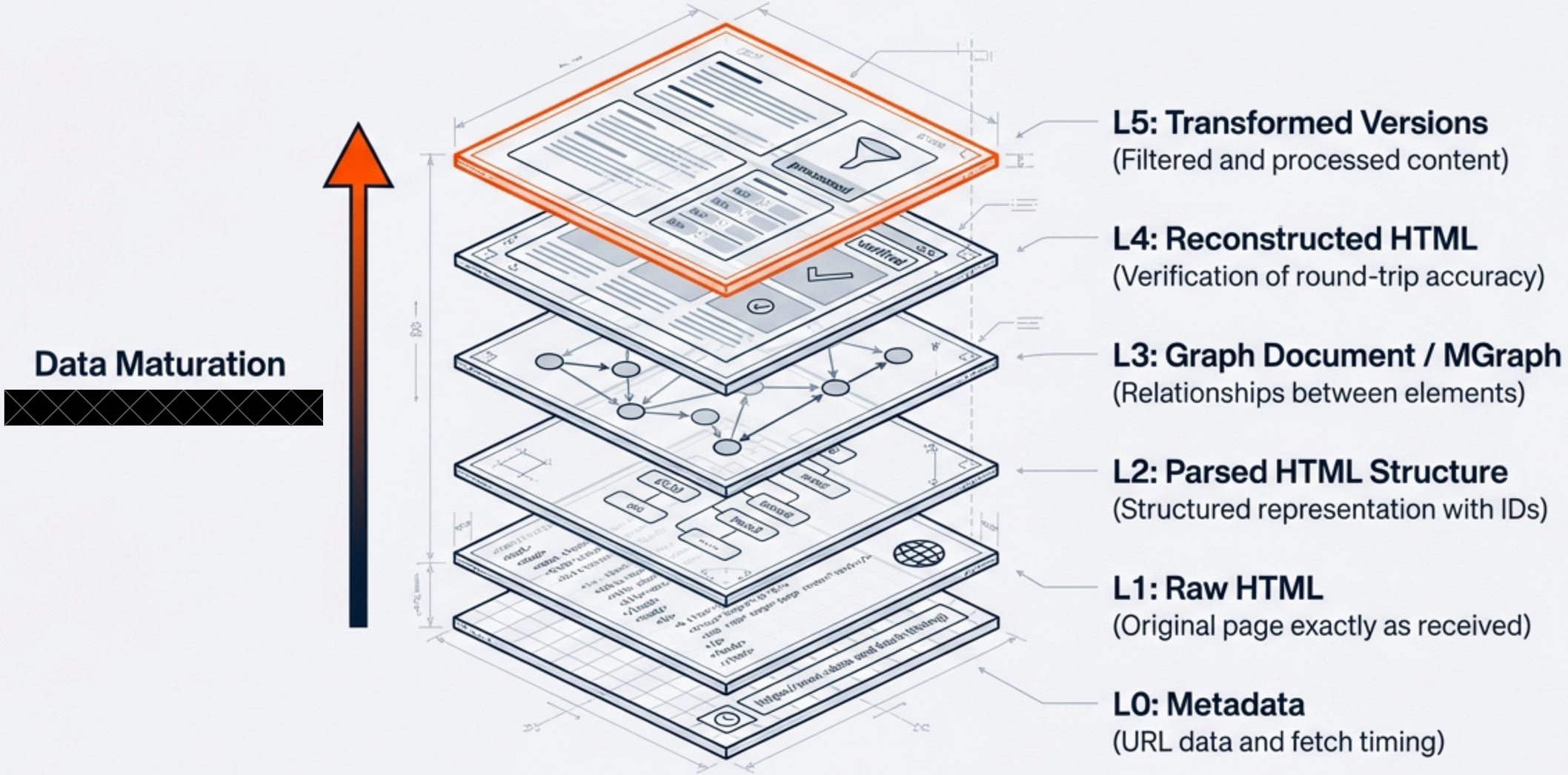
Key Takeaway: This system replaces ad-hoc scraping with a robust architecture designed for high-fidelity data preservation and retrieval.

Ingesting diverse content sources with full context preservation



The LETS Pipeline: A multi-layer architecture for data integrity.

Data Maturation



L5: Transformed Versions
(Filtered and processed content)

L4: Reconstructed HTML
(Verification of round-trip accuracy)

L3: Graph Document / MGraph
(Relationships between elements)

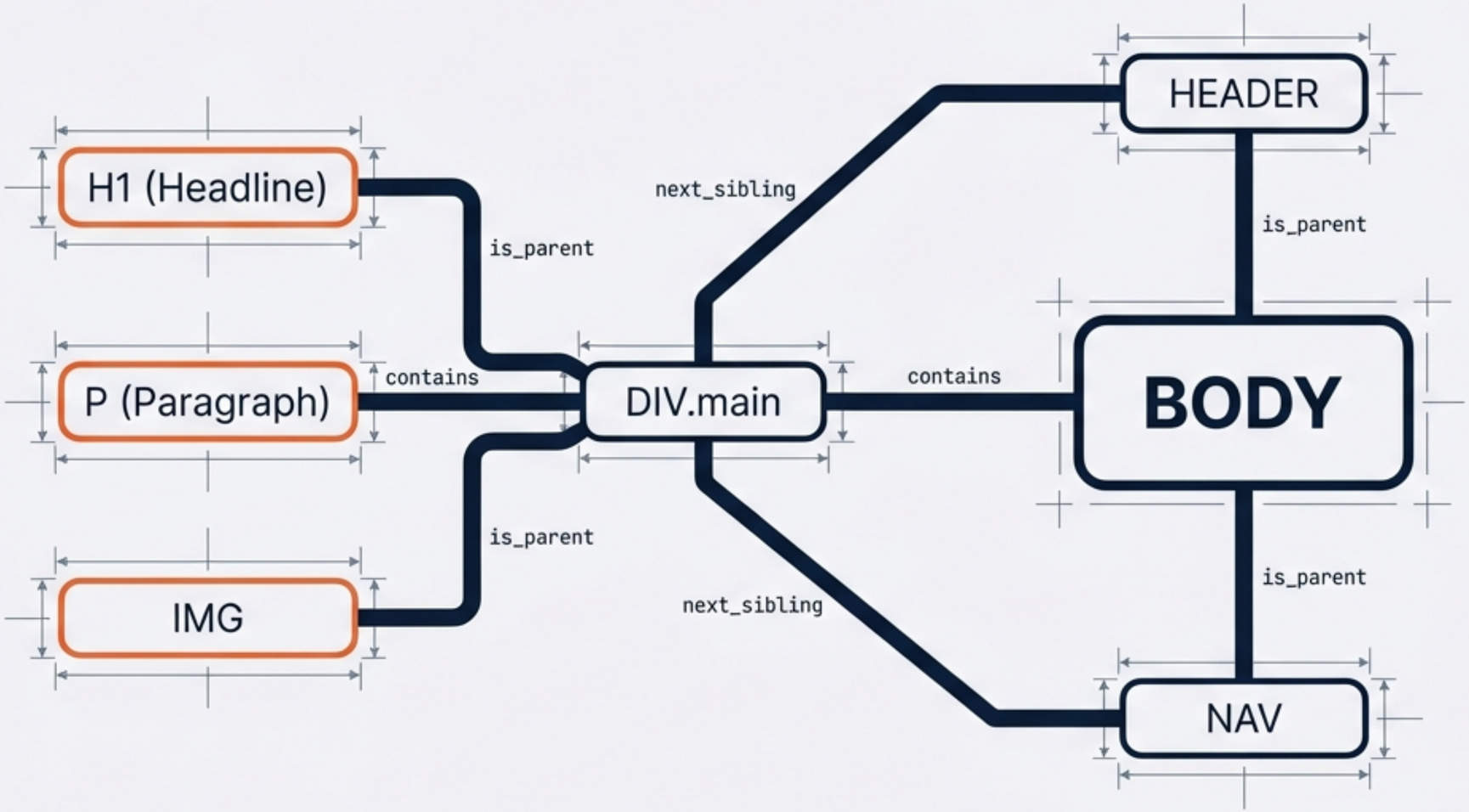
L2: Parsed HTML Structure
(Structured representation with IDs)

L1: Raw HTML
(Original page exactly as received)

L0: Metadata
(URL data and fetch timing)

Key Takeaway: The LETS pipeline ensures data integrity through a structured, multi-layered maturation process from raw capture to transformed, high-fidelity outputs.

Beyond text: Understanding structure via Graph Technology.



Layer L3 (MGraph) maps the relationships between every element on a page.

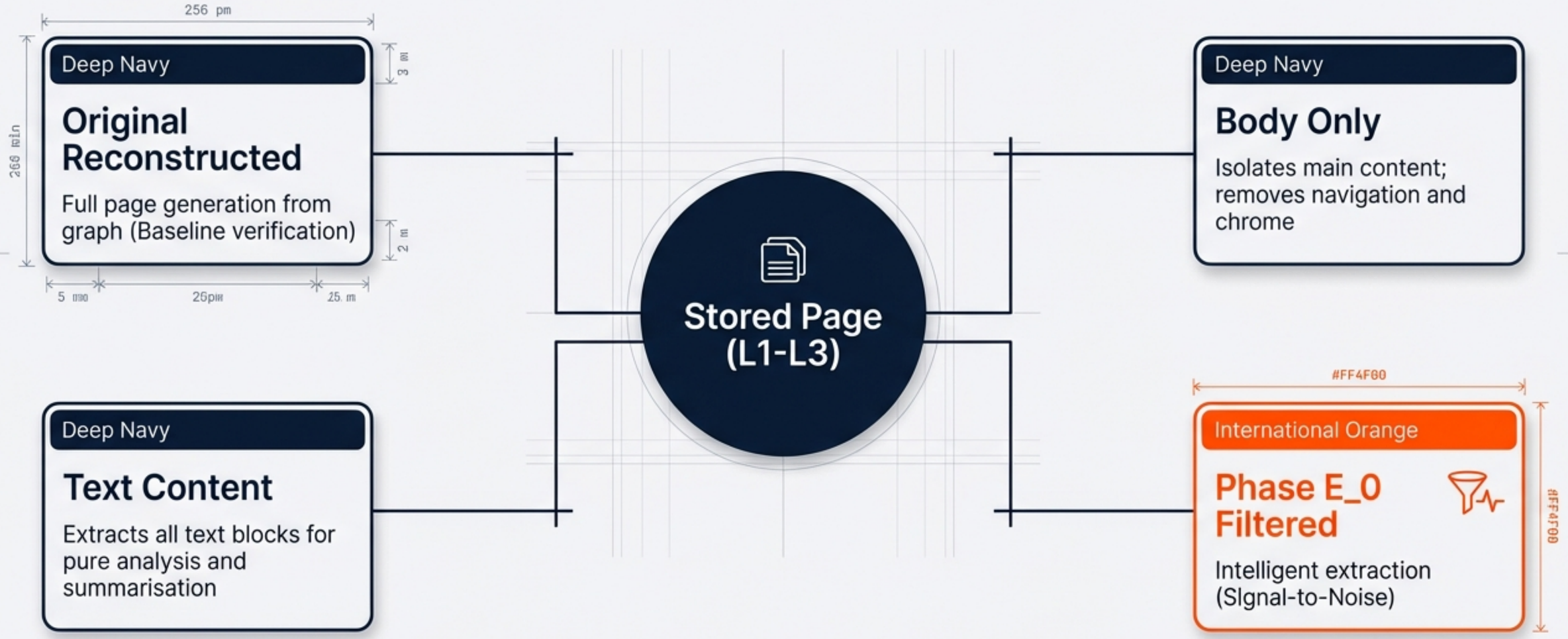
Why It Matters:

- Precise element addressing
- Structural analysis independent of visual rendering
- The ability to reconstruct the page dynamically

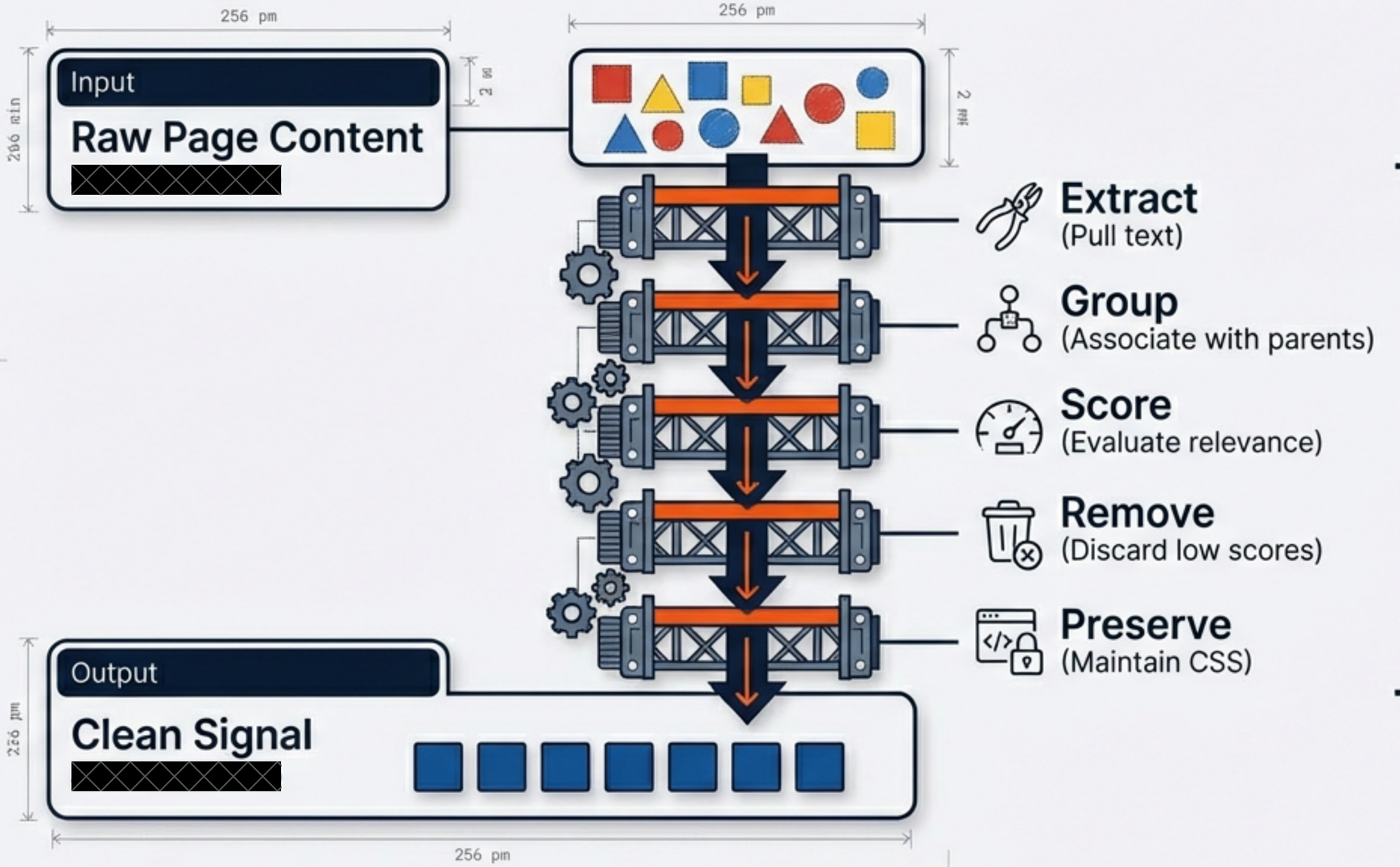
Swiss Engineering Archival

One capture, infinite views.

Once stored, we apply intelligent transformations without re-fetching from the source.



Phase E_0: The Content Filtering Engine.

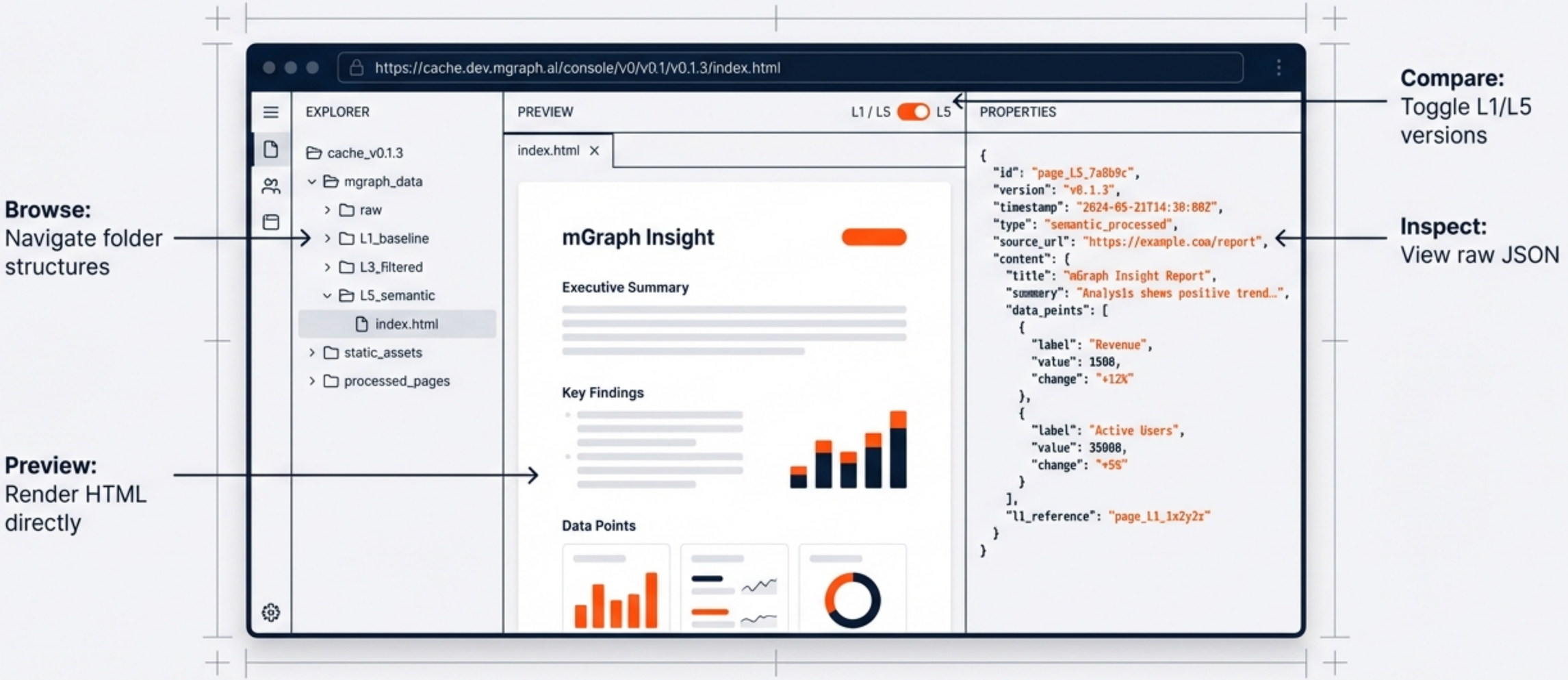


Logic: Strip clutter (ads, nav), keep structural integrity.

Swiss Engineering Archival

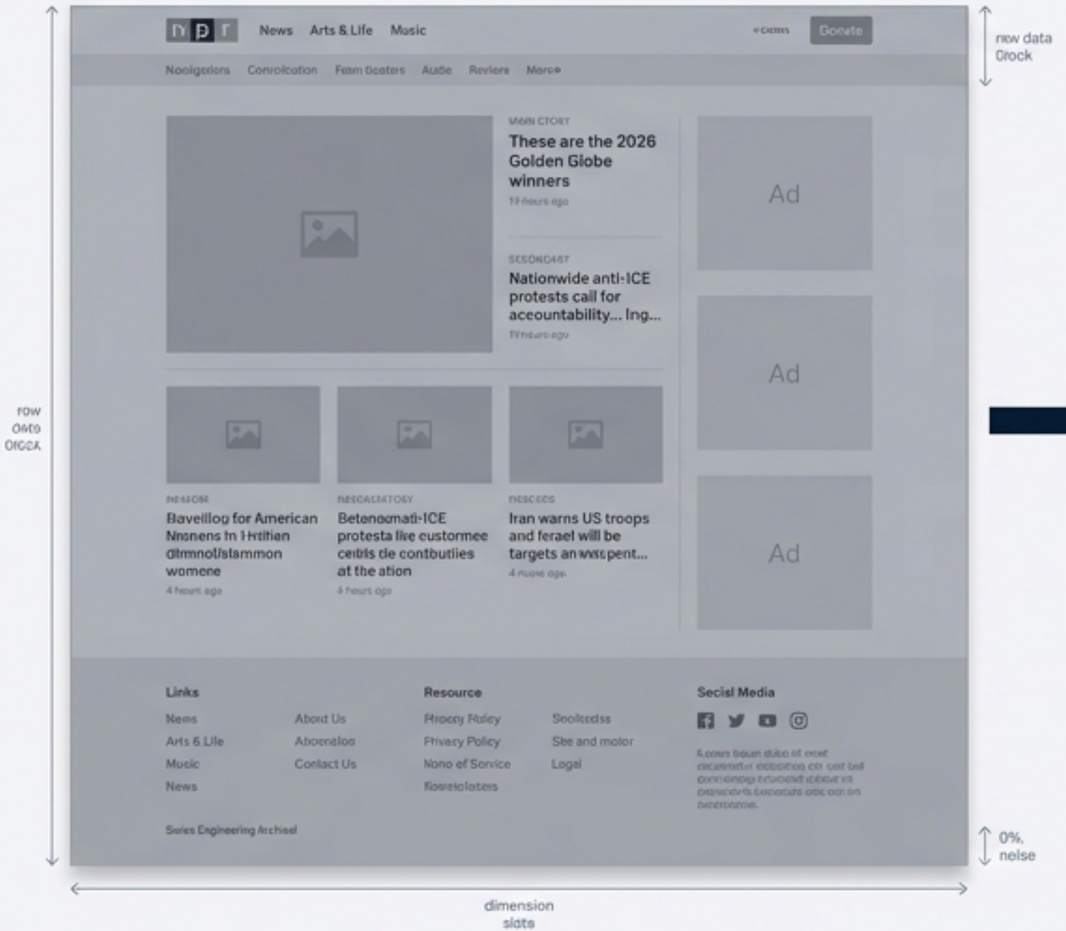
Total transparency via the Cache Browser Console.

Explore, inspect, and compare cached data with precision.



Case Study: Signal extraction on NPR.org.

Input (Raw Web)



Noise Removal

Output (Phase E_0 Filtered)

Validated across
Validated across diverse web architectures.

text.npr.org

News Headlines [redacted]

paulgraham.com

Essays/Blog [redacted]

theintercept.com

Investigative Journalism [redacted]

joelonsoftware.com

Tech Blog [redacted]

bbc.com/sport

Sports News [redacted]

docs.diniscruz.ai

Technical Documentation [redacted]

Status: Captured & Transformed Successfully

Business Value Delivered Today.

Automated Capture

Systematic ingestion of web content replaces manual effort.

Structured Storage

Enabling multiple views and queries on the same data.

Content Filtering

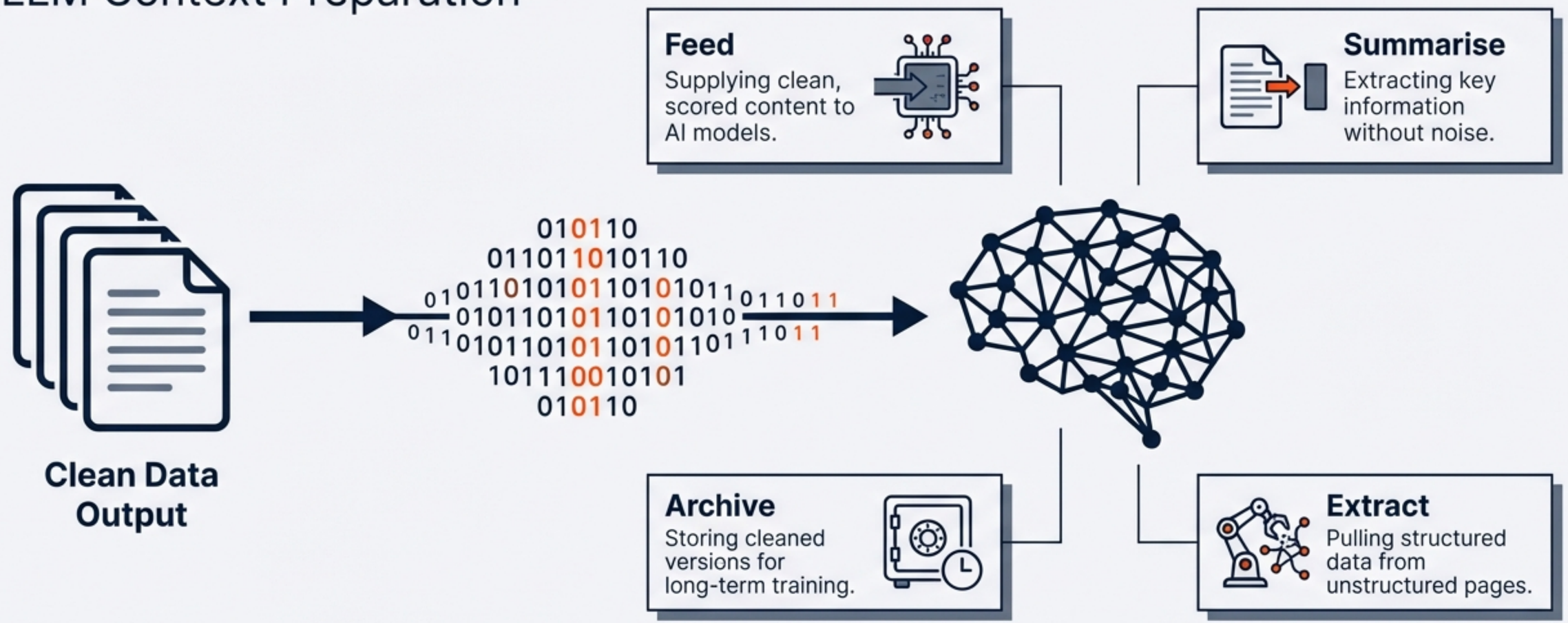
Delivers cleaner, higher-quality data for downstream use.

Visual Inspection

Immediate QA and verification through the Cache Browser.

The Foundation for AI Readiness.

LLM Context Preparation



Roadmap: From Static Capture to Intelligent Monitoring.

ML-Based Filtering

Training models on content relevance.

LLM Integration

AI-driven decisions on capture/retention.

Scheduled Capture

Automatic periodic snapshots (Time-series).

API Access

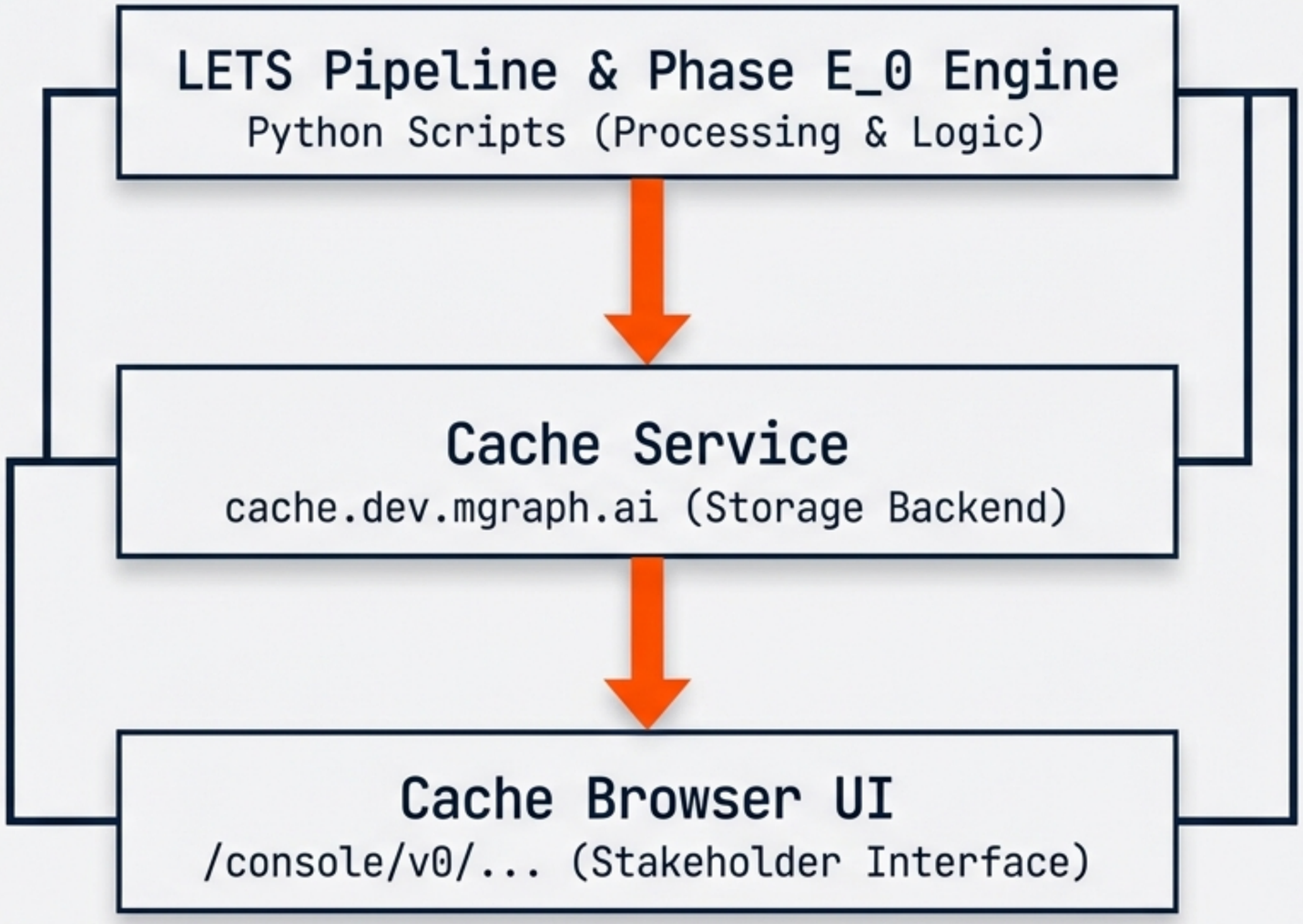
Programmatic access for enterprise tools.

PROGRESS: Present (Solid) → Future (Outlined)

Slate Grey

JetBrains Mono

Technical Infrastructure & Deployment.



Ready to Scale.

✓	**Capture:** Any public web page.
✓	**Store:** Structured, queryable cache.
✓	**Transform:** Multiple processing options.
✓	**Filter:** Intelligent Phase E_0 reduction.
✓	**Visualise:** Web-based inspection.

This infrastructure provides the foundation for automated content processing, AI-ready data preparation, and scalable web archival.