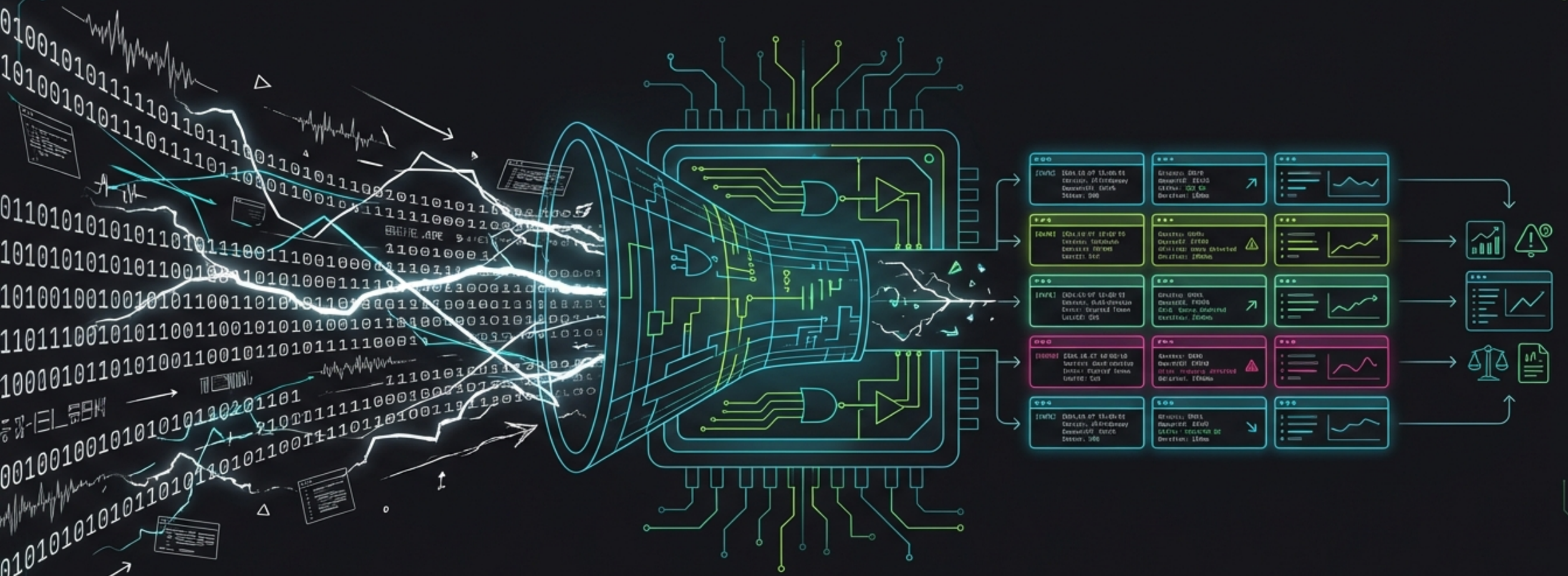
LogFlow Pro: Bridging Raw Logs & Engineering Insights

A structured management layer for AWS CloudWatch Logs.



JetBrains Mono

Product Launch & Technical Deep Dive | v1.0

Operational Rigour for Cloud Logging

Raw CloudWatch access is often ad-hoc, repetitive, and lacks historical context.

LogFlow Pro provides a professional UI for DevOps and Support engineers to manage pre-defined query libraries.

Built on a 'Stateless Backend, Stateful Frontend' architecture to ensure maximum responsiveness.

Centralised Library

Define queries once, reuse indefinitely.

Smart Time

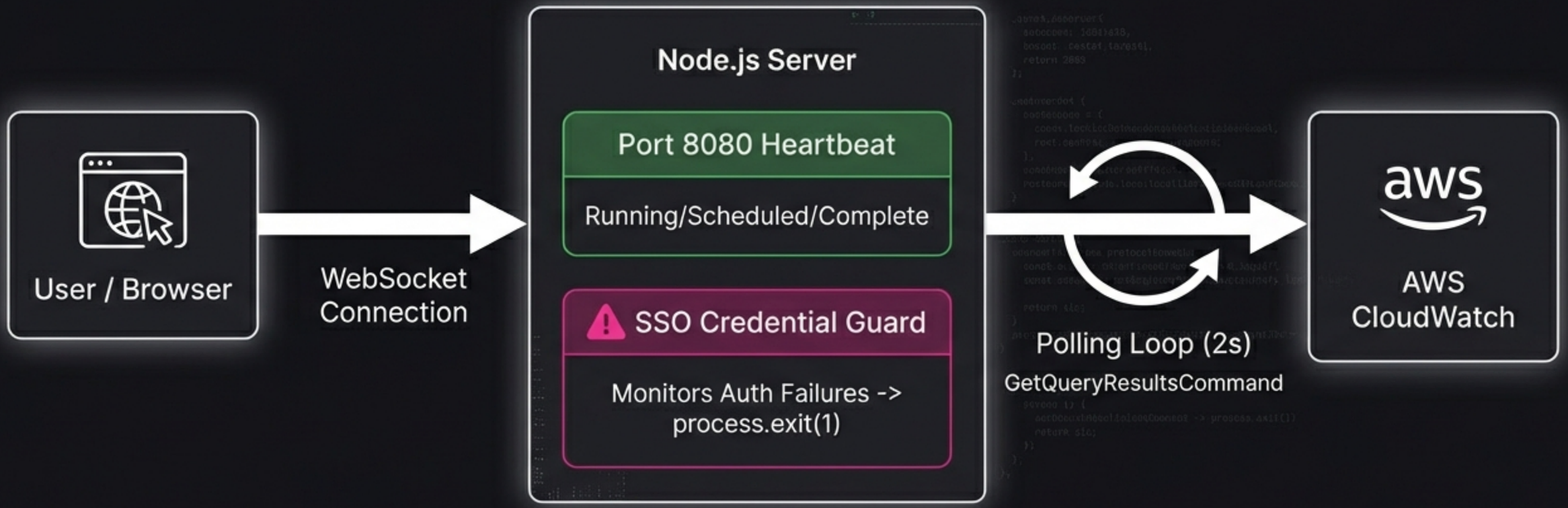
Natural language processing for temporal inputs.

Persistent History

Snapshot results to prevent redundant AWS costs.

Technical Architecture & Data Flow

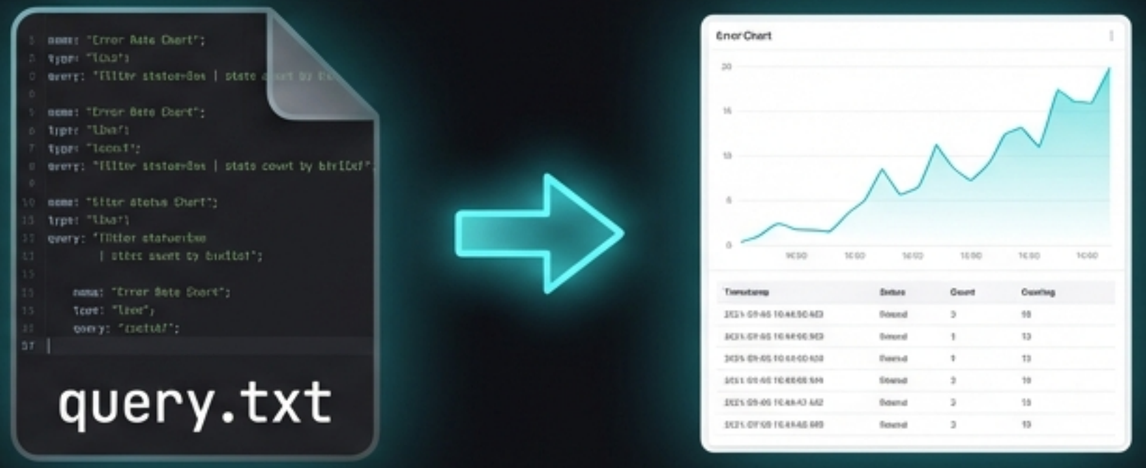
in JetBrains Mono



Configuration as Code: The DSL

LogFlow Pro utilises a file-based Domain Specific Language (DSL).

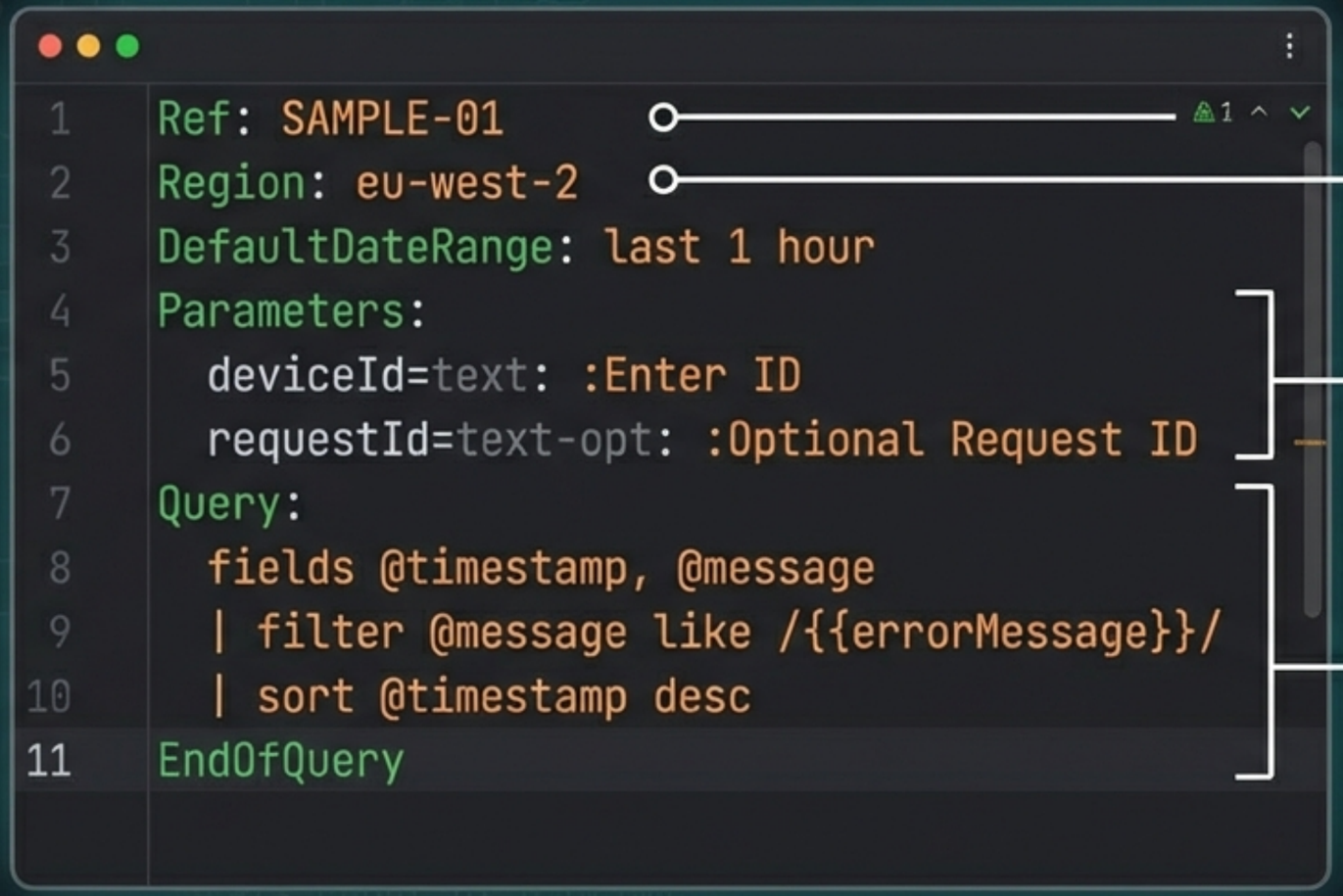
Analysis tools are defined as simple `.txt` files located in the `/queries` directory.



Edit in text, render in UI.

Version control your diagnostic logic just like application code.

Anatomy of a Query File



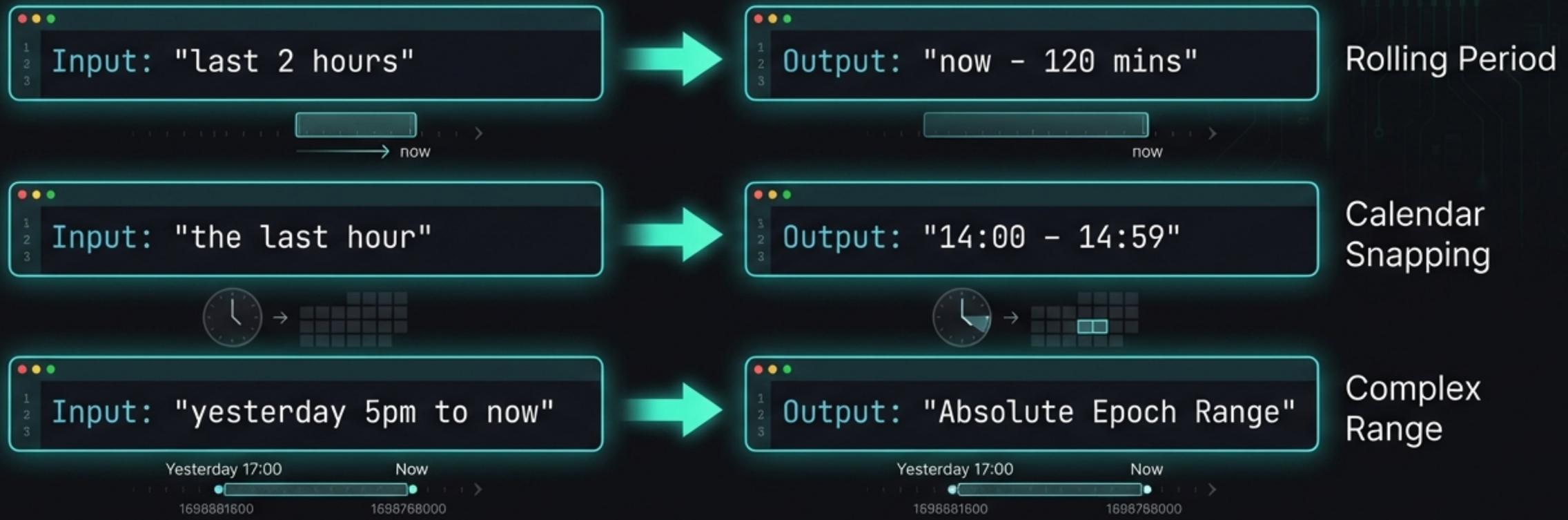
Metadata: Defines identity and AWS target.

Parameters: ‘text’ is Mandatory. ‘text-opt’ is Optional (triggers line omission if blank).

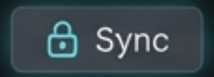
SQL Block: Standard CloudWatch syntax with {{placeholder}} injection.

Smart Time & Natural Language Processing

The “SmartRangeParser.js” engine unifies human inputs with Unix epochs.



UI Integration



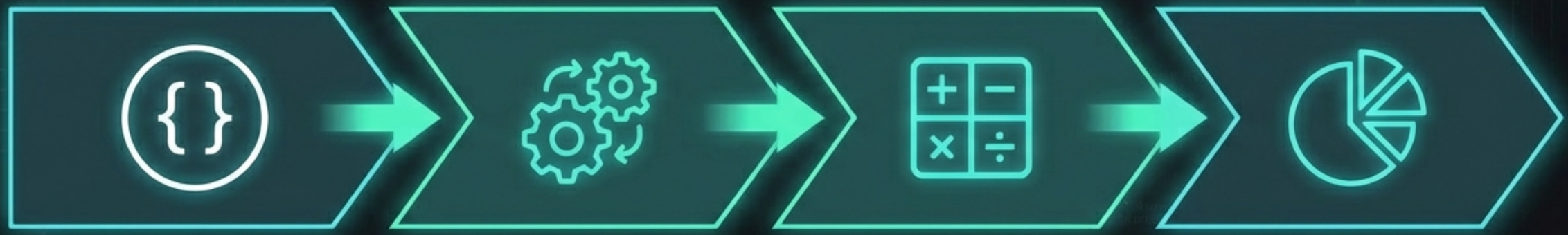
- **Sync Button:** Locks calculated range into absolute timestamps.



- **Chronological Shifting:** Jumps window forward/backward by duration.

Local Aggregation Engine

Transforming flat log rows into dashboards without re-querying AWS.



Receives flat JSON from AWS.

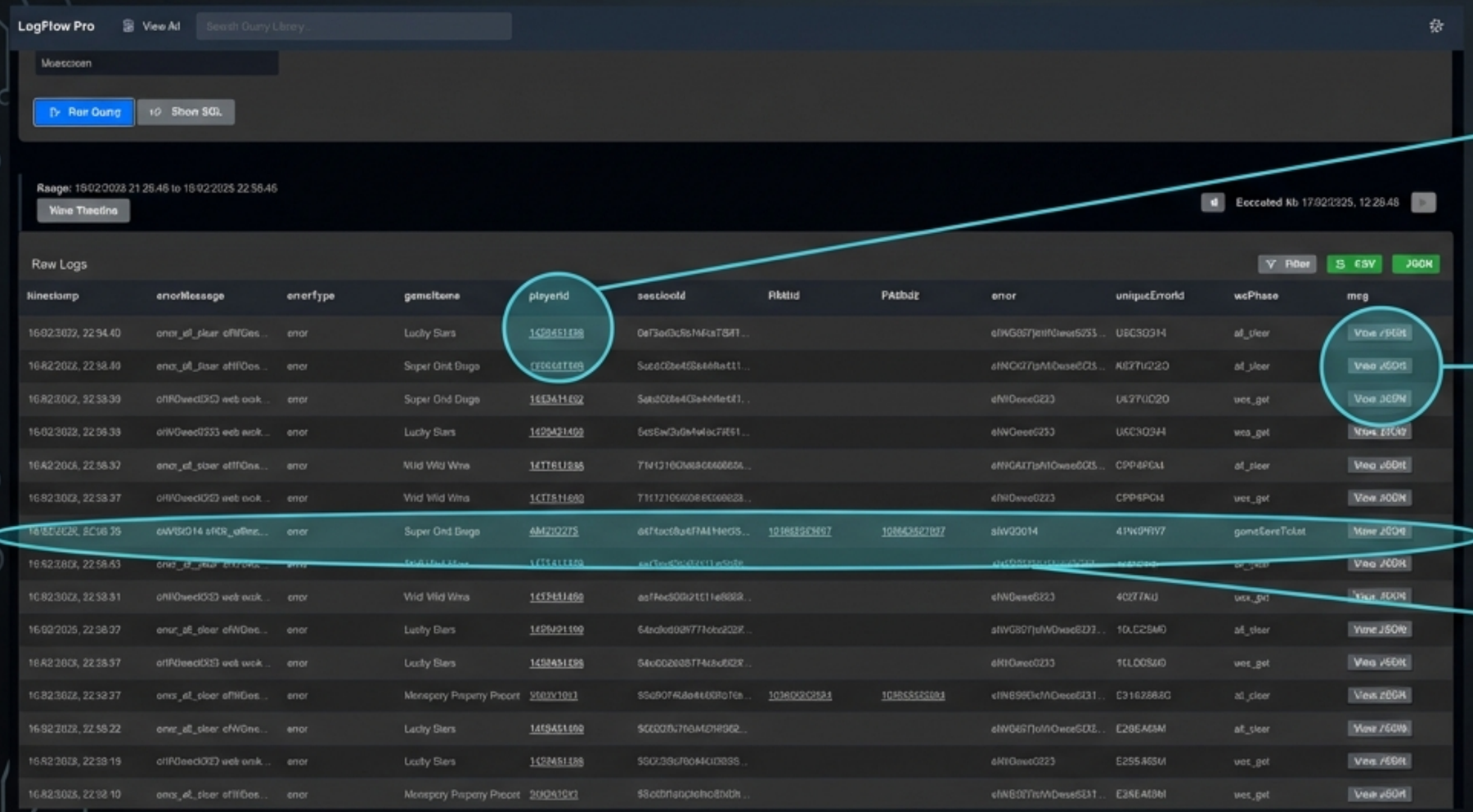
Parses [View Title] blocks from query file.

Performs GroupBy and Aggregate (Count/Sum) in browser.

Draws Chart.js visualisations.

Zero additional AWS API costs.

Interactive Data & Drill-Downs

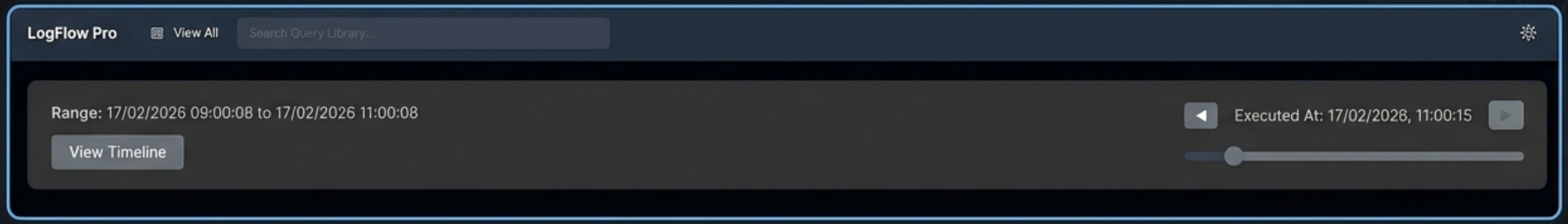


Deep Linking (HREF): Configured columns link to external tools (Splunk, Back Office).

JSON Inspector: Launches JsonTree.js for payload inspection.

Context Aware: Links use row data like .../users/{{playerId}}.

Historical Snapshots & Timeline



Persistence

Executions saved as `JSON` in `/results` with metadata and payload.

Timeline Viewer

SVG **modal** renders historical runs as interactive bars.

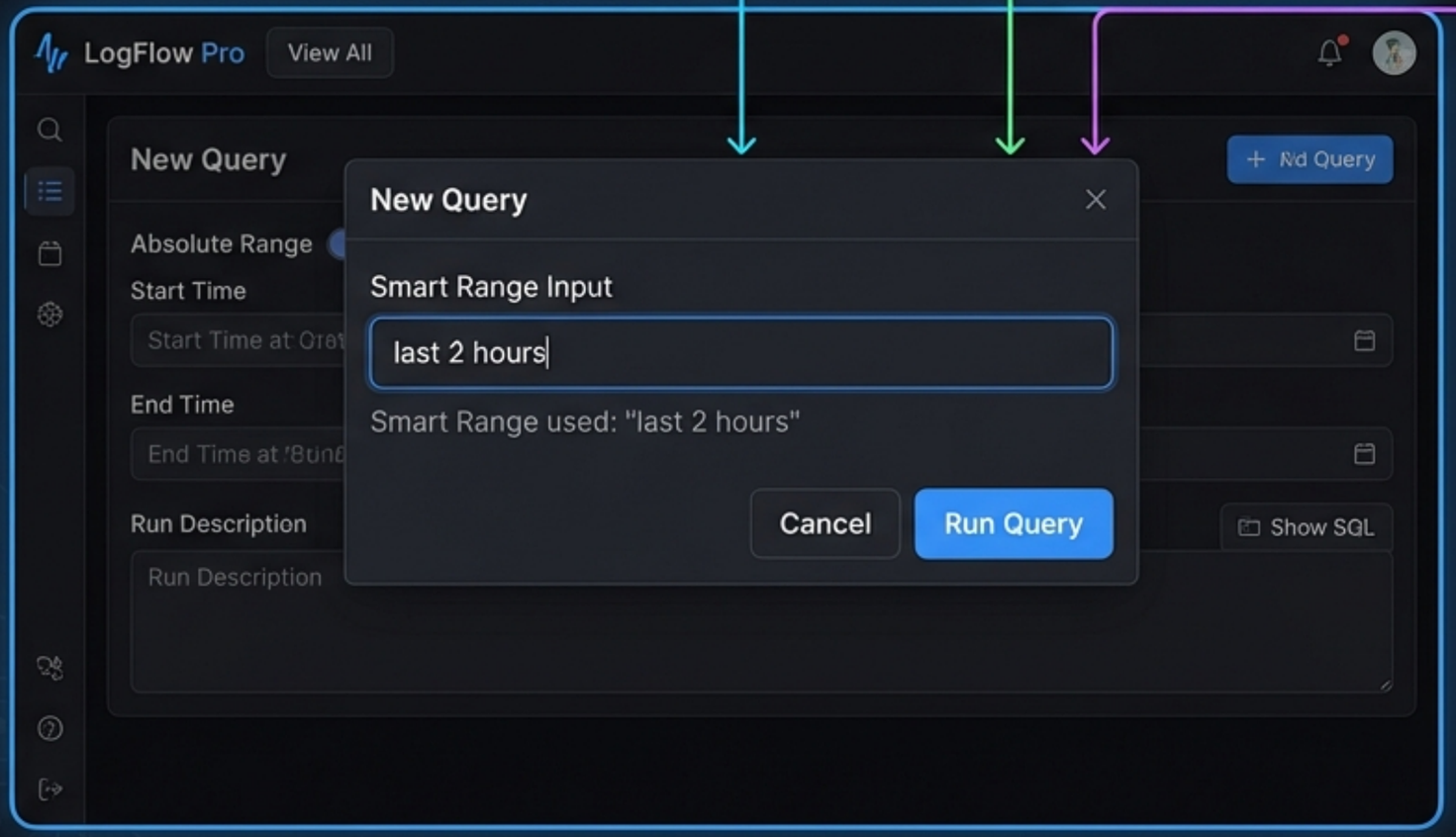
Sticky Header

Keeps temporal context visible while scrolling.

Time travel through previous diagnostics.

Deep Linking & State Management

http://logflow/query?ref=DFE-01&range=last+2h&param=error



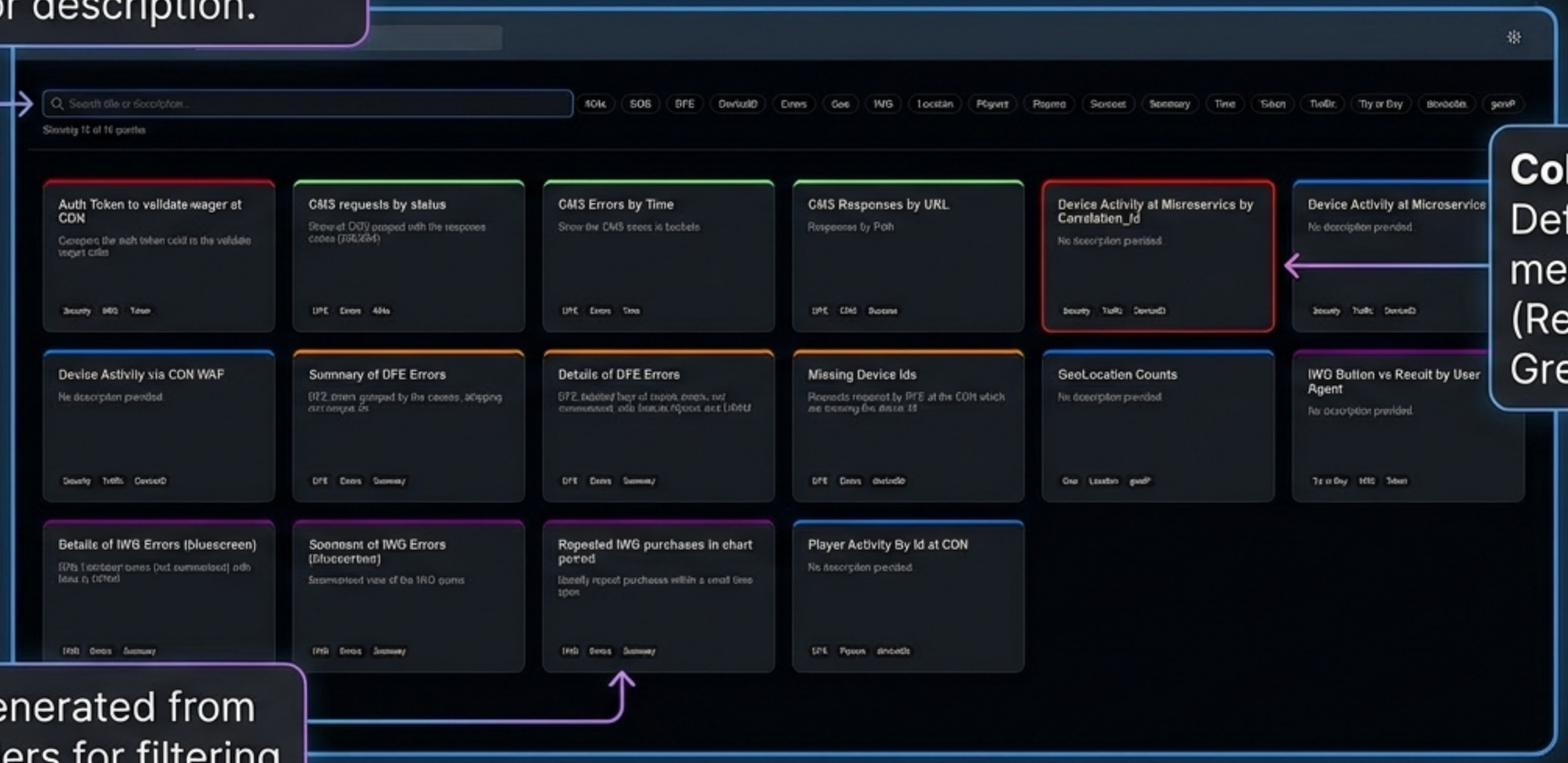
📄 **URL as Source of Truth:** Updates via pushState on every change.

🔗 **Collaboration:** Share 'frozen' analysis states by copying the link.

🌓 **Theming:** Supports persistent Light and Dark modes.

UI Tour: The Query Gallery

Search: Real-time filtering by title or description.

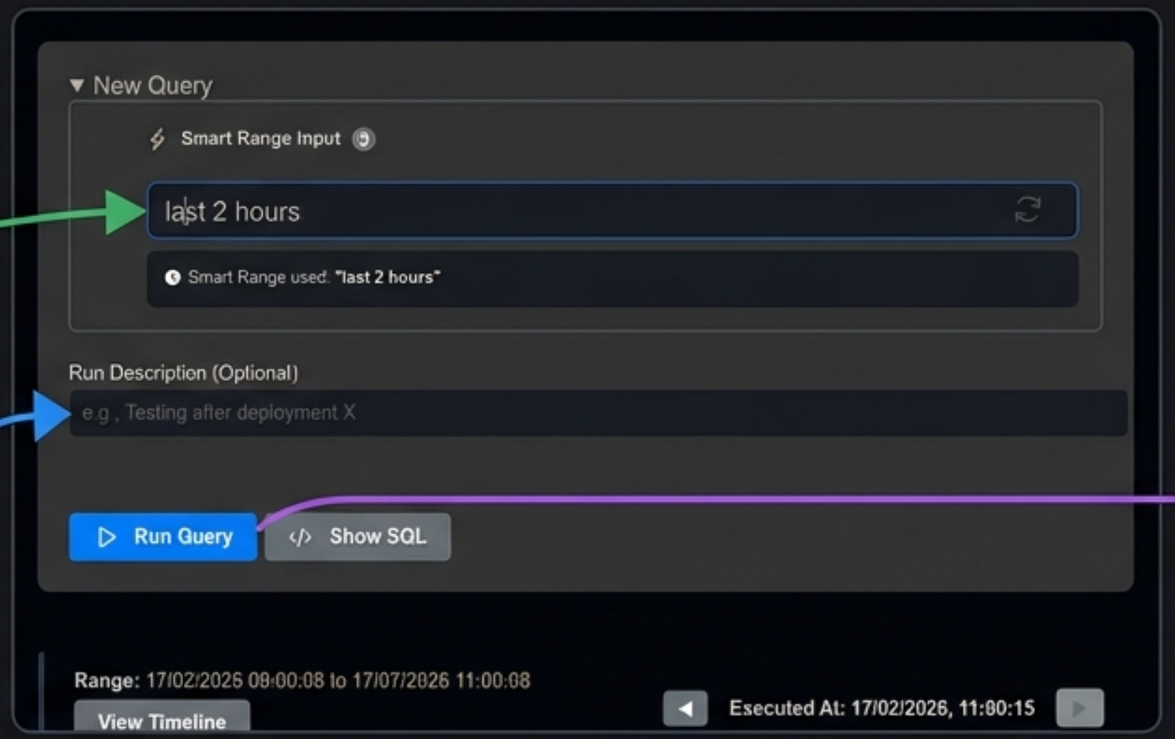


Colour Coding: Defined in metadata (Red=Error, Green=Success)

Tags: Generated from file headers for filtering.

UI Tour: Execution & Parameters

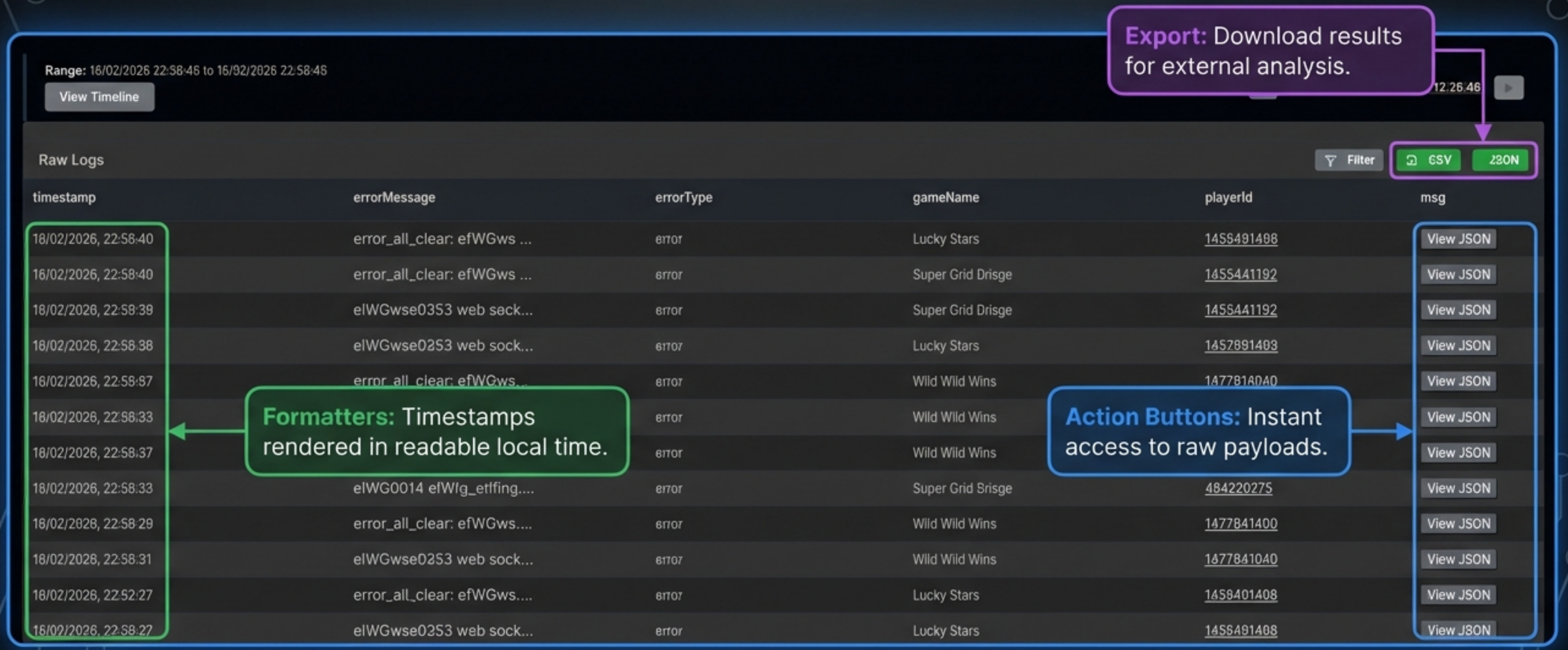
Smart Range Input:
Accepts natural language.



Dynamic Forms:
Inputs generated directly from the DSL.

WebSocket Status:
Real-time feedback.

UI Tour: Visualisation & Formats



Implementation & Stack

Dependencies

Runtime: Node.js (v18.x)

Core: Express & ws (WebSockets)

AWS: @aws-sdk/client-cloudwatch-logs

Frontend: Chart.js, Phosphor Icons, JsonTree.js

Configuration

```
> git clone repo
> npm install
> aws sso login
> node app.js
[Server] Listening on port 3001
[Config] Queries loaded from ./queries
[Config] Snapshots in ./results
>
```

Summary

Structured

Ad-hoc searches → Managed Library

Intelligent

NLP Time Parsing & Smart Params

Efficient

Client-side Aggregation & Persistence

Professional

Polished UI for DevOps Workflows

LogFlow Pro transforms CloudWatch from a data store into an engineering asset.