

A close-up photograph of a person's hands holding a silver stopwatch. The hands are positioned as if ready to start or stop the timer. The background is dark, making the silver of the stopwatch stand out.

# Legion Performance Analytics

May 2022

# Legion Performance Analytics

- Introduction
- Requirements
- Architecture
- Roadmap

# Introduction

## Definition

- Performance : quantified non-functional requirement
- **The test of the machine is the satisfaction it gives you.**— Robert M. Pirsig, *Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values*
- Performance for software
  - Latency
    - ex.: frame time, reaction time, replication time, load time
  - Stability
    - ex.: MTBF, crashes, error logs, memory use
  - Satisfaction
    - ex.: retention, engagement, biometrics, surveys

# Introduction

## Stages of data storage

- in-app: buffered streams of structured events
- **Data Lake**
  - write-friendly format
  - shallow index
- **Data Lakehouse**
  - deeply indexed
  - just-in-time
  - distributed query engine



# Introduction

## Levels of analytics

- in-app: HUD, adjust verbosity of telemetry
- Process stats: MTBF, max memory use, run time
- Deep inspection of a single session: Mem use of every asset, flame graphs for entire process
- Aggregates of high-frequency events over many sessions: Dashboards, Heat maps



# **Legion Performance Analytics**

- Introduction
- Requirements and implications
- Architecture
- Roadmap

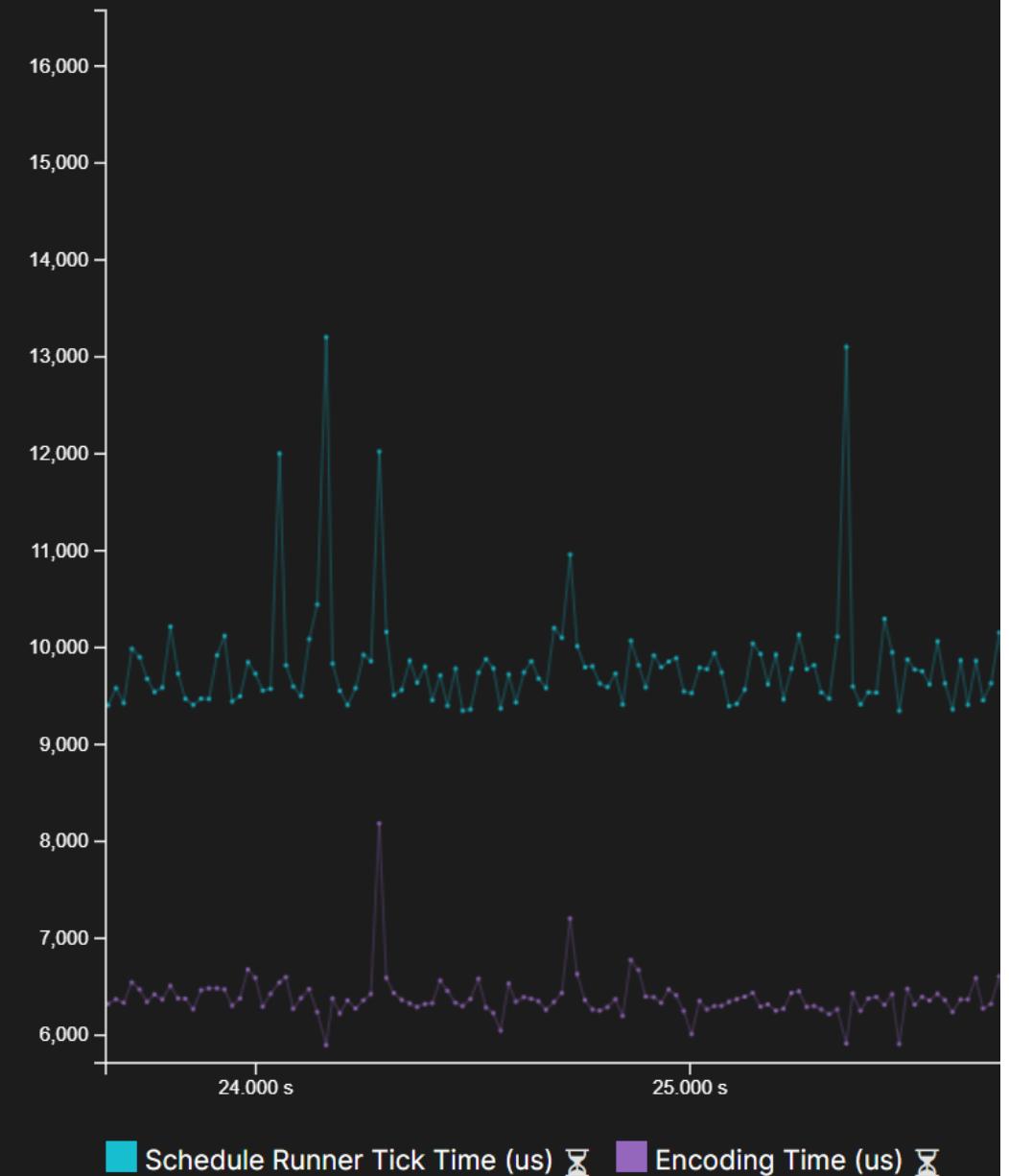
# Requirements

## Frequency of events

- High Frequency: thousands of events per frame
  - begin/end function call: ~40 ns per event
  - begin/end asset-specific scope
  - memory alloc/free



Choose metric ▾



# Requirements

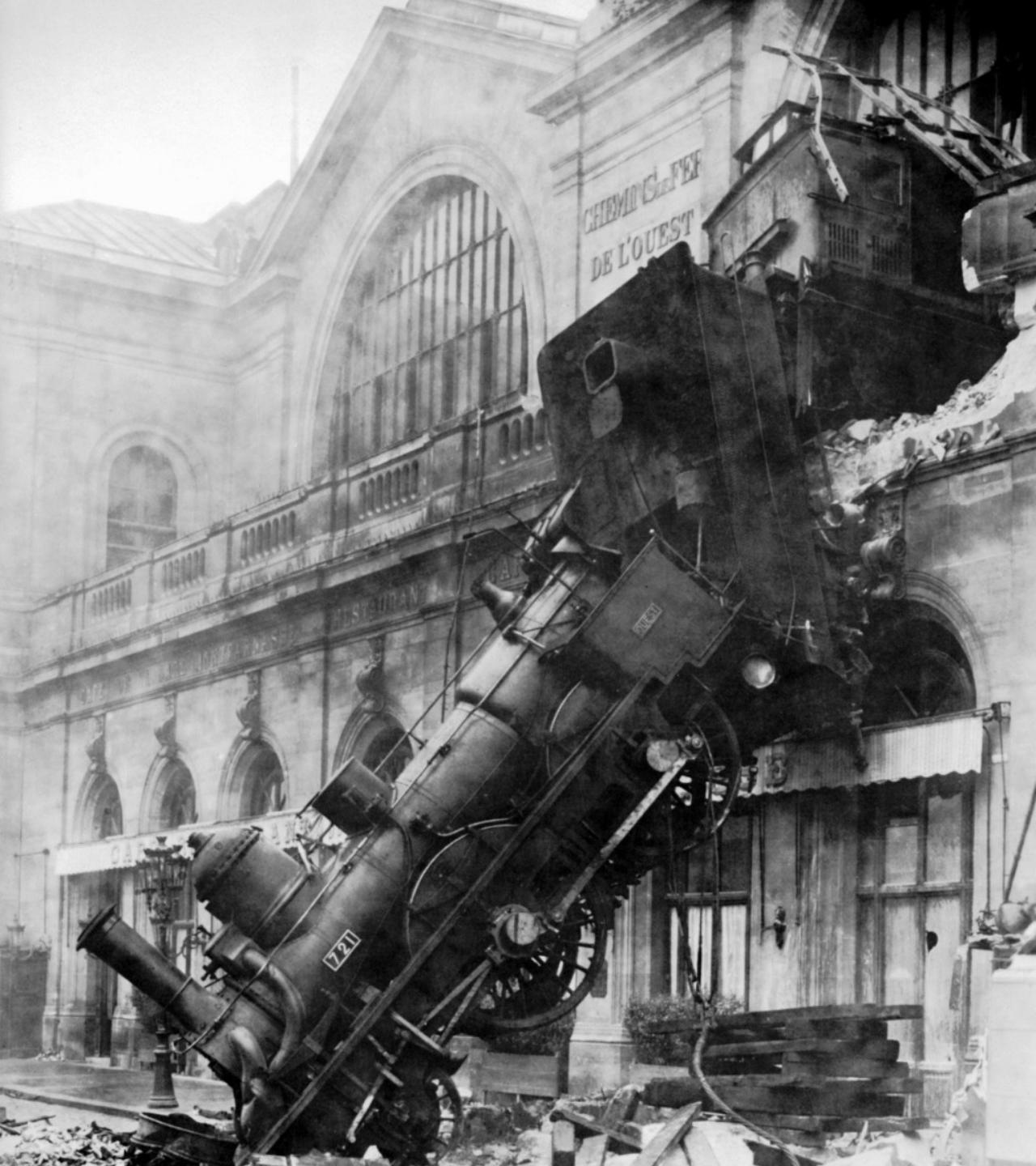
## Frequency of events

- Frame metrics
  - frame time, engine time, render sync
  - player health, #npcs
  - process memory allocated/available
  - i.e. mostly time series

# Requirements

## Frequency of events

- Logs
  - begin/end app state (world loaded, in-play, matchmaking)
  - warnings
  - crashes with callstack



# Requirements

## Generic and Extensible event stream format

- Open/Closed principle: open for extension, but closed for modification
- Adding a feature-specific event should have no impact on ingestion pipeline
- No magic: specific reports/views depend on the presence of specific events
  - tagging of streams to advertise the purpose/suitability
  - i.e. dynamic duck typing
- not limited to time series

# Requirements

## Generic and Extensible event stream format

### Performance characteristics

- write-friendly
  - most work is done with memcpy
  - important size optim: object references
- ingest-friendly
  - store without parsing whole block
  - compressed payload is not decompressed
- generic reader
  - as generic as JSON
  - metadata to decode the writer's memory model

# Requirements

## Understanding distributed applications

- Processes are part of a tree
- Across machines & architectures



# Requirements

5 views to rule over all data

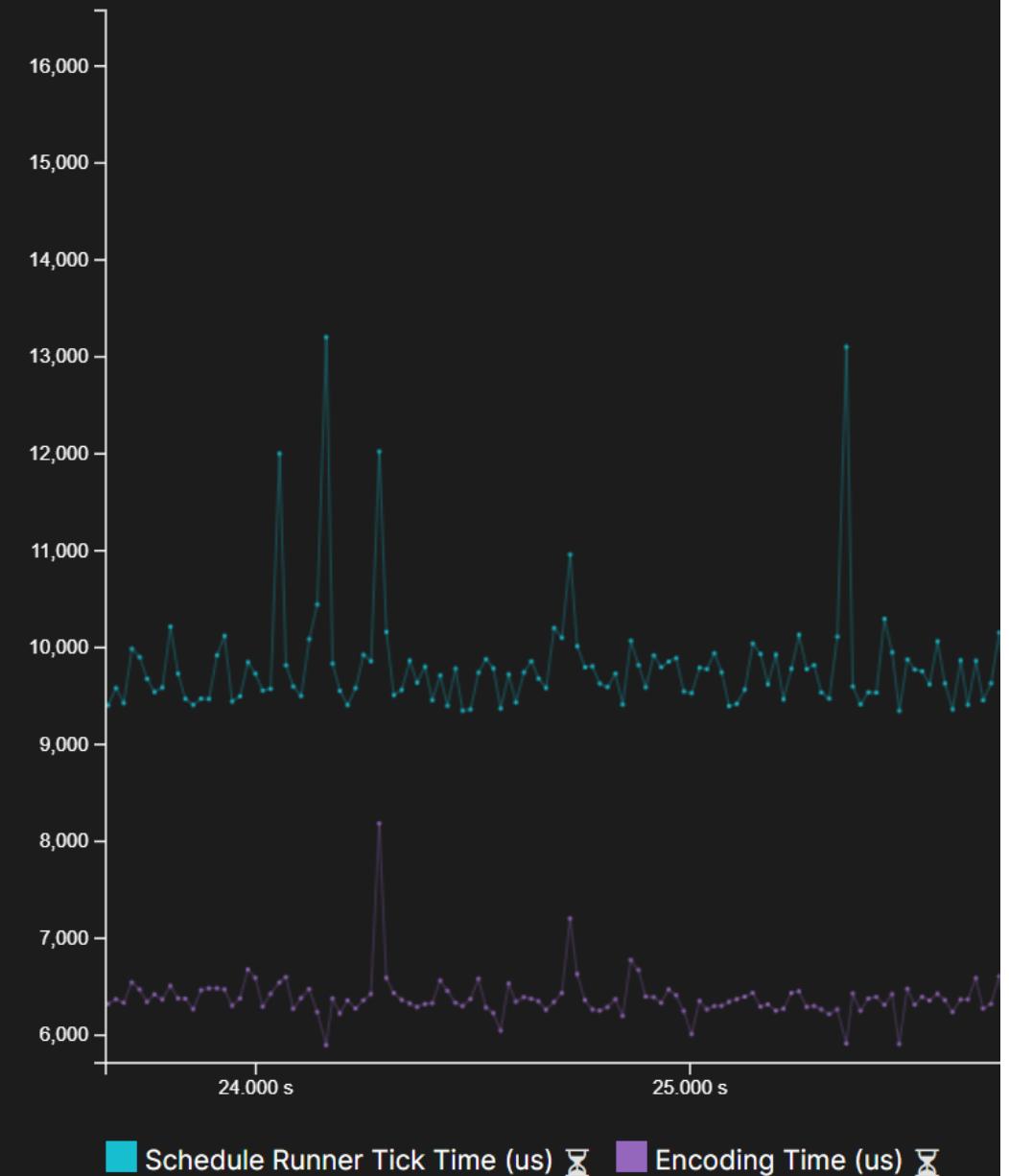
## List / Table / Search

- recent processes
- top crashes
- cpu budget report

ome

OWNER	NAME	TYPE	STATUS	ICON	LAST RUN	OPTIONS
LL-00274597	lsc.exe	Process	Debug	💻	18h 55m 07s	⋮
LL-00497867	editor-srv.exe	Process	Release	💻	6min ago 7h 44m 06s	⋮
LL-00274597	potato.exe	Process	Debug	💻	7min ago 11h 55m 05s	⋮
LL-00497867	cupcake.exe	Process	Debug	🎮	7min ago 13h 11m 04s	⋮
LL-00497867	croissant.exe	Process	Release	💻	8min ago 14h 22m 03s	⋮
LL-00274597	gummybear.exe	Process	Debug	🎮	8min ago 15h 33m 02s	⋮
LL-00274597	lsc.exe	Process	Release	💻	9min ago 1h 44m 01s	⋮
LL-00274597	crepe.exe	Process	Debug	💻	10min ago 2h 55m 02s	⋮
LL-00274597	cookies.exe	Process	Debug	💻	10min ago 5h 21m 03s	⋮
LL-00497867	gaufre.exe	Process	Debug	💻	11min ago 6h 31m 04s	⋮
LL-00497867	carrotcake.exe	Process	Debug	💻	11min ago 9h 41m 05s	⋮
LL-00497867	bananabread.exe	Process	Debug	💻	12min ago 10h 51m 06s	⋮

Choose metric ▾



# Requirements

5 views to rule over all data

## Time series

- Individual frame times over time
- Player health over time
- Cohort engagement over 30 days

# Requirements

5 views to rule over all data

## Graphs & Trees

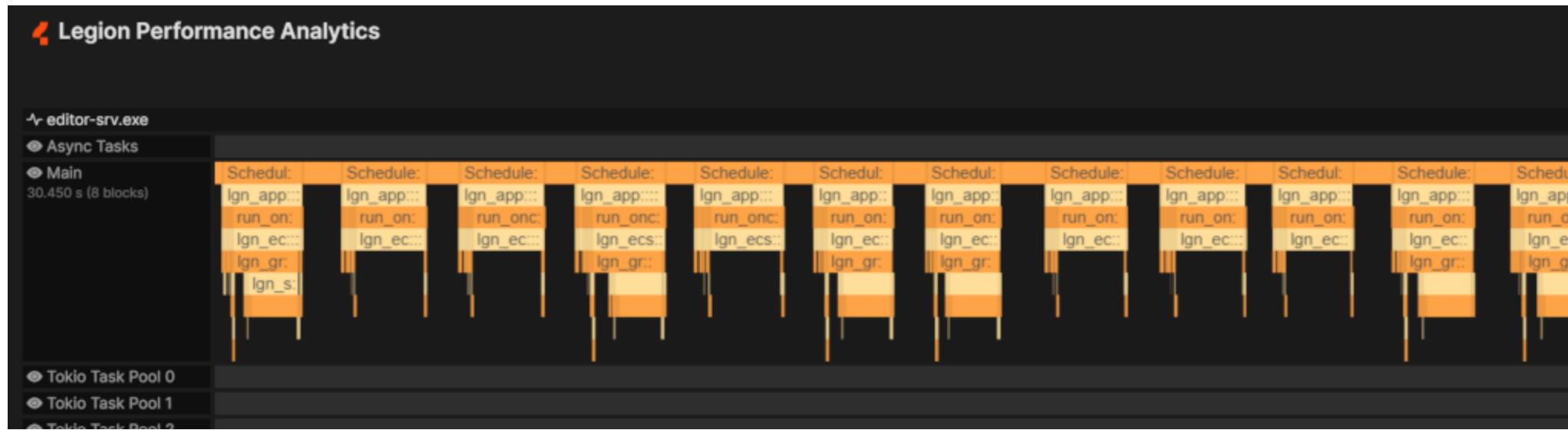
- Cumulative function call statistics
- Loaded object graph



# Requirements

5 views to rule over all data

## Timeline



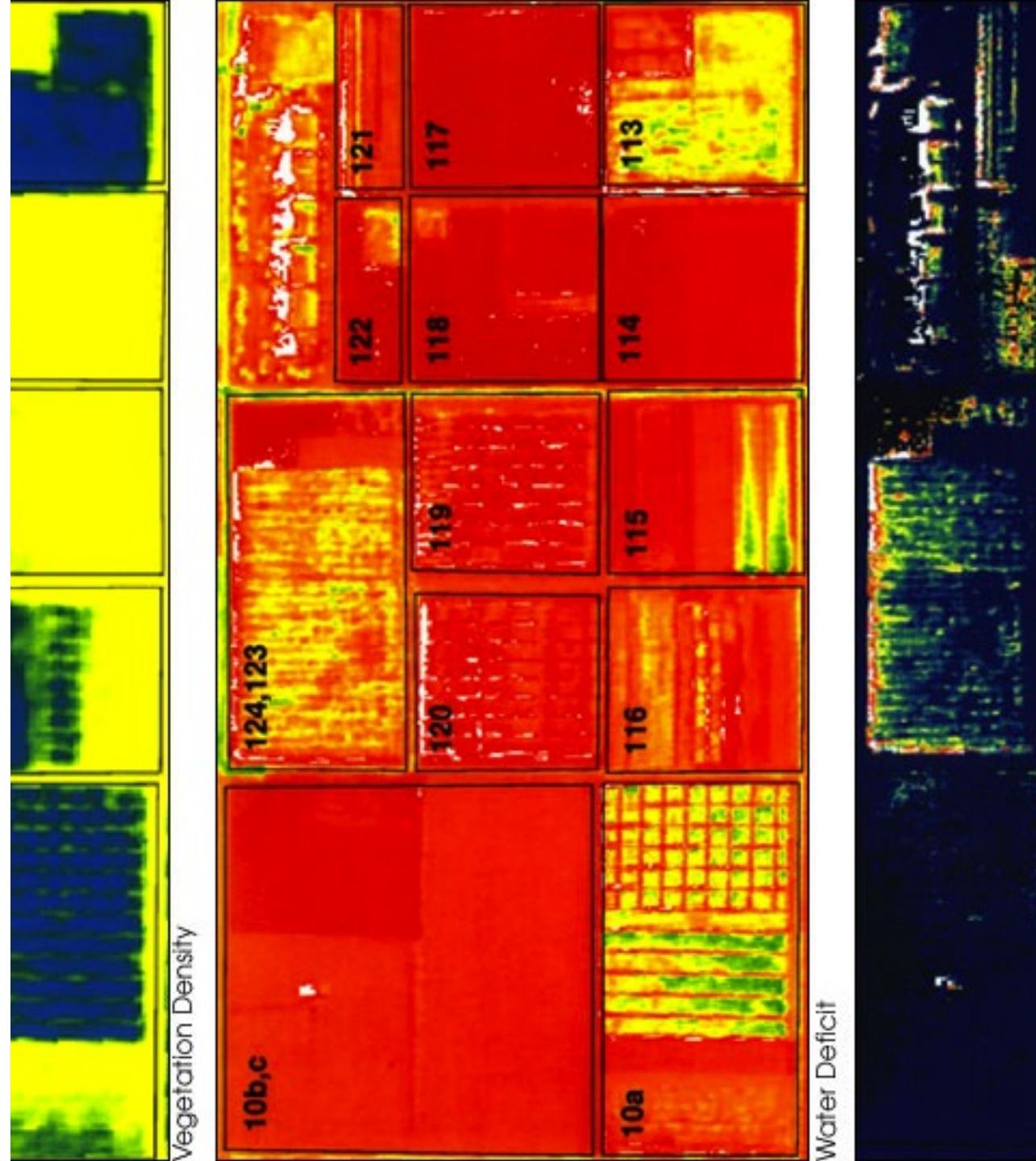
- Function calls per thread
- Audio effects per stream

# Requirements

5 views to rule over all data

## Heatmap

- death map
- geographic slow frames distribution



# Requirements

## non-requirements

- interactive debugging
- per-pixel profiling
- low-level cpu events (L1 cache miss, branch mispredictions, ...)

## not yet

- Video streaming & overlay
- cpu sampling
- context switches



# Legion Performance Analytics

- Introduction
- Requirements
- Architecture
- Roadmap

# Architecture

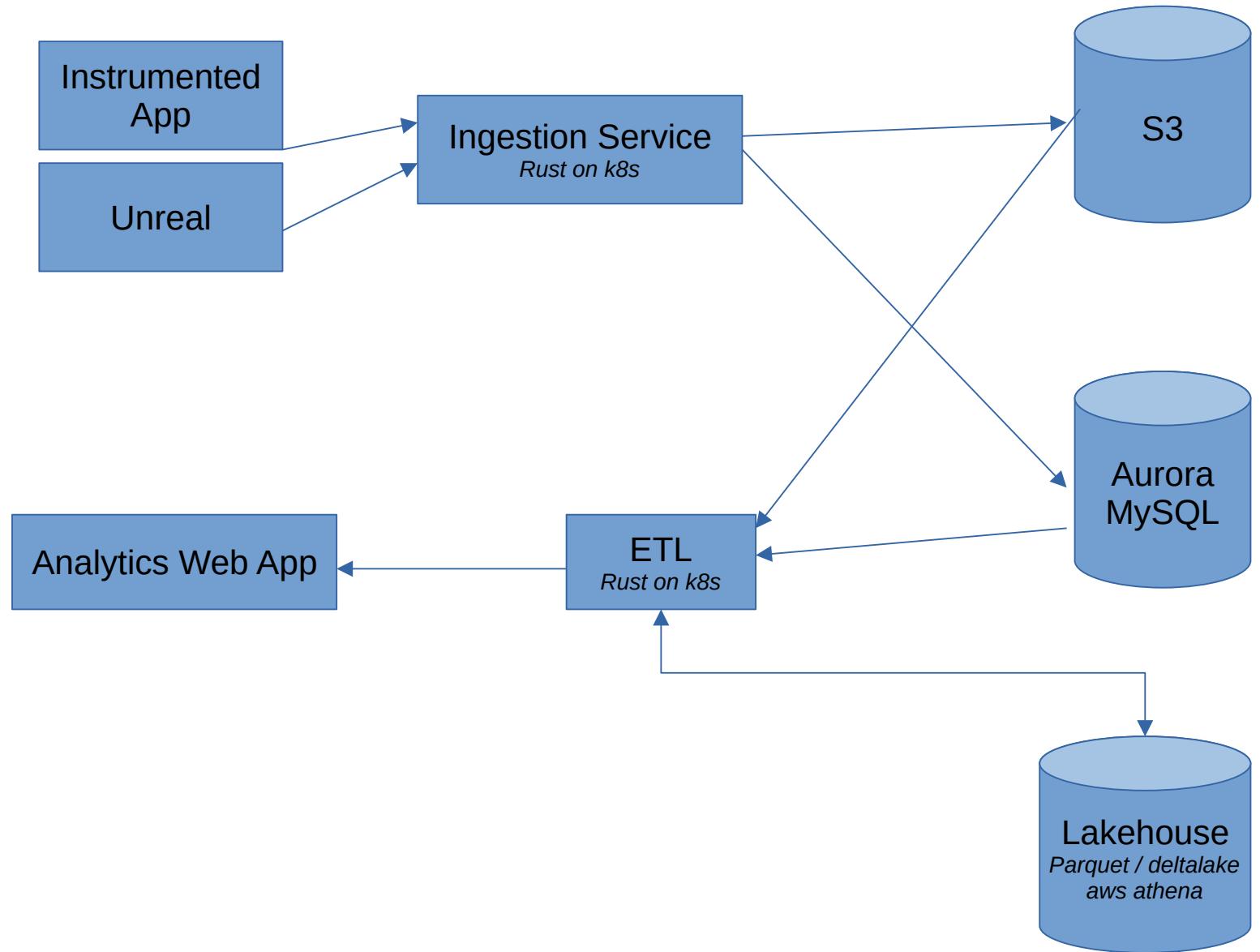
## Object hierarchy

- Process instance
  - Stream
    - Stream block
    - Event



# Architecture

## Online architecture



# Architecture

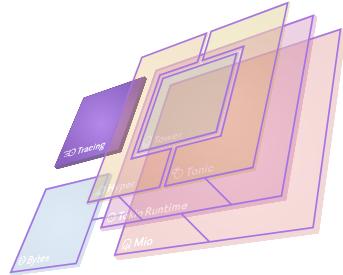
## Integration/reuse of existing solutions

Many ideas in common with `tracing` crate from the `tokio` project.

<https://docs.rs/tracing/0.1.26/tracing/>

But `Collect` trait at the center of the system is a poor fit.

Could support the interface to get visibility into crates that are already instrumented.



# Legion Performance Analytics

- Introduction
- Requirements
- Architecture
- Roadmap

# Progress 2021

## Halloween 2021

- Initial version of client telemetry library
- Local ingestion server (sqlite & files)
- CLI analytics (csv output)

## Christmas 2021

- Web Analytics client (svelte & Canvas)
- Visualization of call tree timeline



# Progress 2022

## January 2022

- Logging improvements, Timeline levels of details

## February 2022

- Cloud infra, Time series, UI redesign

## March 2022

- Time series & Timeline improvements, Async function tracing

# Roadmap

## April 2022

- Unreal telemetry module (MAD)
- Cumulative call graph improvements (Tim)
- Timeline report view (Tim)
- Process List Hierarchy (Tim)
  - ordered by last update
- Auth flow polish (Kevin)

# Roadmap

## May 2022

- Unreal telemetry module (MAD)
- Review regulatory requirements (Liem)
- Infra visibility & alerting UAT (ingestion + analytics)(Liem)
- Logs (dual mode) (Kevin)
  - Paging when not filtering
  - n-first results when there is a filter
- Lakehouse-backed timeline (MAD)

# Roadmap

## Juin 2022

- Lakehouse-backed timeline (continued) (MAD)
- Backups data lake (Liem)
- I10n/i18n (Kevin)
- API key refactor (Julien)
- IAM permissions cleanup for ingestion & analytics (Julien)
- TSC frequency calibration (MAD)

# Backlog

- Async spans stitching/LOD
- Memory tracing and analytics
- VM monitoring / Fluentd output command
- GPU profiling
- Support for Google Cloud Storage
- Landing page
- Object Graphs
  - Why is this texture loaded?

## More Backlog

- Process dashboards
  - monitoring thousands of processes
  - graphs of processes to lists of processes
- Task-based timeline & async span parenting
  - unifying thread-bound and async tasks
- Heat maps
- Real-time logs, metrics and timeline
  - block consolidation

