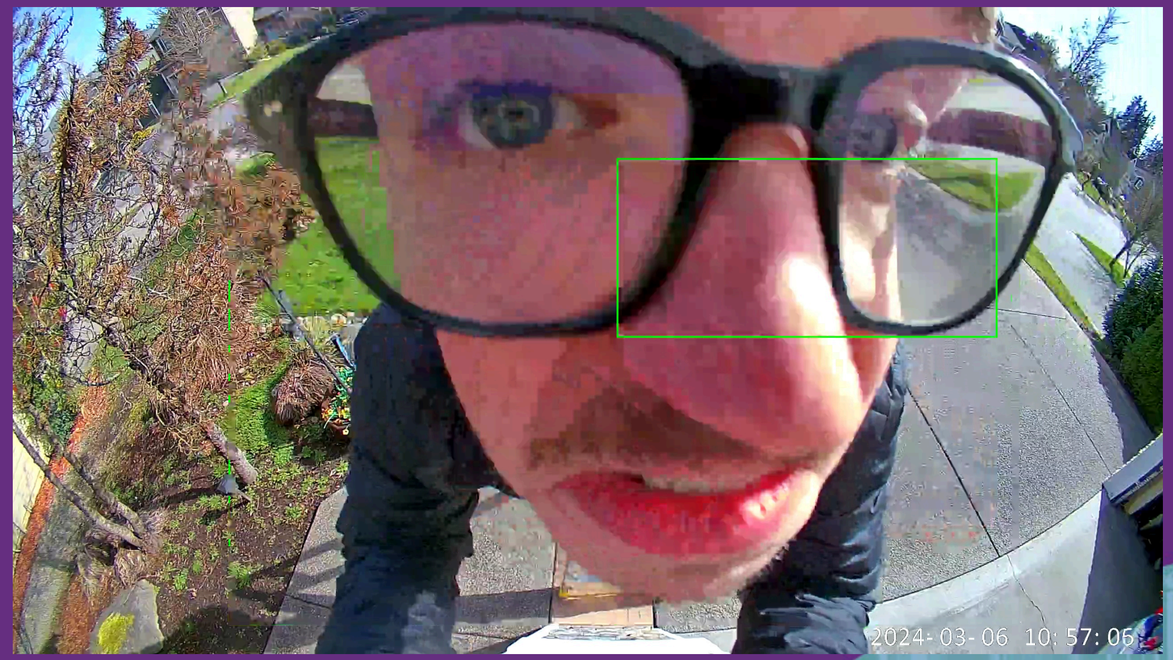
Telemetry-Driven Development

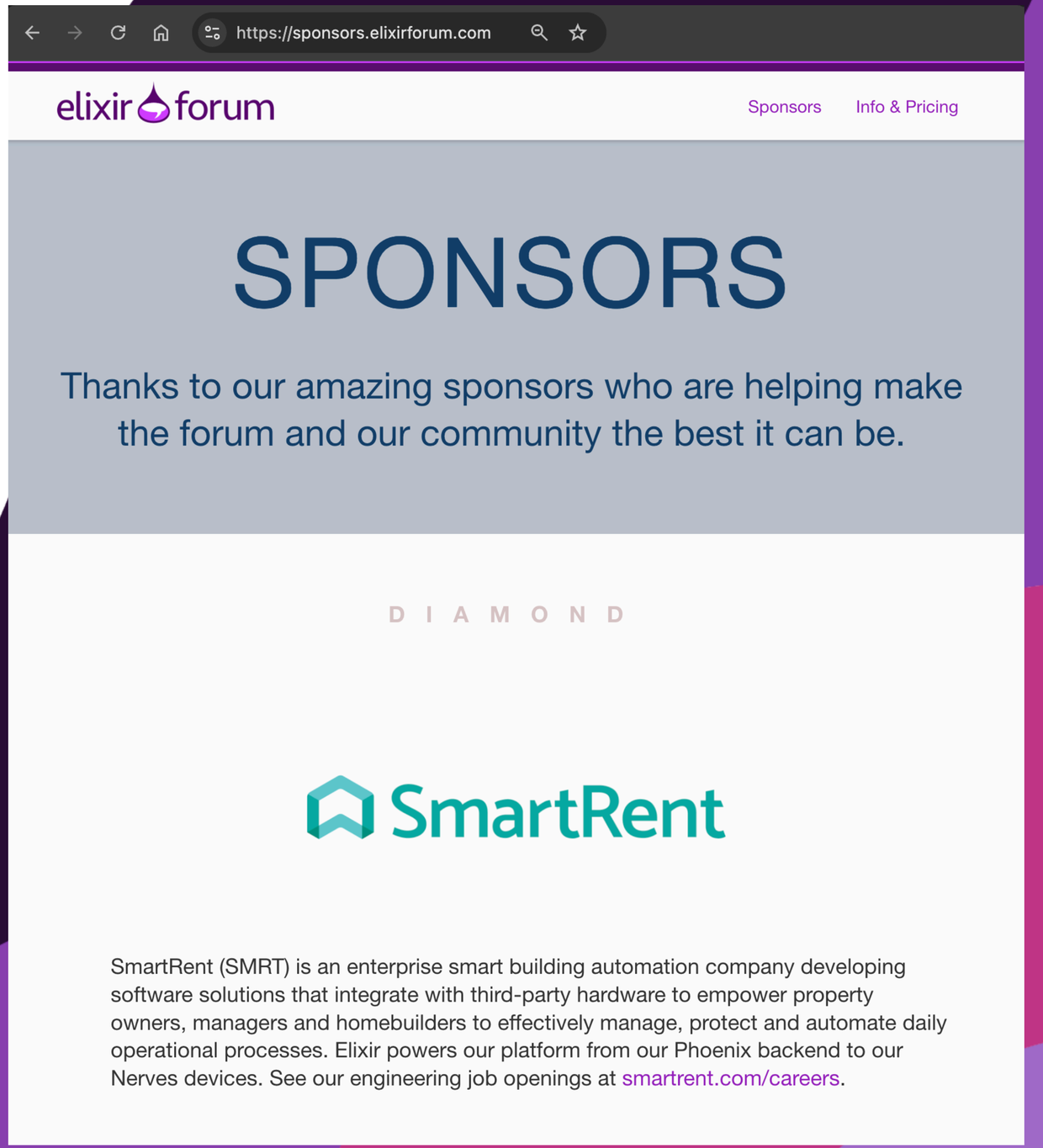
"The Purpose Of the System Is What It Does"

Noah Betzen (@SmartRent)

iex(1)> Node.list(:this)
[:"noah@nezteb.net"]

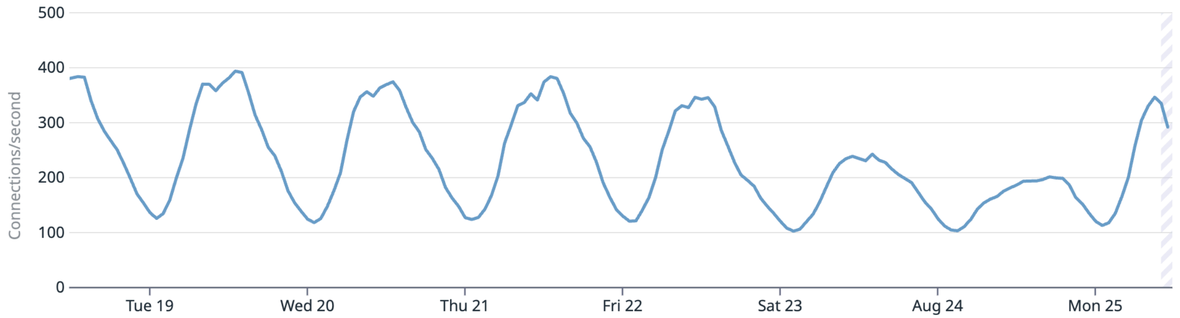


"I have approximate knowledge of many things" - Demon Cat (Adventure Time)


https://sponsors.elixirforum.com
elixir forum
Sponsors
Info & Pricing
SPONSORS
Thanks to our amazing sponsors who are helping make the forum and our community the best it can be.
DIAMOND
SmartRent
SmartRent (SMRT) is an enterprise smart building automation company developing software solutions that integrate with third-party hardware to empower property owners, managers and homebuilders to effectively manage, protect and automate daily operational processes. Elixir powers our platform from our Phoenix backend to our Nerves devices. See our engineering job openings at smartrent.com/careers.

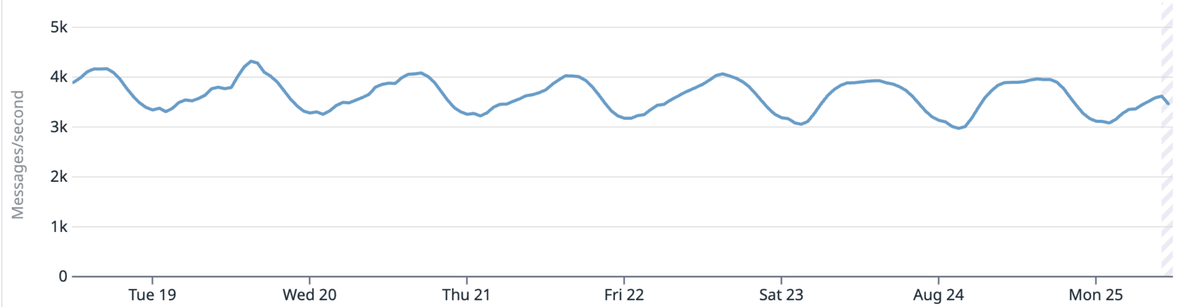
evices in prod

Load Balancer: Active Connections Per Second



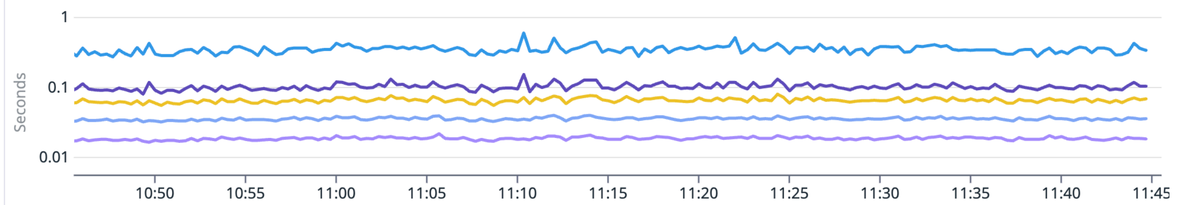
TAGS IN active_connections	AVG	MIN	MAX	SUM	VALUE
*	231 conns/s	103 conns/s	394 conns/s	39.0k conns/s	286 conns/s

Queue: Messages Sent Per Second



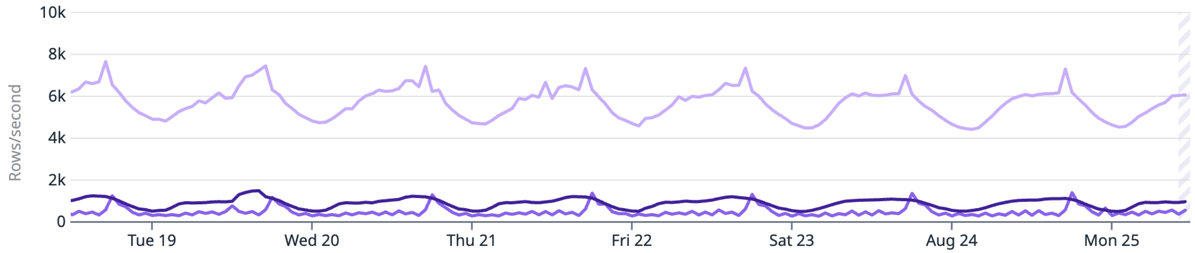
TAGS IN messages_sent	AVG	MIN	MAX	SUM	VALUE
*	3.63k msgs/s	2.98k msgs/s	4.33k msgs/s	614k msgs/s	4.18k msgs/s

Service: Request Latency 1h Past 1 Hour



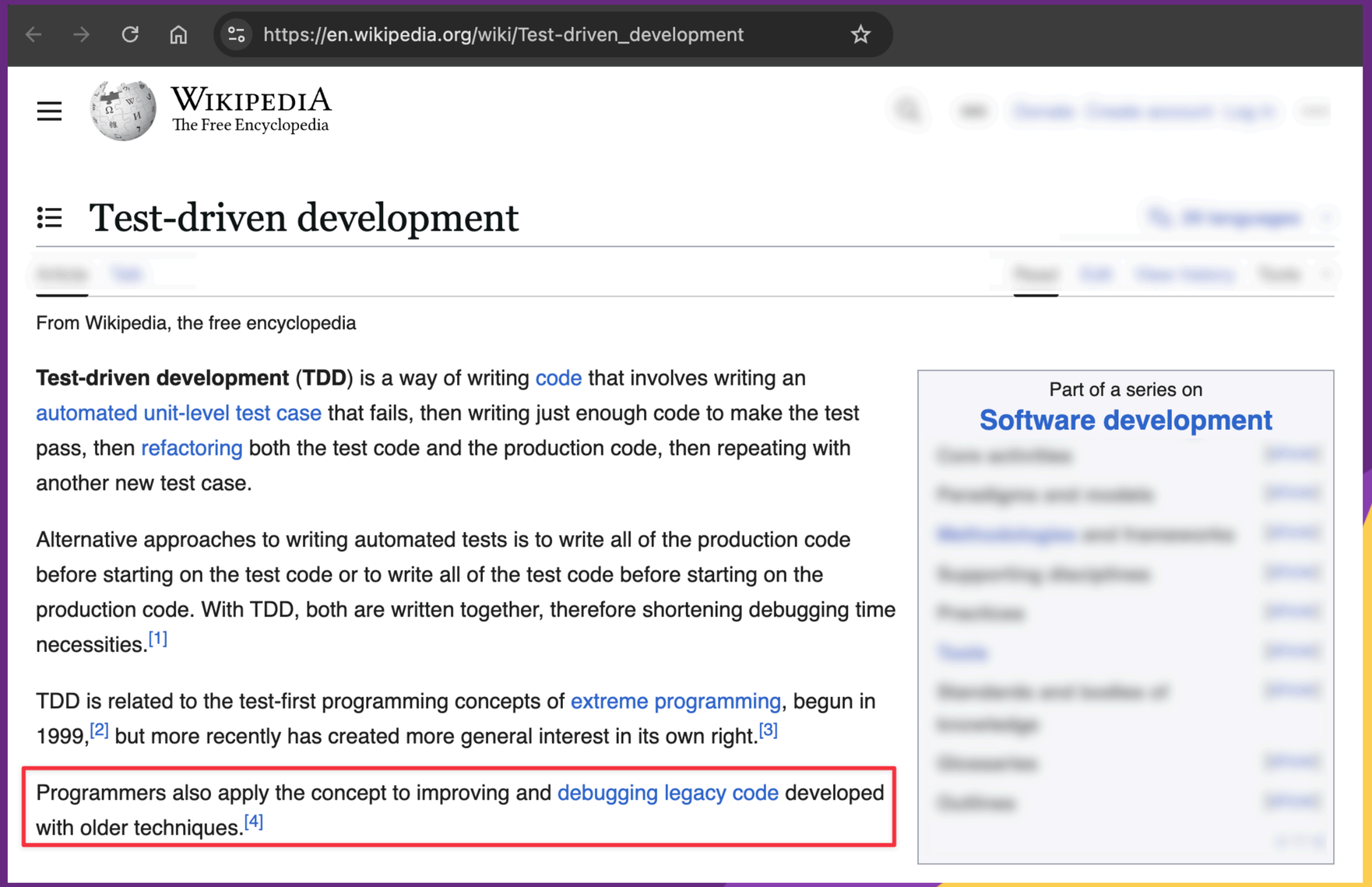
TAGS	METRIC	AVG	MIN	MAX	SUM	VALUE
*	p99	354.2 ms	276.1 ms	608.9 ms	63.41 s	348.4 ms
*	p95	103.3 ms	81.1 ms	155.6 ms	18.50 s	97.7 ms
*	p90	66.3 ms	55.9 ms	81.1 ms	11.87 s	63.3 ms
*	p75	36.0 ms	32.5 ms	40.4 ms	6.44 s	35.7 ms

Database: Inserts / Updates / Deletes

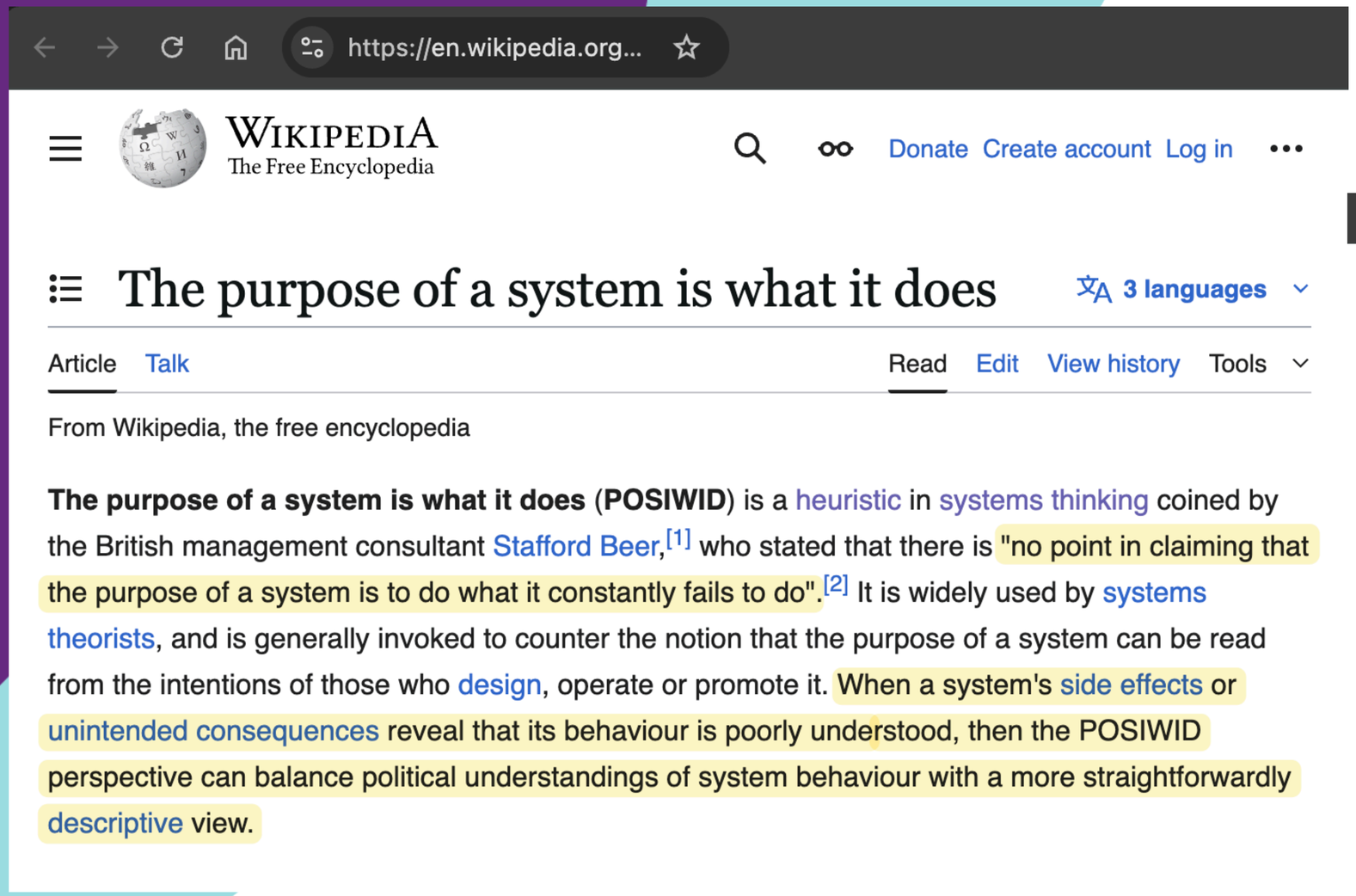


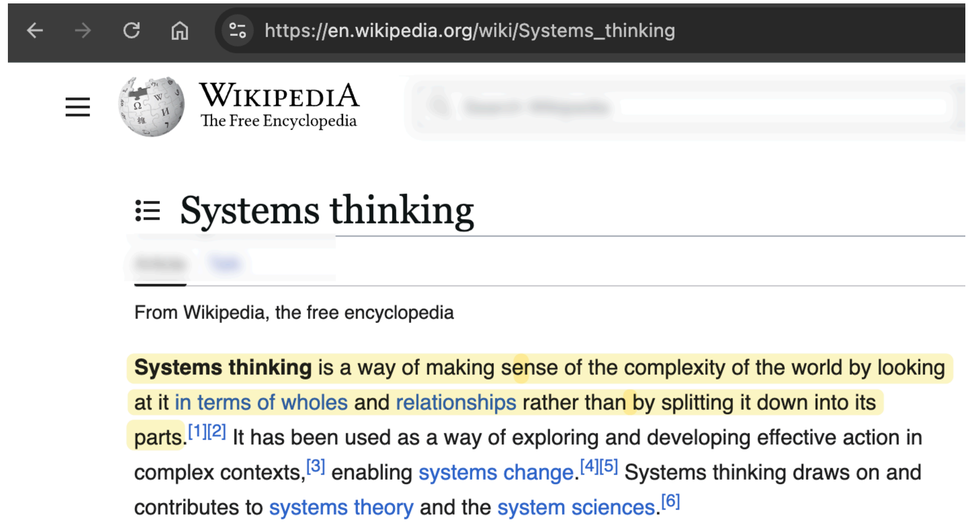
TAGS	METRIC	AVG	MIN	MAX	SUM	VALUE
*	updates	5,712 rows/s	4,435 rows/s	7.67k rows/s	965k rows/s	6,620 rows/s
*	deletes	496 rows/s	293 rows/s	1.42k rows/s	84k rows/s	487 rows/s
*	inserts	913 rows/s	519 rows/s	1.51k rows/s	154k rows/s	1,265 rows/s

Telemetry-driven development?



“Purpose”


https://en.wikipedia.org...
WIKIPEDIA
The Free Encyclopedia
Donate Create account Log in
The purpose of a system is what it does
3 languages
Article Talk
Read Edit View history Tools
From Wikipedia, the free encyclopedia
The purpose of a system is what it does (POSIWID) is a heuristic in systems thinking coined by the British management consultant Stafford Beer,[1] who stated that there is "no point in claiming that the purpose of a system is to do what it constantly fails to do".[2] It is widely used by systems theorists, and is generally invoked to counter the notion that the purpose of a system can be read from the intentions of those who design, operate or promote it. When a system's side effects or unintended consequences reveal that its behaviour is poorly understood, then the POSIWID perspective can balance political understandings of system behaviour with a more straightforwardly descriptive view.


https://en.wikipedia.org/wiki/Systems_thinking
WIKIPEDIA
The Free Encyclopedia
Systems thinking
From Wikipedia, the free encyclopedia
Systems thinking is a way of making sense of the complexity of the world by looking at it in terms of wholes and relationships rather than by splitting it down into its parts.[1][2] It has been used as a way of exploring and developing effective action in complex contexts,[3] enabling systems change.[4][5] Systems thinking draws on and contributes to systems theory and the system sciences.[6]

“The best time to utilize telemetry in your Elixir application was July 3rd, 2021*; the second best time is now.

If you (or an **AI/LLM**) really want to understand the purpose of your system, you need some form of telemetry.”

- Noah Betzen

Versions (12)

1.3.0 Aug 22, 2024

1.2.1 Jan 12, 2023

1.2.0 Jan 03, 2023

1.1.0 Apr 01, 2022

*1.0.0 Jul 03, 2021

Show All Versions

Outline

1. Recap Telemetry + OpenTelemetry
2. Introduce basic local telemetry setup (demo!)
3. MIX_ENV=dev
4. MIX_ENV=test
5. MIX_ENV=prod
6. Challenges combining Elixir + OpenTelemetry
7. Future of telemetry-driven development

“Telemetry”

https://en.wikipedia.org/wiki/Telemetry

WIKIPEDIA
The Free Encyclopedia

Telemetry

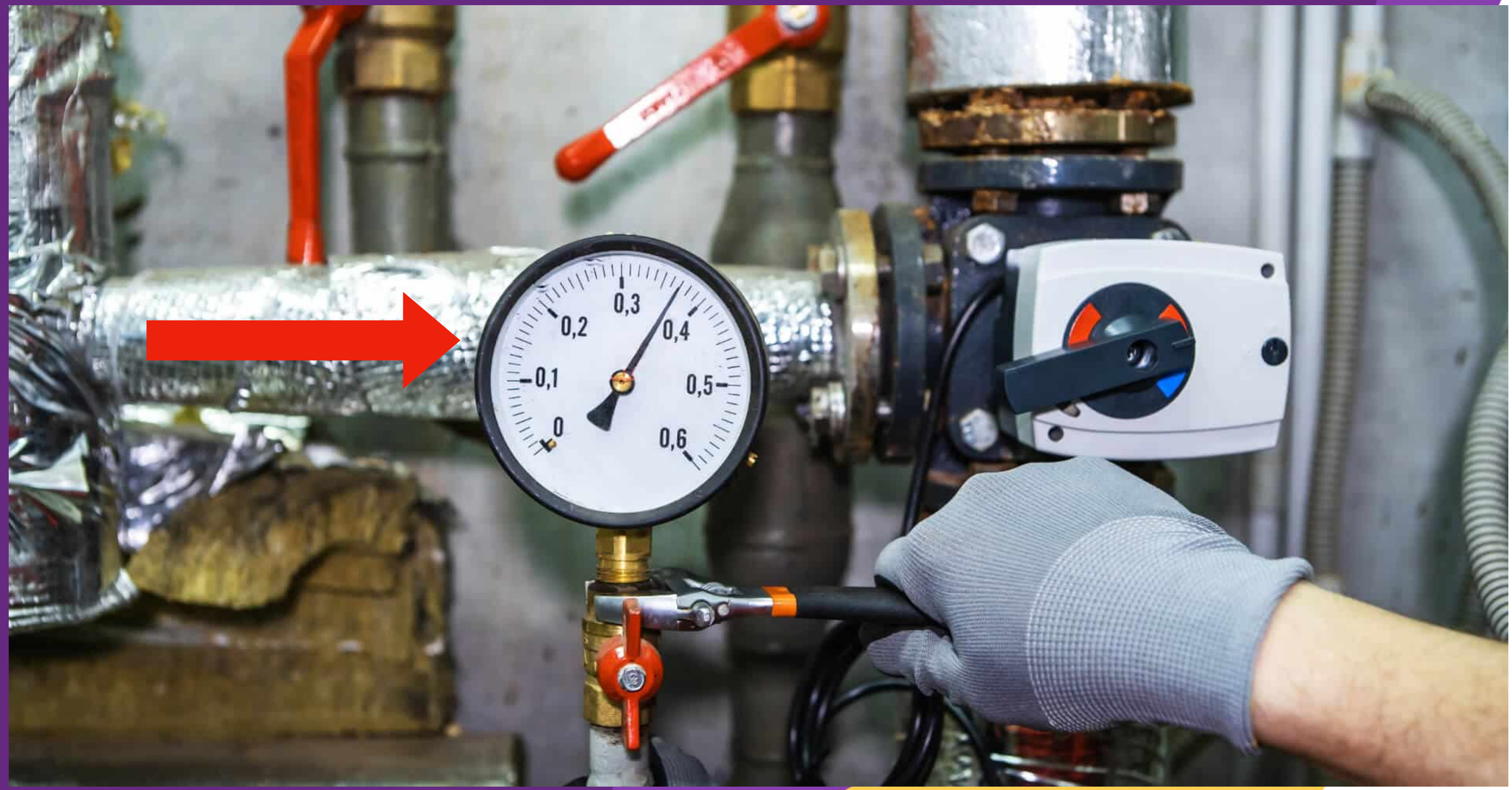
From Wikipedia, the free encyclopedia

Not to be confused with Telemetry (company) or Telemetry (software).

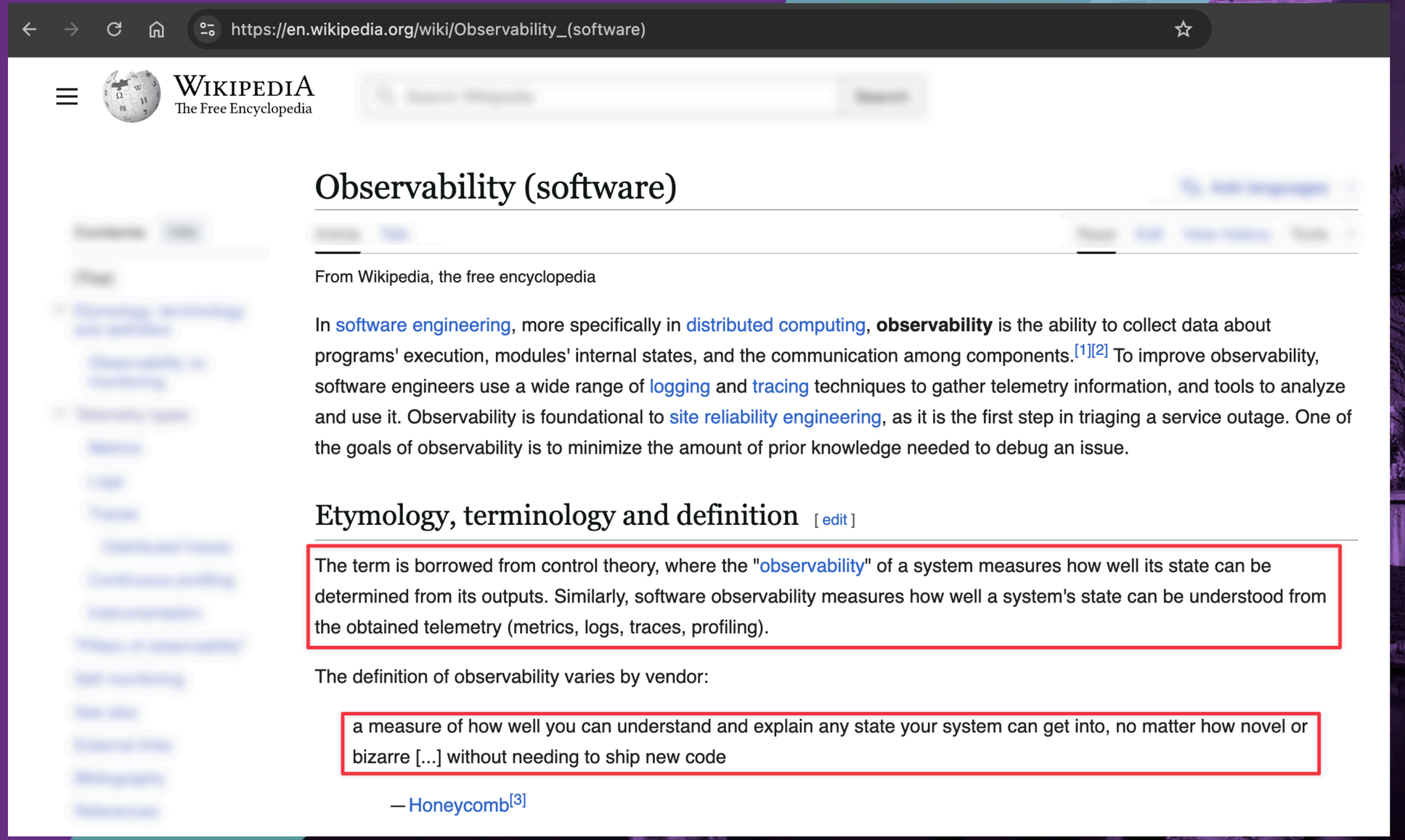
Telemetry is the in situ collection of measurements or other data at remote points and their automatic transmission to receiving equipment (telecommunication) for monitoring.[1] The word is derived from the Greek roots *tele*, 'far off' and *metron*, 'measure' Systems that need external instructions and data to operate require the counterpart of telemetry: telecommand.[2]

Although the term commonly refers to wireless data transfer mechanisms (e.g., using radio, ultrasonic, or infrared systems), it also encompasses data transferred over other media such as a telephone or computer network optical link or other wired communications like power line carriers. Many modern telemetry systems take advantage of the low cost and ubiquity of GSM networks by using SMS to receive and transmit telemetry data.

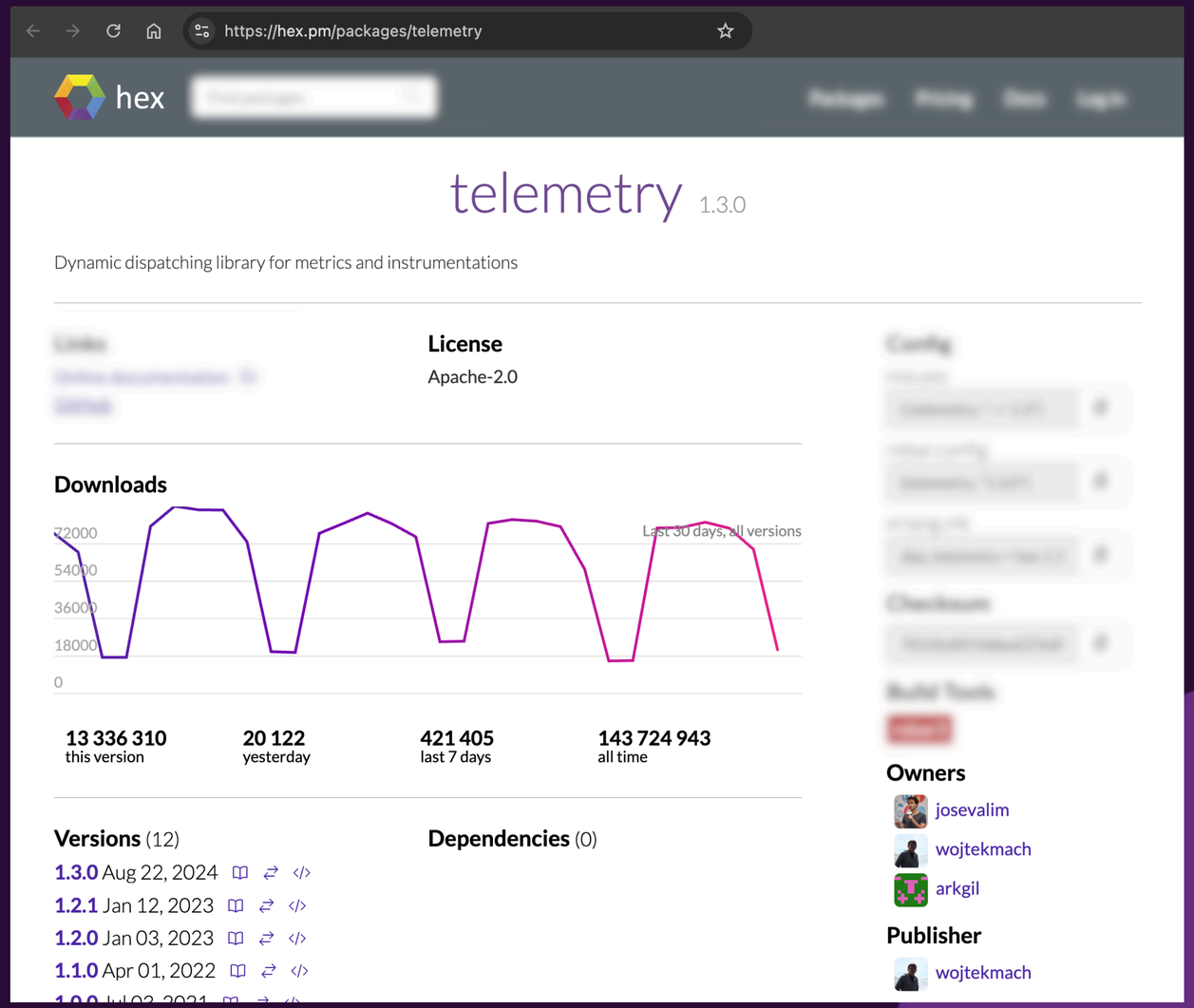
“Telemetry”

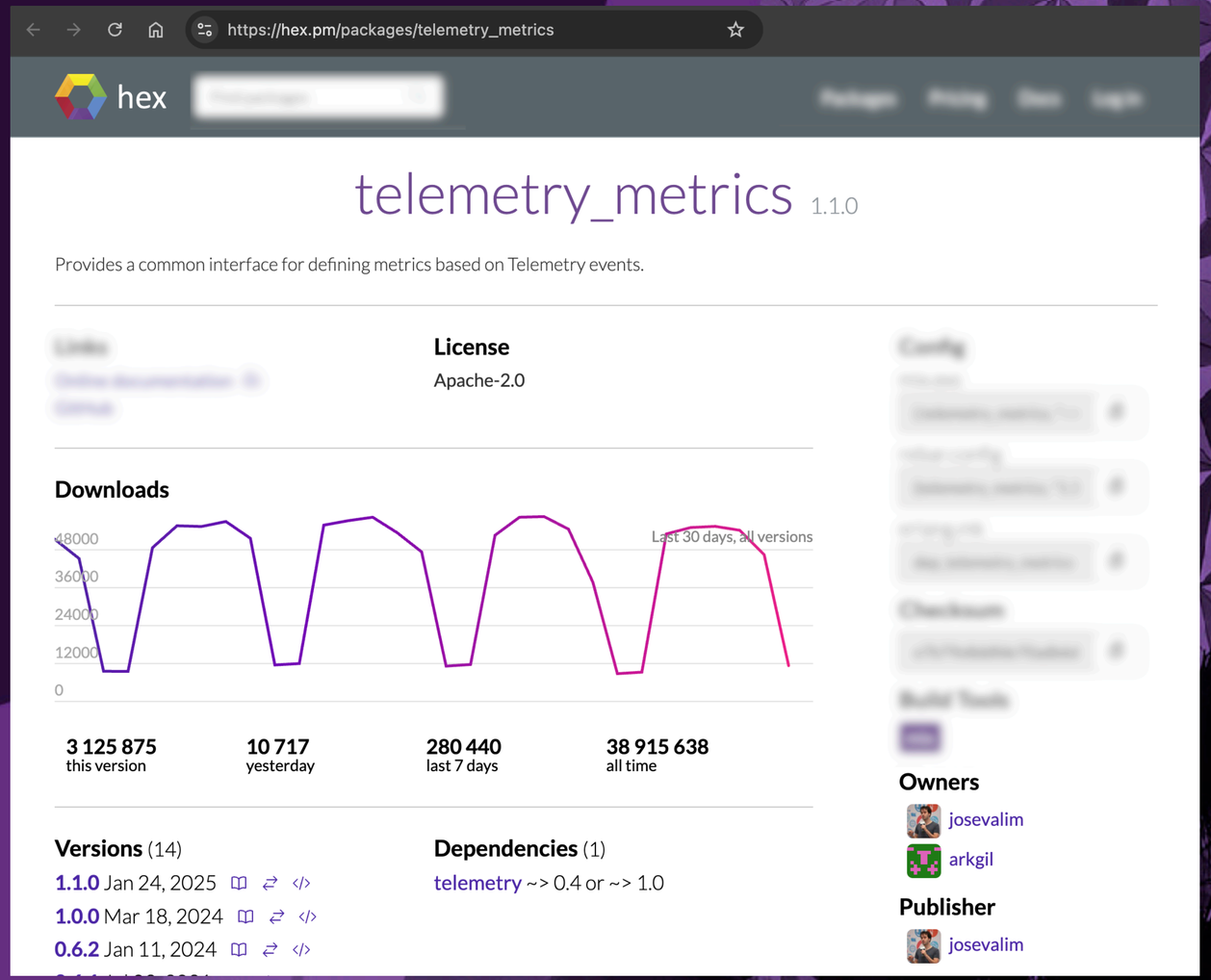


“Observability”



Elixir Telemetry / Metrics



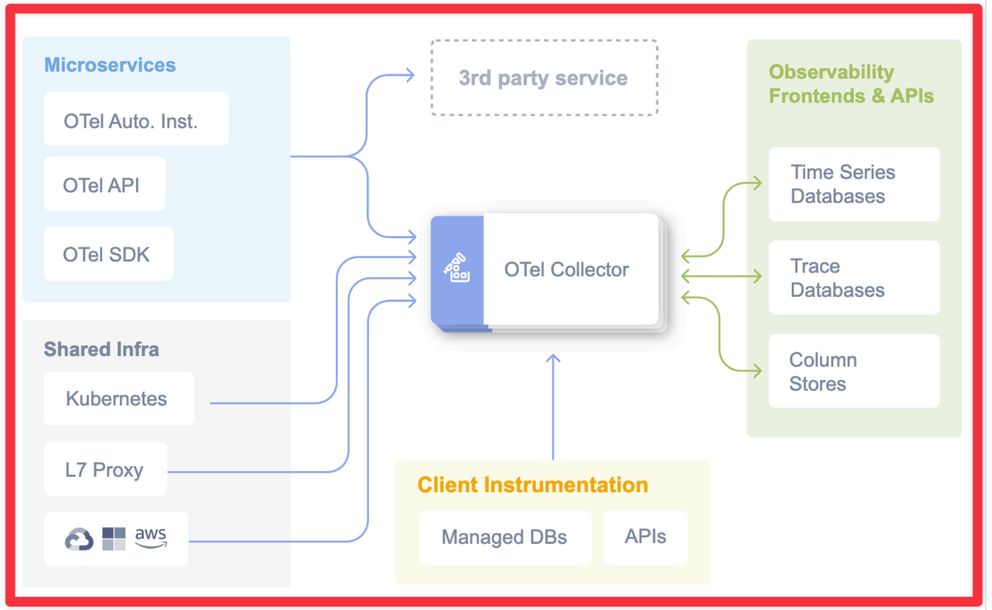


https://opentelemetry.io/docs/

OpenTelemetry

Documentation

OpenTelemetry, also known as OTel, is a vendor-neutral open source Observability framework for instrumenting, generating, collecting, and exporting telemetry data such as traces, metrics, and logs.



OpenTelemetry “Signals”

Signals

Learn about the categories of telemetry supported by OpenTelemetry

The purpose of OpenTelemetry is to collect, process, and export signals. Signals are system outputs that describe the underlying activity of the operating system and applications running on a platform. A signal can be something you want to measure at a specific point in time, like temperature or memory usage, or an event that goes through the components of your distributed system that you’d like to trace. You can group different signals together to observe the inner workings of the same piece of technology under different angles.

OpenTelemetry currently supports:

- Traces
- Metrics
- Logs
- Baggage

Also under development or at the proposal stage:

- Events, a specific type of log
- Profiles are being worked on by the Profiling Working Group.

Elixir + OpenTelemetry

https://opentelemetry.io/docs/languages/erlang/#status-and-releases

OpenTelemetry

- Erlang/Elixir
  - Getting Started
  - Instrumentation
  - Libraries
  - Exporters
  - Propagation
  - Resources
  - Sampling
  - Testing
  - API
  - Examples
  - Registry

Erlang/Elixir

A language-specific implementation of OpenTelemetry in Erlang/Elixir.

This is the OpenTelemetry Erlang/Elixir documentation. OpenTelemetry is an observability framework – an API, SDK, and tools that are designed to aid in the generation and collection of application telemetry data such as metrics, logs, and traces. This documentation is designed to help you understand how to get started using OpenTelemetry Erlang/Elixir.

Status and Releases

The current status of the major functional components for OpenTelemetry Erlang/Elixir is as follows:

Traces	Metrics	Logs
Stable	Development	Development

Packages of the API, SDK and OTLP exporter are published to `hex.pm` as `opentelemetry_api`, `opentelemetry` and `opentelemetry_exporter`.

Repositories

- opentelemetry-erlang: Main repository containing the API, SDK and OTLP Exporter.
- opentelemetry-erlang-contrib: Helpful libraries and instrumentation libraries for Erlang/Elixir projects like Phoenix and Ecto

https://github.com/open-telemetry/opentelemetry-erlang/issues/427

open-telemetry / opentelemetry-erlang

Add logging domain #427

Open

tsloughter opened on Aug 15, 2022 — Member

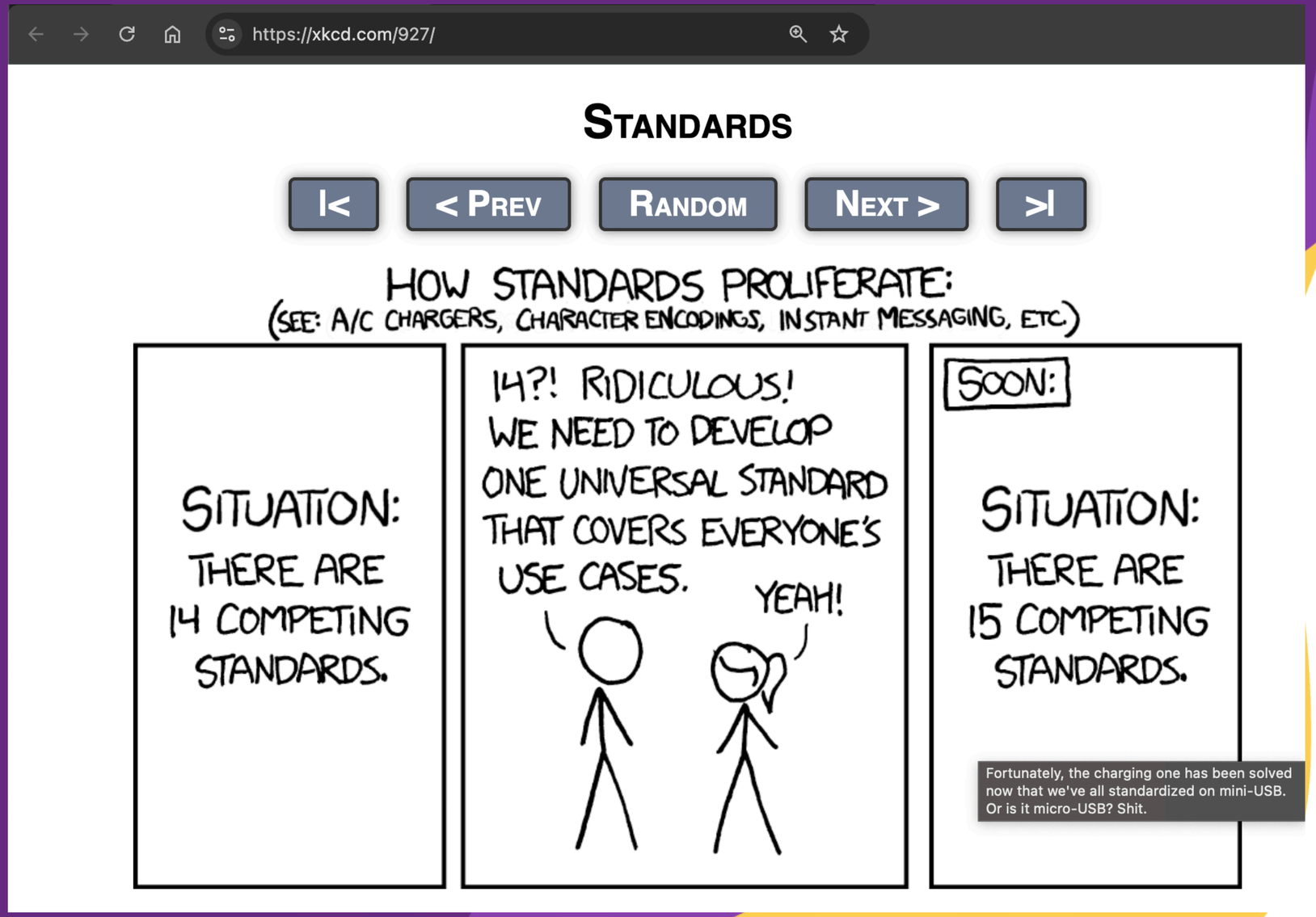
OTP logger supports a metadata key `domain` which makes it easy to filter out logs.

All logs from opentelemetry should include a domain `opentelemetry` and there should be a function to make it easy for users to add the logger filter provided.

We probably just want to have a custom set of `?LOG_` macros instead of putting the metadata manually in every use of the logger macros.

tsloughter added enhancement logs on Aug 15, 2022

Elixir + OpenTelemetry



Elixir + OpenTelemetry

https://hex.pm/packages?search=opentelemetry&sort=recent_downloads

hex opentelemetry

Packages

Exact Match:

opentelemetry 1.5.1
Implementation of stable OpenTelemetry signals
1 467 307 recent downloads

64 Results Found

Sort: Recent downloads

Search Results:

opentelemetry_api 1.4.1
OpenTelemetry API
1 575 433 recent downloads

opentelemetry_exporter 1.8.1
OpenTelemetry Protocol Exporter
1 417 803 recent downloads

opentelemetry_semantic_conventions 1.27.0
OpenTelemetry Semantic Conventions
1 335 905 recent downloads

opentelemetry_telemetry 1.1.2
Bridge library between Telemetry events and OpenTelemetry Erlang
1 215 909 recent downloads

opentelemetry_process_propagator 0.3.0
Tools for opentelemetry context propagation across process boundaries
1 177 519 recent downloads

opentelemetry_phoenix 2.0.1
OpenTelemetry tracing for the Phoenix Framework
1 071 076 recent downloads

https://knowyourmeme.com/memes/mr-bones-wild-ride

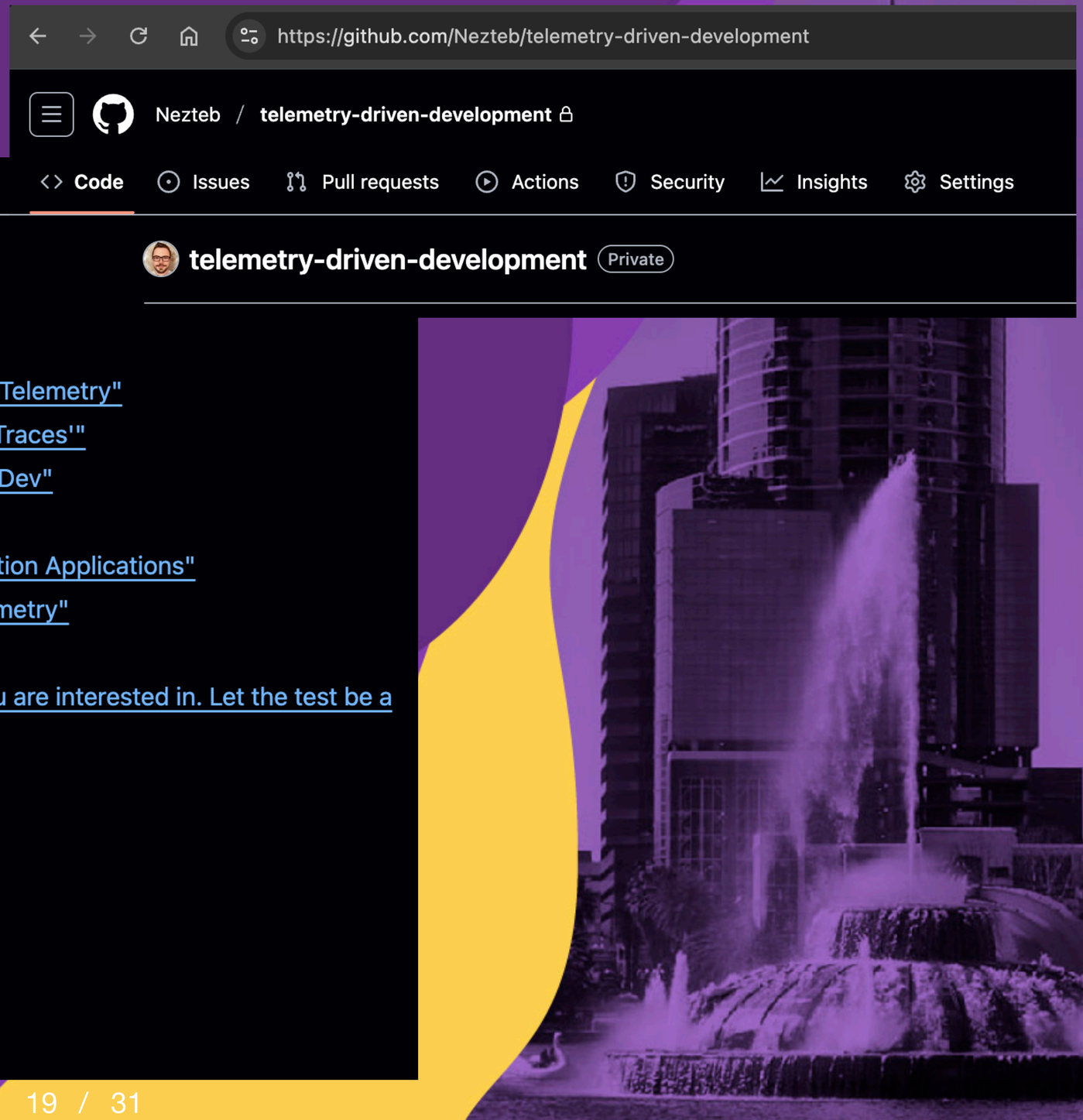
🧘‍♂️ Deep breath

🥤 Water break

Context / Demo

Timeline / Context

- January 19, 2022: "Monitoring Elixir With OpenTelemetry"
- June 22, 2022: "BEAM + Prometheus + Grafana = Observability Heaven"
- November 14, 2022: "Elixir, OpenTelemetry, and the Infamous N+1"
- November 3, 2023: Tweet: "Webinar: Test driven development with OpenTelemetry"
- February 15, 2024: "All you need is Wide Events, not "Metrics, Logs and Traces'"
- February 17, 2024: "Observability for Phoenix using the Grafana Stack in Dev"
- April 4, 2024: Cory O'Daniel DMs me a link to tracetest
- April 16, 2024: "Using OpenTelemetry To Troubleshoot & Monitor Production Applications"
- December 11, 2024: "Instrumenting Erlang and Elixir code with open telemetry"
- December 12, 2024: "Telemetry & Observability for Elixir Apps"
- February 6, 2025: "Let the code to test emit telemetry events of what you are interested in. Let the test be a handler for those telemetry events. " - LostKobrakai
- February 11, 2025: Initial wildfires local telemetry demo
- March 24, 2025: German Velasco asks about tracing in production
- April 27, 2025: "OpenTelemetry: From Desire to Dashboard"
- May 6, 2025: Posts about Elixir pains working with OpenTelemetry
- June 23, 2025: Lars Wikman asks about opinions on Grafana
- June 23, 2025: Łukasz Niemier mentions OpenTelemetry + DuckDB
- June 30, 2025: Jacob Swanner's proposed telemetry talk



Context / Demo

https://hub.docker.com/r/grafana/otel-lgtm

hub

Search Docker Hub ⌘K

Explore / grafana / otel-lgtm

grafana/otel-lgtm Verified Publisher

By Grafana Labs • Updated 1 day ago

An OpenTelemetry backend in a Docker image.

IMAGE

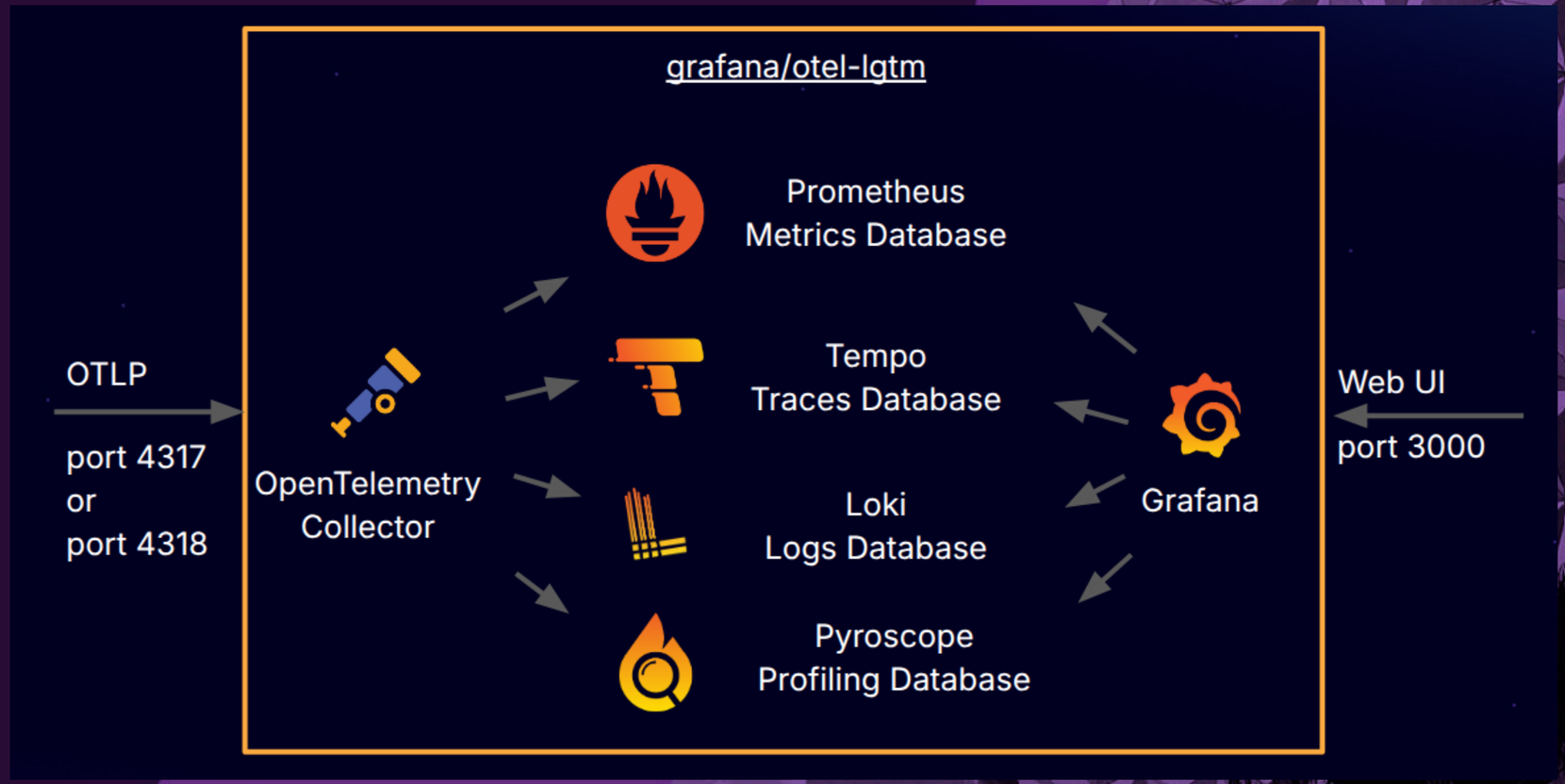
41 500K+

Overview Tags

An OpenTelemetry Backend in a Docker Image

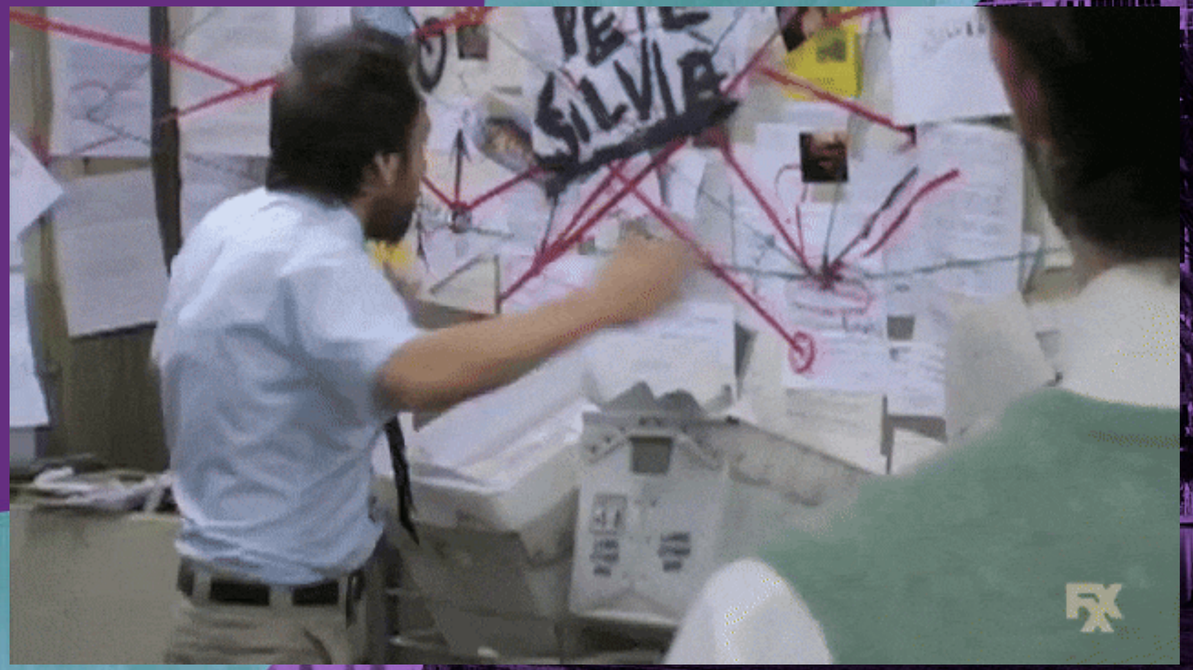
`grafana/otel-lgtm` contains a complete OpenTelemetry backend in a single Docker image:

- OpenTelemetry collector for receiving OpenTelemetry data.
- Prometheus metrics database.
- Tempo trace database.
- Loki logs database.
- Grafana for visualization.



Environments

1. MIX_ENV=dev
2. MIX_ENV=test
    1. Local
    2. CI
3. MIX_ENV=prod
    1. “dev”
    2. “test” / “qa”
    3. “prod”



https://knowyourmeme.com/memes/pepe-silvia


MIX_ENV=dev


MIX_ENV=test

MIX_ENV=prod

“development” “test”, “UAT”, “QA”, “demo”, “prod-us”, “prod-eu”, etc.

🧘‍♂️ Deep breath

🥤 Water break

MIX_ENV	dev	test	prod
Logs?	Yes	As needed	Yes
Metrics?	🤔	🤔	Yes
Tracing?	🤔	🤔	Yes

Environment	“Develop”	“Test”	“US Prod”	“EU Prod”
Logs?	🤔	🤔	Yes	Yes
Metrics?	🤔	🤔	Yes	Yes
Tracing?	🤔	🤔	Yes	Yes

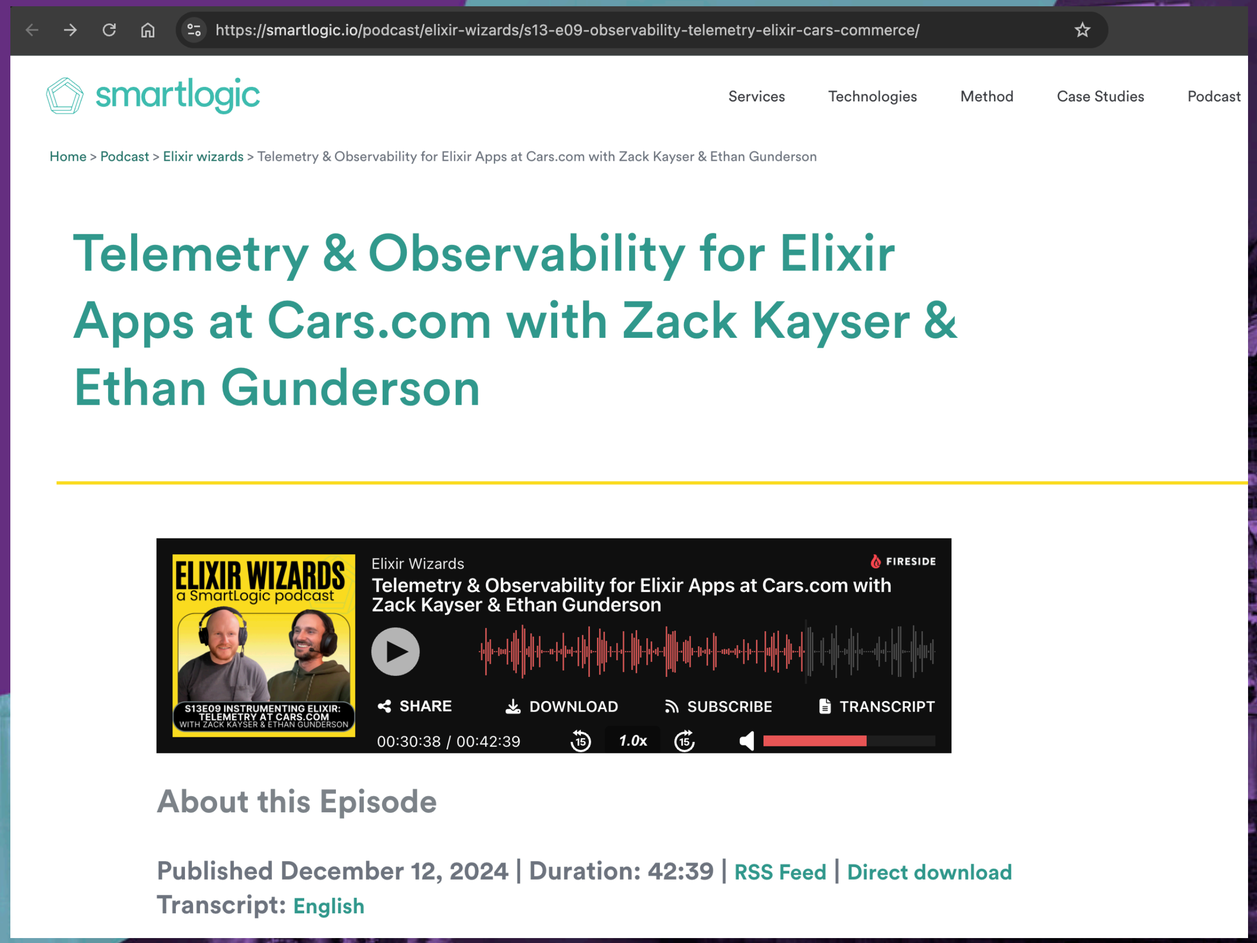
Telemetry Advice

“If you can ingest a hundred percent of your traces, I would just go crazy there and focus all my energy on traces. I wouldn't worry about metrics. I **really** wouldn't worry about logs.

🌶️🌶️🌶️

My ultimate hot take is, ‘**logs are garbage**’. They're super expensive and really low-utility in my opinion.”

- Ethan Gunderson



“Telemetry isn’t free”

- Business value: signal vs noise
  - “Signals”
    - Logging
    - Metrics
    - Traces
- Effort: Encoding, transmitting, parsing, storing, etc.
- Goodhart's Law
  - ”When a measure becomes a target, it ceases to be a good me
- Some ad hoc telemetry data could just be application data
  - `SELECT count(*) FROM <table>`

Future

Status and Releases

The current status of the major functional components for OpenTelemetry Erlang/Elixir is as follows:

Traces	Metrics	Logs
Stable	Development	Development

For releases, including t latest release, see Releases.

Packages of the API, S and OTLP exporter are published to `hex.pm` as `opentelemetry_api`, `opentelemetry` and `opentelemetry_e rter`.

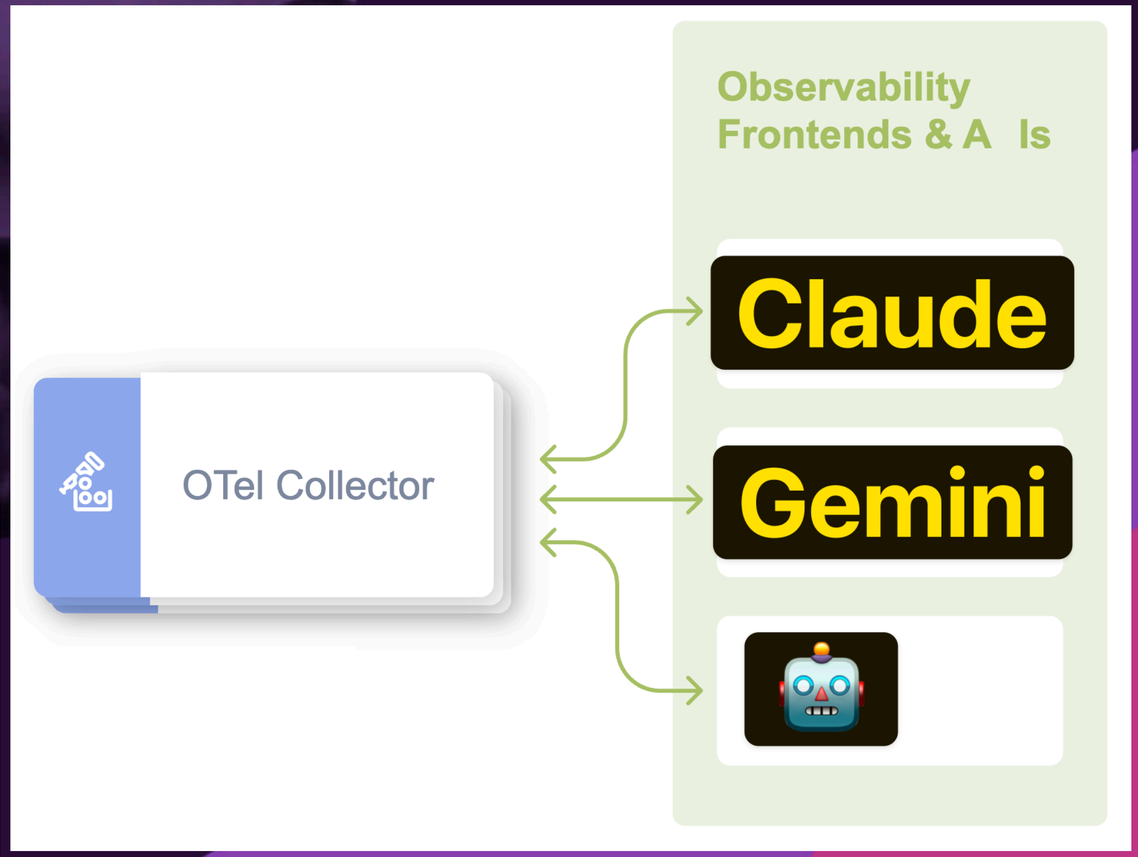
Status an Releases

The current status the major functional components for OpenTelemetry Erlang/Elixir is as follows:

Traces	Metrics	Logs
Stable	Stable	Stable

For releases, including the latest release, see Releases.

Packages of the API, SDK and OTLP exporter are published to `hex.pm` as `opentelemetry_api`, `opentelemetry` and `opentelemetry_exporter`.



Recap

1. Recap Telemetry + OpenTelemetry
2. Introduce basic local telemetry setup (demo!)
3. MIX_ENV=dev
4. MIX_ENV=test
5. MIX_ENV=prod
6. Challenges combining Elixir + OpenTelemetry
7. Future of telemetry-driven development

Q&A

Secret Spicy Take

“If you decide to track product-specific analytics using typical engineering telemetry tooling, you will die.”

There are two hard problems in Computer Science:

0. Naming things
1. Cache invalidation
2. Off-by-one errors