

Apache Flink and Prometheus: better together to improve the efficiency of your observability platform at scale

Lorenzo Nicora

(he/him)

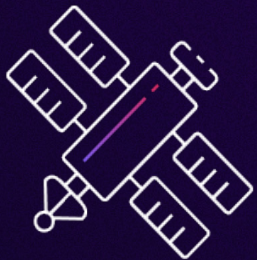
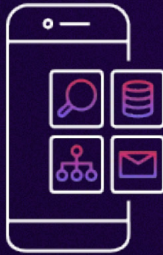
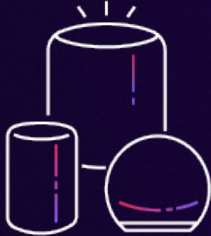
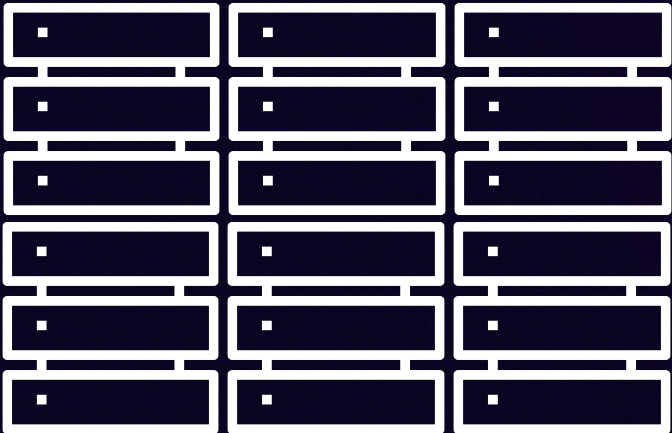
Sr. Streaming WW Solution Architect
Amazon Web Services

Hong Liang Teoh

(he/him)

Sr. Software Dev Engineer + Apache Flink Committer
Amazon Web Services

What do we want to observe?



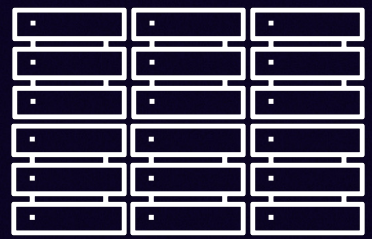
Real-time dashboards
Alerting



Real-time dashboards
Alerting



Observing servers



Prometheus Exporter
/metrics

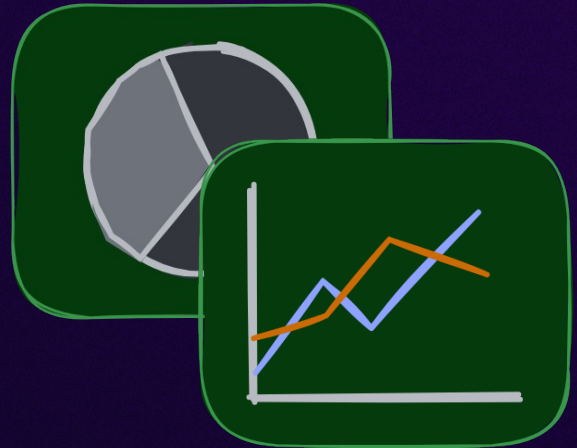
scrape



PromQL

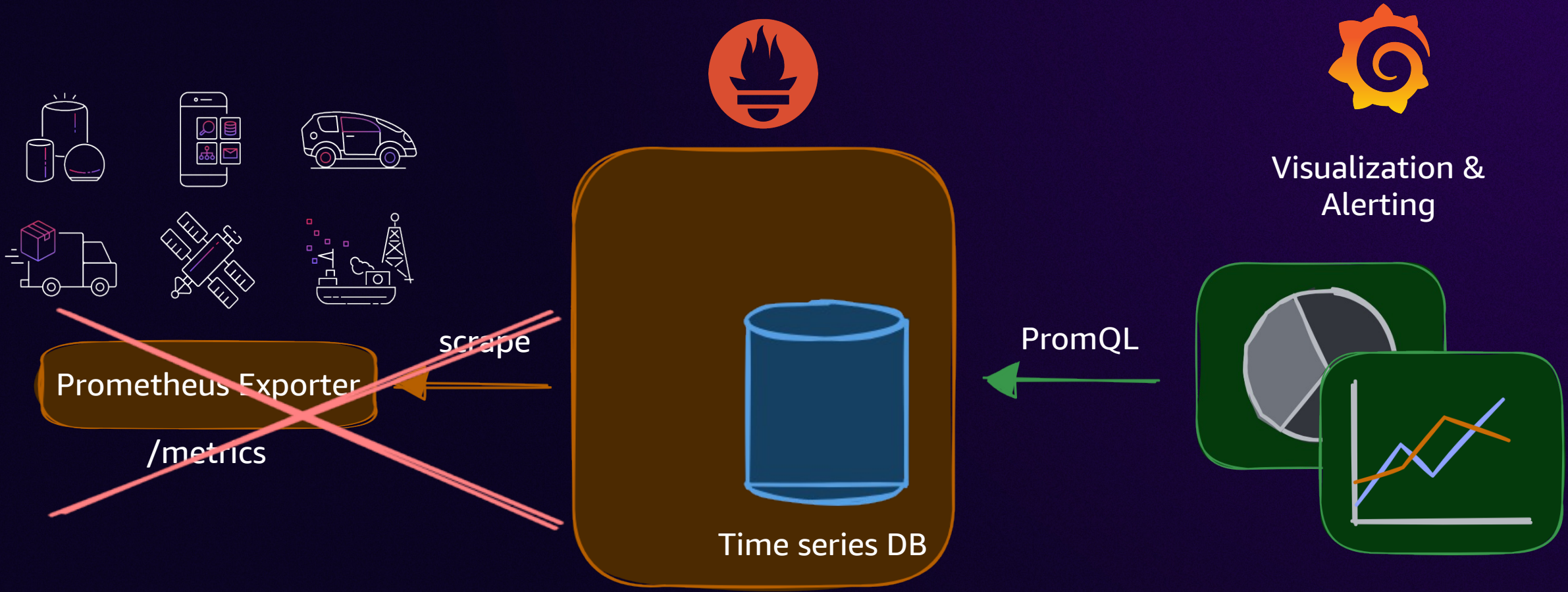


Visualization & Alerting



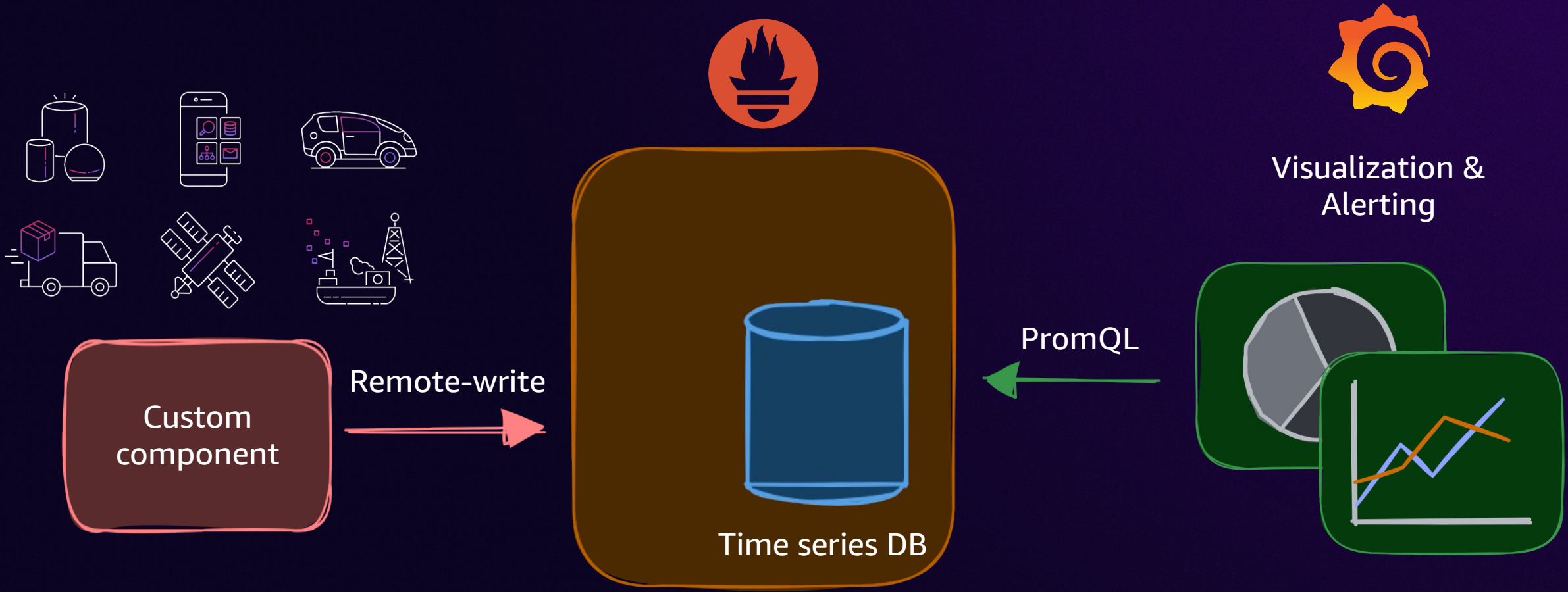
Pull model

Observing IoT devices



Pull model

Observing IoT devices



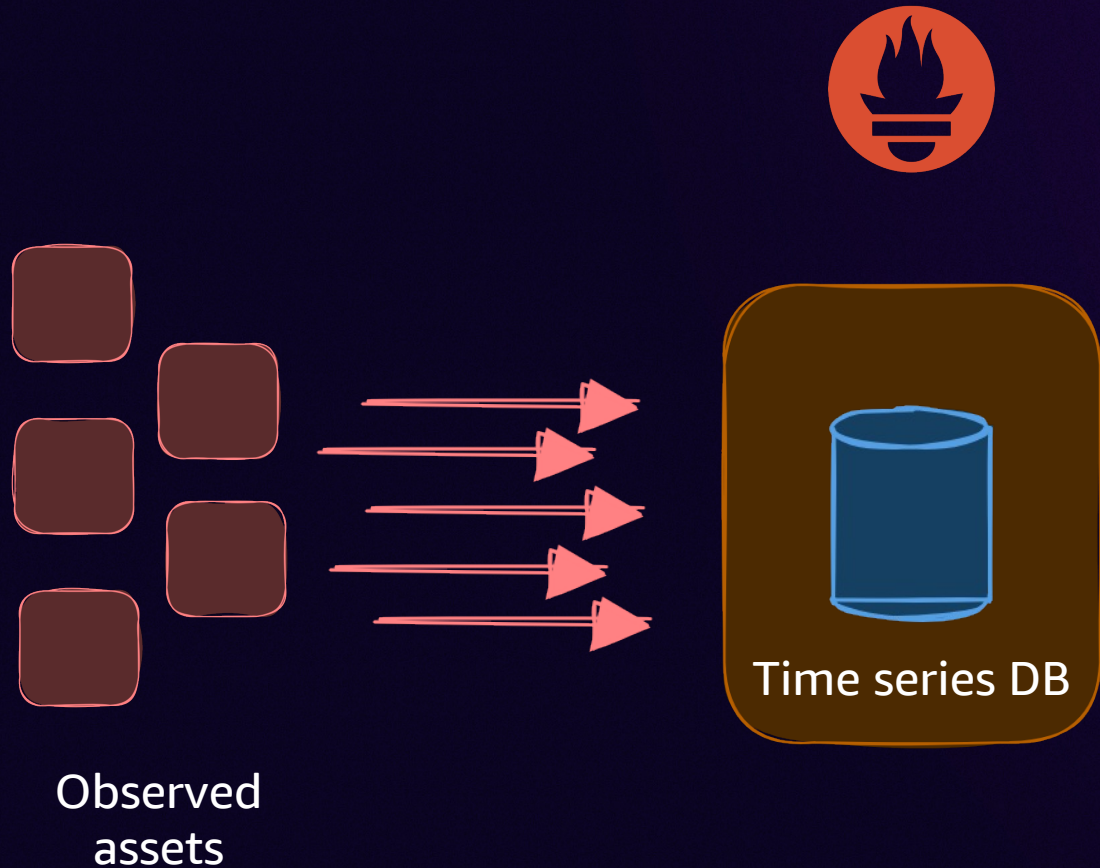
Push model

Querying Prometheus

PromQL

- Filtering and aggregating time series in real time
- Dedicated set of functions
 - `rate()`, `deriv()` . . .
- No joins
- Limited custom aggregations
- Slow performance for high cardinality data

Observing IoT devices

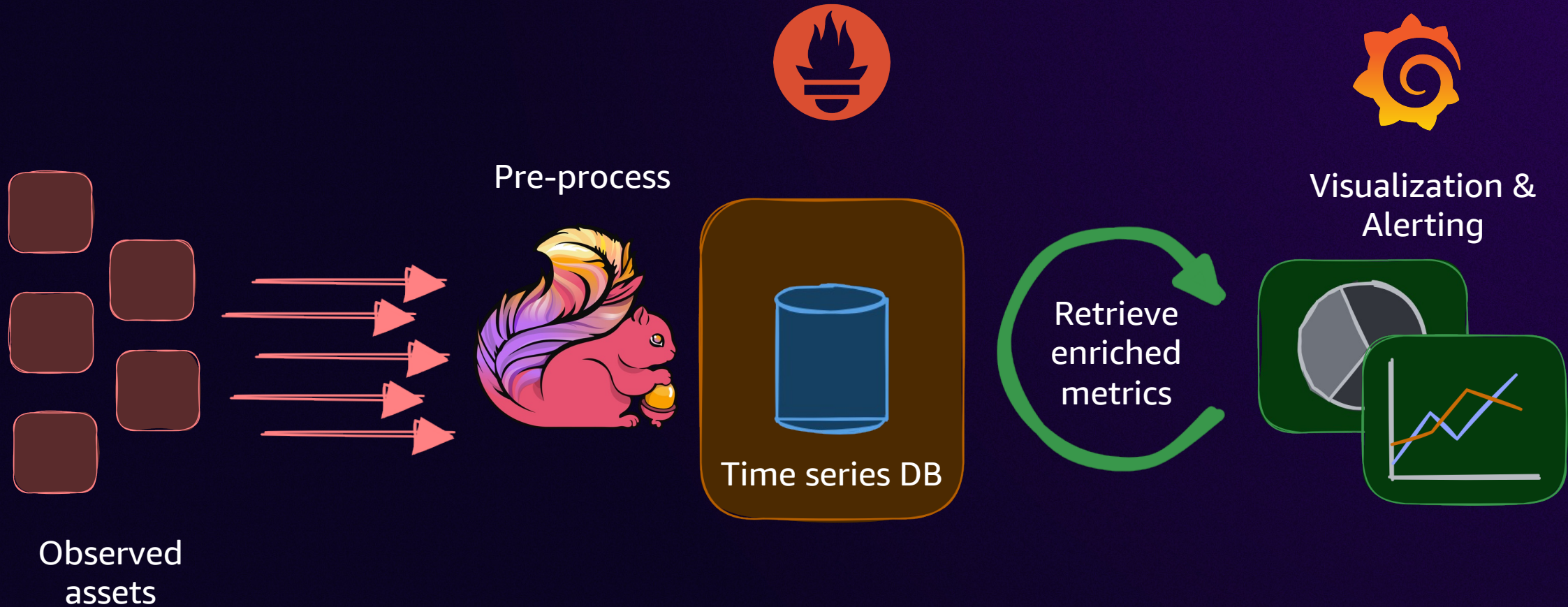


- **Push-model only**
- **Very high cardinality**
- **High frequency**
- **Lack of contextual info**

Process on query



Process on write (pre-process)



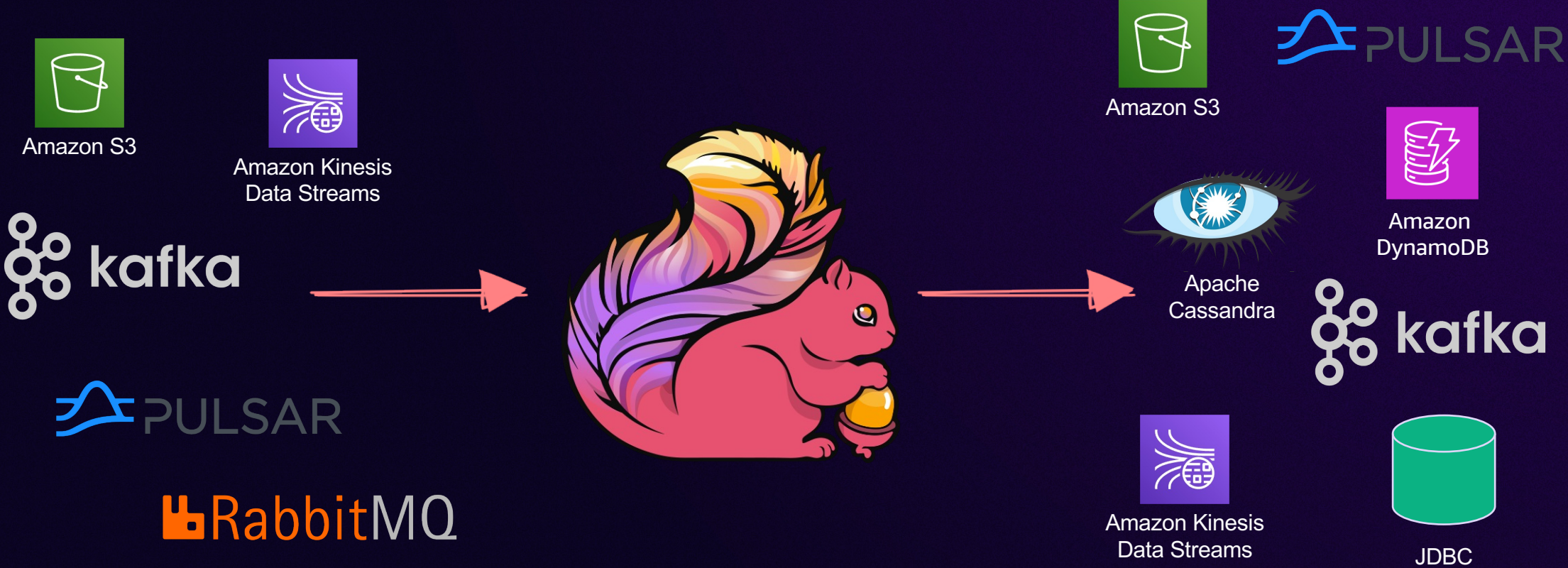
Apache Flink as operational metrics pre-processor



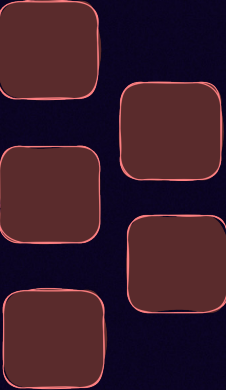


“Apache Flink is **framework**
and a **distributed processing engine**
for **stateful computation**
over **unbounded and bounded**
data streams.”

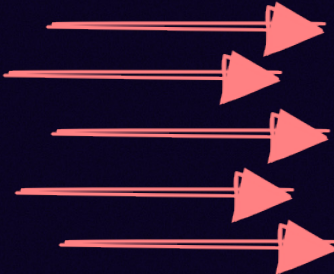
Multiple connectors



Architecture with Flink



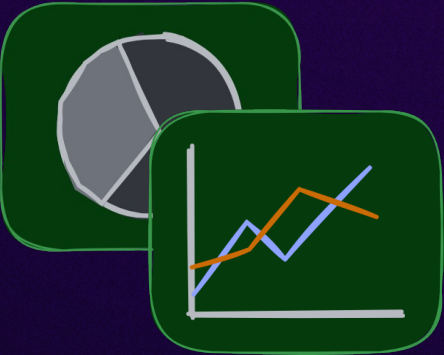
Observed assets



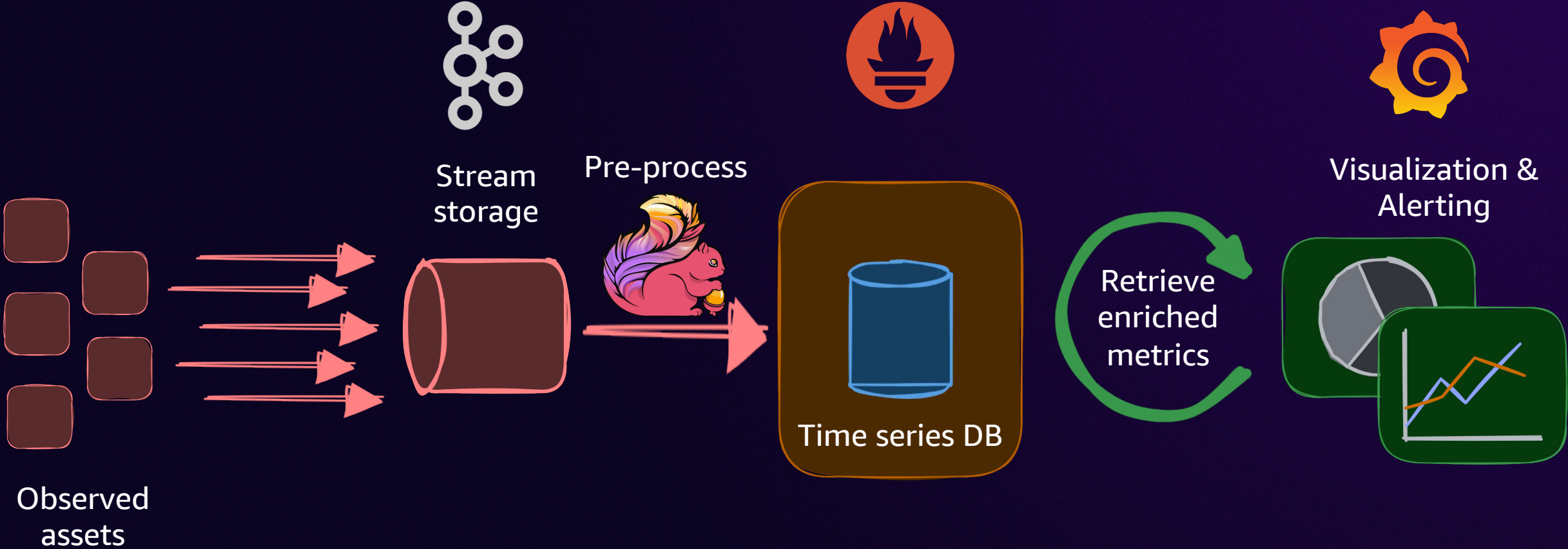
Pre-process



Visualization & Alerting



Architecture with Flink



Pre-processing operational metrics with Apache Flink



- Reduce cardinality
- Reduce granularity
- Filtering
- Enrichment – add dimensions
- Derived metrics

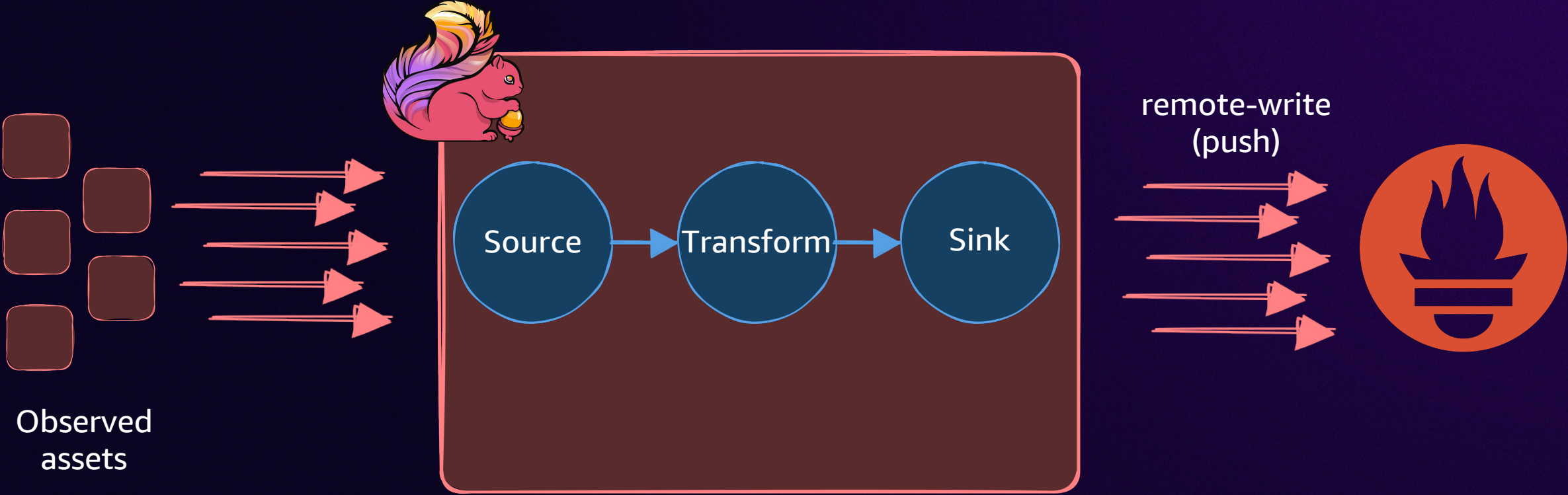
Simplify
scaling

Simplify
analysis

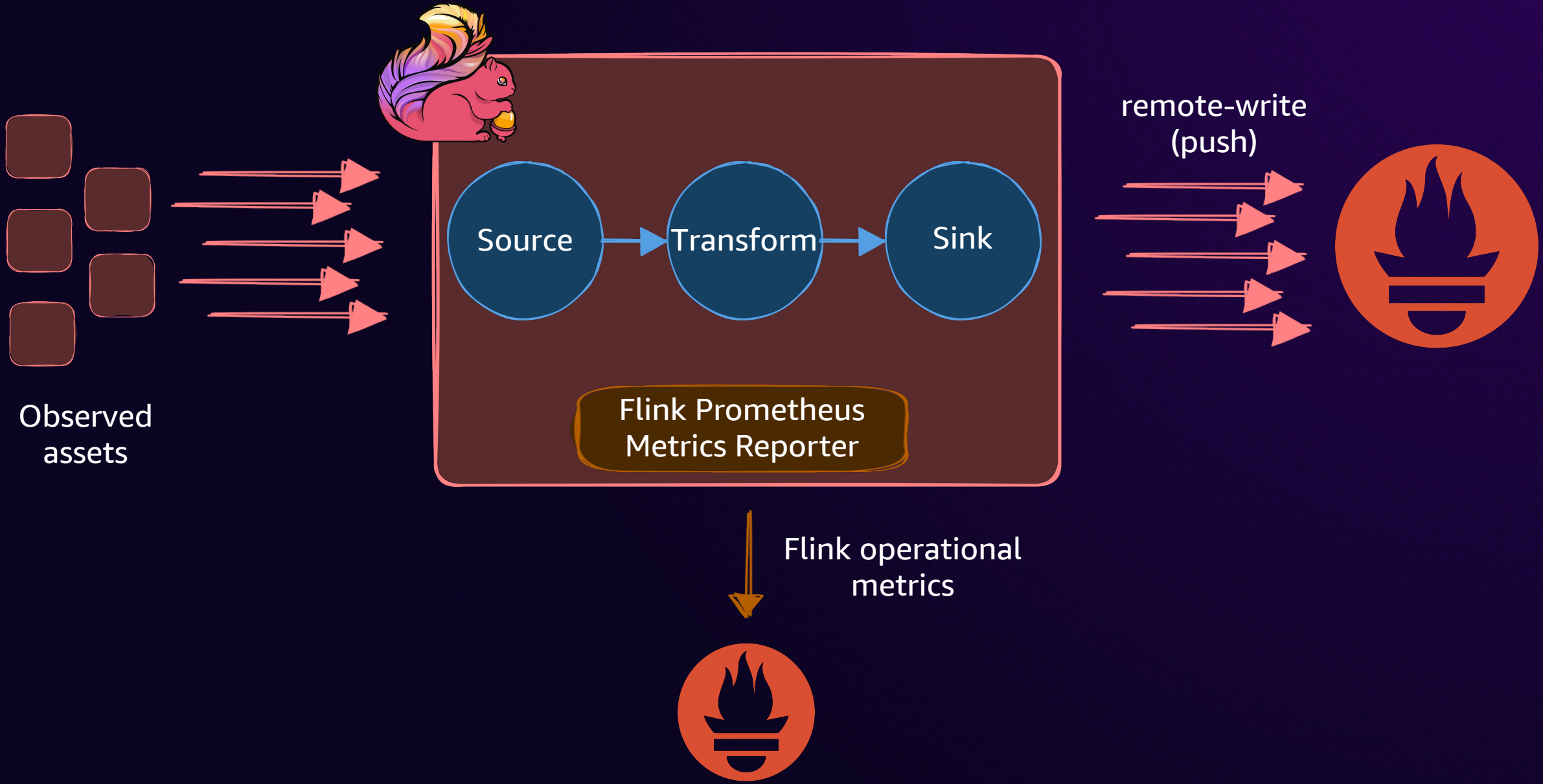
Apache Flink + Prometheus Integration



Apache Flink + Prometheus Integration



Can't we use the Flink Prometheus Reporter?



Flink Prometheus Connector

- New addition to Apache Flink Connector ecosystem
- Released: November 2024
- Supports Apache Flink 1.19+



A screenshot of the Prometheus Sink documentation page. The page has a white background with a dark blue header. The title is "Prometheus Sink". Below the title, there is a paragraph explaining that the sink connector can be used to write data to Prometheus-compatible storage using the Remote Write interface. Another paragraph states that the backend must support the Remote Write 1.0 standard API and have the Remote Write endpoint enabled. A callout box contains a note that the connector is not for internal Flink metrics. Below this, there is a section for Maven dependencies with a code block. The code block shows the Maven dependency for the connector. At the bottom, there is a "Usage" section with a paragraph explaining the builder class and providing a link to a request signer.



Prometheus



Prometheus Sink

This sink connector can be used to write **data** to Prometheus-compatible storage, using the [Remote Write](#) Prometheus interface.

The Prometheus-compatible backend must support [Remote Write 1.0](#) standard API, and the Remote Write endpoint must be enabled.

This connector is not meant for sending internal Flink metrics to Prometheus. To publish Flink metrics, for monitoring health and operations of the Flink cluster, you should use [Metric Reporters](#).

To use the connector, add the following Maven dependency to your project:

```
<dependency>
  <groupId>org.apache.flink</groupId>
  <artifactId>flink-connector-prometheus</artifactId>
  <version>1.0.0-1.20</version>
</dependency>
```

Usage

The Prometheus sink provides a builder class to build a `PrometheusSink` instance. The code snippets below shows how to build a `PrometheusSink` with a basic configuration, and an optional [request signer](#).

Tested at scale

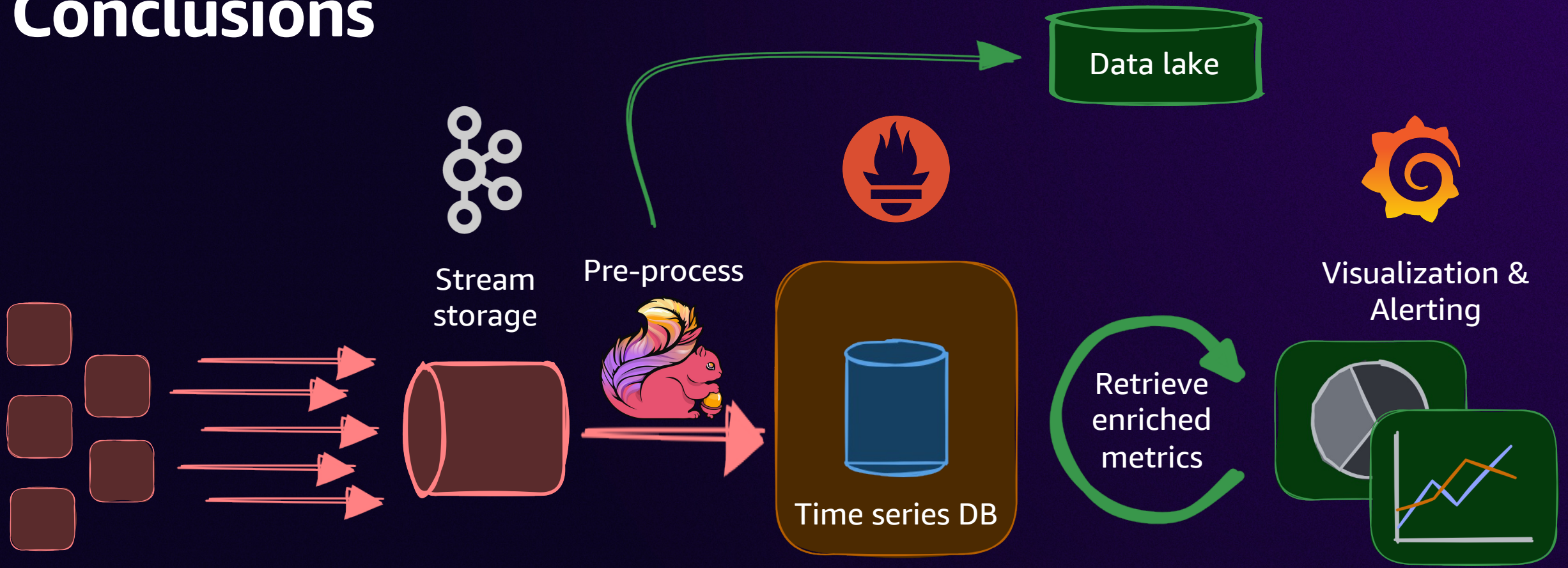
Tested up to
1M events/sec
with metric cardinality of 1M metrics
(we did not test further)

Tradeoffs

- A *yes-code*TM solution → Java
- Run the Flink job → Flink cluster (or managed service)

Conclusions

Conclusions



Observing
at scale

Flink
as pre-processor

Real time dashboard
and alerting



Flink Prometheus
connector
documentation



Connected
vehicles demo
GitHub repo

Thank you + Q&A

Hong Liang Teoh

LinkedIn: <https://www.linkedin.com/in/hongteoh/>

GitHub: @hlteoh37

Lorenzo Nicora

LinkedIn: <https://www.linkedin.com/in/nicus/>

GitHub: @nicusX

Demo

