Experiences with Datalogging to InfluxDB at the European XFEL

Dr. Gero Flucke, *et al.*
European XFEL GmbH

13th NOBUGS Conference: “**N**ew **O**pportunities for **B**etter **U**ser **G**roup **S**oftware”

Paul Scherrer Institute, Villigen, Switzerland

September 19-22, 2022

Outline

- Data logging in the **event-driven Karabo control system**
- **Mapping** Karabo to InfluxDB
- InfluxDB **backend setup**
- **Hiccups** on the way
- Conclusions

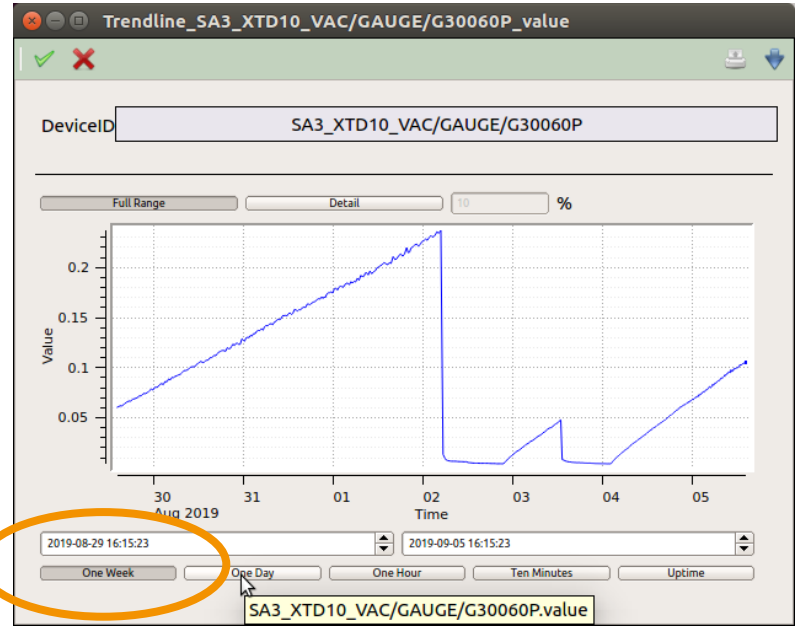
Karabo: Device Based Communication via a Message Broker

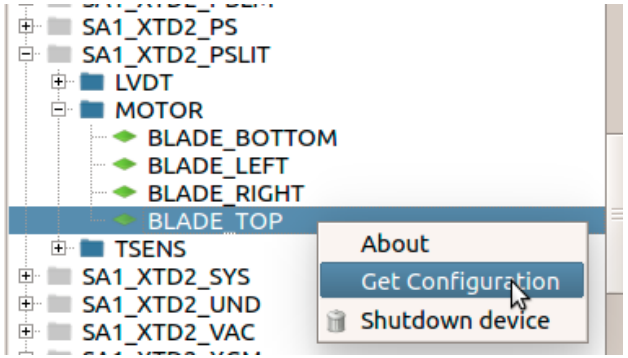
Self-describing
Karabo Devices

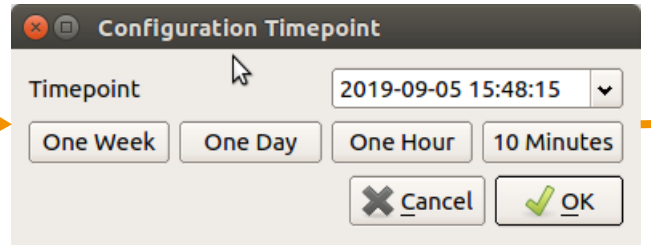
- Equipment control,
  e.g. motors, valves,…
- Detectors
  - e.g. cameras
- Online data analysis
- Data Logging
- Other system services
  - GUI entry point
  - DAQ for big/scientific data (not shown)

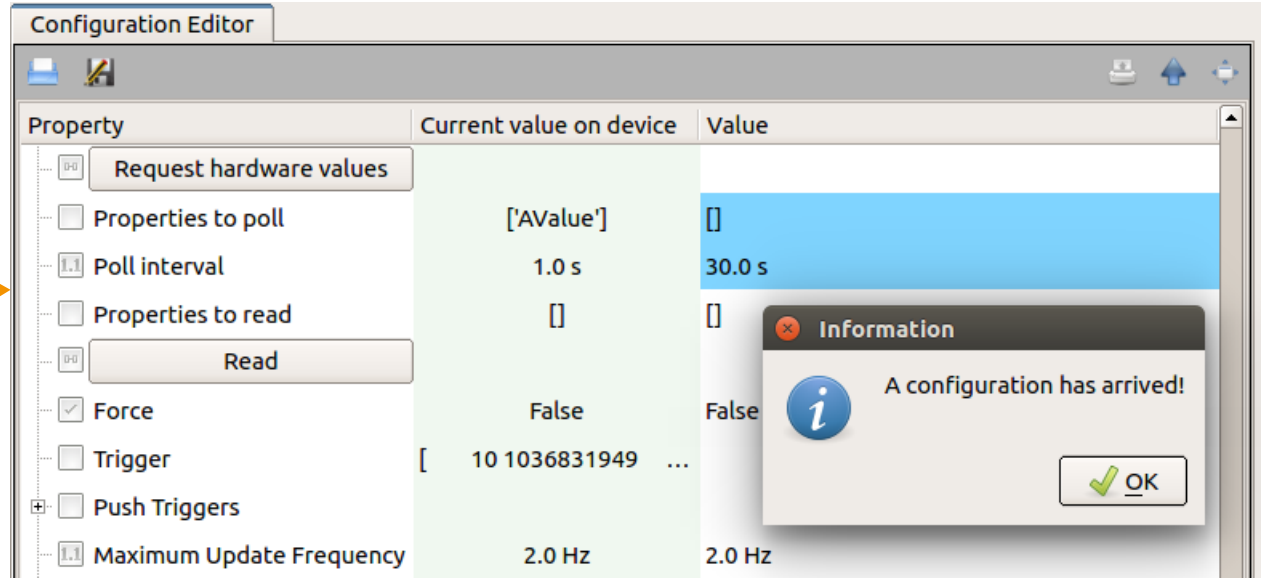
Karabo Data Logging

- Main use cases:
  - Historic data for trendlines: scalar property vs time.
  - Historic configurations: all device properties at a point in time
  - Base of “Config & Recovery” (project in beta stage)



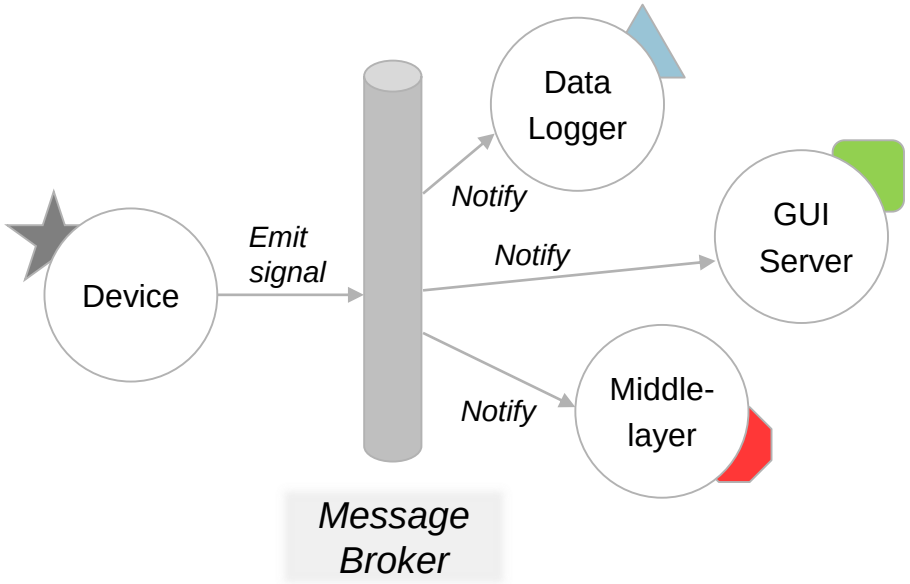






Karabo: Event-Driven Broker Communication

- Cross device signal/slot subscription
  - Devices subscribe slots to a remote “signal”.
- When signal is “emitted”, all subscribed slots are called.
  - Single message to the broker
    - ► Avoids publishing overhead for “popular” devices
  - Regular **polling obsolete**.



- Used by Data Logger Devices
  - Logging task distributed among a handful of “loggers”
  - Each logger subscribes to the “property update” and “schema update” signals of its share of devices
- Logged are
  - Properties
    - ► Configurations (e.g. hardware port)
    - ► Conditions (e.g. values read from hardware like a temperature)
  - Self-description (schema)
  - Timestamps as published by device

Karabo Data Logging: How?

- Original, custom-made solution from 2014:
  - Ascii file backend (one directory per logged device)

```
20200719T184128.573059Z|1595184088.573059|804227972|actualPosition|FLOAT|35.91043||VALID
```

  - Drawbacks
    - ► Human readable data does not scale well: routine access only for 3 months
    - ► Indexing (for fast retrieval) needed custom implementation
    - ► No support for statistical treatment
      (data reduction by simple down sampling: e.g. every 2nd data point only)
    - ► Reading back data can be slow

- Time series database is better suited to event-driven Karabo:
  - Logging to InfluxDB:
    - ► Karabo prototype in 2018
    - ► Serious development in 2019 and 2020
    - ► In production since summer 2020
    - ► Migrated all data from January 2020 onwards

Mapping Karabo to InfluxDB

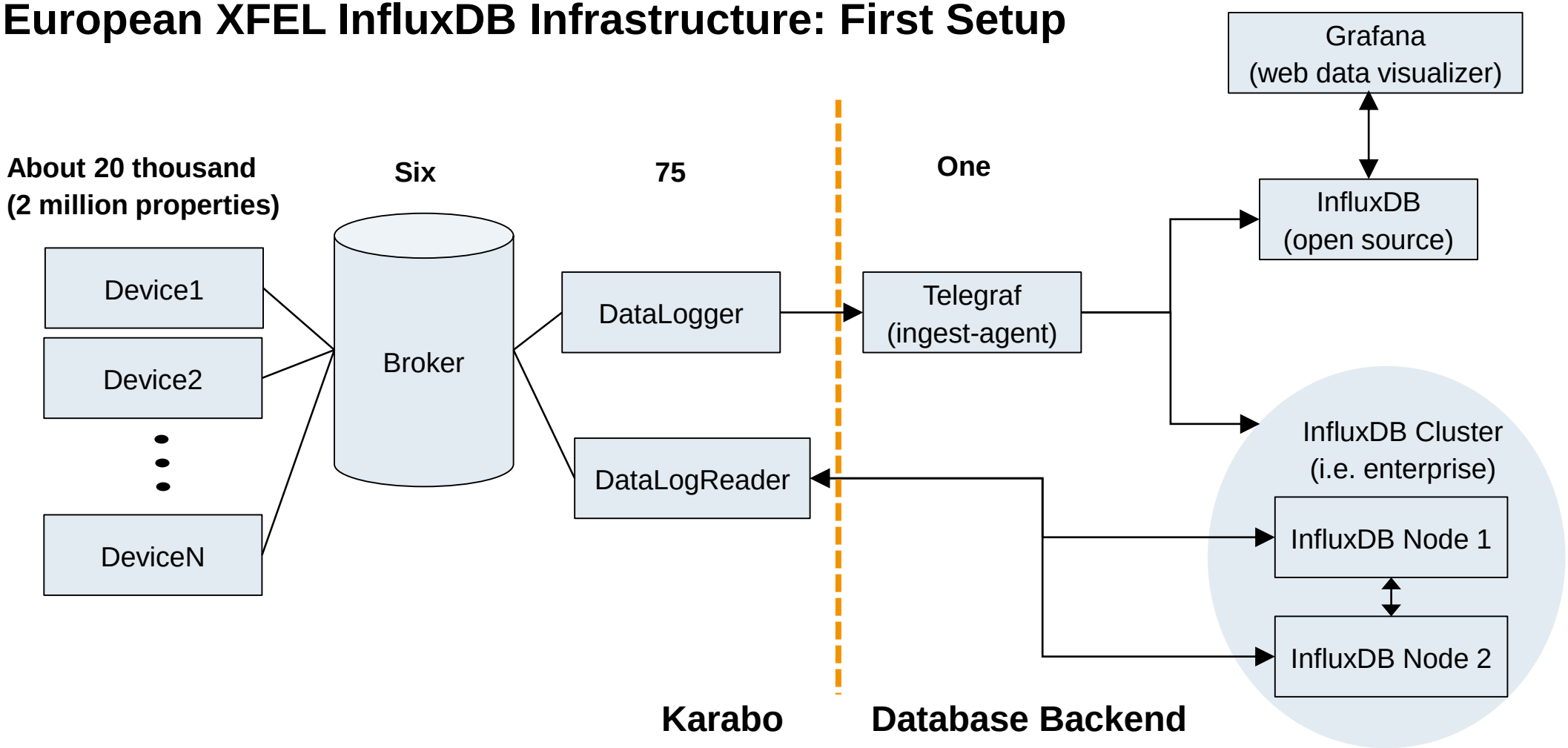
- Karabo device ➔ InfluxDB *measurement*
- Device property ➔ InfluxDB *field*
- Fields must have unique type within a *shard* (i.e. time period) in InfluxDB
  - Karabo device self-description may in principle change any time
    ➔ append type to field name
  - INF/NAN cannot be stored as float
    ➔ extra field *framerate.actual-FLOAT_INF*
- Timestamps are received from the Karabo device
  - Store millisecond precision in InfluxDB
  - If device provides, EuXFEL photon train ID stored as well

InfluxDB Measurement:

time	cameraId-STRING	framerate.actual-FLOAT	…
2022-09-16T10:54:04.12Z	192.67.56.13	0.	
2022-09-16T10:57:05.42Z		10.23554	
2022-09-16T10:57:15.45Z		10.34765	
…			…

Types: Karabo	InfluxDB	
BOOL	Boolean	
FLOAT, DOUBLE	Float (special NaN treatment)	
[U]INT8, … [U]INT32, INT64	Integer	
UINT64	Integer re-interpreted as int64	⚠
STRING	String (mangling escape characters)	⚠
VECTOR_[U]INT8, …	Comma separated string	Karabo-less interpretation
VECTOR_STRING	Base64 encoded JSON string	
CHAR, VECTOR_CHAR (raw data), VECTOR_HASH (table), SCHEMA (self-description)	Base64 encoded (Karabo) binary	Karabo-specific

European XFEL InfluxDB Infrastructure: First Setup



Twelve Communication "Topics"
(6 Instruments, 3 Photon Beamlines, 3 Optical Laser)

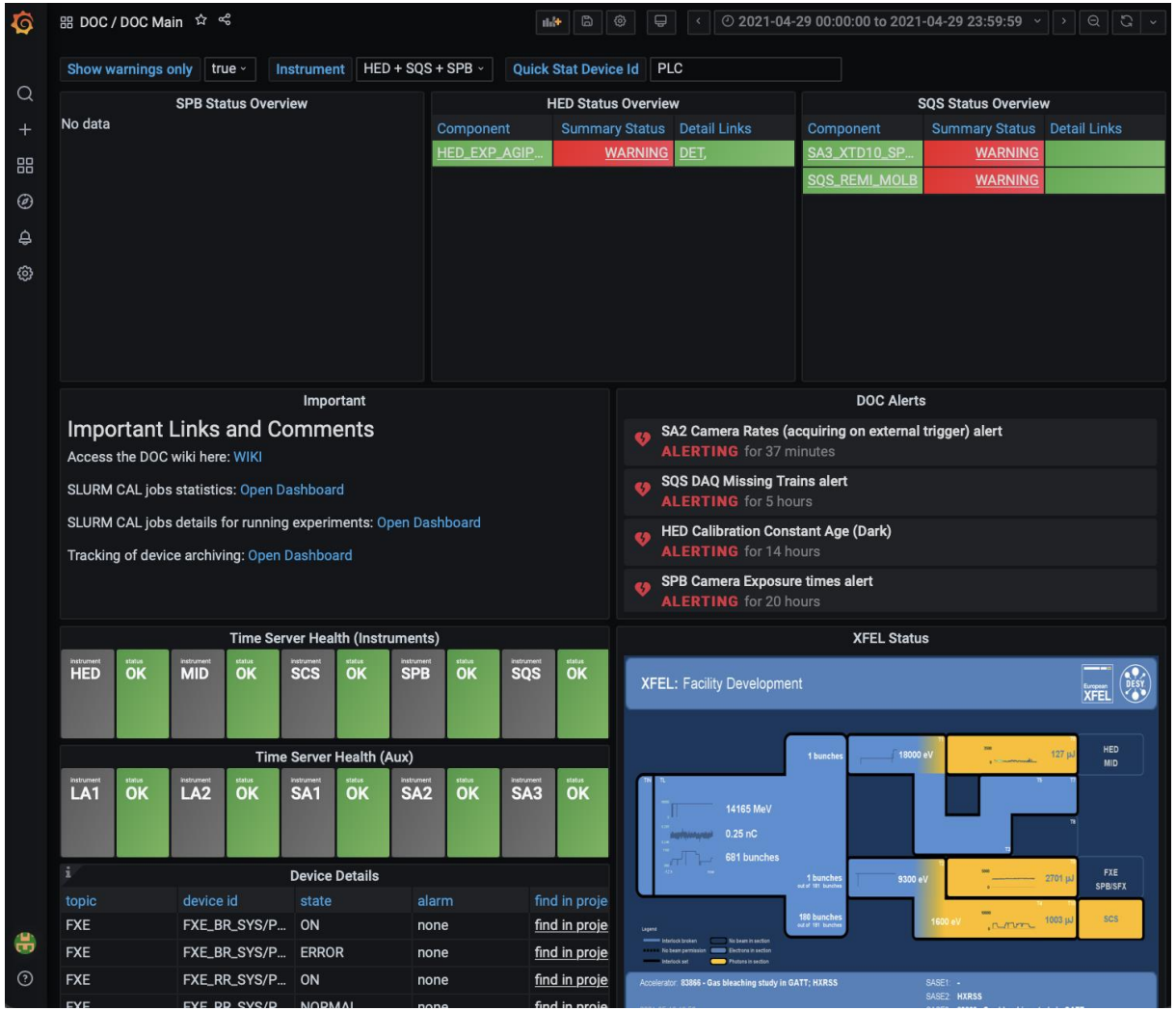
Twelve InfluxDB Databases

Enhancement of Operational Experience with Karabo

- Can provide much longer data availability than previously (was 3 months)
  - 3 years retention period to be become active end of 2022
- Can provide averaging instead of “random” down sampling if data reduction needed
  - More uniform view of data if requested for two slightly different intervals
  - Spikes short in time do not slip unnoticed
- Much faster retrieval of historic trendline data, irrespective how recent (within the 3 years)
  - About 1 second for data of 1 week that updates every 5 seconds (incl. averaging)
- Availability of data for Grafana web frontend opens doors beyond the control system
  - Many visualization options for detailed data analysis
  - **New use case**: Facility monitoring in Data Operation Centre
    ► Data available for monitoring with about 30 seconds delay

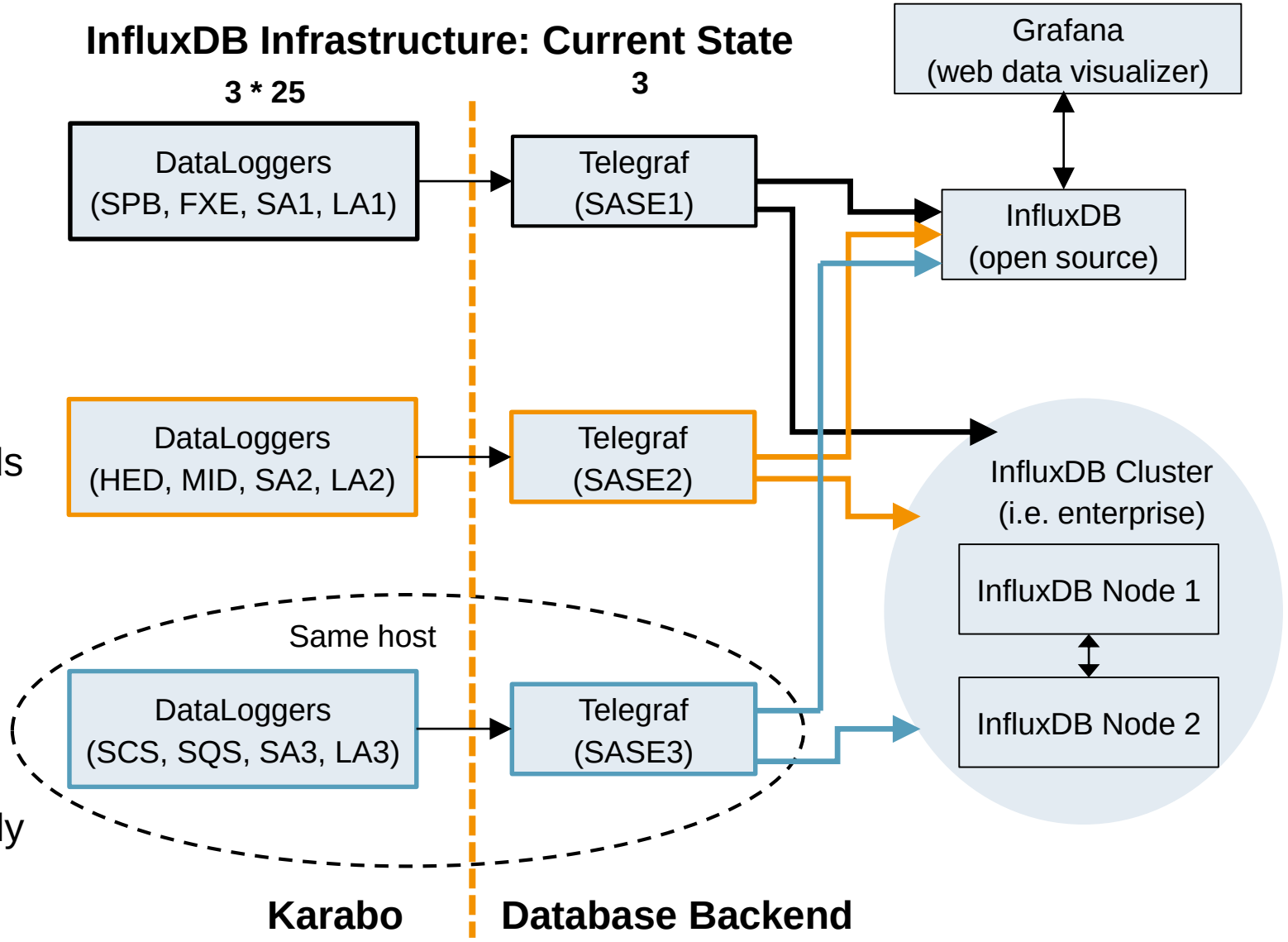
Enabling Centralised Monitoring

- EuXFEL Data Operation Centre: **DOC**
- Unified monitoring and support as a co-effort
  - Controls, Electronics, Data Analysis, IT & Data Management, Detector Operation
- Monitors overall system health via many Grafana dashboards
  - An unforeseen use case when developing InfluxDB logging



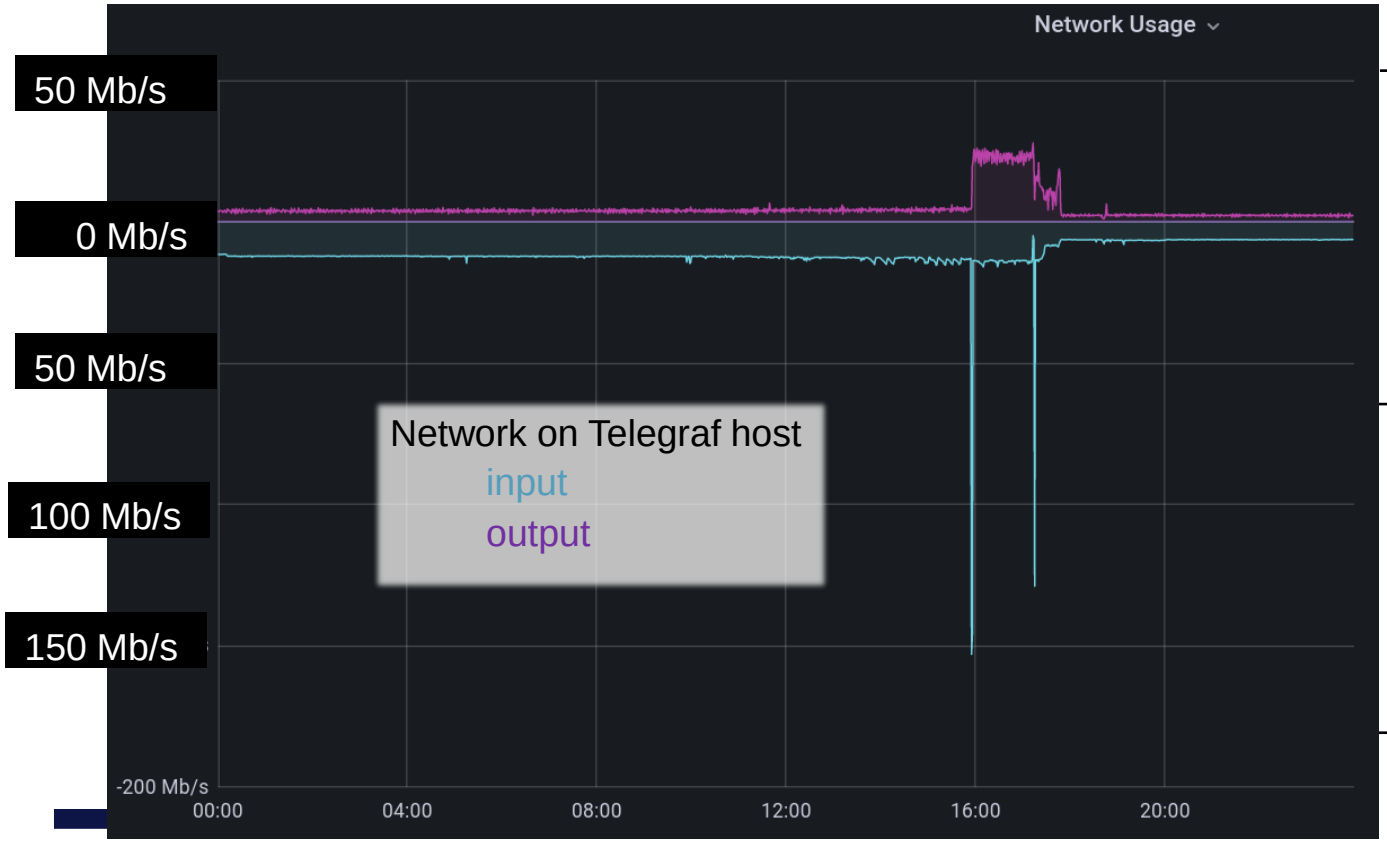
Hiccups on the Way

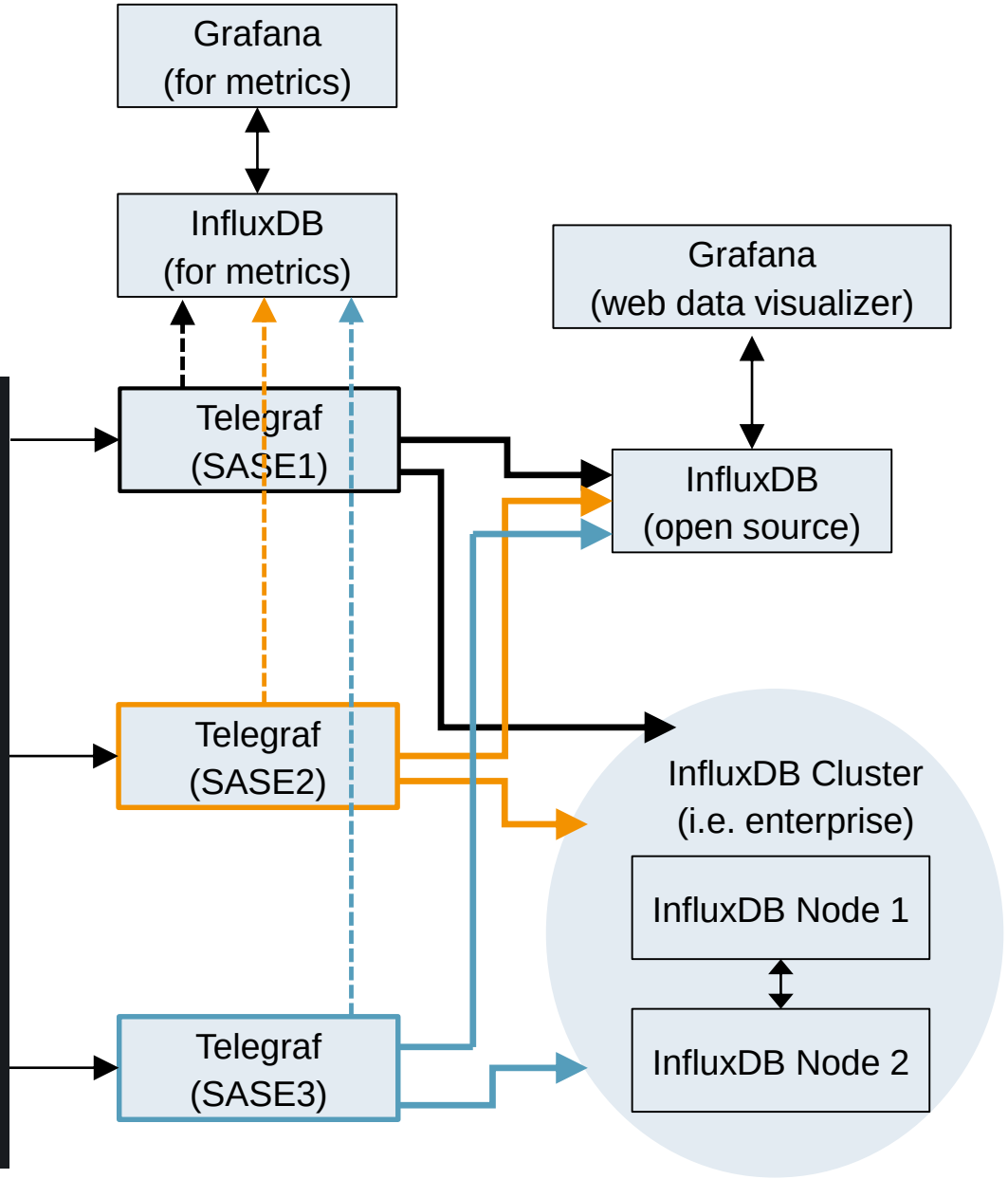
- Beware! Influx is a “sharded” time series database:
  - Data with future (wrong) timestamp compromises performance
  - Had a case with timestamps months in the future:
    - ► Delayed data availability:
      > 1 hour instead of 30 seconds
- “Out of a sudden”, Telegraf failures happened from time to time
  - 1 ➔ 3 Telegraf instances
    - ► separate beam lines
  - Still happened
    - ► at least with partial outage only



Monitor InfluxDB

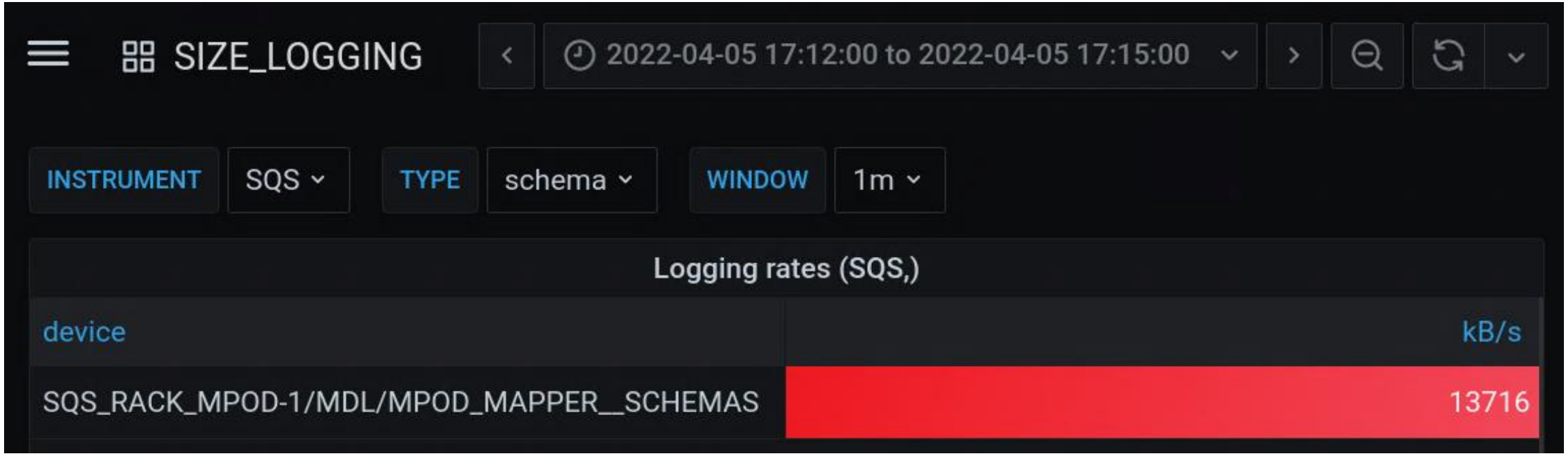
- Metrics of involved servers monitored
  - Network, memory, CPU, etc.
- Stored in an independent InfluxDB
  - Incident on April 5th shows clear network input peak





Analyse Existing Data

- Before the system goes down, some data is stored!
- Flux queries in Grafana allow programming advanced queries:
  - Schema data of a device had a rate of >13 MB/s, averaged over 1 minute!



Findings, Fixes and Protections

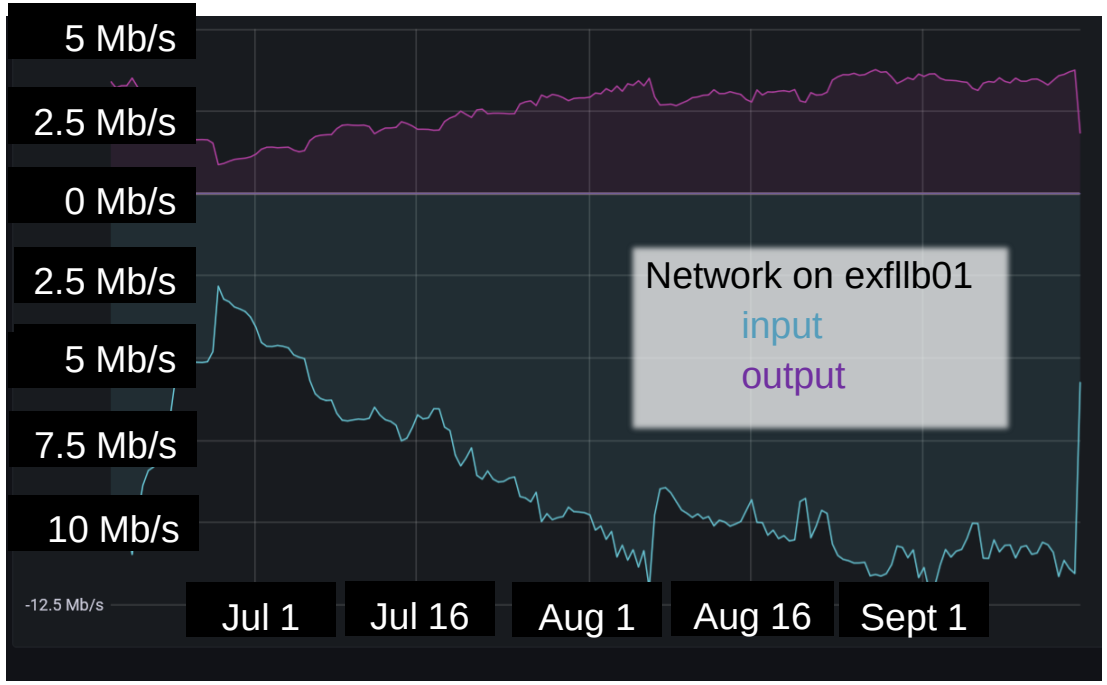
- InfluxDB could not cope with data rate from Telegraf
  - Replied with http error to Telegraf which tried to send data again, and so on…
  - New data arrives at Telegraf and fills its data buffer
    - ► Note: data in buffer is lost if process restarted

- Traced two bugs that together caused very high data rates to storage backend
  - Karabo device level: code updated device schema N=180 times in a loop
    - ► should publish once outside the loop when all information is gathered
  - Karabo data logger *appended* to a buffer that stores the serialized schema
    - ► data was sent $N^2/2$ times – note: single schema can easily be 500 kB
    - ► should fill buffer from scratch

- Incidents showed vulnerability of event driven data logging
  - A single ill-behaving device compromised the logging system
  - New protections on Karabo logger ingestion service:
    - ► refuse vectors longer than 4 * 2700 elements
    - ► refuse property or schema if update rate surpasses 5 MB/s (averaged over 5 seconds)
    - ► Keep track of refusals to identify ill-behaving devices

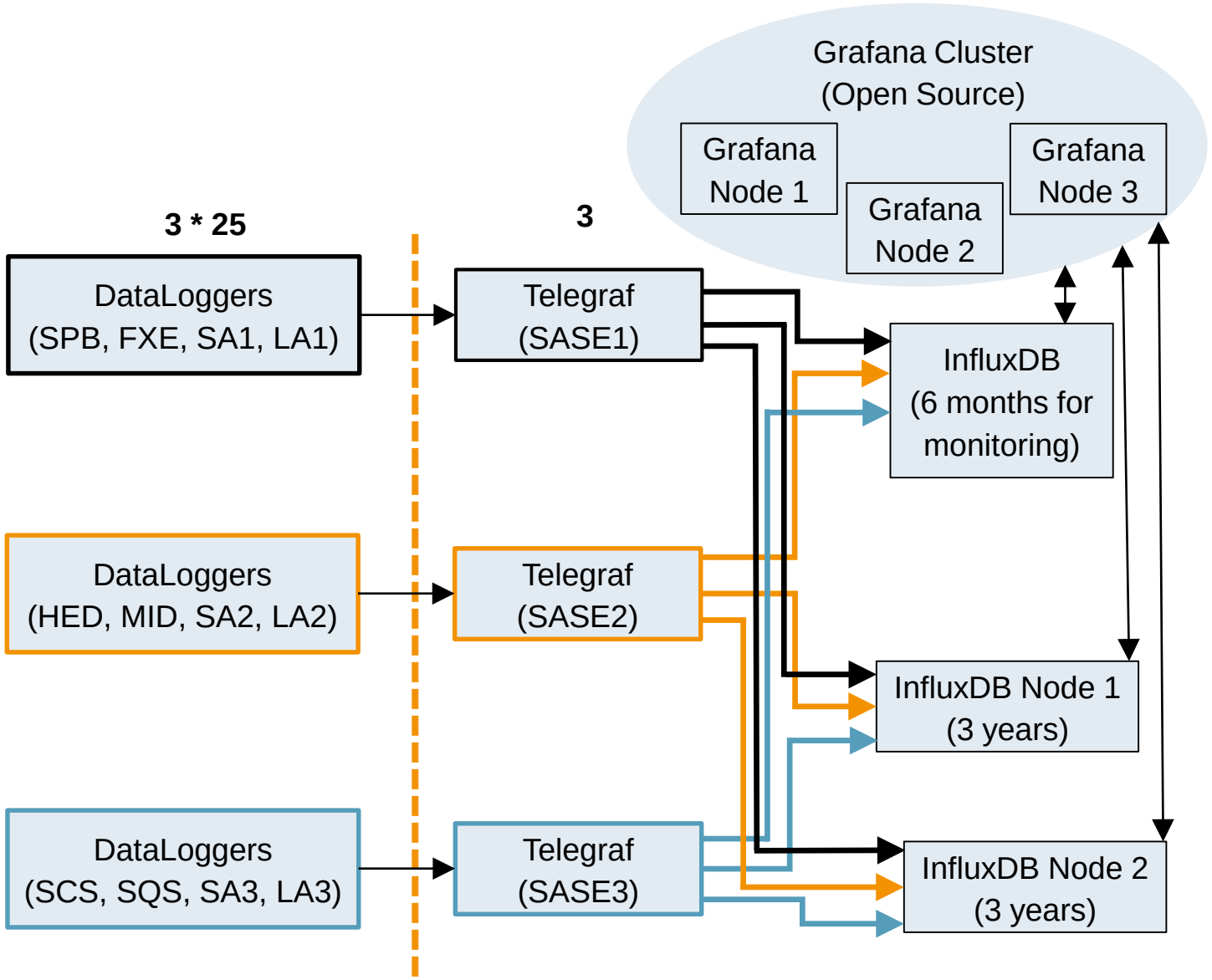
InfluxDB @ EuXFEL in Numbers

- Each Telegraf instance receives roughly 10 Mb/s
  - Duplicated (but zipped) output rate of about 3.5 Mb/s
  - Less when summer maintenance period started (end of June)
- > 240 billion property updates stored in InfluxDB
  - Increase per month: ~ 10 Billion property updates
  - 25 TB per InfluxDB node (Sept. 2022)



Future Steps

- Split cluster into individual nodes
  - Telegraf directly writes to all three
- Reduced data retention (6 months) for InfluxDB dedicated to monitoring
  - Full 3 years available for analysis
- Upgrade to InfluxDB 1.9 enterprise edition
  - Same version everywhere, incl. test instances
  - Quick InfluxData support response
  - May move indexing from RAM to disk
- Grafana visualization consolidated
  - Upgrade to cluster of 3 nodes



Conclusions

- Since 2020, EuXFEL's beam line and instrument control logs slow control data to InfluxDB
  - Both configurations and conditions
  - 3 * 25 TB of influx data on disk, current data input rate about 3 * 10 Mb/s

- Data access experience:
  - Quick retrieval of historic trendline data
  - Availability in Grafana for monitoring enabled EuXFEL to setup its Data Operation Centre

- Logging data to InfluxDB fits well Karabo's event driven approach
  - Automatically logging only when data changes
  - But needs protection against ill-behaving noisy devices
    - ► And - no bugs,  please…

Logging to InfluxDB boosted **availability, accessibility and usage** of historic control data at the European XFEL

Authors

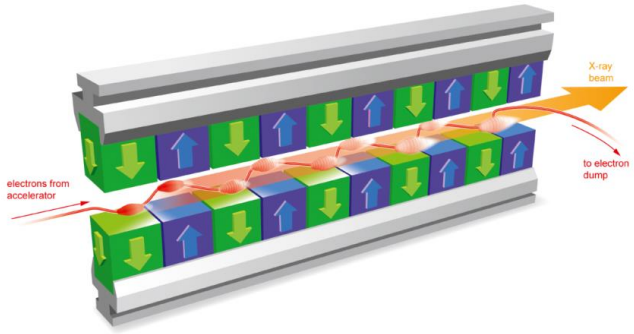
- Members of the **European XFEL** Groups for **Controls** and **IT/Data Management**:
  - G. Flucke, N. Anakkappalla, J. Bin Taufik, V. Bondar, R. Costa, W. Ehsan, S. Esenov, A. Garcia-Tabares, G. Giovanetti, D. Göries, S. Hauf, D. G. Hickin, I. Karpics, A. Klimovskaia, A. Parenti, G. Previtali, A. Samadli, A. Silenzi, M. Smith, F. Sohn, M. Staffehl, J. Szuba, C. Youngman

- **References**
  - Hauf, Steffen, et al. "The Karabo distributed control system." Journal of synchrotron radiation 26 (2019): 1448-1461. doi:10.1107/S1600577519006696
  - InfluxDB, https://www.influxdata.com
  - Grafana, https://grafana.com
  - Flucke, Gero, et al. "Karabo data logging: InfluxDB Backend and Grafana UI“, in proceedings of ICALEPCS2021, JACoW Publishing, doi:10.18429/JACoW-ICALEPCS2021-MOBL04

BACKUP

The Home of Karabo: European X-ray Free Electron Laser (XFEL):

- Linear electron accelerator
  - 10 Hz of “trains” of up to 2700 pulses
  - run by DESY
- Undulators
  creating X-ray laser photons

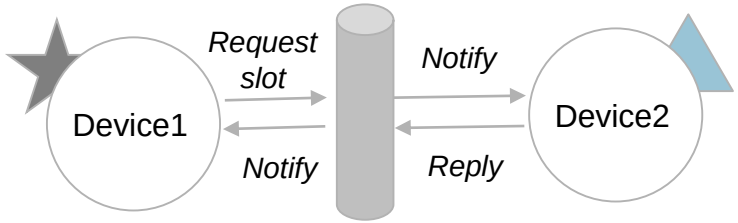


- Photon beam steered
  - through 3 tunnels
  - to 6 instruments

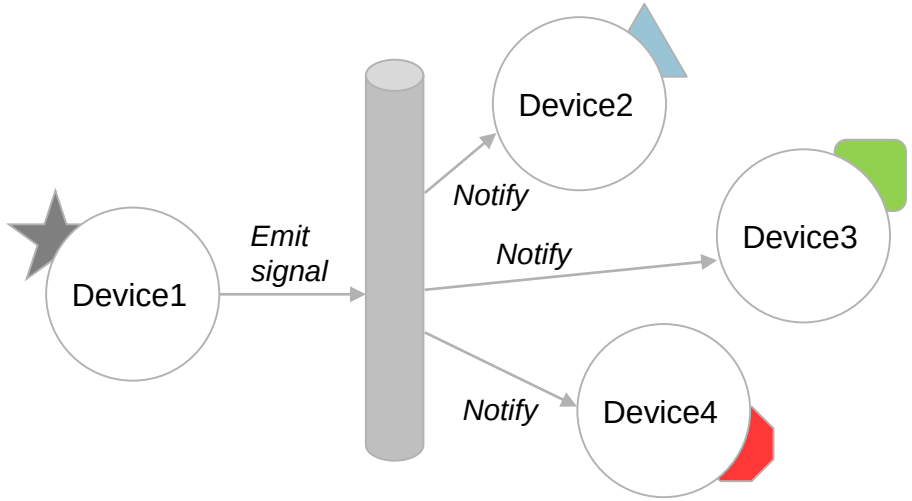
Karabo:
Designed and developed for control, data acquisition, analysis

Karabo Communication Patterns

- Request/reply
  - Device registers methods as “slots”.
  - Call from remote
    - ► with up to four arguments and return values.



- Publish/subscribe
  - Devices subscribe slots to a remote “signal”.
  - When signal is “emitted”,
    all subscribed slots are called.
    - ► No publishing overhead for “popular” devices
    - ► **Karabo framework is completely event-driven**:
      regular **polling obsolete**.



InfluxDB – Estimate of EuXFEL Needs Before Installation

Load	Field writes per second	Queries per second	Unique series
Low	< 5 thousand	< 5	< 100 thousand
Moderate	< 250 thousand	< 25	< 1 million
High	> 250 thousand	> 25	> 1 million
Probably infeasible	> 750 thousand	> 100	> 10 million

From https://docs.influxdata.com/influxdb/v1.7/guides/hardware_sizing/#general-hardware-guidelines-for-a-single-node