1st NFDI4Energy Conference
20 - 21 February 2024
Leibnizhaus, Hannover

Metadata Extraction from a Huge Time Series Database

Andreas Schmidt, Mohamed Anis Koubaa, Nan Liu, Karl-Uwe Stucky, and Wolfgang Süß

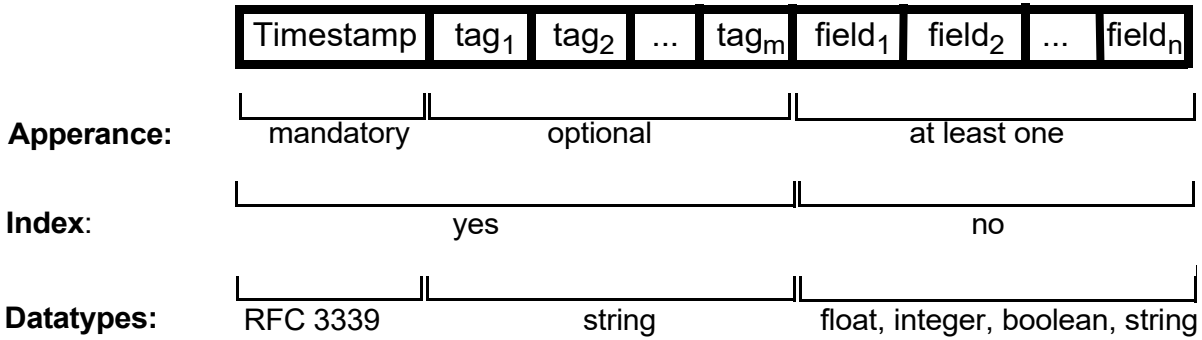
Institute for Automation and Applied Informatics (IAI)

Outlook

- Background
  - Influx Database at KIT
  - Zeitgeist
- Concept of MetadataExtractor
- Architecure
- First results
- Planned conceptual extensions
- Summary and Outlook

Influx Database

- Schema-free database, optimized to store time based datasets (precision: ns)
- Standalone/Cluster (sharding/replication)
- Up to 700.000 datasets/s for a single node
- data organized in *measurements* (comparable to a table)



- Retention policy, Continuous queries
- Query languages:
  - InfluxQL (SQL-like, limited range of functions)
  - Flux (stream-based)
- Programmatic interface: REST-API as well as multiple language bindings

Zeitgeist

- Easy publication of time series data in RO-Crate Format (data & metadata)
- Metadata comes from Influx Metadata API & manually specified (ORCID, ROR, licence, QUDT.org, description)
- Interactive specification of data to export/publish
  - restrictions on time range
  - restrictions on tag values
- Screenshots or live demo ...

Demo ...

- Actual limitation: Only one time series per export
- Goal: Bundle of multiple semantically related time series in one RO-Crate
- Question: How can we support the user to find related time series?

Zeitgeist - Export Configurator

Datasource:

Configuration: efficio_raw (DB: fm_efficio_mirror) ☐ (force to read metadata from Influx DB)

database: fm_efficio_mirror
Measurement: efficio_raw
Time-range: [2020-12-31T23:15:00Z .. 2024-02-20T18:03:16Z]
Duration (days): 1145
Fields: displayDelta, displayEnd, displayStart, value

Export Definition:

Metadata:

ORCID: 0000-0003-2785-7736

Wolfgang Süß
KIT

License: BSD Zero Clause License

Server side parameter specification: ☐

Output format:

- ◉ RO-Crate 1.1
- ○ CSV

Description :

bla bla bla ... metadata ohah !!!!

Filters:

Time-Interval:

Start: 2024-02-20T17:03:16Z (UTC)

End : 2024-02-20T18:03:16Z (UTC)

Tag-Filter:

- ☐ SubCounter
- ☑ building
  - 444
  - 445
  - 448
  - 449
- ☐ deactivated
- ☑ energyType
  - Brauchwasser
  - Druckluft
  - Durchflussmenge
  - Fehlwasser
- ☐ factor
- ☐ id
- ☐ installationPlace
- ☐ measurementUnit
- ☐ measuringValueType
- ☐ meteringType
- ☐ name
- ☐ parentName
- ☐ pointType
- ☐ raster
- ☐ unit

build export-URL

Export Definition:

Metadata:

ORCID: 0000-0003-2785-7736

Wolfgang Süß
KIT

License: BSD Zero Clause License

Server side parameter specification[?]: ☑

Output format:

- (•) RO-Crate 1.1
- ( ) CSV

Description [edited]:

```
bla bla bla ... metadata ohah !!!!
```

Filters:

Time-Interval:

Start: 2024-02-20T17:03:16Z (UTC)

End : 2024-02-20T18:03:16Z (UTC)

Tag-Filter:

- ☐ SubCounter
- ☑ building
  - 441
  - 443
  - 444
  - 445
- ☐ deactivated
- ☑ energyType
  - Brauchwasser
  - Druckluft
  - Durchflussmenge
  - Fehlwasser
- ☐ factor
- ☐ id
- ☐ installationPlace
- ☐ measurementUnit
- ☐ measuringValueType
- ☐ meteringType
- ☐ name
- ☐ parentName
- ☐ pointType
- ☐ raster
- ☐ unit

build export-URL

Download URL:

https://localhost/zeitgeist/export.php/fm_efficio_mirror.efficio_raw.65d51da560add.zip add to cache

(config file)

InfluxQueryString:

```
select *
from "efficio_raw"
where time >= '2024-02-20T17:03:16Z'
and time <= '2024-02-20T18:03:16Z'
and ("building" = '445' or "building" = '449')
and ("energyType" = 'Druckluft')
order by time
```

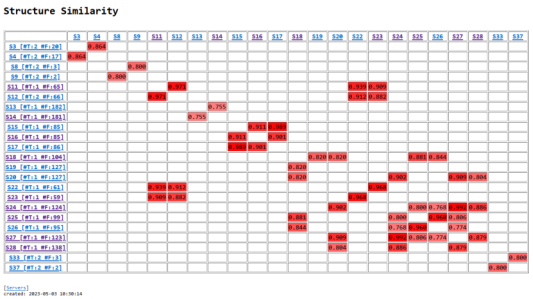
Zeitgeist
RO-Crate Result

Metadata Extraction

General Idea:

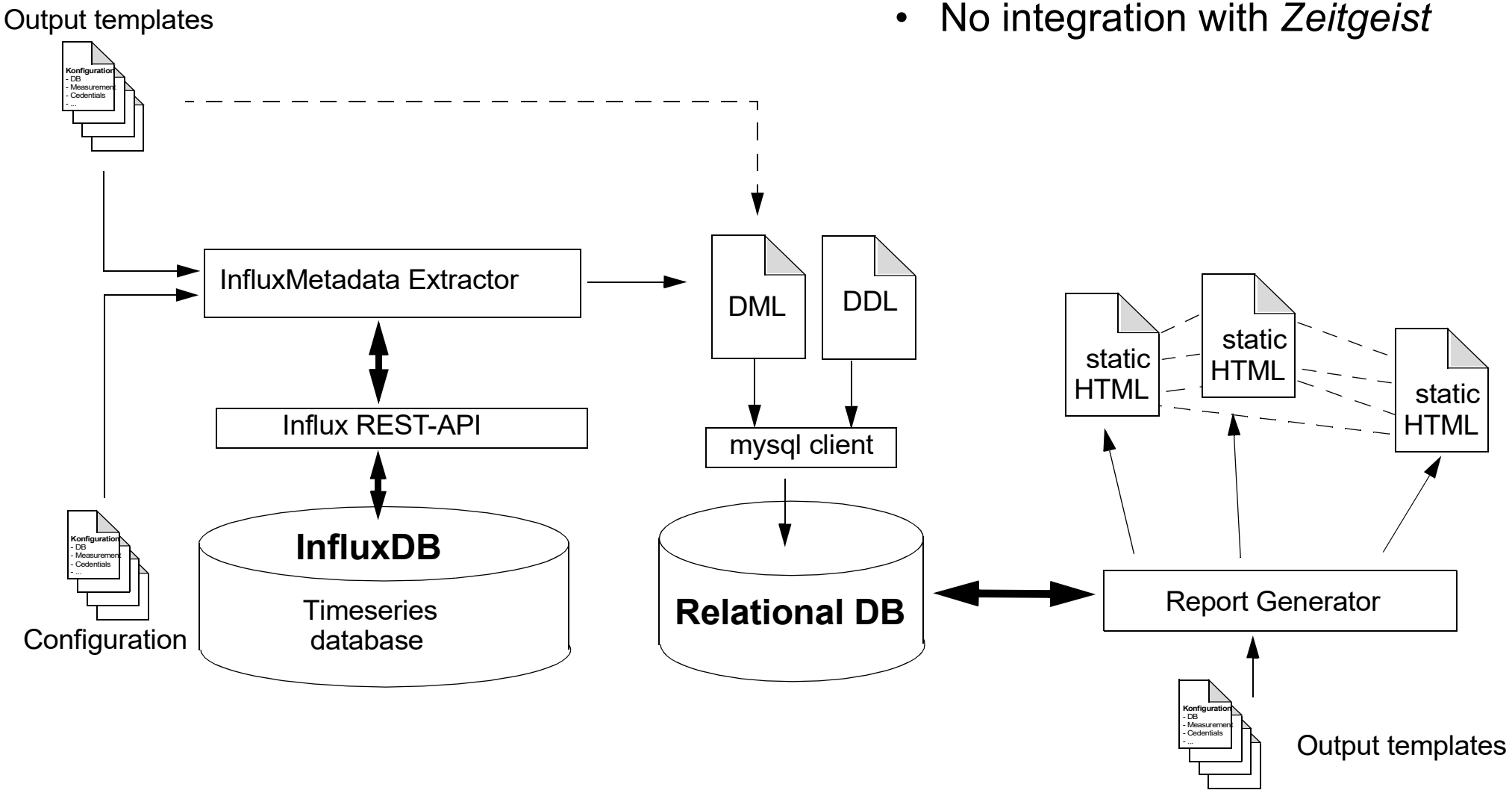
- Build a tool that extracts the available metadata from the Influx DB Cluster at KIT
  - Extract all databases and measurements
  - Extract possible tag keys and values
  - Extract field names and their datatypes
  - Extract range of numerical field values
  - Extract cardinality (how many different combinations of tag values)
- Analyse data structures to find similar data/semantically related data
- Live example (navigation inside results)

Demo ...



Actual Architecture

- For human inspection only
- No integration with *Zeitgeist*



First Findings

- 20 different databases
- temporal scope between 0 and 15 years
- Large number of modeling variants
  - up to 5000 measurements per database
  - up to 180 different fields per measurement
  - up to 26 tags per measurement

- Extreme examples:
  - measurement name combined with UUID
  - one measurement for a whole working group
  - for a single dataset: only one (of many fields) contains a value

First Results

- 20 different databases
- temporal scope between 0 and 15 years
- Large number of modeling variants
  - up to 5000 measurements per database
  - up to 180 different fields per measurement
  - up to 26 tags per measurement

- Examples:
  - measurement name combined with UUID
  - one measurement for a whole working group
  - for a single dataset: only one (of many fields) contains a value

direct consequence of schema free database

nevertheless, there are recommendations on how to build schemas

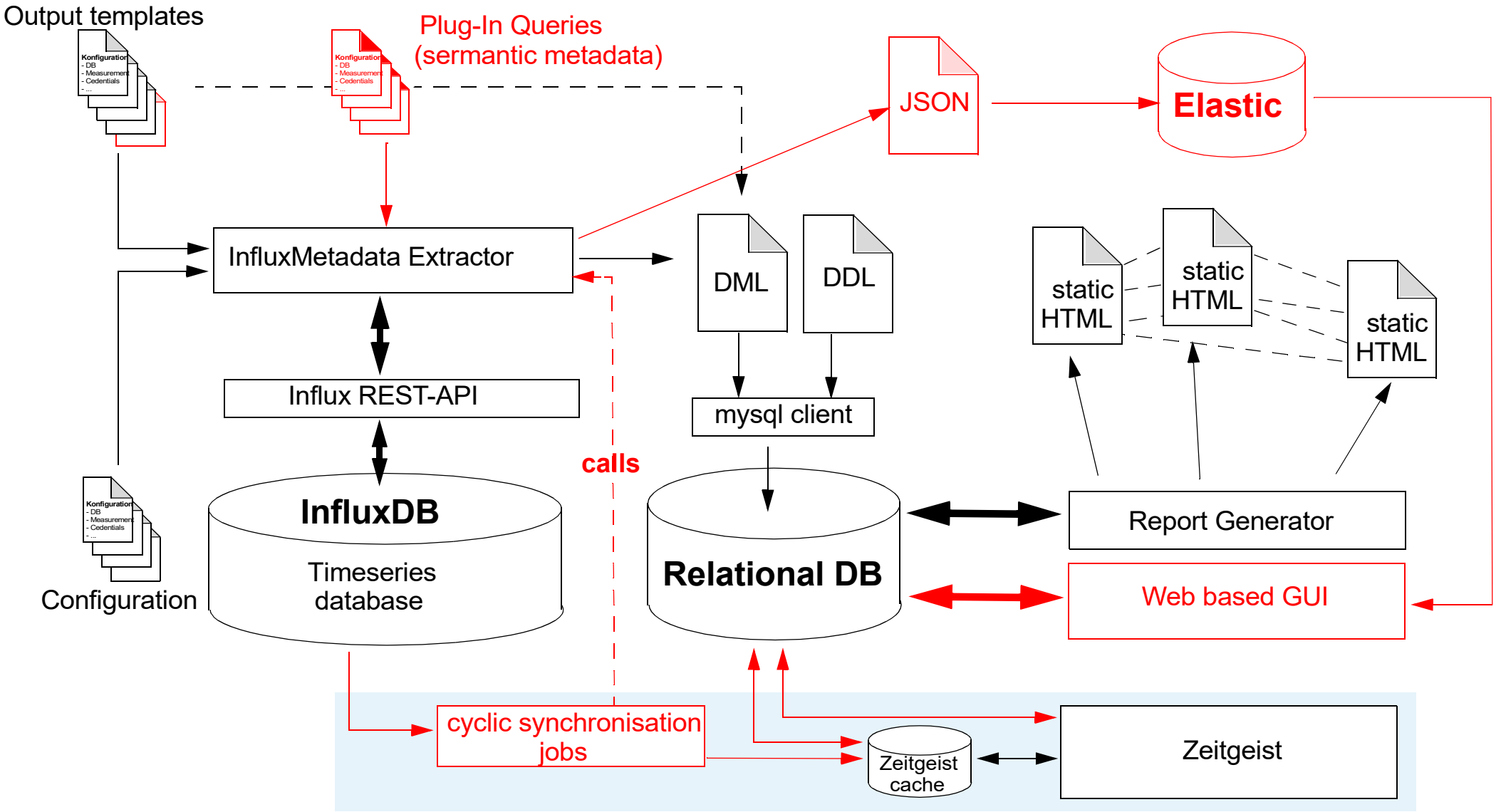
First Results

- Searching for relationships between data records using the data structure of a measurement is difficult
  - several unrelated time series in one measurement
  - related time series in multiple measurements
  - probably evolution of data structures over time

- More sophisticated algorithms needed:
  - search for tags, fields that occur together in a dataset
  - examination of the recorded interval
  - Look for similar values of tags (i.e. a sensor name, location)
  - Integrate the user experience (domain expert)
  - ...

  Question: can a „relatedness measure“ be calculated?

Planned Architecture



Performance Aspects

- Influx Meta API
  - Metadata calls are typically fast (< 0.05 sec.), but there can be exceptions:
  - Exceptions:
    - Extraction of field names can be slow (up to 30 s) - depending on number of shards
    - Extraction of aggregates (min/max/count) can be expensive (up to 5 s) on many (~100) fields
  - But: data structures typically change slowly
  - Cyclic job that queries the Influx DB and updates metainformation in relational database (+ Zeitgeist metadata cache)

Summary

- Tool for extraction of meta information from a schema-free time series database
- Primary goal: search for related/similar data (based on structure) for our *Zeitgeist* tool
- The results are currently presented as static web pages in which you can navigate
- Difficult to search for relationships between data sets based on the structure of a measurement
- In order to be used meaningfully in the context of *Zeitgeist*, more in-depth analyses than the analysis of Influx Meta-API results are necessary

- Secondary goal: Stand-alone tool to analyze the structure of a schema free InfluxDB

Outlook

- Next version of MetadataExtractor will cover:
  - Data set related structural analysis (incl. structure migration)
  - Considering the recording times
  - Integration of domain expert knowledge
  - Interactive frontend with search capabilities
  - Near realtime statistics (cyclic analysis)

- Further ideas
  - calculate measure about modelling quality