StreamConnect: Ingesting Historic and Real-Time Data into Unified Streaming Architectures

Philipp Zehnder and Dominik Riemer
FZI Research Center for Information Technology

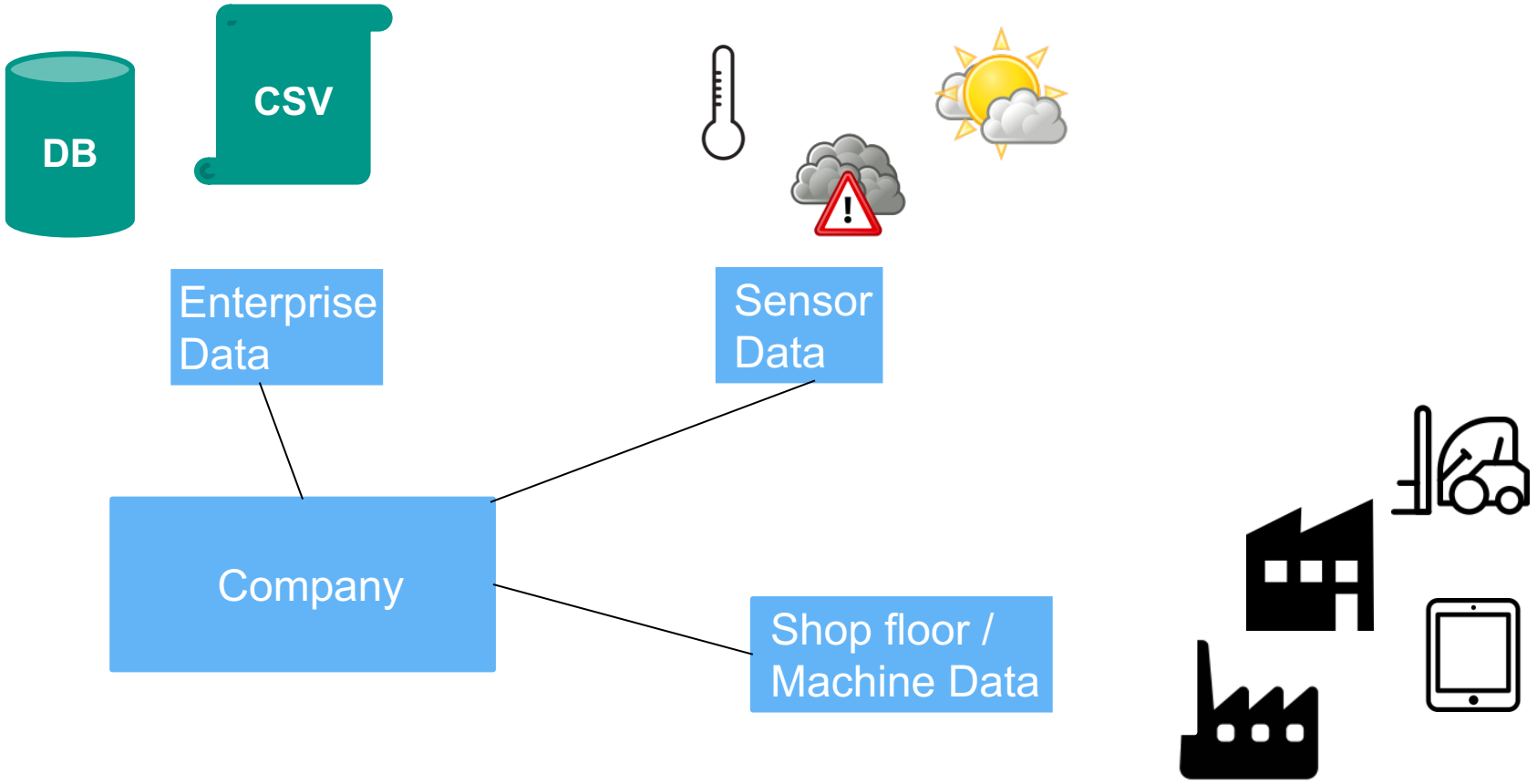
Workshop: WSP ISWC2017
22.10.2017

Motivation

Motivation:
Ingesting Heterogeneous Data Sources

Example

A manufacturing company wants to use all available data sources



Motivation (Current Situation)

Data Sources

- Specific adapters for real-time and historic data
- Often implemented for special use cases and not re-usable in another context [1]
- High technological understanding required to implement such adapters

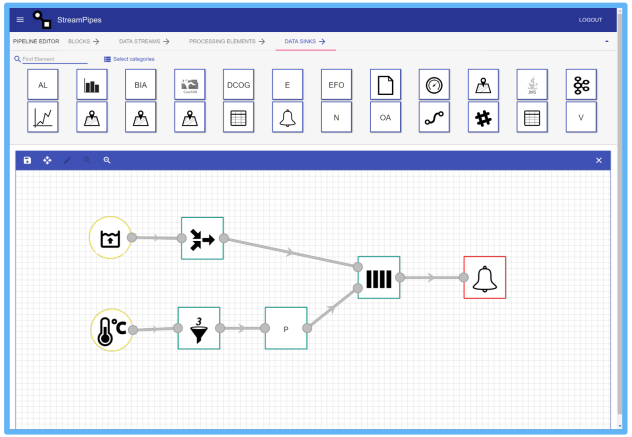
Related Work

- Xgsn [4] Sensor middleware for OpenIoT [5]
  - Our approach is independent from runtime format, can be used with any broker or processing engine
- ETL Tools (Talend) [6]:
  - Does not use semantics of data

Data Processing

- Unified processing engines are capable of processing historic and real-time data [2]

- Self-Service solutions for stream processing (e.g. StreamPipes) [3]



Motivation
(To Be Situation)

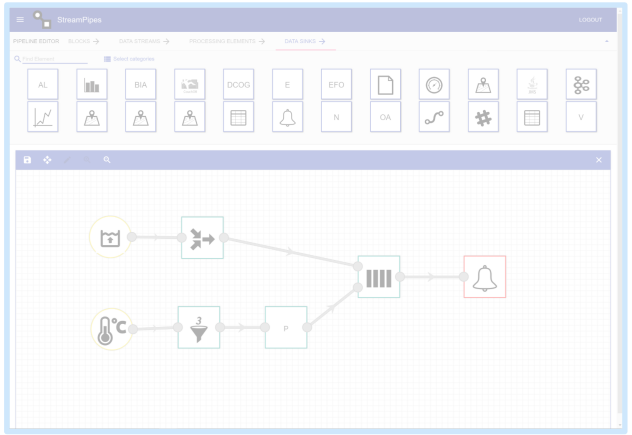
Data Sources

- Unified adapters for real-time and historic data
- Re-usable adapters for different domains and different kind of data
- Domain expert should be able to configure data ingestion from new data sources
- Adapter should be able to semi-automatically preprocess data

Data Processing

- Unified processing engines are capable of processing historic and real-time data [2]

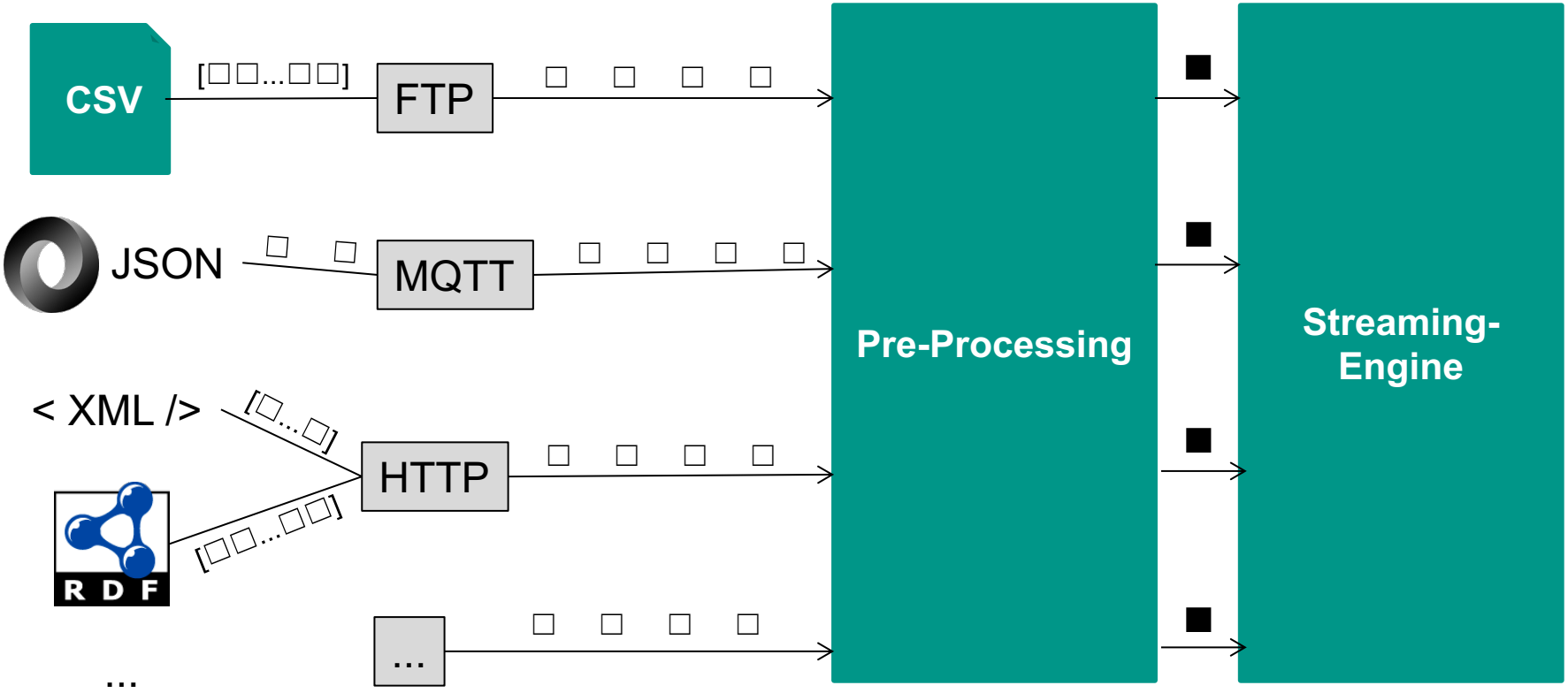
- Self-Service solutions for stream processing (e.g. StreamPipes) [3]



Objective

Objective

Creating an adapter framework that can be used by domain experts and that is capable of handling historic data and real-time data streams.



Challenges

Challenges

Technical challenges and requirements when developing an adapter concept

Temporal Aspect

Adapters need to be able to handle both real-time and historical data.

Adapter Configuration

A solution must provide a higher level of abstraction to enable domain experts to configure adapters, which still requires a lot of technical understanding.

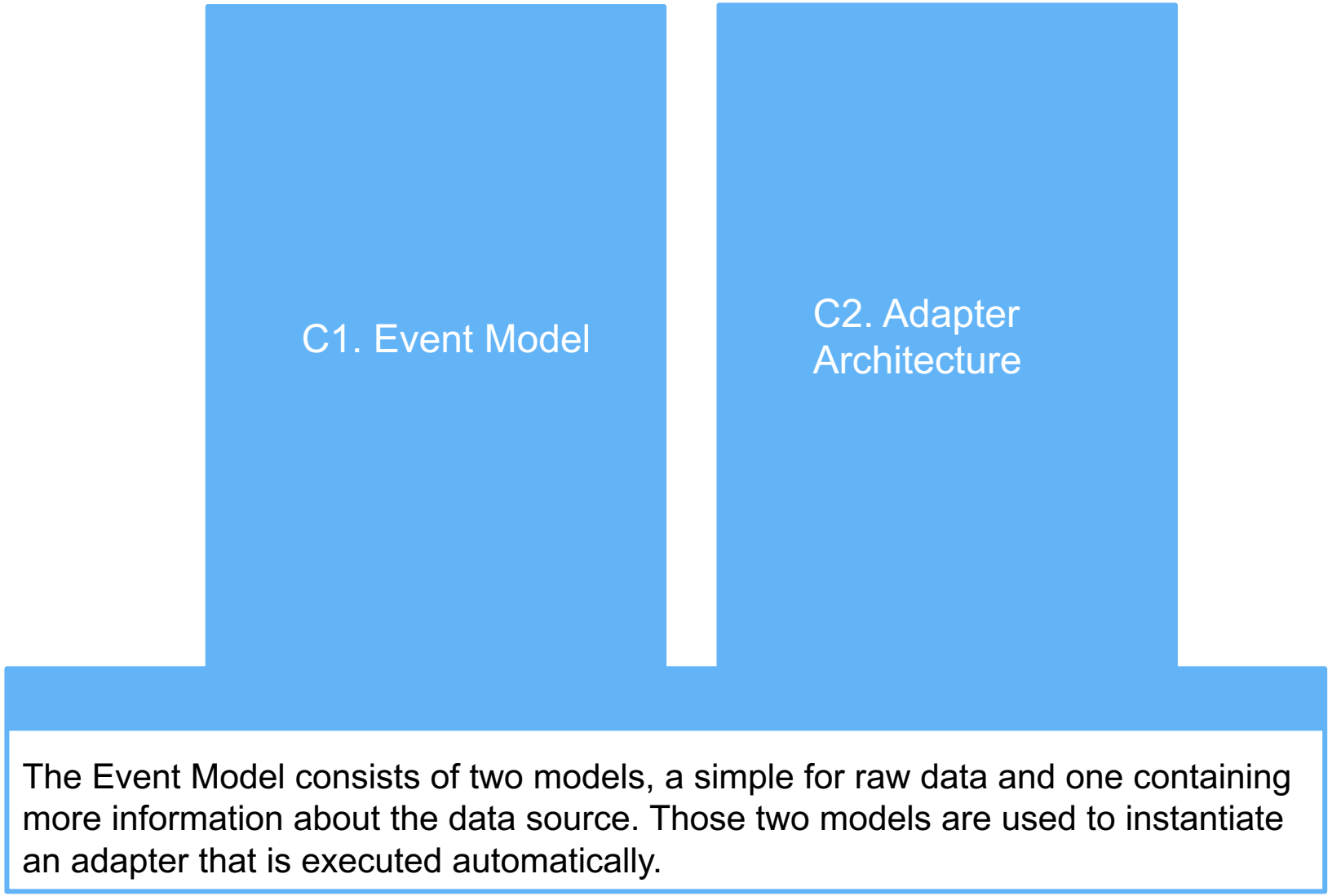
Data Cleaning

Since adapters are often long running processes, they should ensure that the quality of the data does not change over time.

(Edge)Pre-Processing

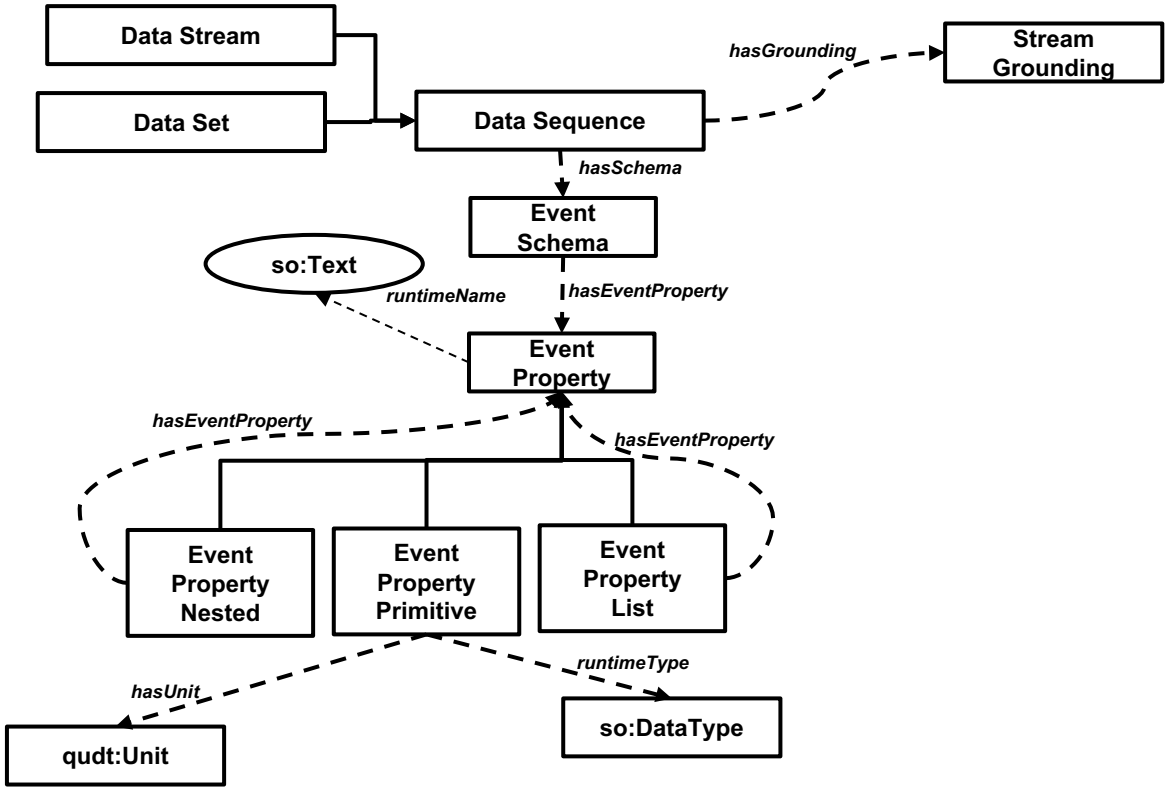
Simple pre-processing steps such as filtering, transforming or aggregating data should be executed locally close to the sensor to avoid sending noisy data to the messaging system.

Contributions

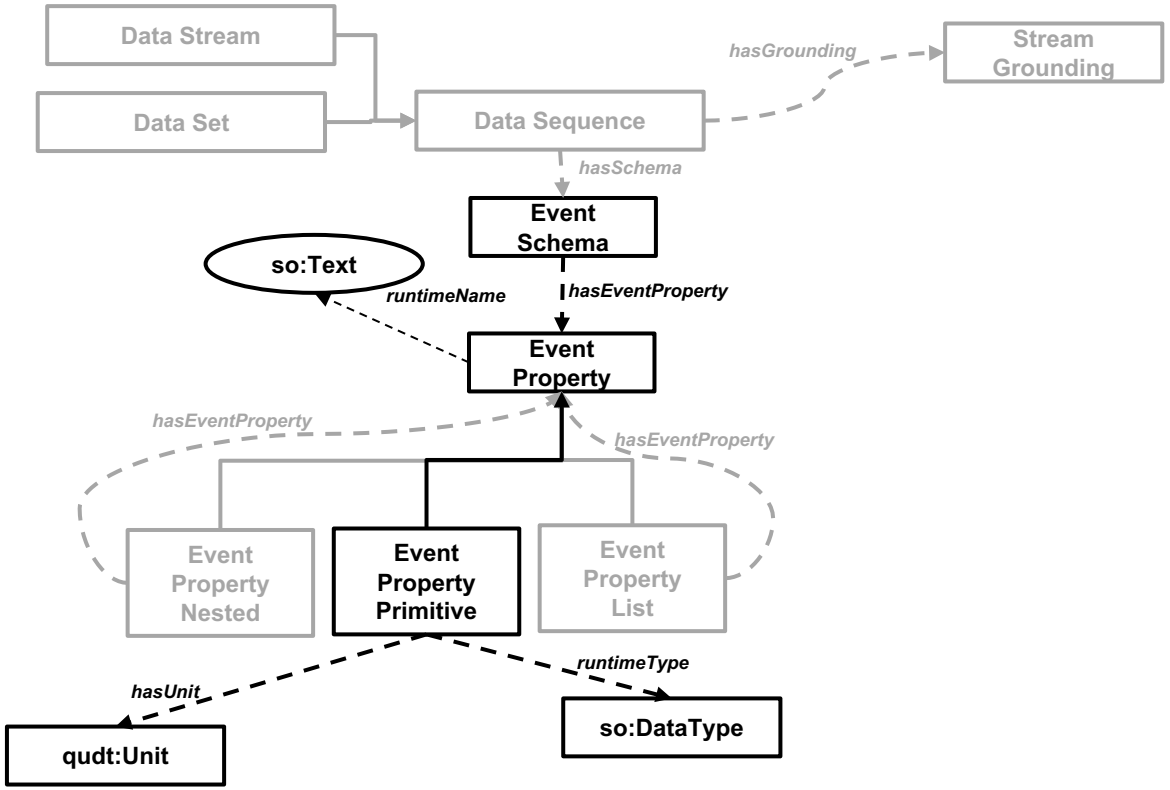


Event Model

Event Model (Raw Data)



Event Model (Raw Data)



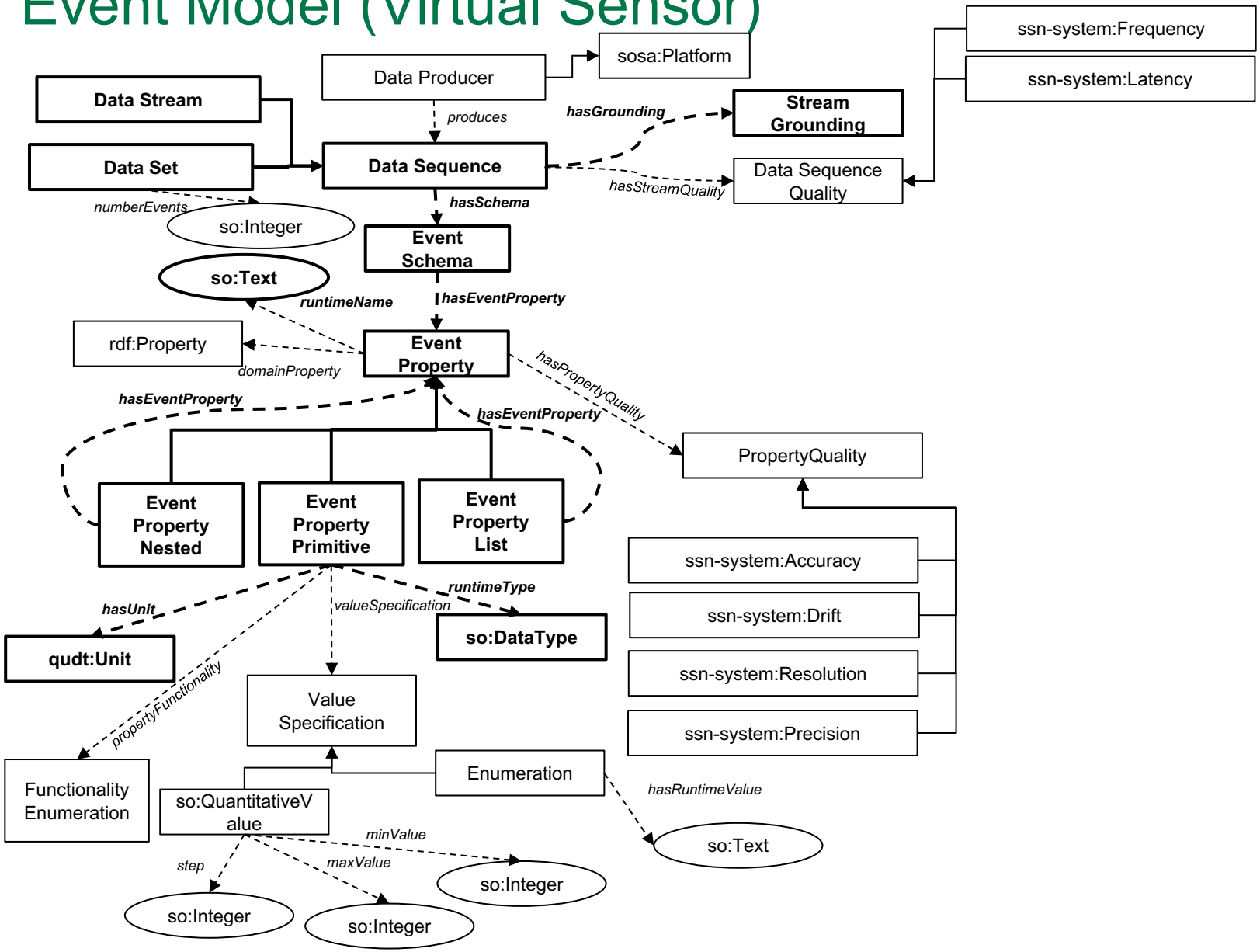
Event Model:

```
<eventSchema1> a :EventSchema
    :hasEventProperty    <prop1>,
                         <prop2>

<prop1> a :EventPropertyPrimitive
    :runtimeName    “timestamp”
    :runtimeType    so:Text

<prop2> a :EventPropertyPrimitive
    :runtimeName    “temperature”
    :runtimeType    so:Integer
    :hasUnit.       qudt:DegreeCelsius
```

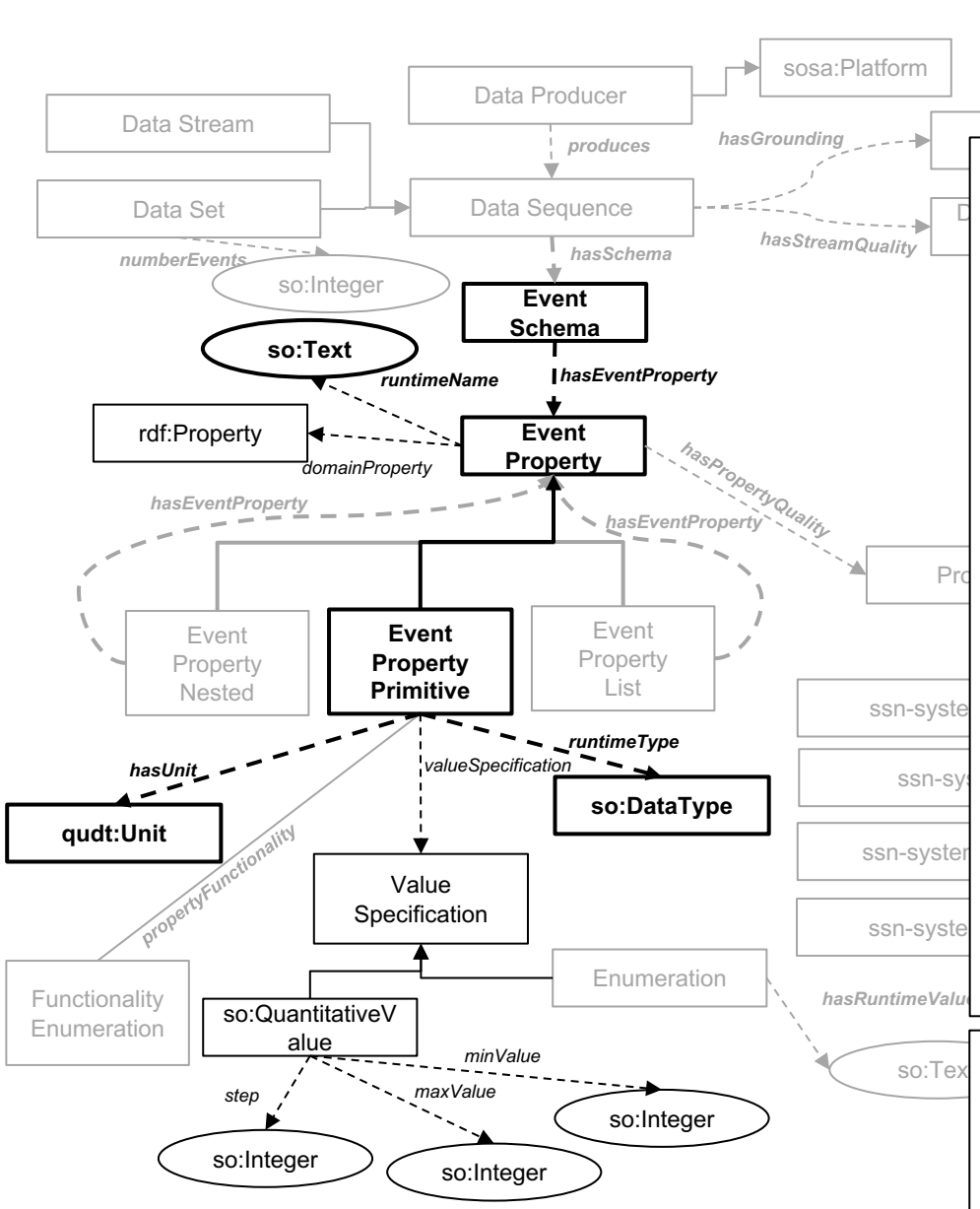
Runtime Event:

```
{
    ”timestamp”: “2017-10-22T16:00Z”
    “temperature”: 32
}
```

Event Model (Virtual Sensor)



Event Model (Virtual Sensor)



Event Model:

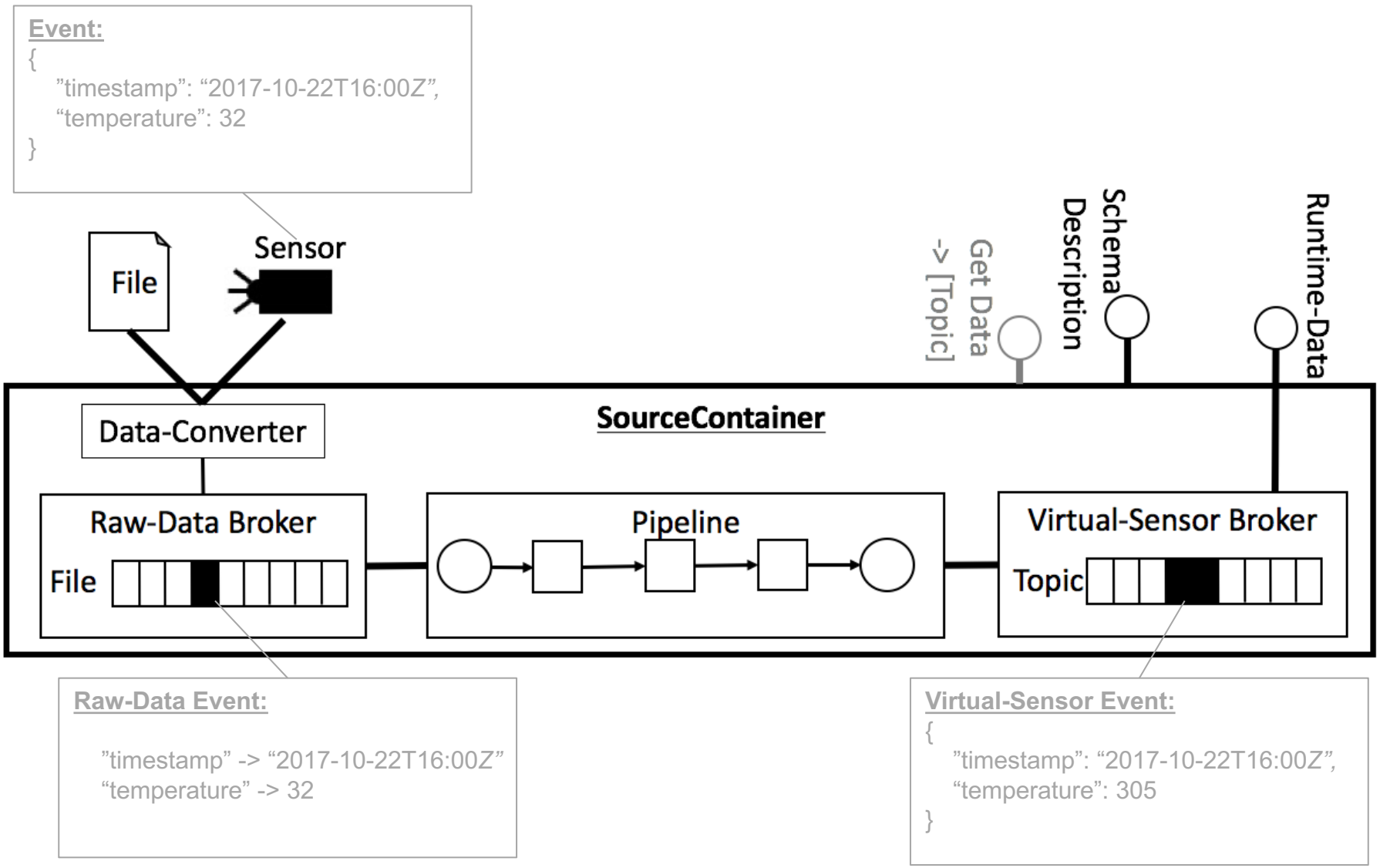
```
<eventSchema1> a :EventSchema
   :hasEventProperty    <prop1>,
                        <prop2>

<prop1> a :EventPropertyPrimitive
   :runtimeName      “timestamp”
   :runtimeType        so:Text
   :domainProperty    so:DateTime

<prop2> a :EventPropertyPrimitive
   :runtimeName        “temperature”
   :runtimeType          so:Integer
   :hasUnit.                qudt:Kelvin
   :domainProperty      sp:temperature
   :valueSpecification <valSpec>

<valSpec> a so:QuantitativeValue
   :step              1
   :minValue  263
   :maxValue 313
```

Runtime Event:

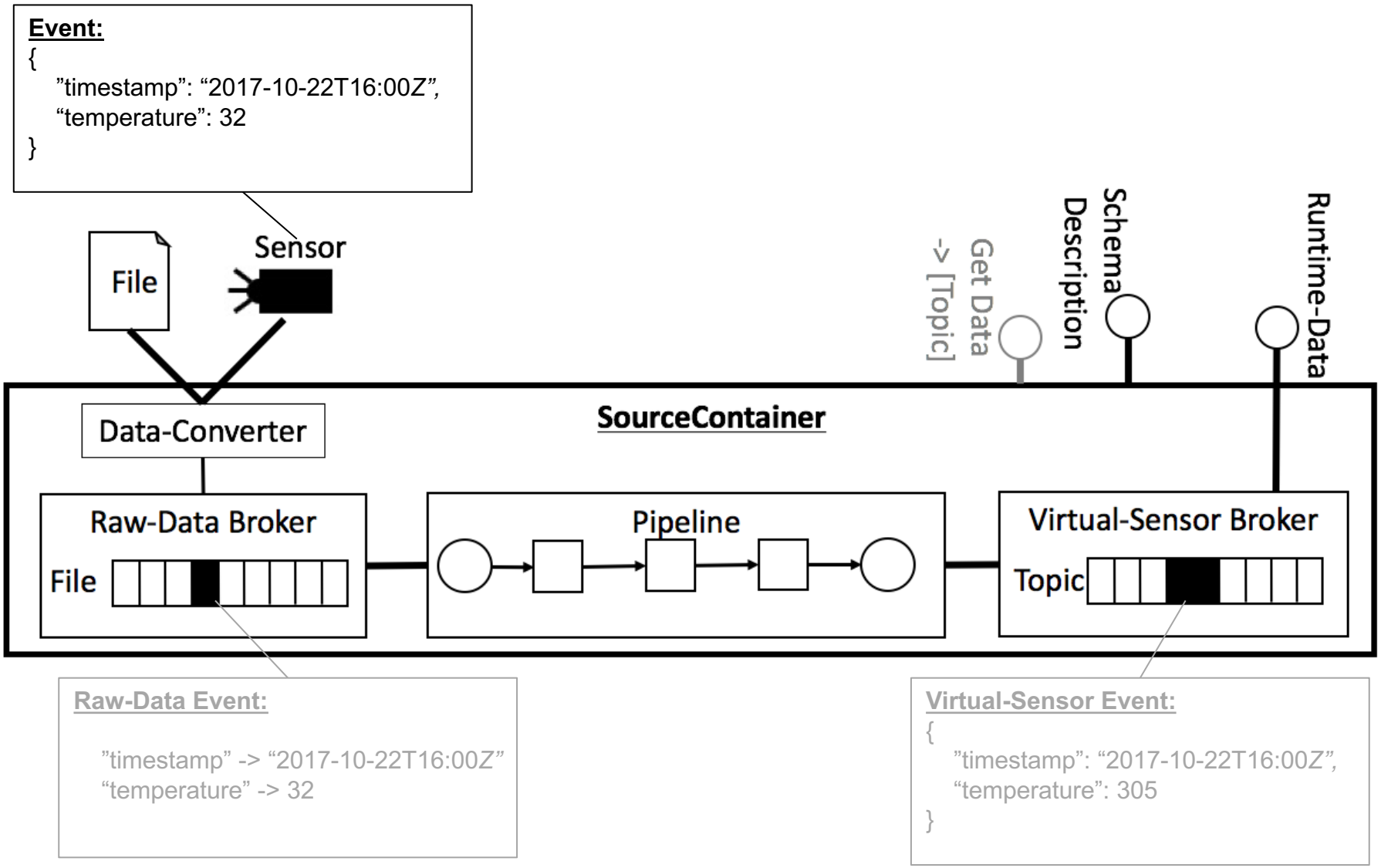
```
{
   ”timestamp”: “2017-10-22T16:00Z”
   “temperature”: 305
}
```

Adapter Architecture

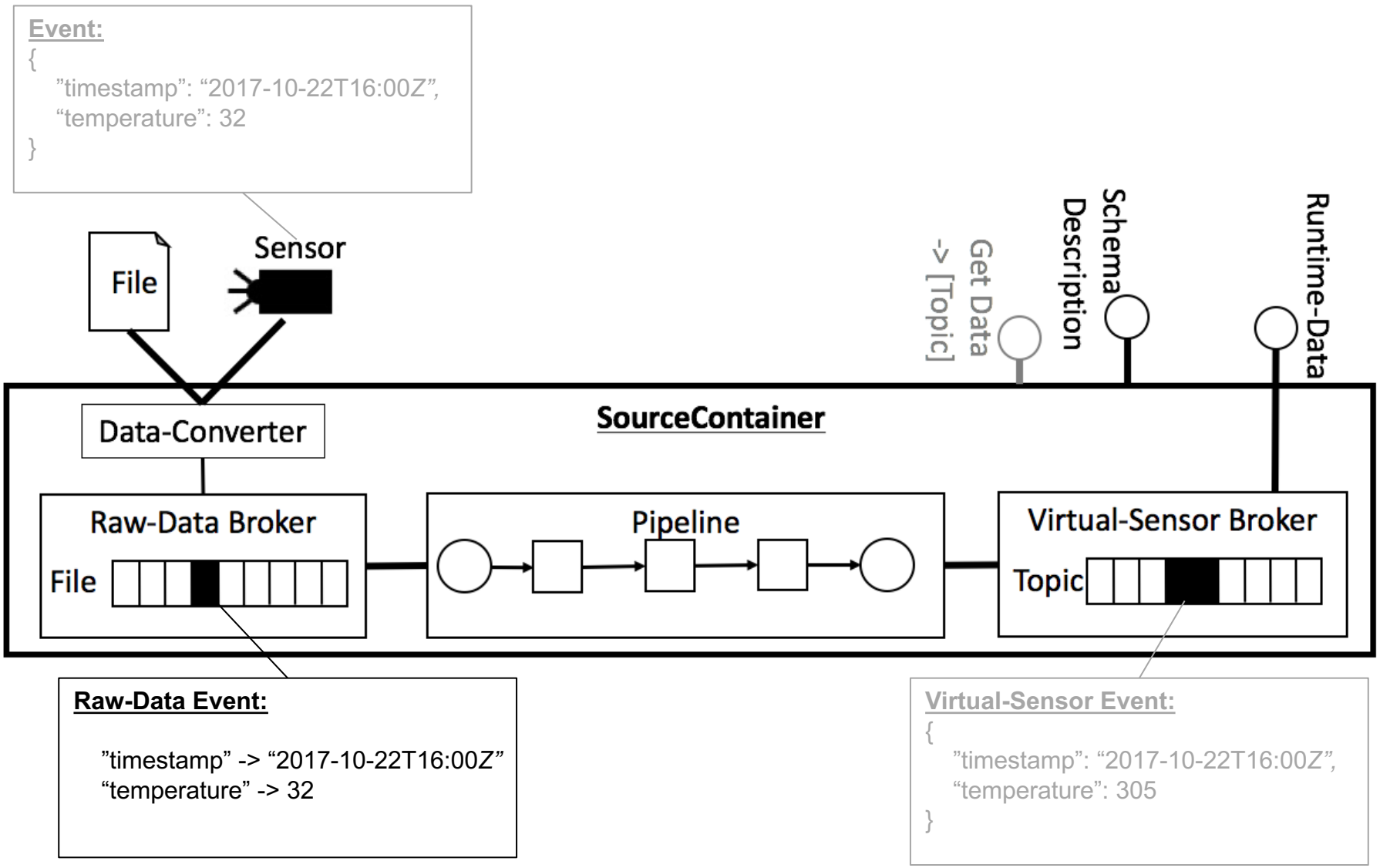
Adapter Architecture



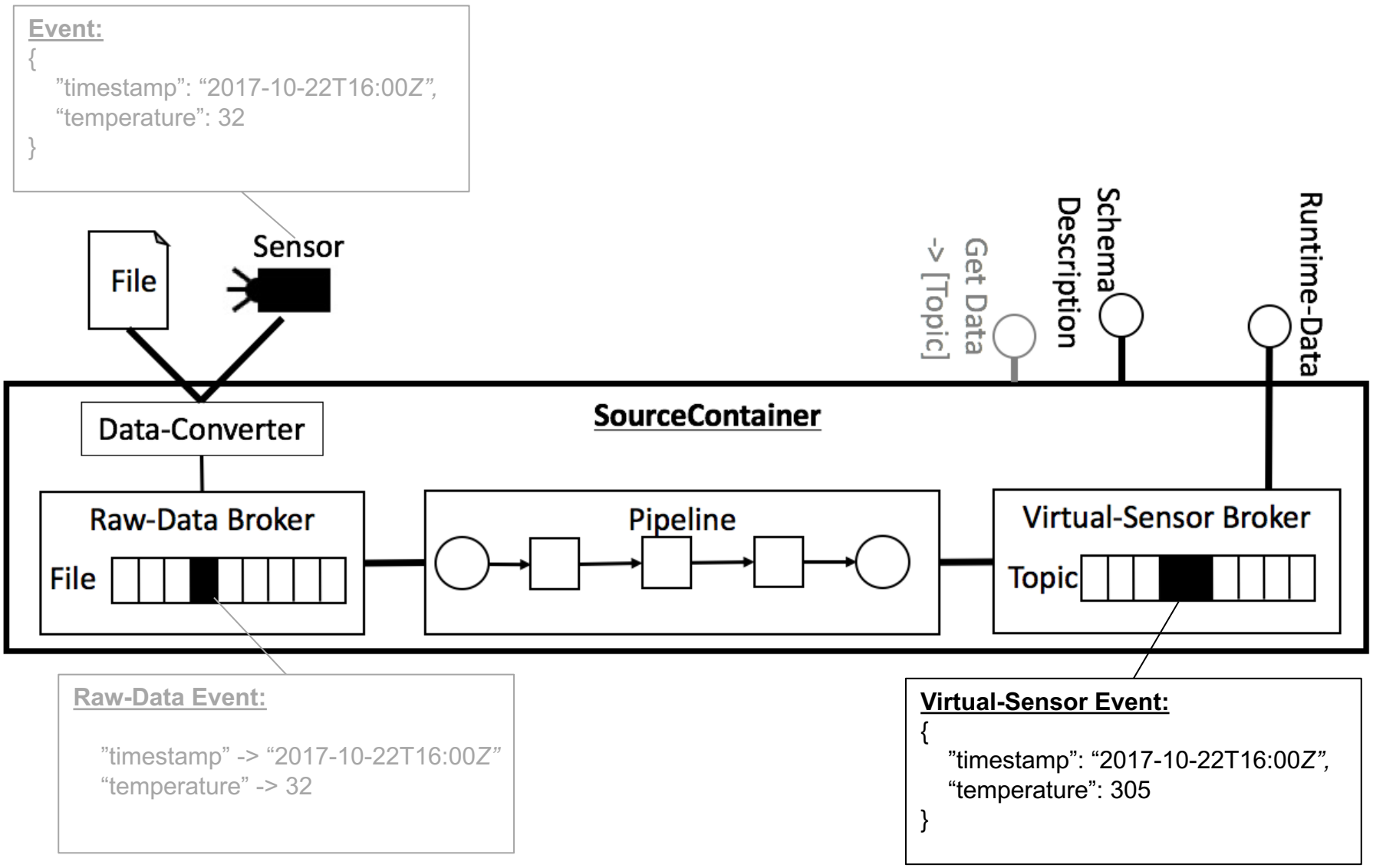
Adapter Architecture



Adapter Architecture



Adapter Architecture



Data Converter

Design Time

User defines the raw event model step by step. First the **protocol** must be configured, then the **format** defined.

Protocol Configuration

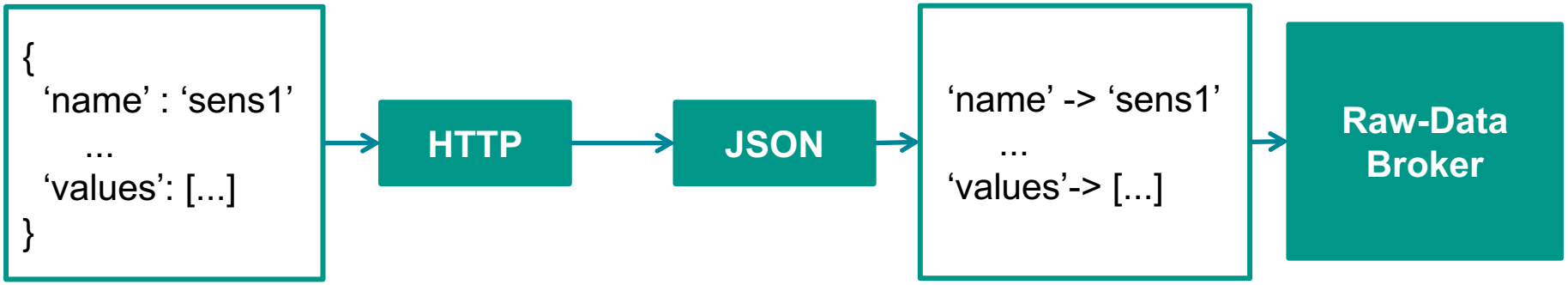
- Select one of the existing protocols (e.g. HTTP)
- Provide information like the URL of the endpoint

Format Configuration

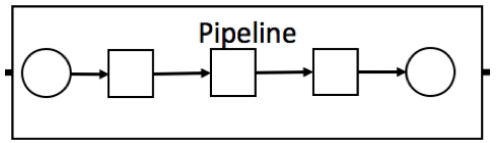
- Set some format-specific parameters (e.g. delimiter of CSV)
- System tries to guess those by extracting some sample data.

Run Time

- Protocol is used to create a connection to the data source
- Information of the format is needed to transform data into the internal representation



Raw Data Pipeline



Goal

Use the semantic models to semi-automatically generate a data pipeline for cleaning and harmonizing the data based on user preferences.

Structural Transform Agent

- Transforms the raw-data structure into the structure of the virtual sensor
- Mappings between raw-data model and virtual-sensor based on Event Properties
- Example Event:

Raw:

‘sensor’ ->
‘temp’ -> 32

Virtual-Sensor:

‘temperature’ -> 32

Unit Transform Agent

- Uses the information about the units between raw-data and virtual-Sensor and transforms values when necessary
- Example Event:

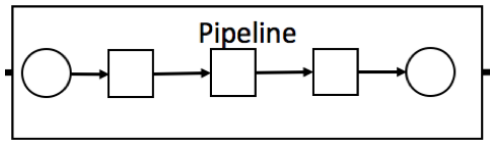
Raw: (Degree Celsius)

‘temperature’ -> 32

Virtual-Sensor: (Kelvin)

’temperature’ -> 305

Raw Data Pipeline



Filter Agent

- Ensures data is produced according to the semantic description
- Using the value specification or qualities to apply filter rules on data

- Example:

Value Specification:

- min: -10
- max: 20

Event:

```
'temperature': 100
```

Frequency Reducer Agent

- Agent changes frequency of data
- Frequency is reduced when the values rarely change and increased again when they change

- Example:
- Sensor measuring power consumption
  - Sampling: 100Hz
  - Reduced to 1/sec

  - When machine idle not as many events should be sent

Conclusion / Open Problems

Conclusion

Contribution

- Presented an adapter framework to ingest real-time and batch data into stream processing engines
- A lightweight model for raw-data and an extended version for virtual sensors

Approach

- Model instances are defined by domain experts
- Adapter Architecture which is configured with the model instances
  - First accessing the data
  - transform into internal format
  - pipeline transforms data into virtual sensor format
  - data published on a message broker
- Presented different agents of the adapter pipeline

Current State / Open Problems

Current State

- Framework for domain experts, but still focused on manual configuration
- Specified a vocabulary
- Prototypical implementation

Future Work / Open Problems

- Evaluation
  - Conduct a user study to ensure the framework can be used by domain experts and a sufficient number of formats and protocols are supported.
- Further automate the modelling process
  - Machine learning approach to find domainProperty based on runtime name (e.g. column name in table)
- Already using some rules based on semantics (e.g. unit transformation)
  - How much more can be inferred out of the semantics of the data?

References

- [1] S. Bischof, C. Martin, A. Polleres, and P. Schneider, Collecting, In- tegrating, Enriching and Republishing Open City Data as Linked Data. Cham: Springer International Publishing, 2015, pp. 57–75. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-25010-6_4
- [2] R. C. Fernandez, P. R. Pietzuch et al., “Liquid: Unifying nearline and offline big data integration,” in CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA,, 2015.
- [3] D. Riemer, F. Kaulfersch, R. Hutmacher, and L. Stojanovic, “Streampipes: Solving the challenge with semantic stream processing pipelines,” in Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems, ser. DEBS ’15. New York, NY, USA: ACM, 2015, pp. 330–331. [Online]. Available: http://doi.acm.org/10.1145/2675743.2776765
- [4] J.-P. Calbimonte, S. Sarni, J. Eberle, and K. Aberer, “Xgsn: An open-source semantic sensing middleware for the web of things.” in TC/SSN@ ISWC, 2014, pp. 51–66.
- [5] J. Soldatos, N. Kefalakis et al., “Openiot: Open source internet-of-things in the cloud,” in Interoperability and open-source solutions for the internet of things. Springer, 2015, pp. 13–25.
- [6] http://talend.com/