apache streampipes

Apache StreamPipes for Pythonistas

IIoT data handling made easy!

Tim Bossenmaier, Sven Oehler

About Us

StreamPipes Committers

Current committers, sorted by last name:

Data Engineer @inovex

PMC

@bossenti

Tim Bossenmaier

Student @Bytefabrik.AI

Committer

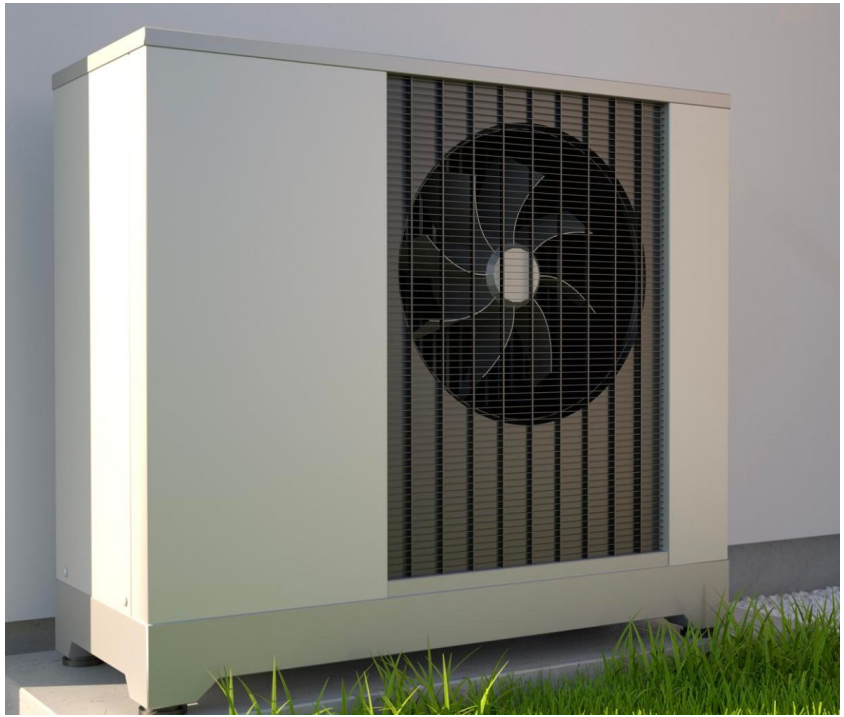
@SvenO3

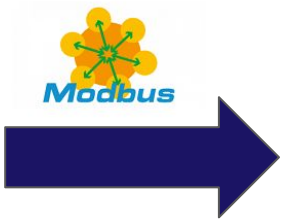
Sven Oehler

... and many more!

Remember last autumn & winter

There is a need to modernize





Let’s take a look at Modbus

- invented 1979
- communication protocol for programmable logic controllers (PLCs)
- supports only 4 types of objects:
  - coil & discrete input (1 bit)
  - input register & holding register (16 bit)
- registers have addresses

not really fun or convenient to work with

That’s exactly where StreamPipes comes in

apache streampipes

apache streampipes

version 0.91.0

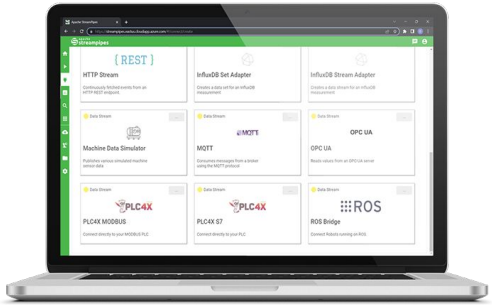
"An open source industrial IoT toolbox to enable non-technical users to connect, analyze and exploit (IoT) data streams."

- Connect
- Pipeline Editor
- Dashboard
- Data Explorer
- Notifications

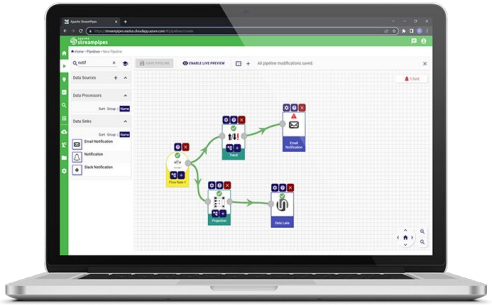
StreamPipes Core Features

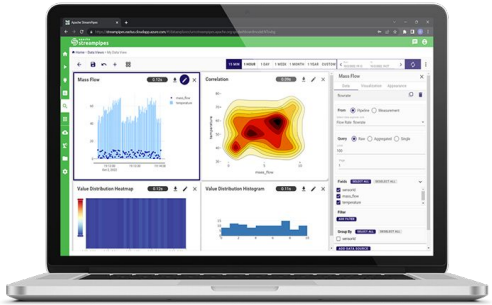
apache streampipes

1 Quickly connect IIoT data streams

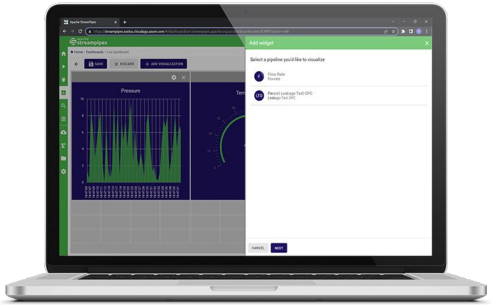


Pipelines to harmonize & analyse data 2





3 Visually explore IIoT data



Live dashboard 4

Feature Walkthrough

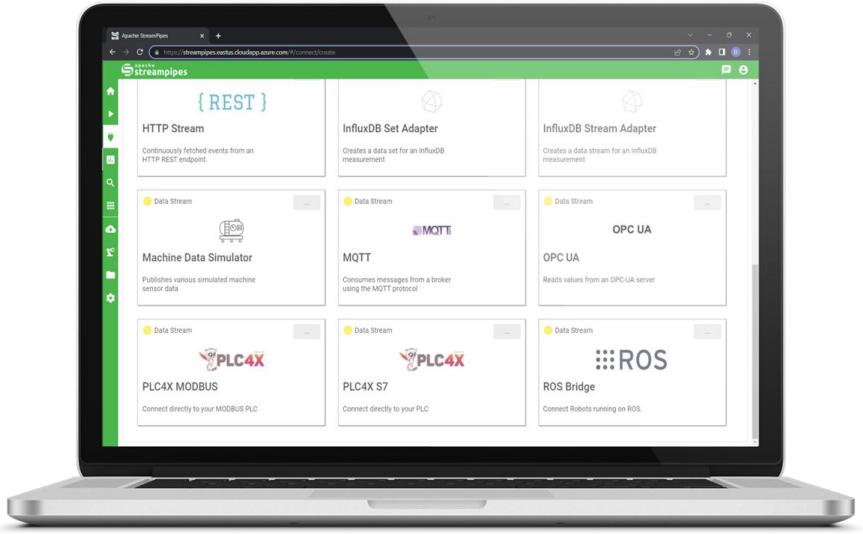
Connect

Quickly connect industrial data sources in real-time

> 20 adapters for industrial protocols

Pre-processing rules for data harmonization

Advanced schema configuration options



Feature Walkthrough

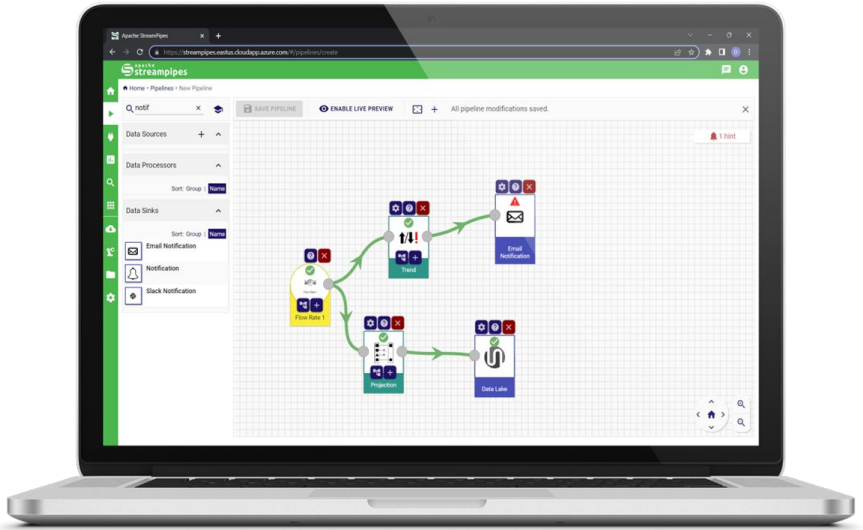
Pipeline Editor

Flexible toolbox for data harmonization & analytics

From thresholds to AI models

>100 algorithms + data sinks

SDK for extensions

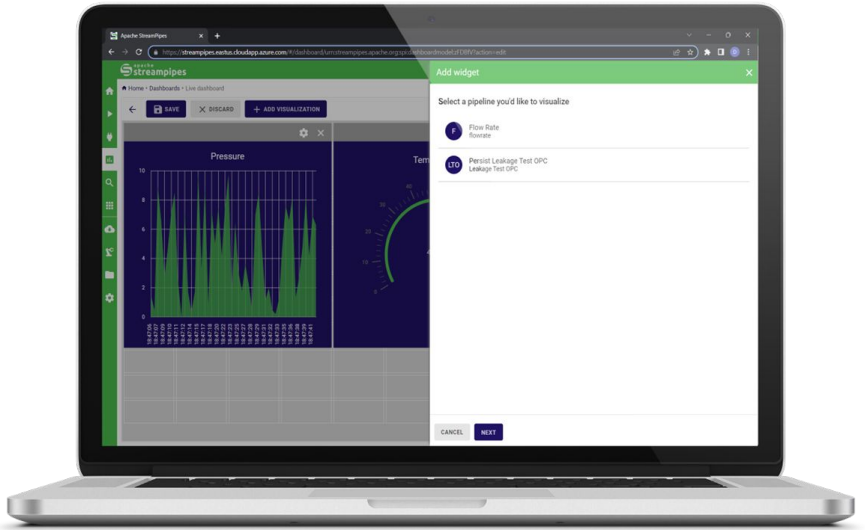


Feature Walkthrough

Live Dashboard

Visualization of live metrics & KPIs (e.g., at shopfloor level)

Widgets and easy configuration

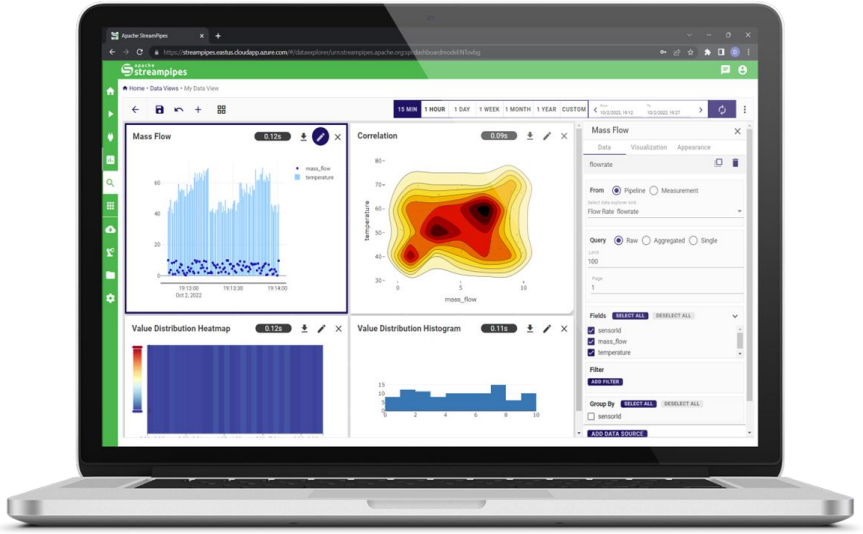


Feature Walkthrough

Data Explorer

Visual analytics of historical data

Quickly explore data and find correlations



Pipeline Editor: build data pipelines

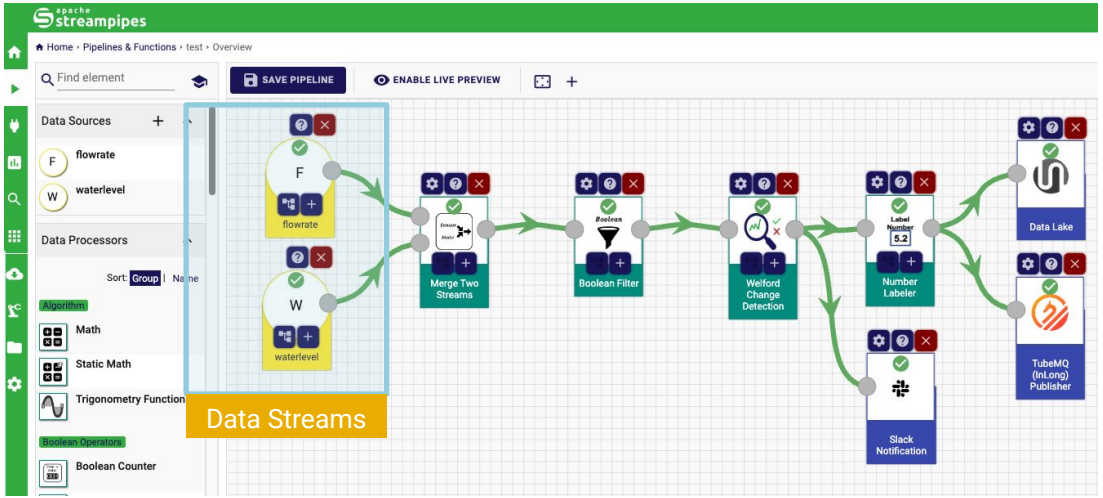
Data Streams

Pipeline = Data Streams

DAG + Data Processors

+ Data Sinks

Data stream. Ordered sequence of **events** generated by an instance of a **StreamPipes Connect** Adapter while connected to an event source (e.g. a robot arm via ROS).



Pipeline Editor: build data pipelines

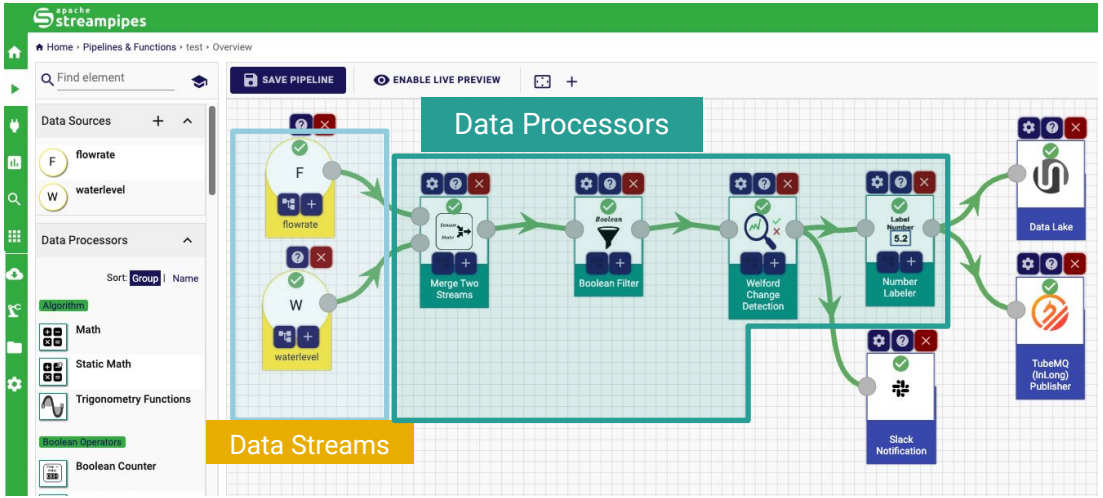
Data Processors

Pipeline = Data Streams

DAG + Data Processors

+ Data Sinks

Data processors. Apply a **function** to **one or more** incoming event streams (transform, filter, enrich, etc.) and produce an outgoing event stream.



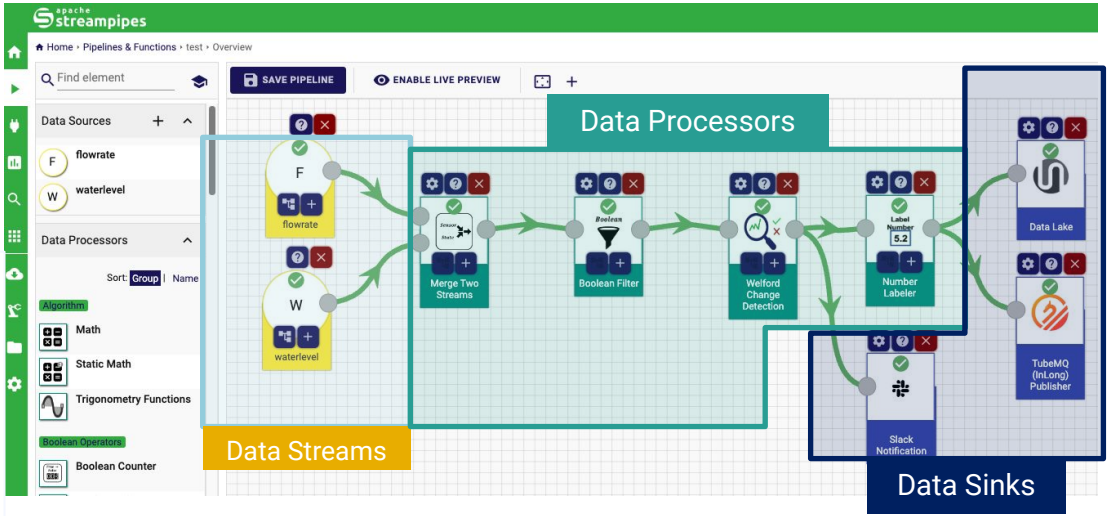
Pipeline Editor: build data pipelines

Data Sinks

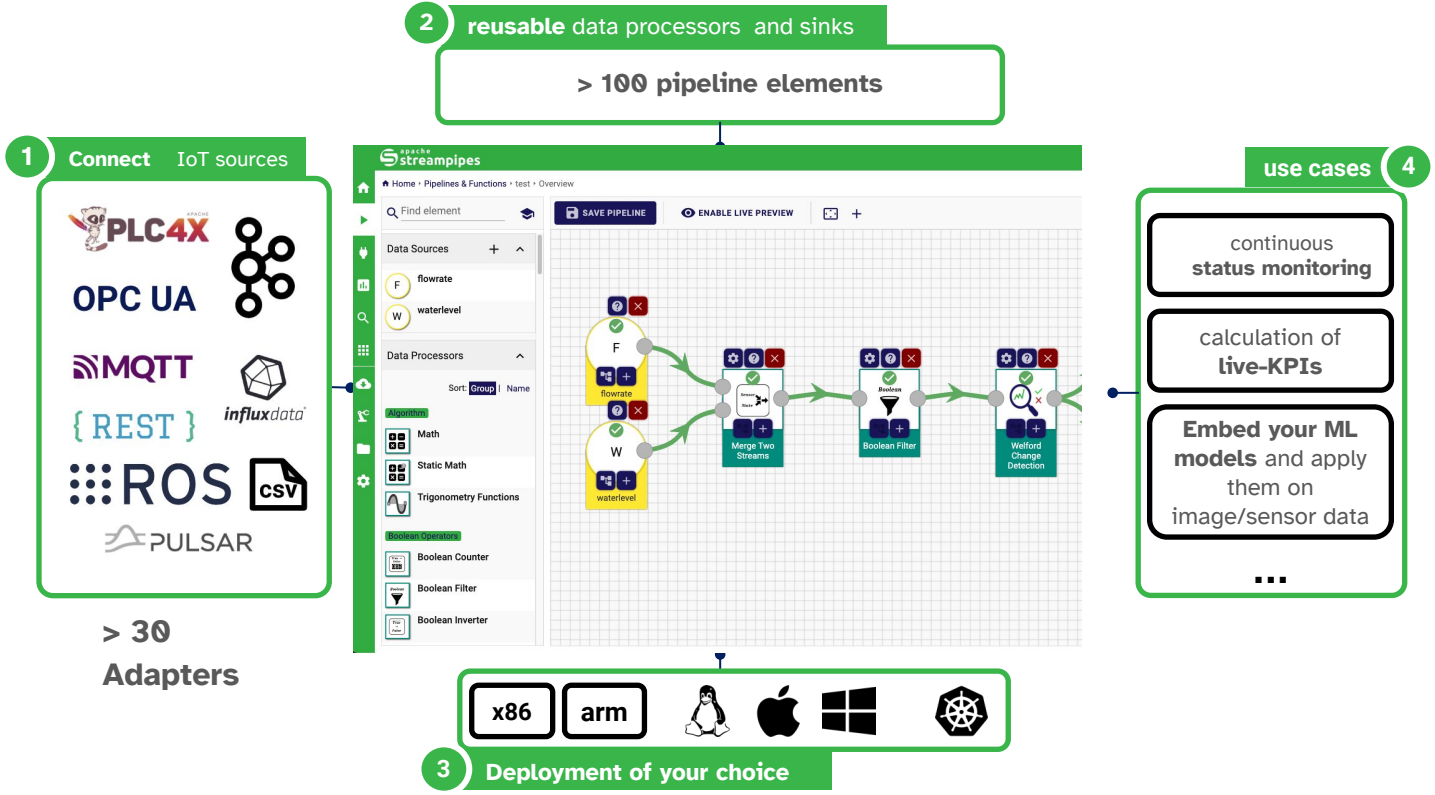
Pipeline = Data Streams
DAG + Data Processors
+ Data Sinks

Data sinks. Mark the end of a pipeline and send the incoming event stream to:

- **internal** sinks (Dashboard, Datalake)
- **external** sinks (Kafka, IoTDB, ROS, etc.)



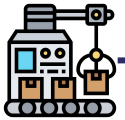
How does StreamPipes work?



End-2-End IIoT Data Analytics

apache streampipes

OPC-UA



MQTT

Modbus

Connecting
Adapter

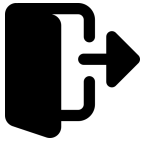
Processing
Pipelines /Functions

Data Handling
Time-Series DB
Message Broker

GUI
Pipeline Editor
Data Explorer

Programmatically
Java SDK
Python SDK
Online ML

Third Party Systems
Kafka, Rocket MQ, IotDB, ...



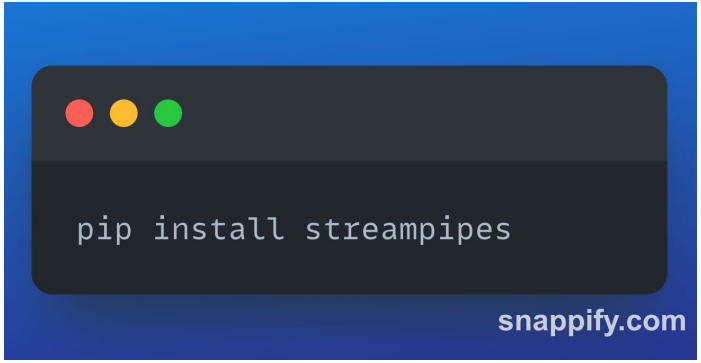

Hands-On

StreamPipes Python - Motivation

"Specifically with StreamPipes Python, we want to address the amazing data analytics and data science community in Python and benefit from the great universe of Python libraries out there."

StreamPipes Python

- freshly released with StreamPipes 0.91.0
- available on PyPI (conda-forge will follow soon)



- documentation can be found at

  https://streampipes.apache.org/docs/docs/python/latest/

StreamPipes Python Features

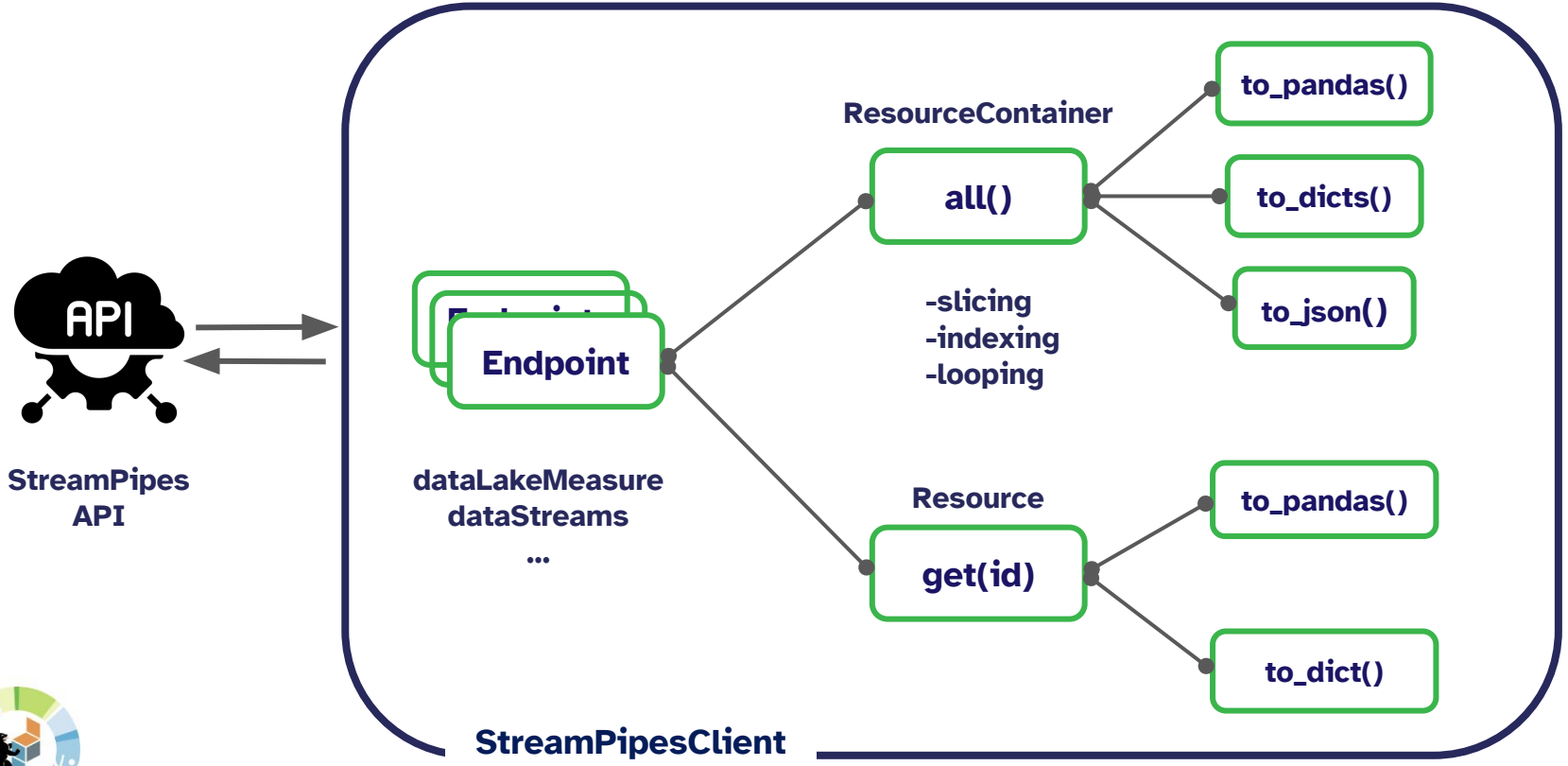
🔐 StreamPipes Client

- connects to a StreamPipes instance
- allows to interact with the StreamPipes API

🛠️ StreamPipes Functions

- lightweight processing elements
- only exist at runtime
- allow to define custom functionalities based on live data, e.g. Online ML

StreamPipes Python Architecture



StreamPipes Python Client - Example

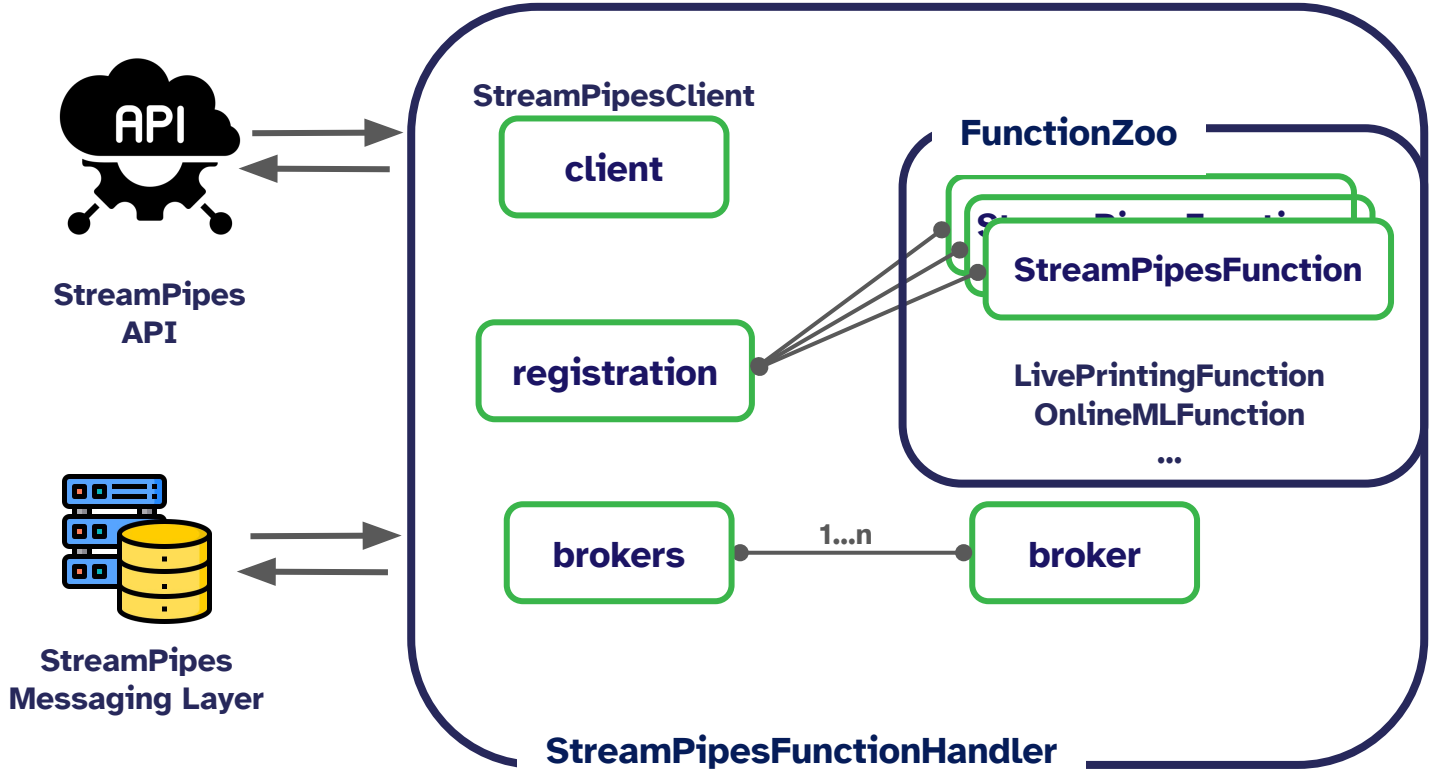
demo.py

```python
from streampipes.client import StreamPipesClient
from streampipes.client.config import StreamPipesClientConfig
from streampipes.client.credential_provider import StreamPipesApiKeyCredentials

config = StreamPipesClientConfig(
    credential_provider = StreamPipesApiKeyCredentials.from_env(
        username_env="USER",
        api_key_env="API-KEY",
    ),
    host_address = "localhost",
    https_disabled = True,
    port = 80
)

client = StreamPipesClient(client_config=config)

# get all available datat lake measures
measures = client.dataLakeMeasureApi.all()

# get amount of retrieved measures
len(measures)
```

snappify.com

To see more visit the tutorials section on our documentation

StreamPipes Python Architecture

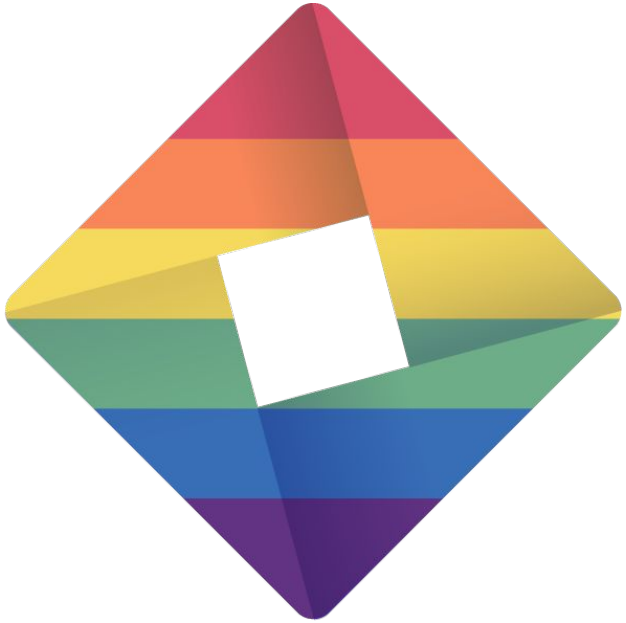



Hands-On

Visit us @inovex

apache streampipes

inovex is diversity sponsor of this year's PyConDE



come and visit us at our booth

apache streampipes

Tim Bossenmaier

inovex GmbH

tim.bossenmaier@inovex.de

Sven Oehler

Bytefabrik.AI GmbH

sven.oehler@bytefabrik.ai

Thank you for your attention and interest ❤️

Find the talk's materials on GitHub:
https://github.com/bossenti/pycon-23-streampipes-pythonistas

Enjoy the conference!