apache streampipes

IIoT analytics made easy with Apache StreamPipes

Dominik Riemer | PPMC Apache StreamPipes

IoT Track | Oct 03, 2022

This talk

- Project Overview
- Walkthrough
- Feature Tour
  - Connect industrial data
  - Visually explore data
  - Create pipelines
  - Use data in external systems
  - Extend StreamPipes
- Future Work / How to contribute

About me

- from Karlsruhe, Germany
- Open Source Enthusiast
  - Co-initiator & PPMC member @StreamPipes
  - Committer @PLC4X
- currently co-founder & CEO of Bytefabrik.AI
- previously research division manager @FZI

Bytefabrik.AI

Project Overview

Apache StreamPipes (incubating)

Mission Statement

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

Apache StreamPipes

Project facts

originally created by FZI Research Center for Information Technology, first presented at ACNA19

Incubating project since December 2019

4.5 Apache releases

(release 0.70. under vote)

13 PPMC members

7 committers

Apache StreamPipes

Integration with other ASF projects

Connectivity

Connectors

Visualizations

Apache StreamPipes

What's new in release 0.70.0?

- Improved Data Explorer
- Management of assets and resources
- Export/Import of StreamPipes resources
- Event preview in StreamPipes Connect
- > 50 bug fixes and improvements

Walkthrough

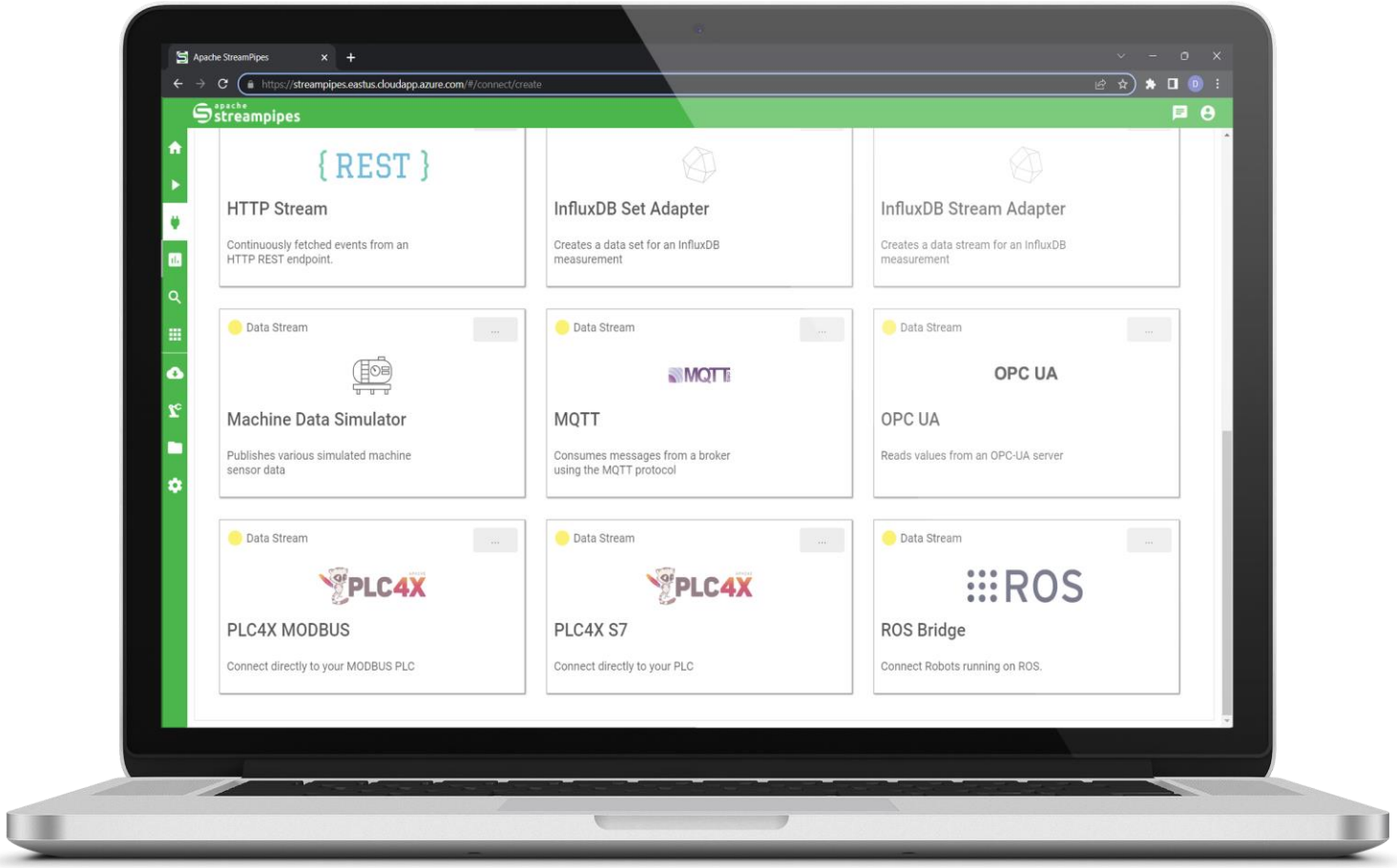
Apache StreamPipes

Connect

Stream/Set adapter library

Pre-processing rules

Schema configuration



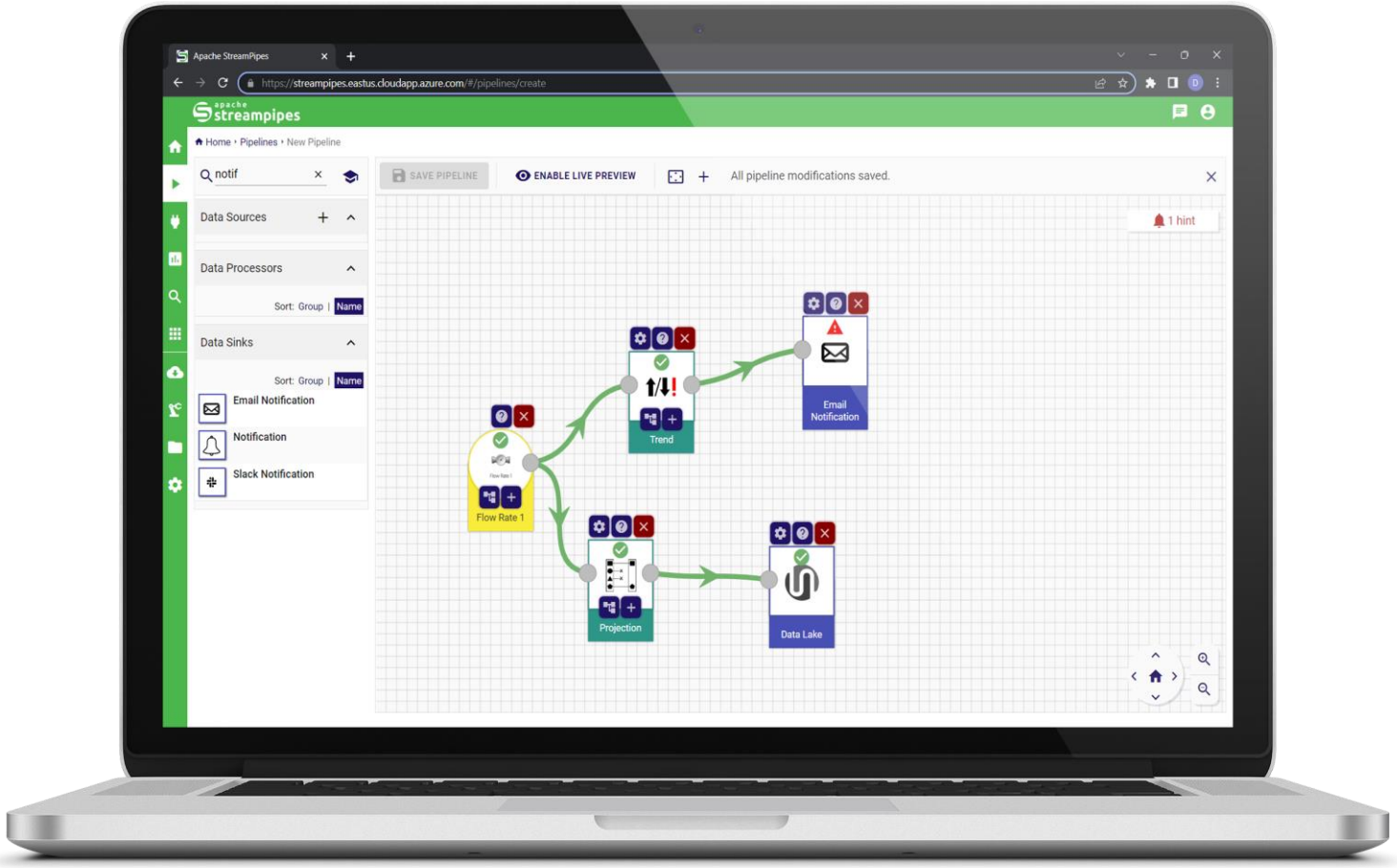
Apache StreamPipes

Pipeline Editor

Flexible toolbox for data harmonization & analytics

From thresholds to AI models

> 100 algorithms and SDK for extensions

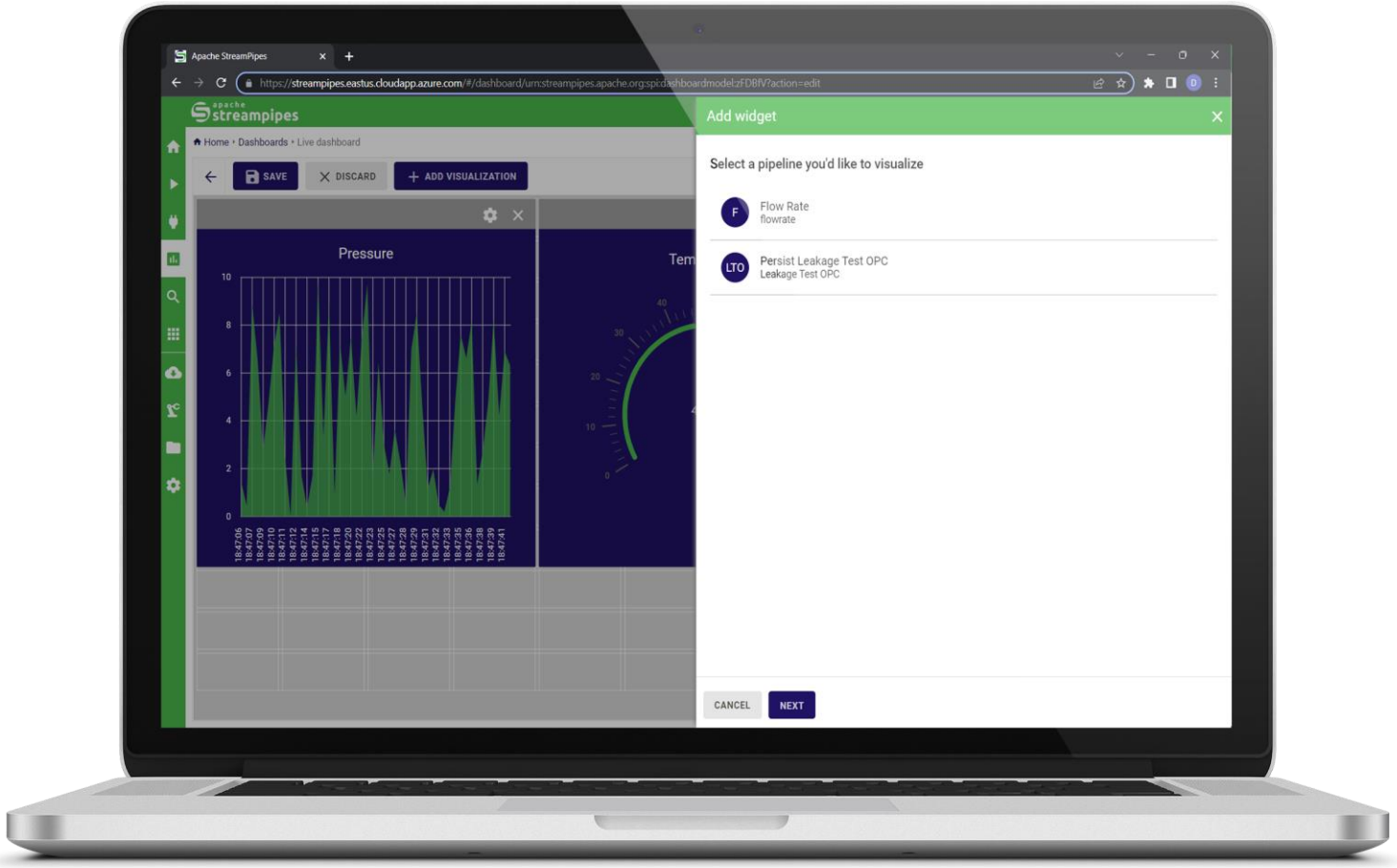


Apache StreamPipes

Live Dashboard

Visualization of live metrics & KPIs

Widgets and easy configuration

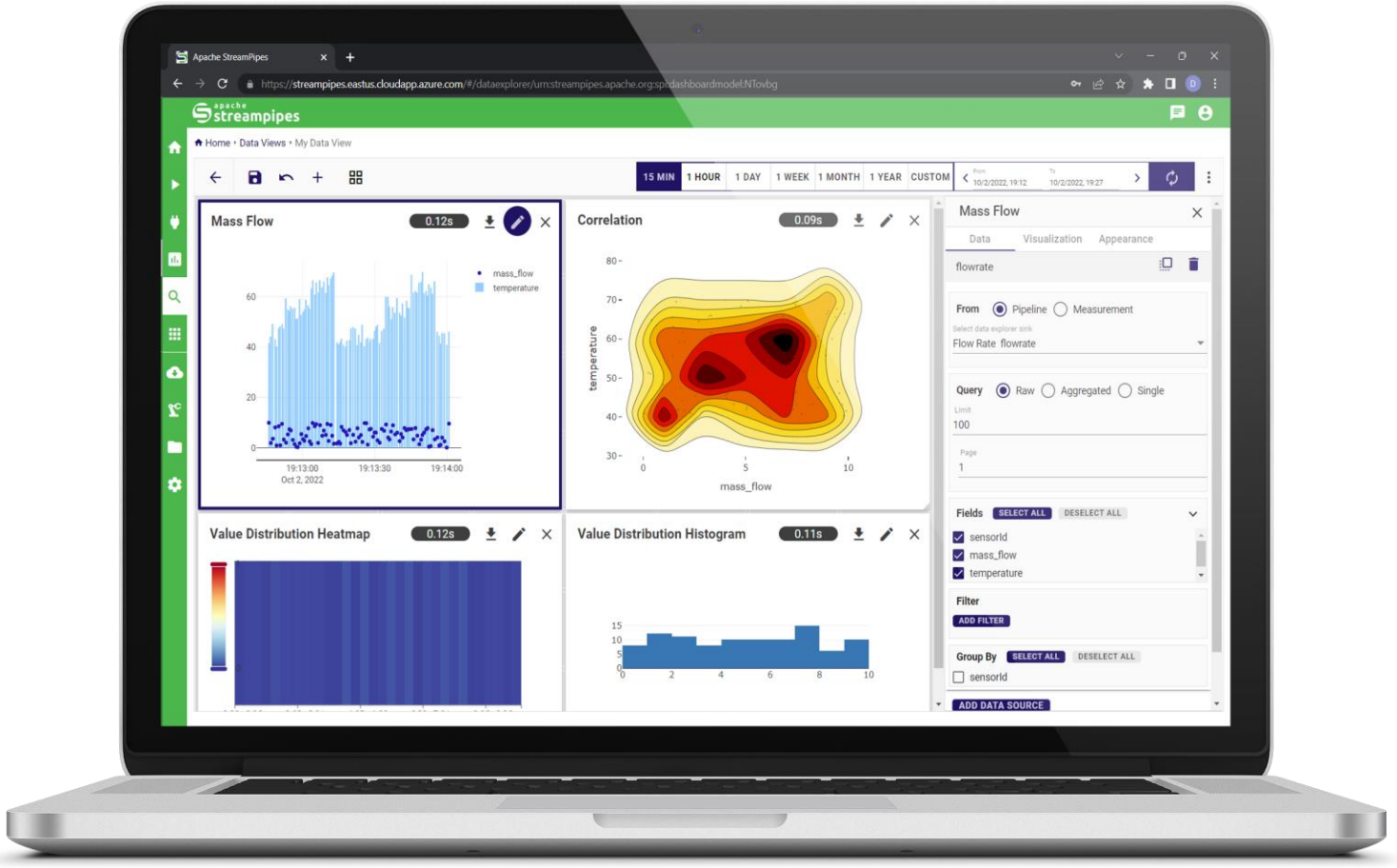


Apache StreamPipes

Data Explorer

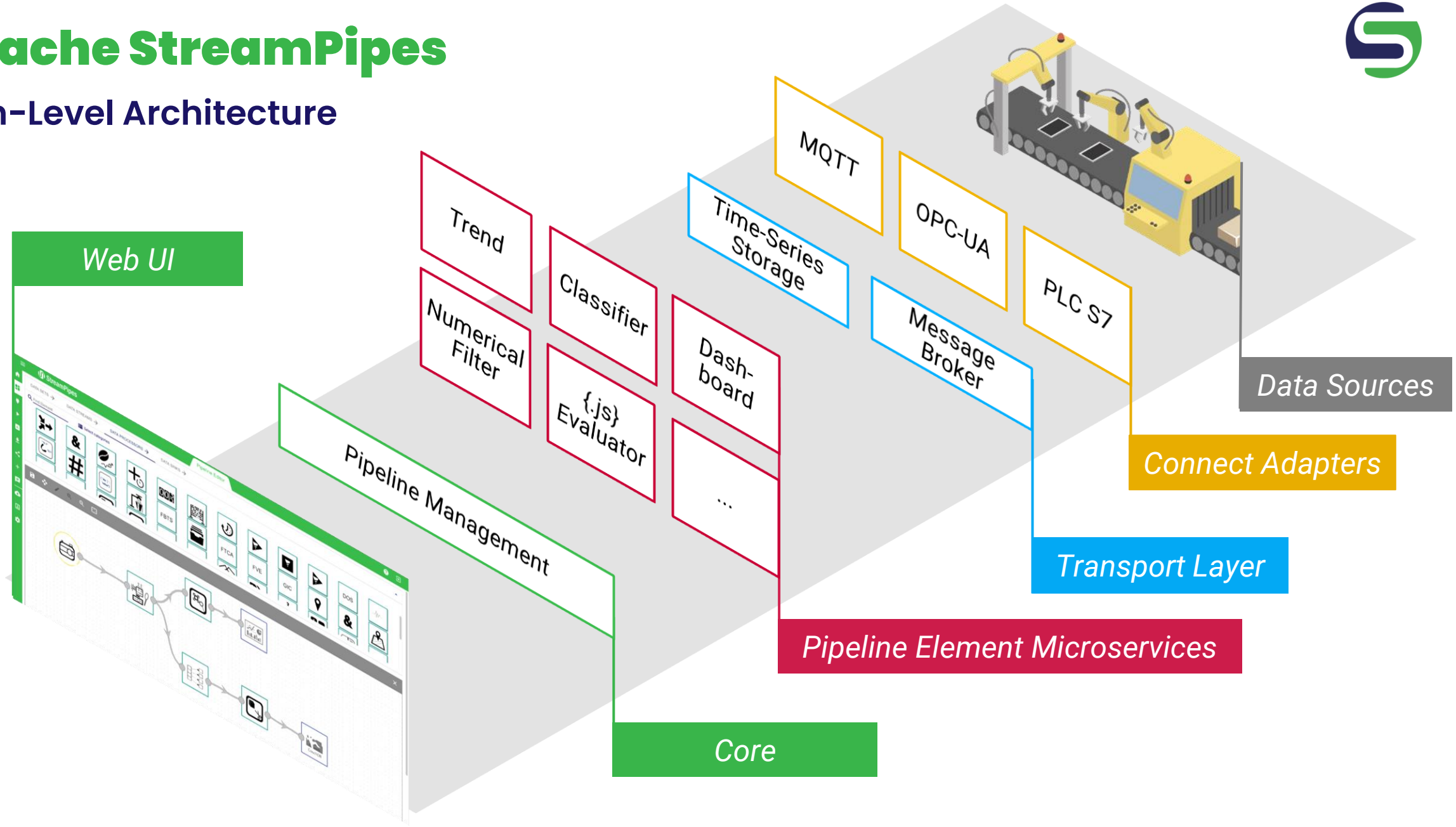
„Visual Analytics“

Quickly explore data and find correlations



Apache StreamPipes

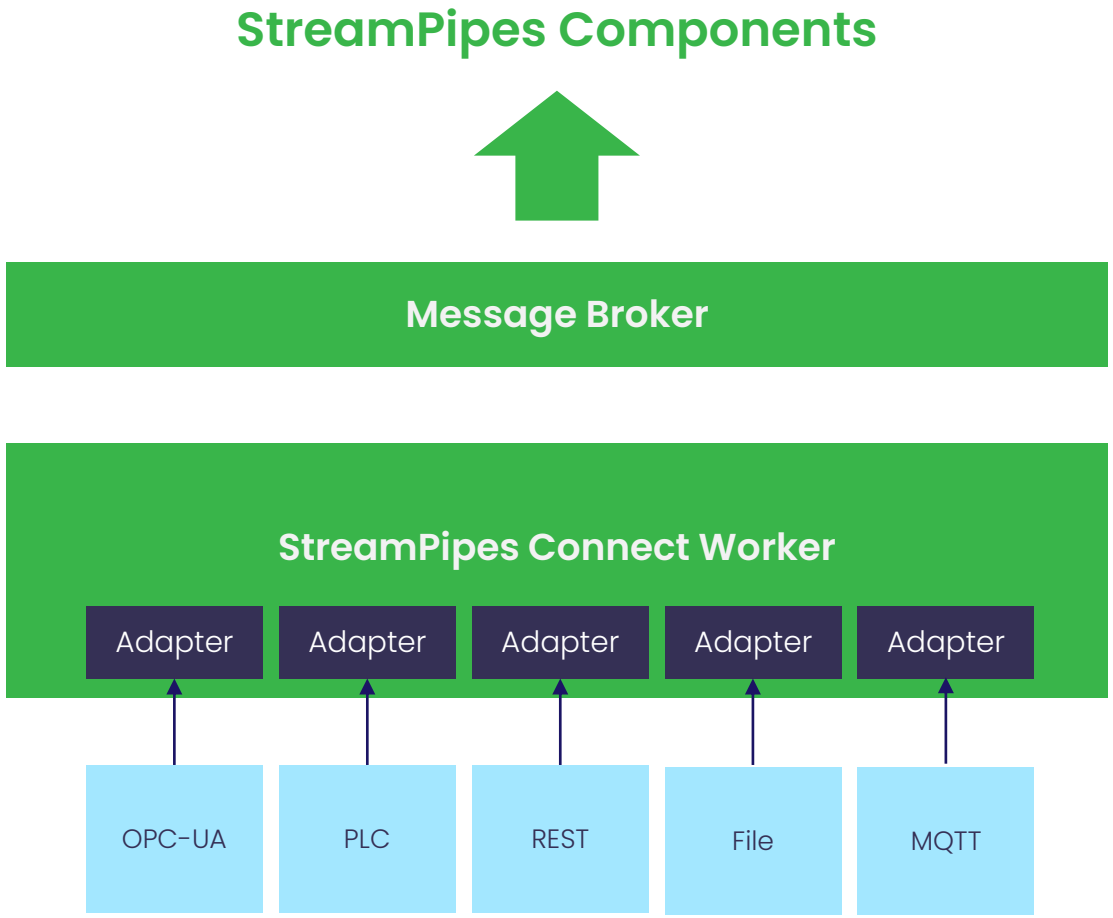
High-Level Architecture



Connect industrial data sources

Connect data

Architecture



Connect data

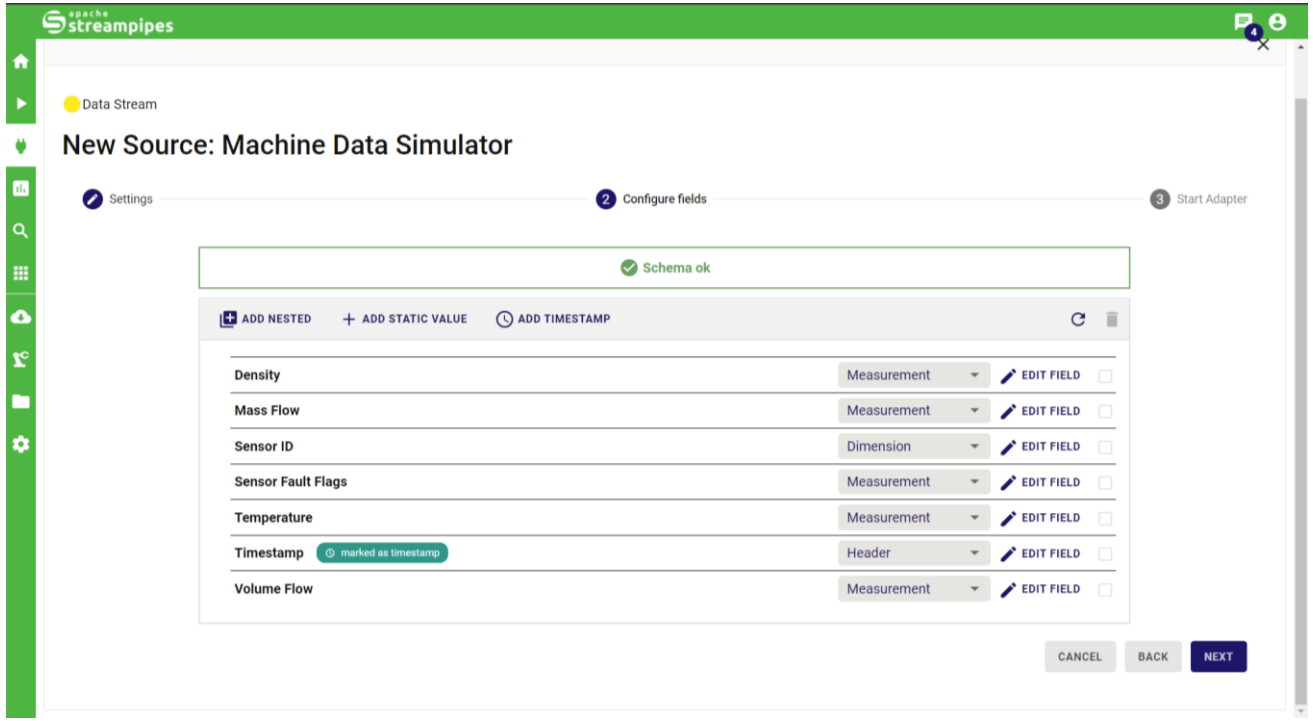
Process

1. Basic configuration
   - Protocol-specific settings
   - Access rights
   - Frequency
2. Format Definition
   - e.g., JSON or CSV
   - Format-specific settings
3. Schema transformation
   - Datatypes
   - Lightweight semantics
   - Pre-processing rules
4. Start Adapter
   - Direct start
   - Convenience methods

Demo!

Connect data

Schema editor: Features



Events always need a timestamp!

ggf. Transformation anwenden

Category

Measurement
Dimension
Header

Schema

Runtime Name (key at runtime)
Semantic Type (e.g., oil temperature)
Field Data Type (e.g., integer)

Value transformation

Correction Value (Factor)
Math Operator (Multiply, Add, Substract)

Unit transformation

Input unit
Output unit

Visualize & explore industrial data

Visualize & explore data

Tools

Live Dashboard

Simply consume live data, e.g., shop floor monitoring

Data Explorer

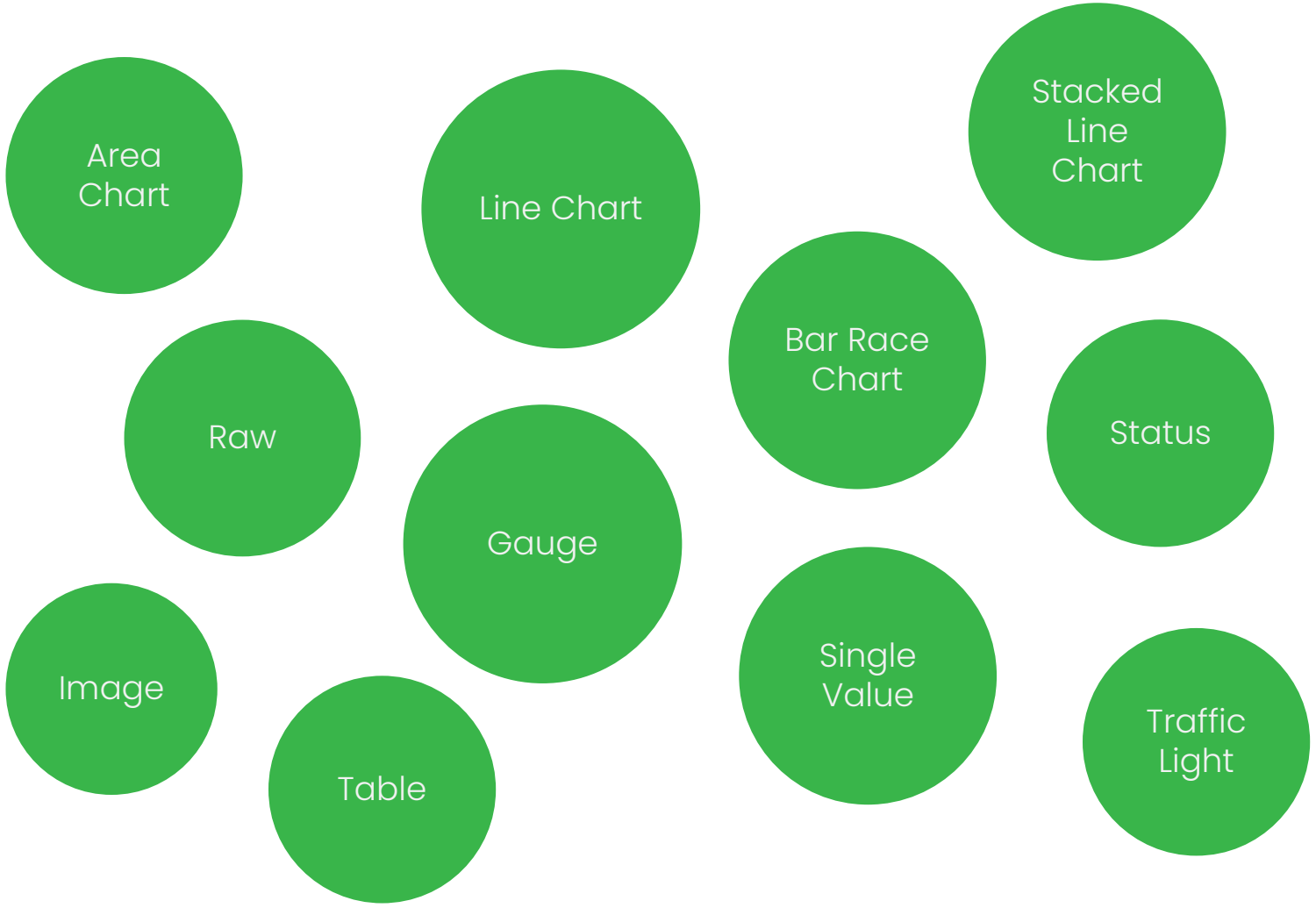
Explore data interactively and facilitate data understanding

Live Dashboard

Available visualizations & process

Requirement:
Persisted data stream

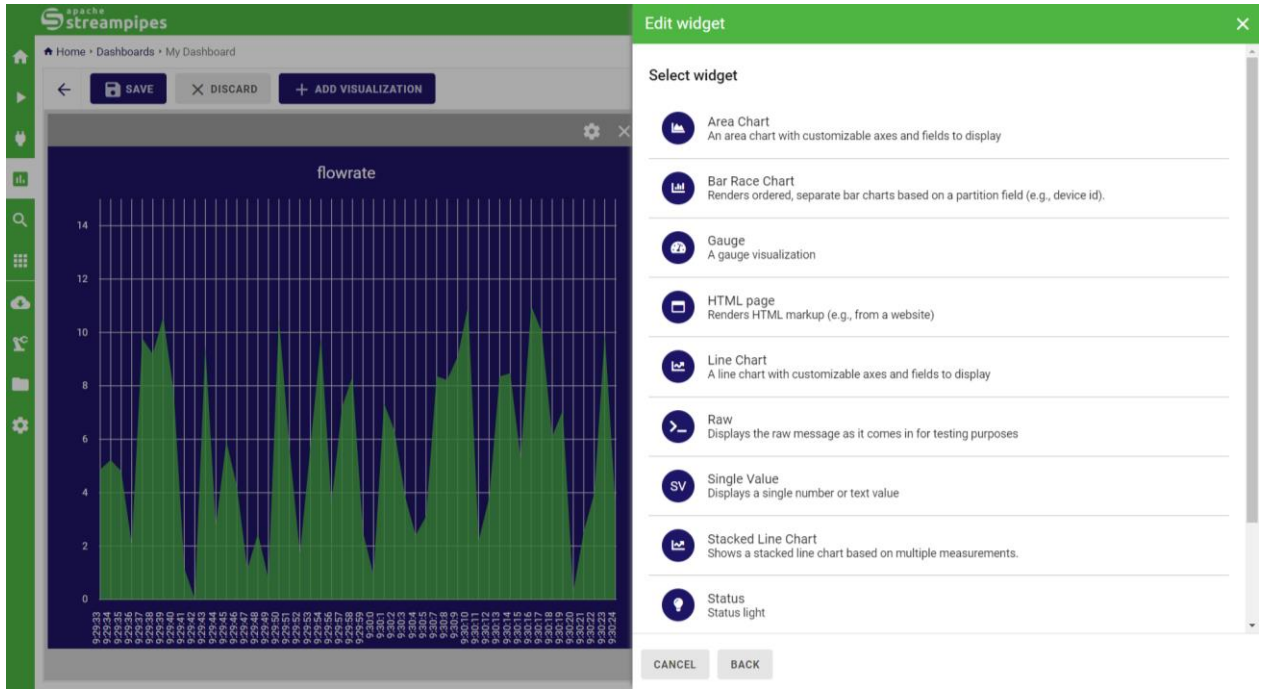
1. Create dashboard
2. Choose pipeline
3. Choose widget
4. Configure widget



Demo!

Live Dashboard

Summary



Available Visualizations

Depend on the underlying event schema

Attention: Changes

Widgets might need to be reconfigured in case the underlying schema changes

Widgets without pipeline are shown as incomplete

Data Explorer

Visualizations & process

Requirement:
Persisted data stream

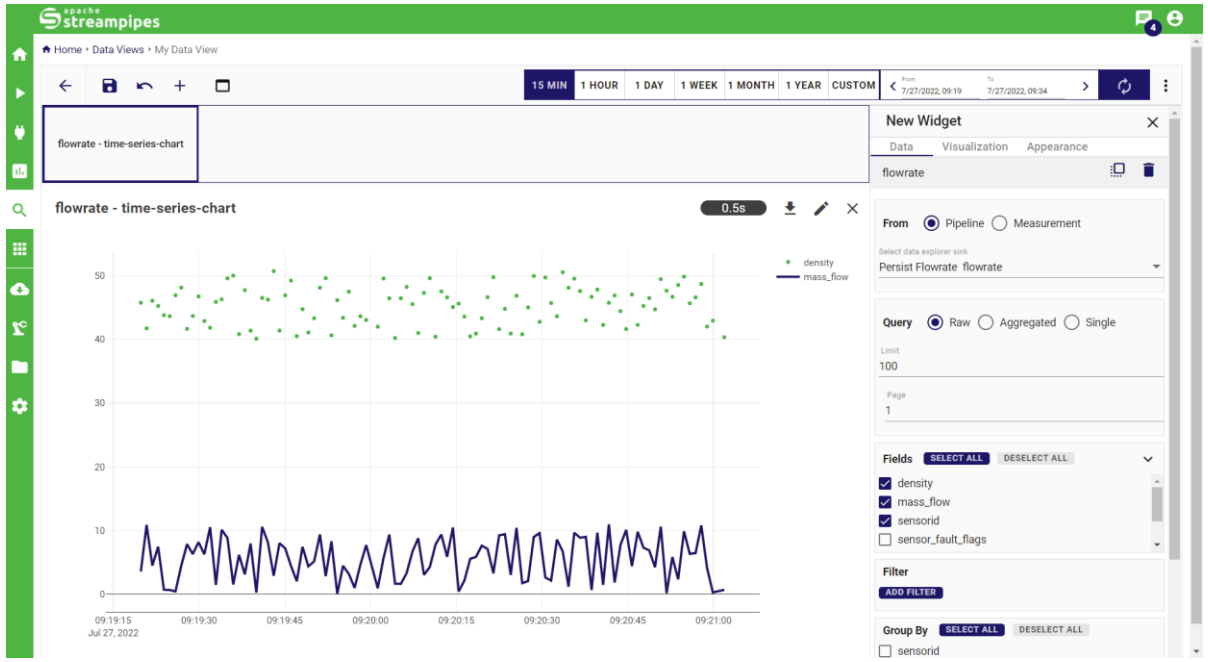
1. Create data view
2. Configure data and visualization
3. Choose time range

Table	Heatmap Values over time	Time Series Line, Scatter, Scattered Line, Bar, Symbol
Map	Image	
Distribution Histogram, Heatmap, Pie	Indicator Single values, delta	2D Correlation Scatter, Density Chart

Demo!

Data Explorer

Features



Configuration

Data configuration
Visualization configuration
Appearance configuration

Time Selection

Always applies for all widgets

Views

Grid View („dashboard style“)
Slide View („powerpoint style“)

Data Volume

Warning when queriy returns too many data items which would overload browser, can be manually overriden

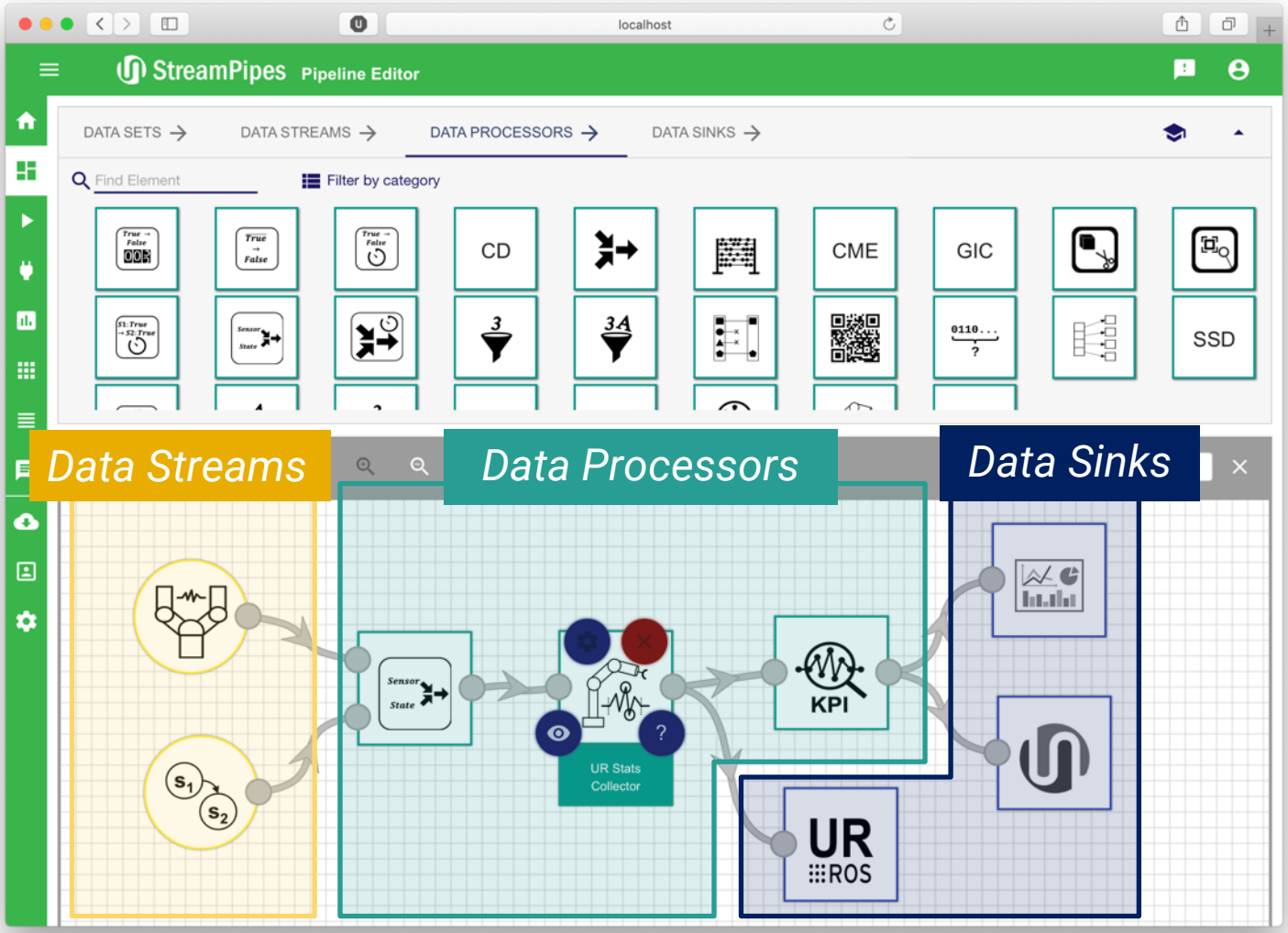
Create data analytics pipelines

Pipelines and pipeline element types

Pipeline = Data Streams

\+ Data Processors

\+ Data Sinks



Pipelines

Create pipeline

1. Choose data stream
2. Choose processor or sink
3. Configure element
4. Start pipeline

Goal

Flexibly harmonize and analyze data streams

Approach

Drag and drop interface to connect data with algorithms and sinks

Verification

The system prevents the creation of pipelines with incompatible data structures

Demo!

Pipelines

Features

Pipeline Editor

Element selection

Sort by name/group
(average) documentation included
Recommender
Compatible elements

Debugging

Live Preview

Convenience

Pipeline Element Templates
Auto Layout

Pipeline Details

Overview

Overview of pipelines, pipeline actions (start, modify, delete)
Only non-running pipelines can be edited

Monitoring

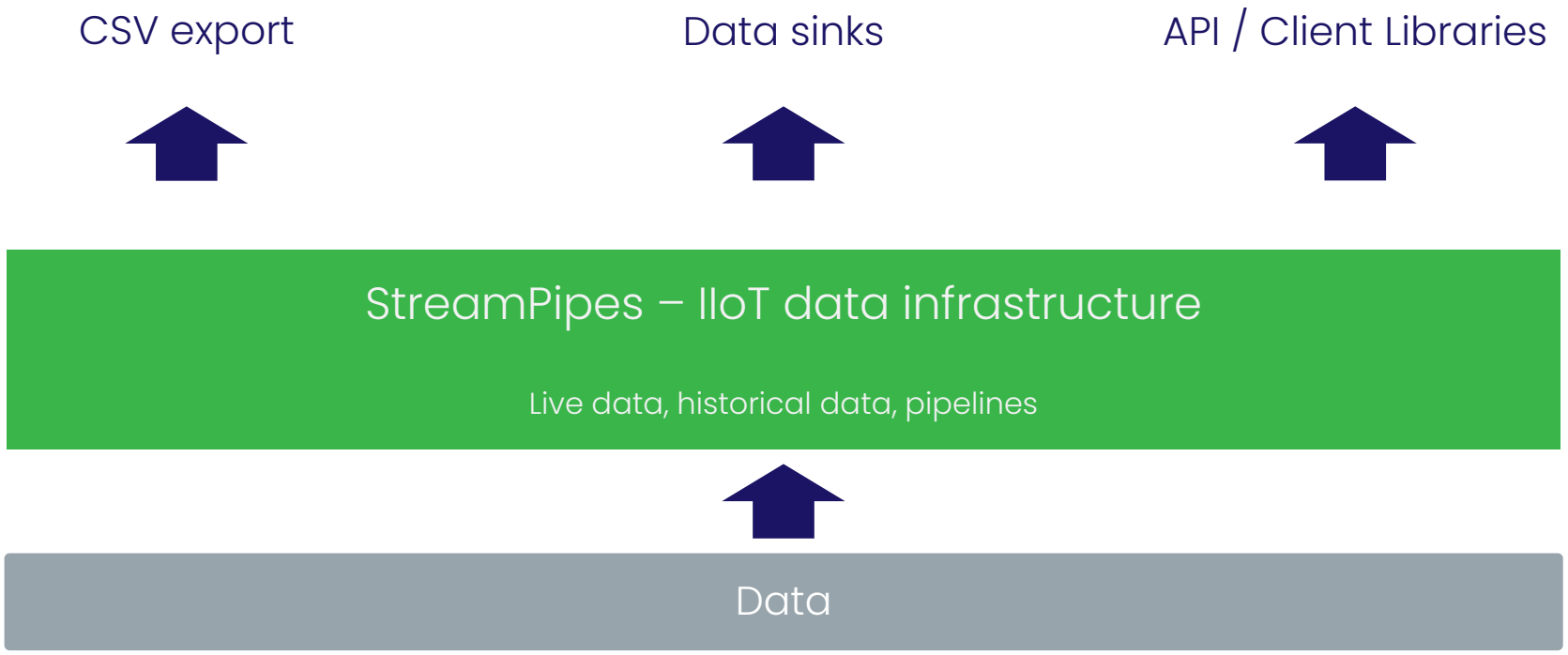
Overview of running pipelines, exceptions and metrics (in progress)

Quick Edit

Edit pipeline element configurations

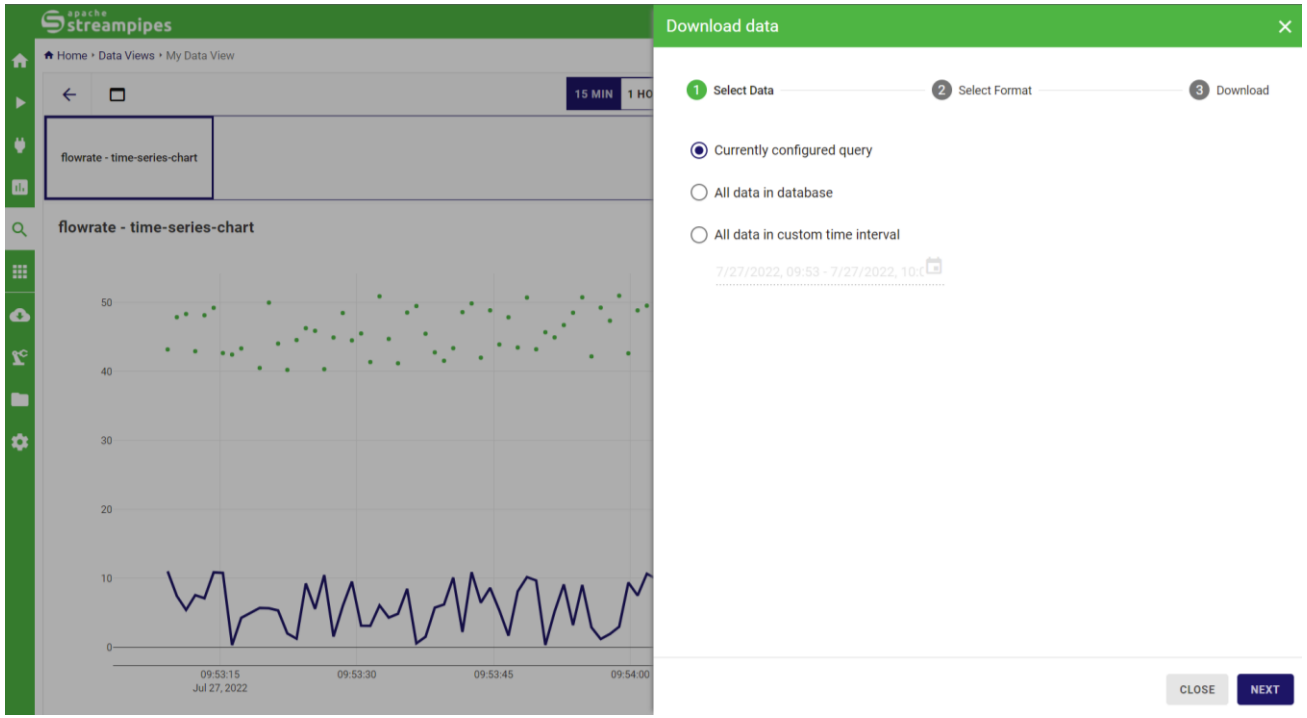
Use data in external systems

Interaction with external systems



Data Export

Data Explorer



CSV-Export

From query
Complete data
Filter by time

Exported data

Matches query, not necessarily the visualization

Additional opportunities

Data forwarding using pipelines

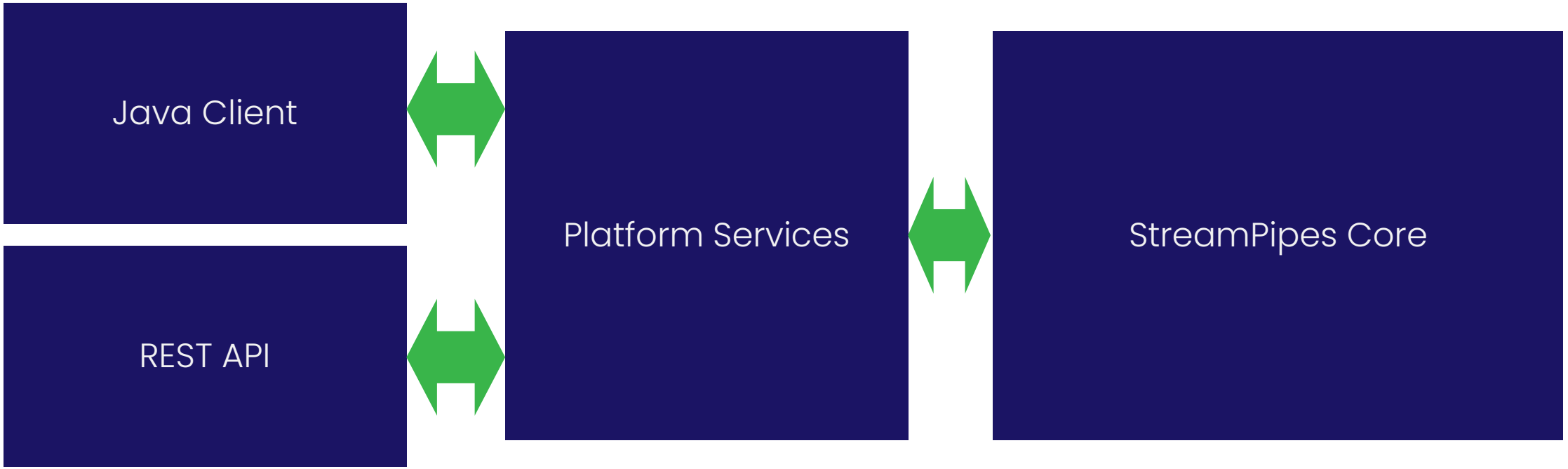
Platform API

Accessing StreamPipes features from external applications

Applications

- Change pipeline lifecycle from external apps
- Modify pipelines from external apps
- Provide live data to external apps
- Load historical data into external apps

API Communication



StreamPipes Client

Java library

```java
public static void main(String[] args) {
  StreamPipesCredentials credentials = StreamPipesCredentials
          .from(System.getenv("user"), System.getenv("apiKey"));

  // Create an instance of the StreamPipes client
  StreamPipesClient client = StreamPipesClient
          .create("localhost", 80, credentials, true);
}
```

```java
// Get all pipelines
    List<Pipeline> pipelines = client.pipelines().all();

    // Start a pipeline
    PipelineOperationStatus message = client.pipelines().start(pipelines.get(0));

    // Get all data streams
    List<SpDataStream> dataStreams = client.streams().all();

    // Subscribe to a data stream
    client.streams().subscribe(dataStreams.get(0), event -> MapUtils.debugPrint(System.out, "event",
  event.getRaw()));
```

How to extend StreamPipes

Ways to extend StreamPipes

Adapters and pipeline elements using SDK

UI extensions using microfrontends

Software Development Kit

Ziel

Extend StreamPipes with additional pipeline elements, e.g., adapters for specific protocols, or sinks

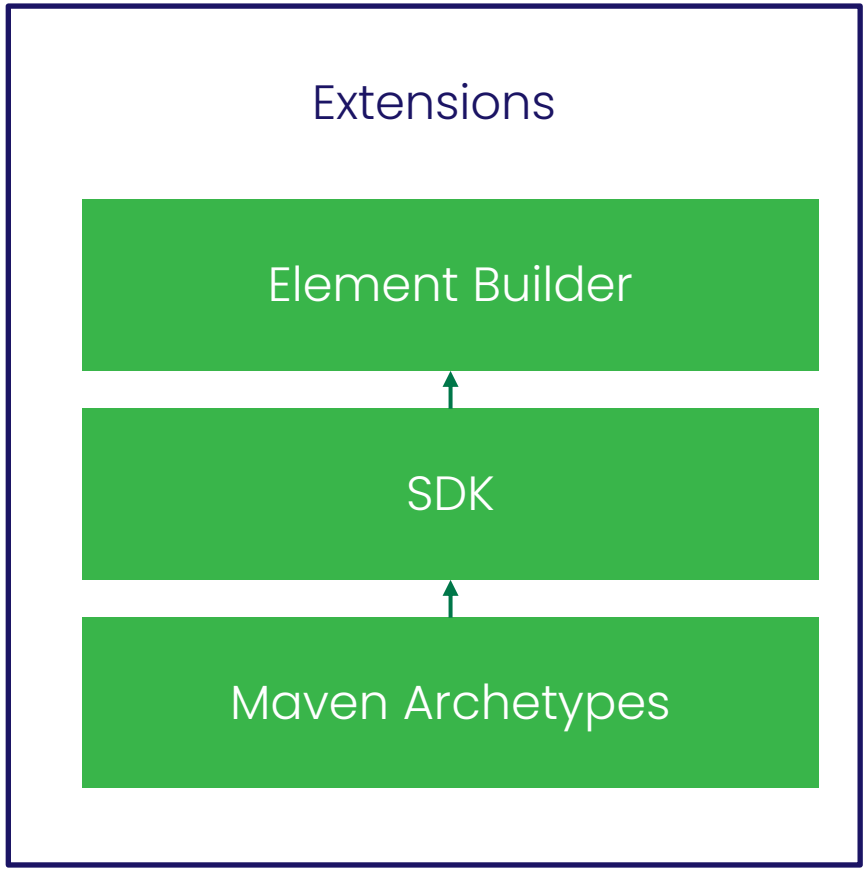
Features

Java-based SDK

Fully configurable UI elements

Maven archetypes

Installation at runtime



UI Extensions

Microfrontends

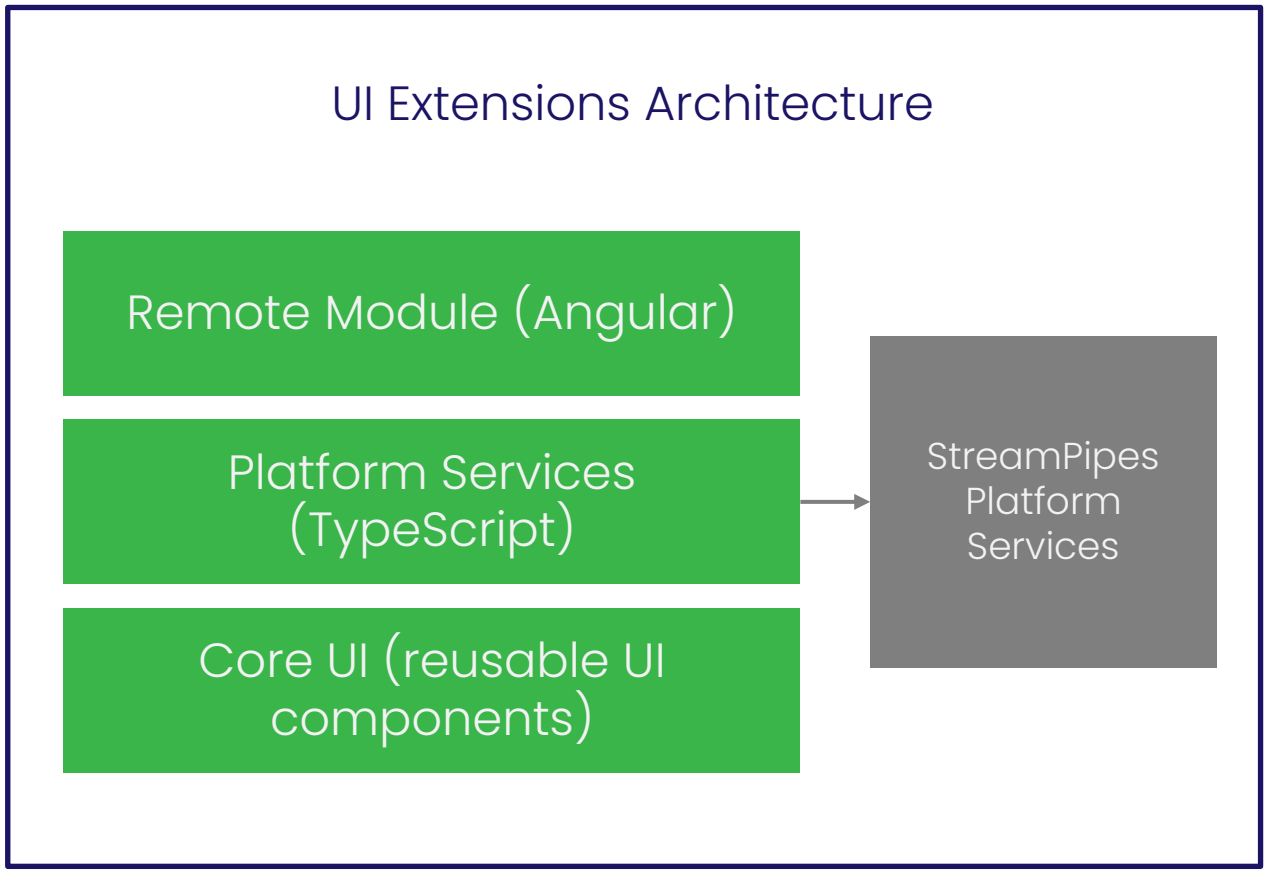
Goal

Create UI addons for specific use cases which build upon StreamPipes services

Features

Typescript SDK for all platform features (e.g. pipelines, data, data explorer views, ...)

Dedicated storage API for data bound to a specific extension



Roadmap / How to contribute

Roadmap

Current community work

Revival of python wrapper

Improved pipeline & adapter monitoring

Improved documentation

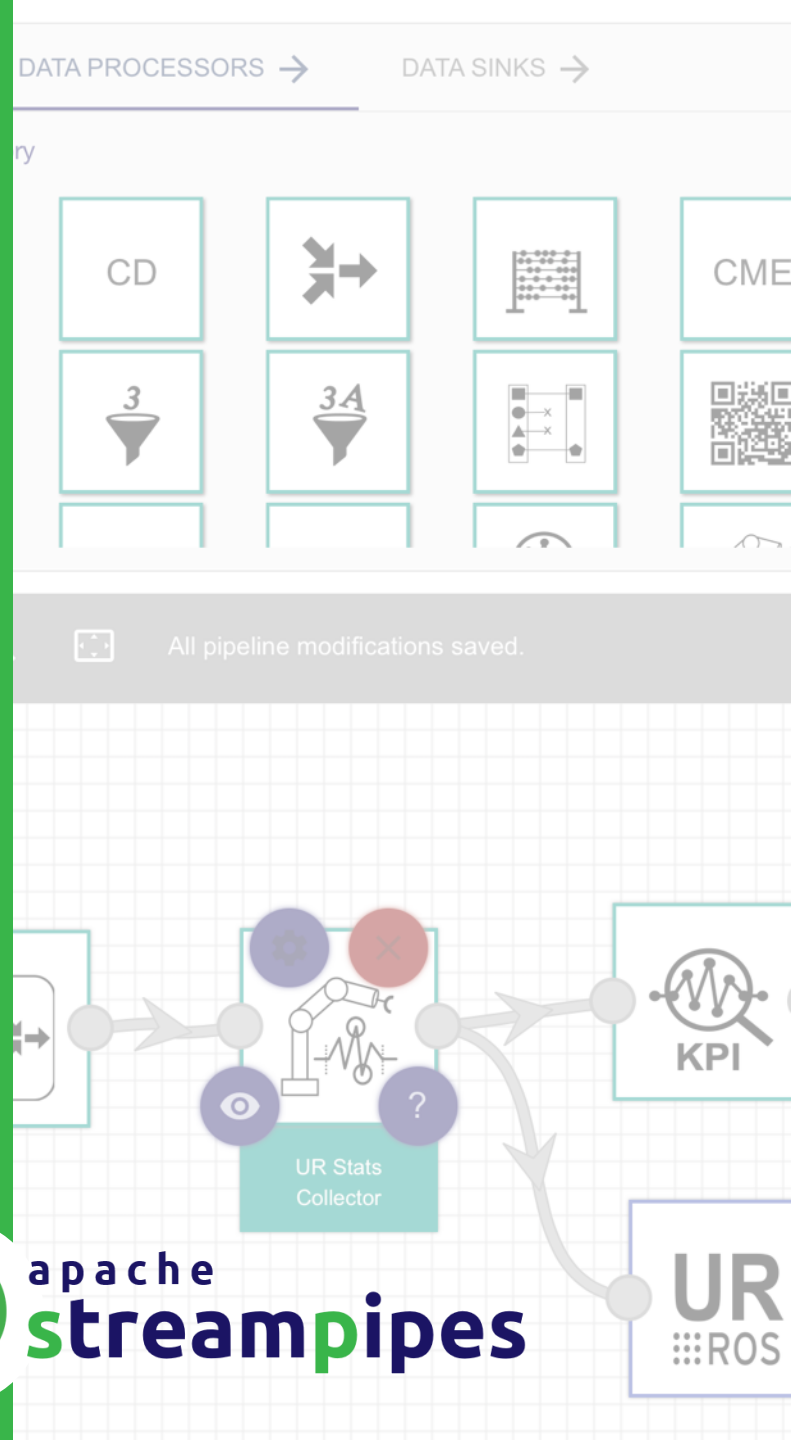
Want to get involved?

How to contribute

- Check our **issues in Jira** to get started
- Subscribe to **mailing lists** and get involved in the discussions
- Help us grow the community
- Looking to contribute?
  - Core/UI features
  - Extensions (adapters, processors, sinks)
  - Docs/Blog posts

Let‘s connect!

- streampipes.apache.org
- streampipes.apache.org/docs
- apache/incubator-streampipes
- @streampipes



apache streampipes