

IIoT-Datenanbindung und -analyse leicht gemacht

Tim Bossenmaier, Philipp Zehnder

Über uns



Tim Bossenmaier



Data Engineer @inovex



PMC



@bossenti



Philipp Zehnder



CTO @bytefabrik.ai



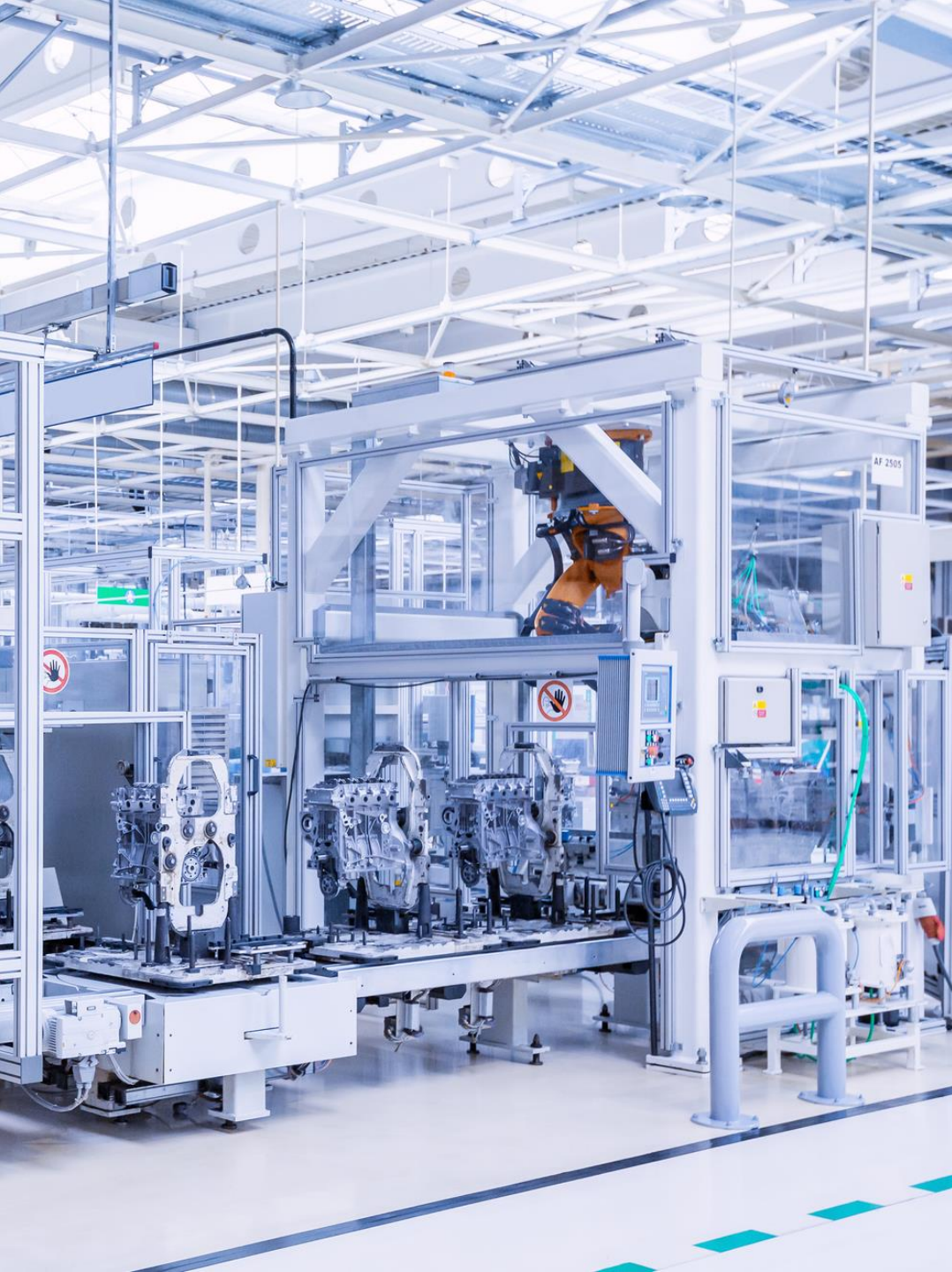
PMC



@tenthe

...und viele mehr!





Dieser Vortrag



Motivation

Systemübersicht

Feature Tour

Daten anbinden

Daten explorieren

Pipelines erstellen

Datenexport

StreamPipes erweitern

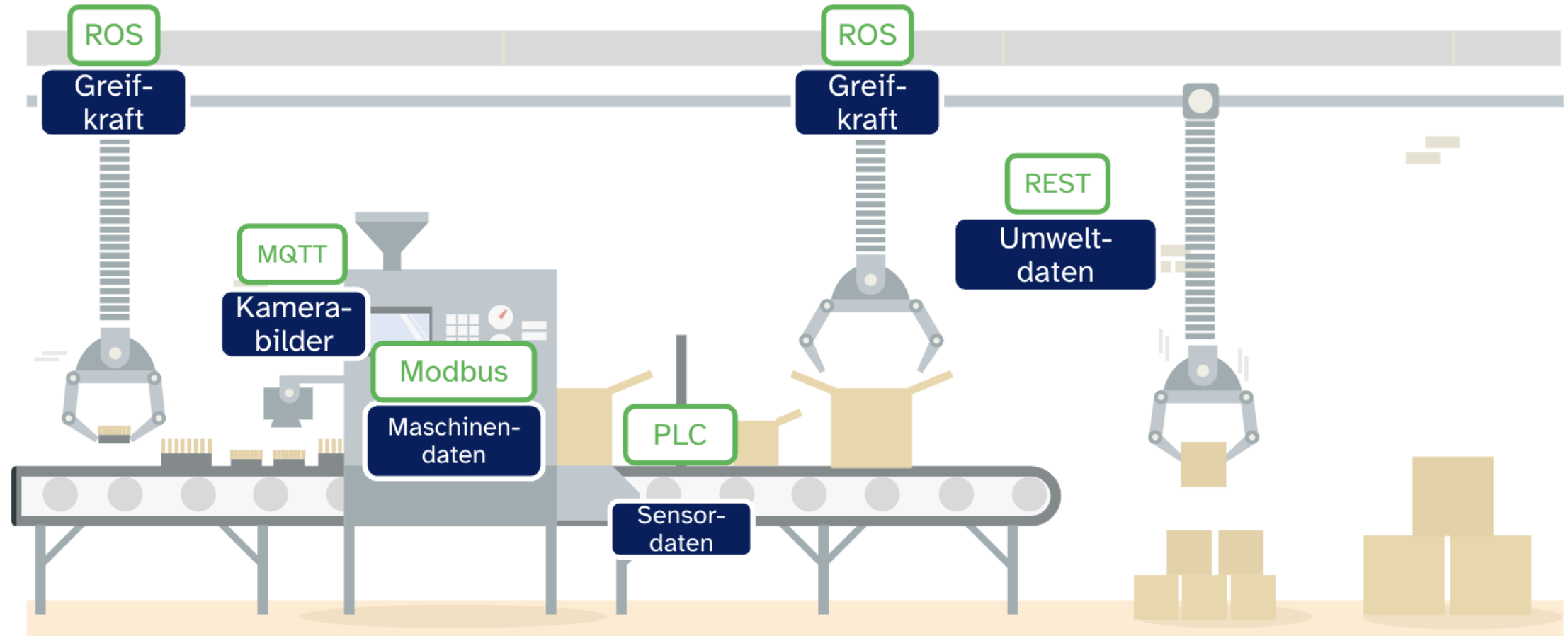
Roadmap

Motivation & Projektübersicht

IIoT in der Produktion



Datenströme überall



**Wie kann ich Maschinen- und Anlagendaten
einfach nutzbar machen?**

...vor allem als Nicht-IT-Experte?

Apache StreamPipes

Mission Statement



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



Apache StreamPipes

Projektübersicht



Ursprünglich entstanden
am FZI Forschungszentrum
Informatik, Karlsruhe

Apache Incubator seit 11/2019
Apache TLP seit 11/2022

7 Apache Releases

14 PMC Member

12 Committer

23 Contributor





Apache StreamPipes

Integration mit anderen ASF-Projekten

IIoT-Datenanbindung



Konnektoren



Visualisierungen



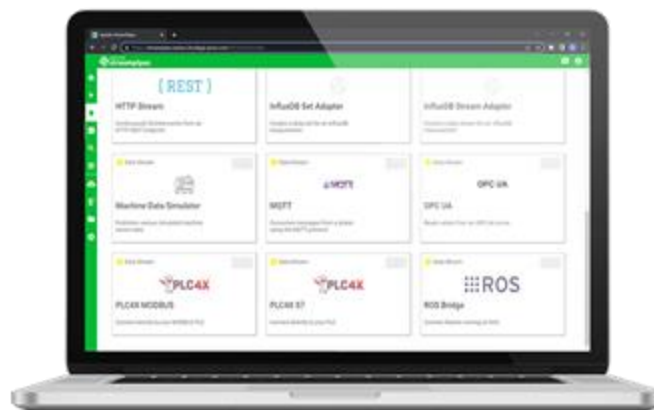
Systemübersicht

Module



1

Quickly connect IIoT data streams



Pipelines to harmonize & analyse data

2



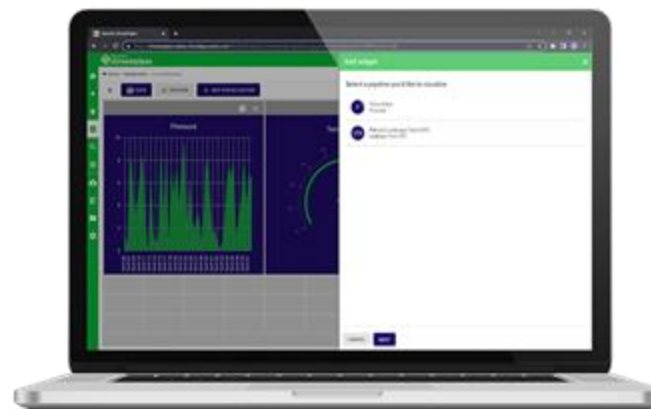
3

Visually explore IIoT data



Live dashboard

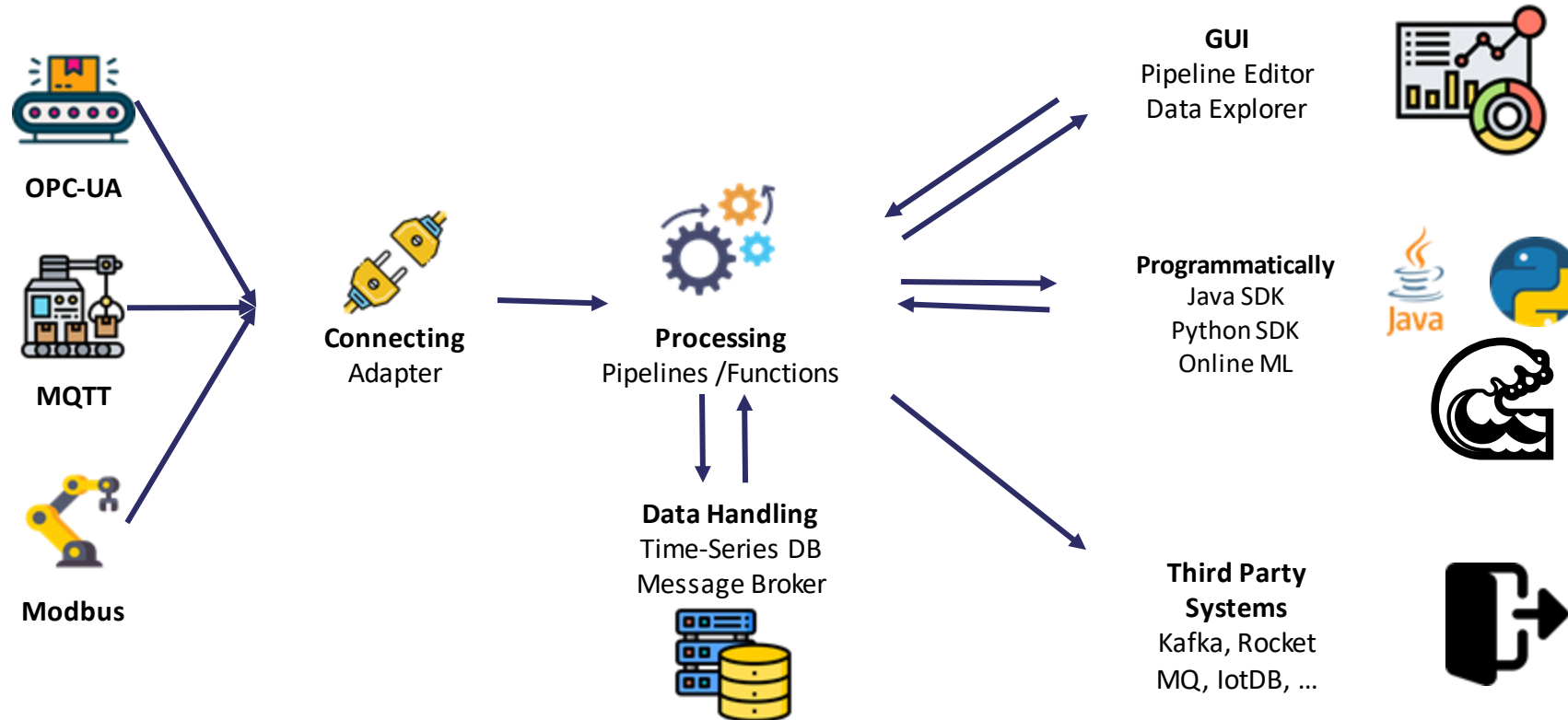
4





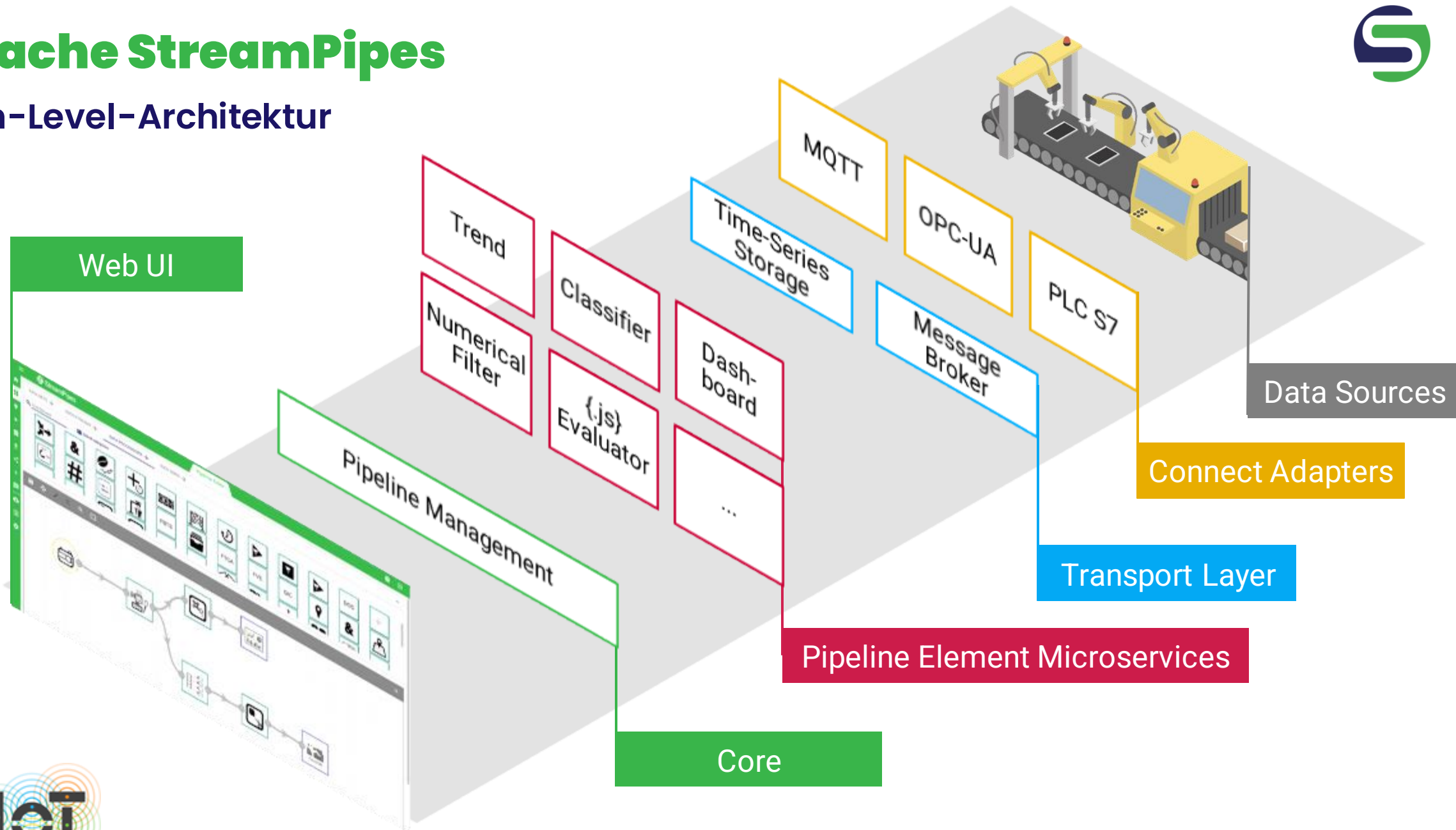
Apache StreamPipes

Datengewinnung und Datennutzung einfach gemacht



Apache StreamPipes

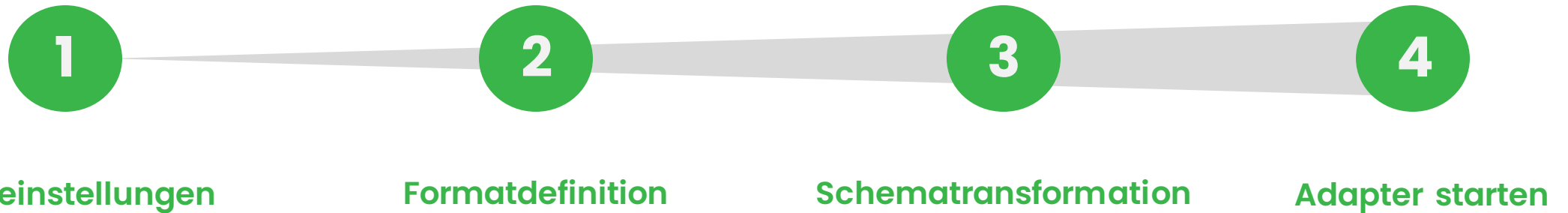
High-Level-Architektur



**Industrielle Daten
einfach anbinden**

Datenanbindung

Prozess



Protokollspezifische Einstellungen
Zugriffsrechte
Frequenz

Formatspezifische Einstellungen
z.B. JSON oder CSV

Datentypen
Leichtgewichtige Semantik
Vorverarbeitungsregeln

Direkter Start
Persistierung



A young boy and girl are sitting at a wooden desk, looking at a silver laptop. The boy, on the left, is wearing a dark blue t-shirt with 'YAMAHA' printed on it and has his arms raised in excitement. The girl, on the right, is wearing a green and white striped shirt and is pointing at the laptop screen. Both children have wide, joyful expressions. A green rectangular box with the word 'Demo!' in white text is overlaid in the center of the image. The background shows an office-like setting with a green wall and a red exit sign.

Demo!

**Industrielle Daten
einfach explorieren**

Daten visualisieren & explorieren



Tools

Live Dashboard

Live-Daten
konsumieren, z.B. für
Shopfloor-Monitoring

Data Explorer

Daten visuell
explorieren und
Datenverständnis
fördern





Data Explorer

Visualisierungen & Prozess

Voraussetzung:
Persistierte Daten

1

Data View erstellen

2

Daten konfigurieren &
visualisieren

3

Zeitspanne auswählen

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

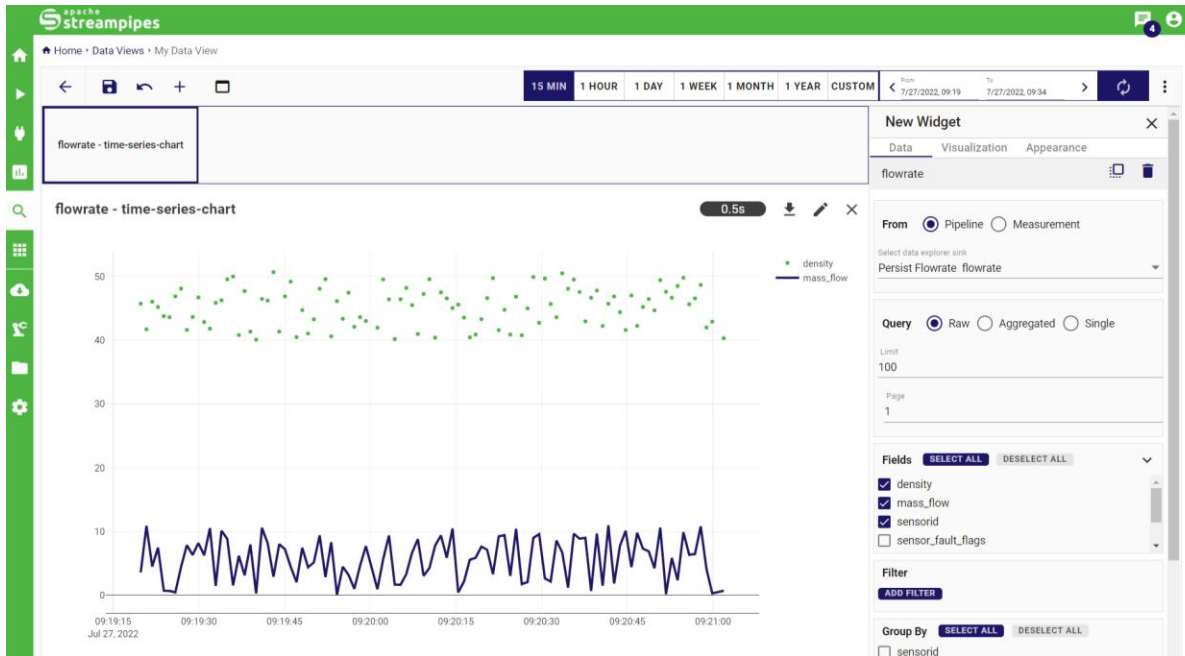
A young boy and girl are sitting at a wooden desk, looking at a silver laptop. The boy, on the left, is wearing a dark blue t-shirt with 'YANKEES' printed on it and has his arms raised in excitement. The girl, on the right, is wearing a green and white striped shirt and is pointing at the laptop screen. Both children have wide, joyful expressions. A green rectangular box with the word 'Demo!' in white text is overlaid in the center of the image. The background shows an office-like setting with a green wall and a red exit sign.

Demo!



Data Explorer

Zusammenfassung



Konfiguration

Datenkonfiguration

Visualisierungskonfiguration

Darstellungskonfiguration

Zeitauswahl

Gilt immer für alle Widgets in einer View

Views

Grid View („Dashboard-Stil“)

Slide View („Powerpoint-Stil“)

Datenvolumen

Warnung bei Auswahl zu hoher Datenmenge
oder automatische Datenreduktion



Datenanalysepipelines erstellen

Pipelines und Pipeline-Elemente



Pipeline

=

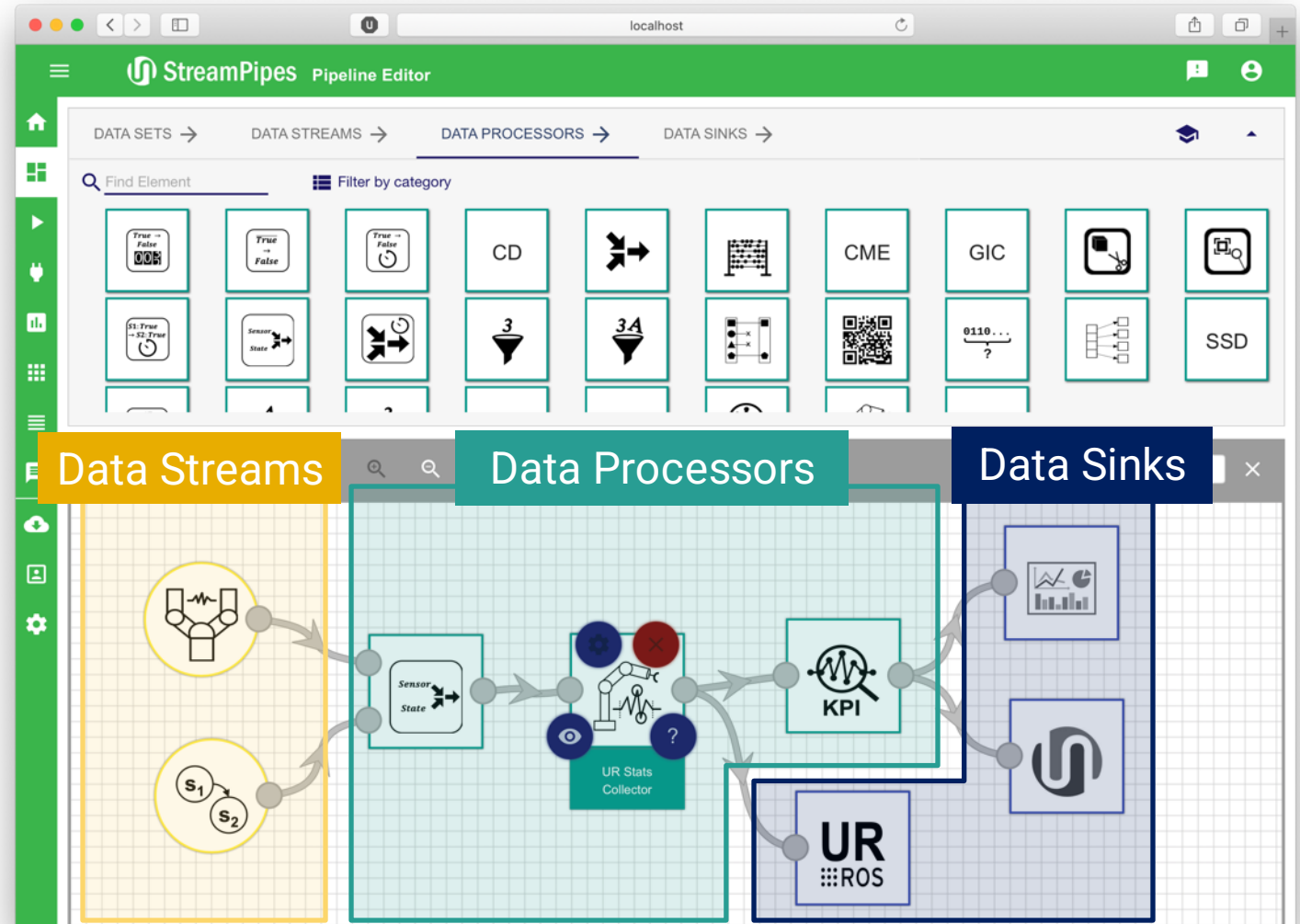
Data Streams

+

Data Processors

+

Data Sinks





Pipelines

Pipeline erstellen

1

Datenstrom auswählen

2

Prozessor oder Senke
auswählen

3

Pipeline-Element
konfigurieren

4

Pipeline starten

Ziel

Datenströme flexibel harmonisieren und analysieren

Vorgehen

Drag-and-Drop-Interface

Automatische Überprüfung

System verbietet die Verbindung von Pipeline-Elementen, die zueinander inkompatibel sind

A photograph of two young children, a boy and a girl, sitting at a wooden desk. The boy, on the left, is wearing a dark blue t-shirt with 'YAMAHA' printed on it and has his arms raised in excitement. The girl, on the right, is wearing a green and white striped shirt and is pointing at a silver laptop. Both children have wide, joyful expressions. A green rectangular box with the word 'Demo!' in white text is overlaid in the center of the image. The background shows an office-like setting with a green wall and a red exit sign.

Demo!



Pipelines

Features

Pipeline Editor

Elementauswahl

Sortieren nach Name/Gruppe
(meistens) mit Dokumentation
Anzeige kompatibler Elemente

Debugging

Live-Vorschau

Convenience

Pipeline Element Templates
Auto Layout



Pipeline-Details

Übersicht

Übersicht von Pipelines und Aktionen
Bearbeitung von nicht-aktiven Pipelines

Monitoring

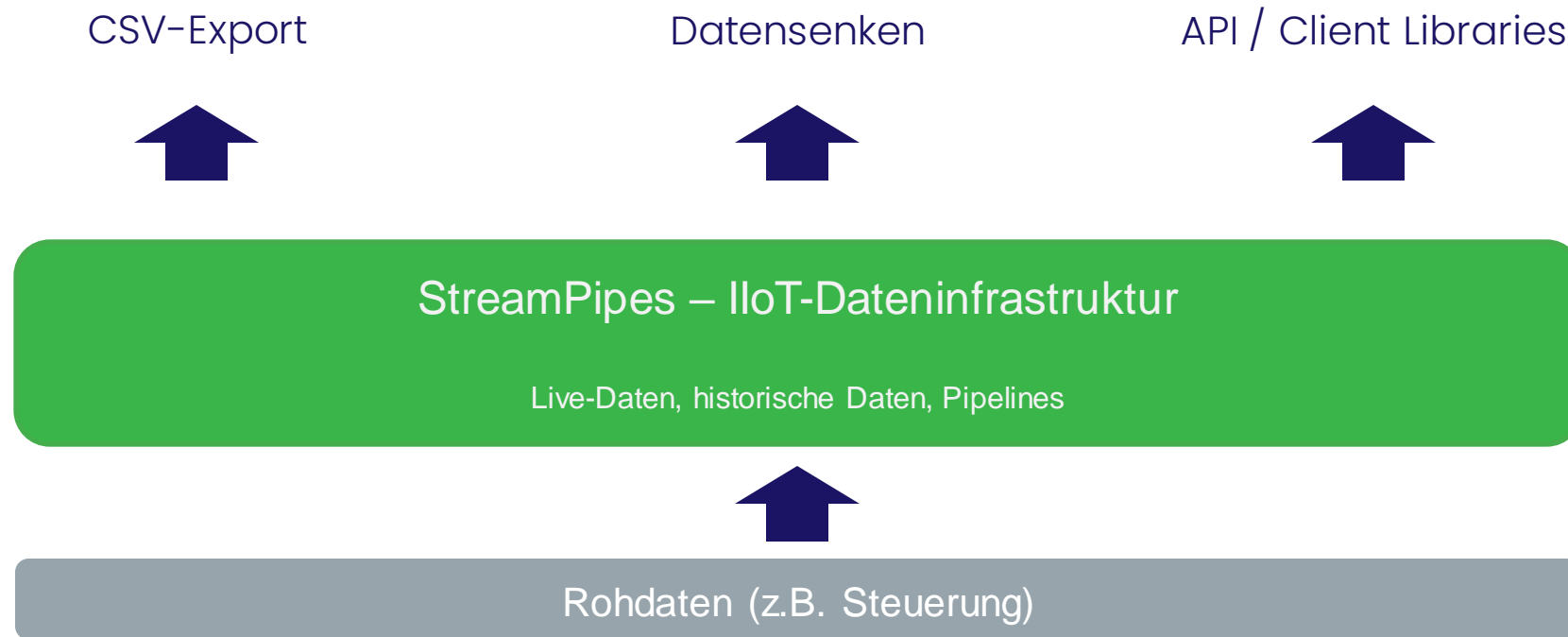
Übersicht über Pipelines, Fehler und Metriken

Quick Edit

Konfigurationen schnell anpassen

Daten extern nutzen

Interaktion mit externen Systemen



StreamPipes-Features aus externen Anwendungen nutzen

Anwendungen

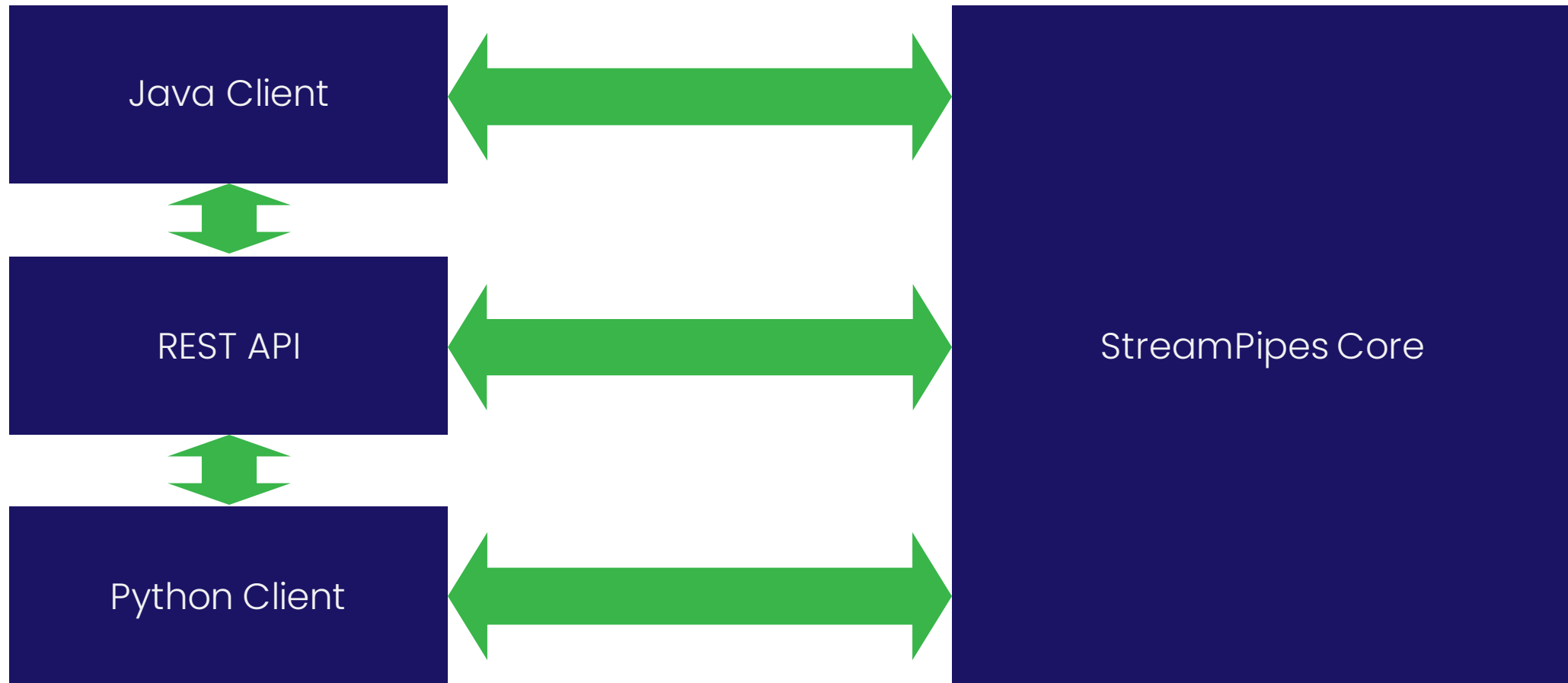
Pipeline-Lifecycle verändern

Pipelines bearbeiten aus externen Anwendungen (z.B. Schwellwerte setzen)

Live-Daten in externen Anwendungen beziehen

Historische Daten in externe Anwendungen laden

API-Kommunikation



StreamPipes Client

Java library



```
1 public static void main(String[] args) {
2     StreamPipesCredentials credentials = StreamPipesCredentials
3         .from(System.getenv("user"), System.getenv("apiKey"));
4
5     // Create an instance of the StreamPipes client
6     StreamPipesClient client = StreamPipesClient
7         .create("localhost", 80, credentials, true);
8 }
```

```
1 // Get all pipelines
2 List<Pipeline> pipelines = client.pipelines().all();
3
4 // Start a pipeline
5 PipelineOperationStatus message = client.pipelines().start(pipelines.get(0));
6
7 // Get all data streams
8 List<SpDataStream> dataStreams = client.streams().all();
9
10 // Subscribe to a data stream
11 client.streams().subscribe(dataStreams.get(0), event -> MapUtils.debugPrint(System.out, "event",
    event.getRaw()));
```





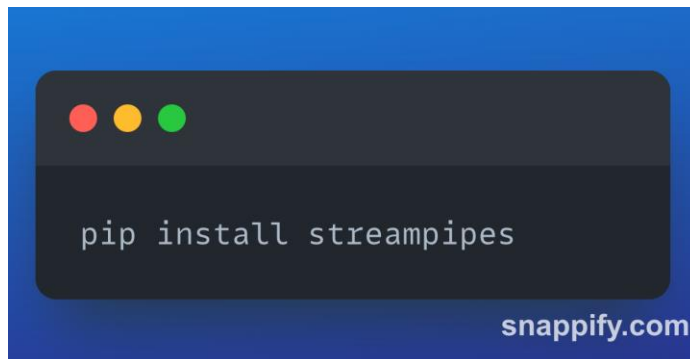
StreamPipes Client

Python Library

Übersicht

Verbindungsstück um Datenanbindung via StreamPipes mit dem großartigen Universum an Datenverarbeitungs-, Datenanalyse- und Datenvisualisierungs-Bibliotheken in Python

Verfügbar auf PyPI



Features

Client

Komfortable Interaktion mit StreamPipes API

Abruf von Daten aus Data Lake

Functions

Leichtgewichtige Prozessoren

Erlauben Interaktion mit Datenströmen

Integration mit Online ML Bibliothek River



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

streampipes.apache.org | [@streampipes](https://twitter.com/streampipes) | github.com/apache/streampipes

StreamPipes erweitern

StreamPipes-Erweiterungen entwickeln



Adapter und Pipeline-Elemente
mittels SDK

UI-Erweiterungen mittels
Microfrontends



Software Development Kit



Ziel

Erweiterungen von StreamPipes mit neuen Adaptern oder Pipeline-Elementen

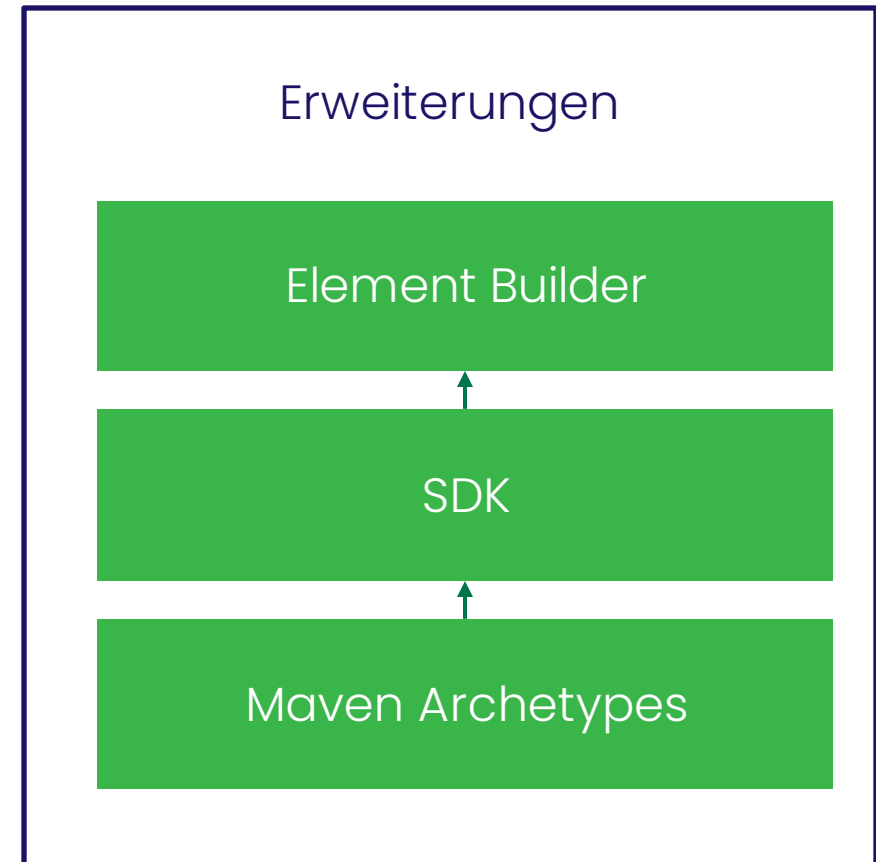
Features

Java-basiertes SDK (+ Functions für Python)

Vollkonfigurierbare UI-Elemente

Maven archetypes

Installation zur Laufzeit





UI-Erweiterungen

Microfrontends

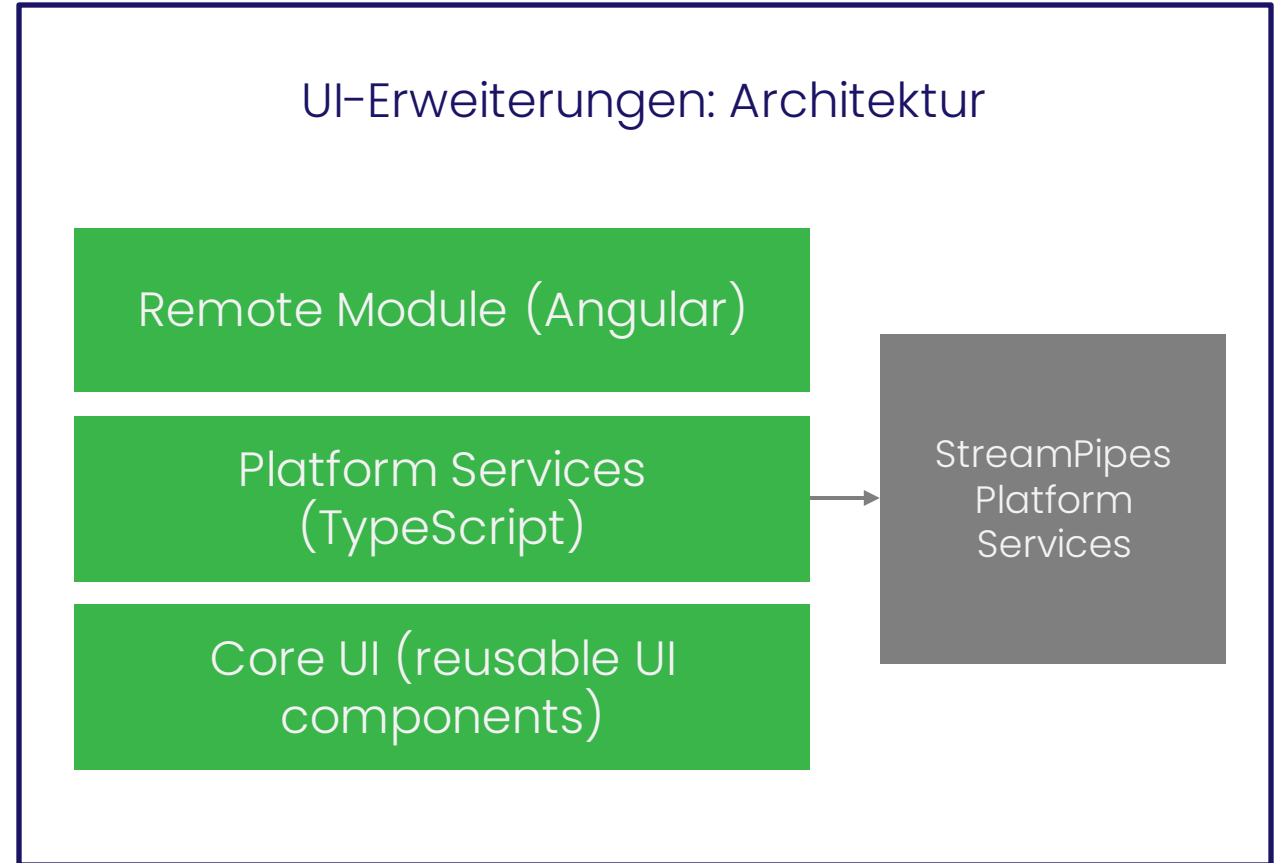
Ziel

Eigene UI-Sichten für interne Anwendungen bereitstellen

Features

Typescript-SDK für alle Plattformfeatures wie Datenschnittstellen oder Data Views

Storage-API für die Speicherung von anwendungsinternen Konfigurationen



Roadmap

Roadmap

Aktuelle Schwerpunkte



Python Library

Verbessertes
Monitoring von
Pipelines & Adaptern

Austausch
Zeitreihendatenbank
(IoTDB)



Wir freuen uns über Unterstützung!



Mitarbeiten



Ausprobieren



Feedback



Weitererzählen



Use Cases



Let's connect!



streampipes.apache.org



streampipes.apache.org/docs



[apache/streampipes](https://github.com/apache/streampipes)



[@streampipes](https://twitter.com/streampipes)



Vielen Dank!

Tim Bossenmaier

inovex GmbH

tim.bossenmaier@inovex.de

Philipp Zehnder

Bytefabrik.AI GmbH

philipp.zehnder@bytefabrik.ai

