

APACHE STREAMPIES MIKROSERVISNA ARHITEKTURA

STEFAN MARINKOVIĆ

ANA MILENKOVIĆ

ŠTA JE APACHE STREAMPIPES?

- Samouslužni industrijski IoT alat
- Omogućava ne-tehničkim korisnicima da povežu, analiziraju i istraže IoT tokove podataka



TOK PROCESA

Integriše setove i tokove podataka

korišćenjem ugrađenih StreamPipes biblioteka
sa podrškom za generičke protokole

kao što su HTTP, Kafka, MQTT

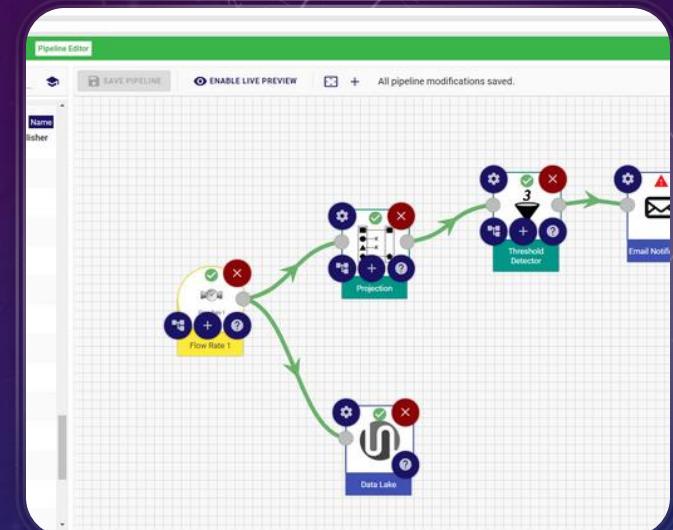
Povezivanje

Analiza

Eksplotacija

Harmonizuje i analizira podatke korišćenjem
postojećih real-time algoritama i daje mogućnost
kreiranja novih algoritama

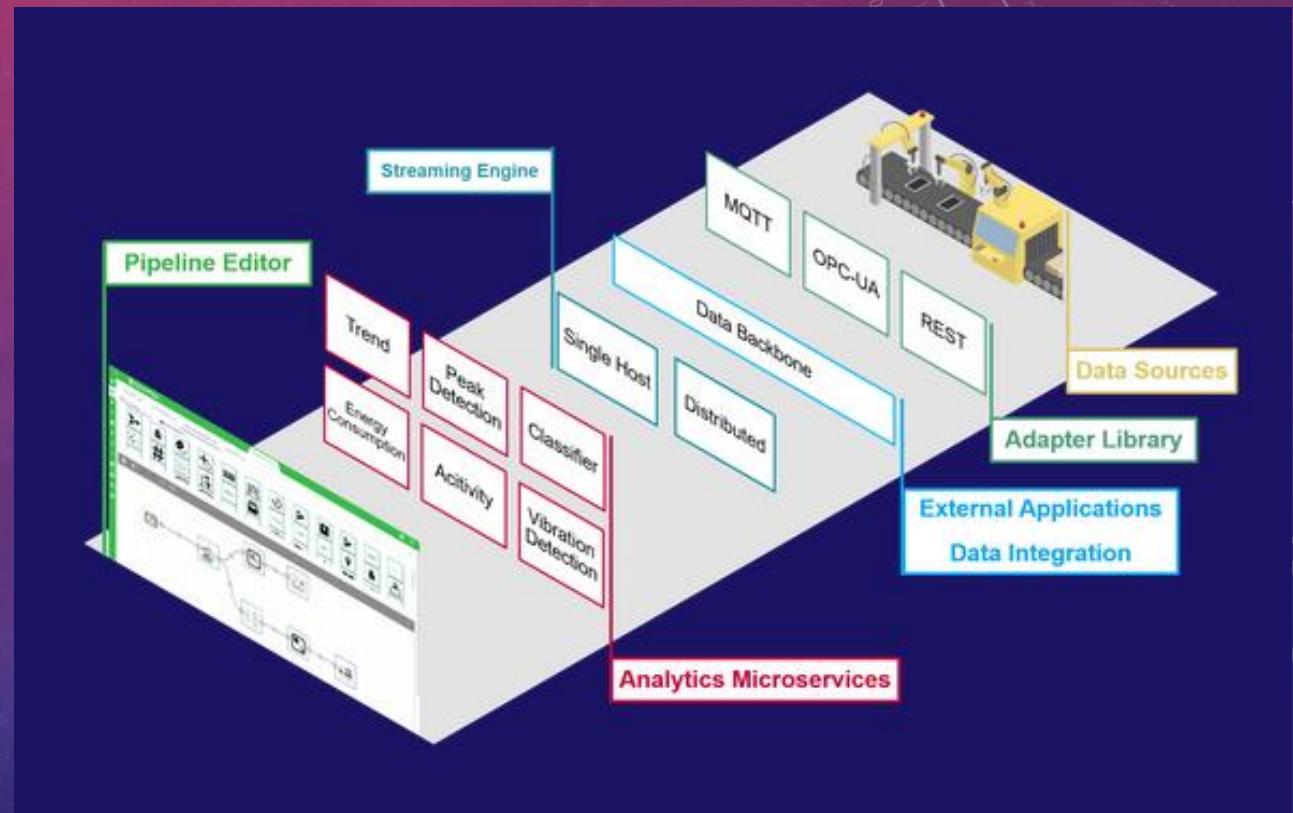
Notifikacije, dashboard, slanje
podataka third-party sistemima
ili eksternim servisima



ADAPTERS		
Kafka	Data Stream (Apache Kafka)	Data Set (File Set)
Apache	Data Stream (Apache Pulsar)	Data Set (HTTP Server)
REST	{ REST } (HTTP Server)	{ REST } (HTTP Set)

SAMOUSLUŽNA IOT ANALITIKA PODATAKA

- IoT analitika podataka je dostupna svima koji koriste Streampipes
- Grafički interfejs i modeler dopušta biznis analitičarima da definišu procese u pipeline-u, bez potrebe za data ekspertima



GLAVNE ODLIKE APACHE STREAMPIPES

FLEKSIBILAN PIPELINE

La ko za korišćenje

Inteligentno

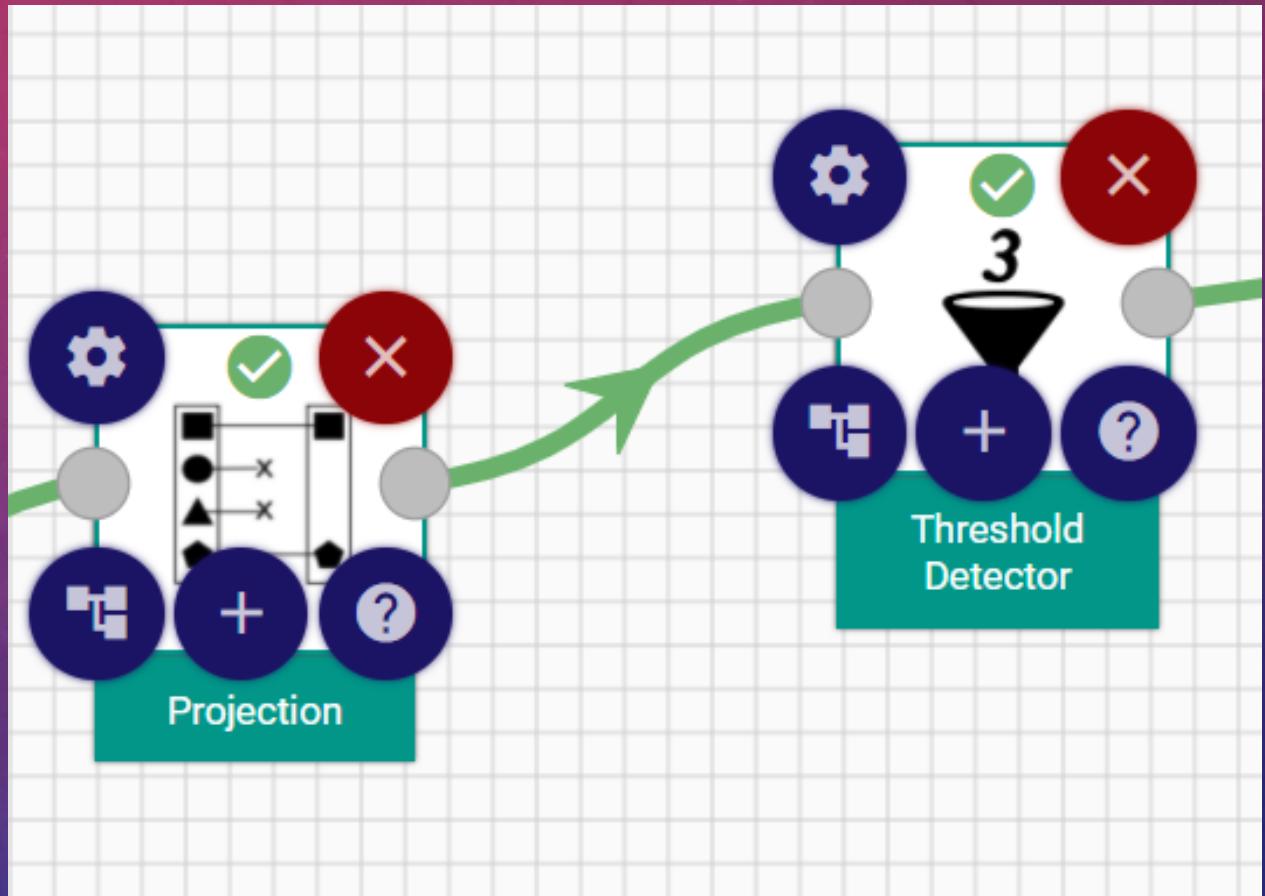
Moćno

Lako

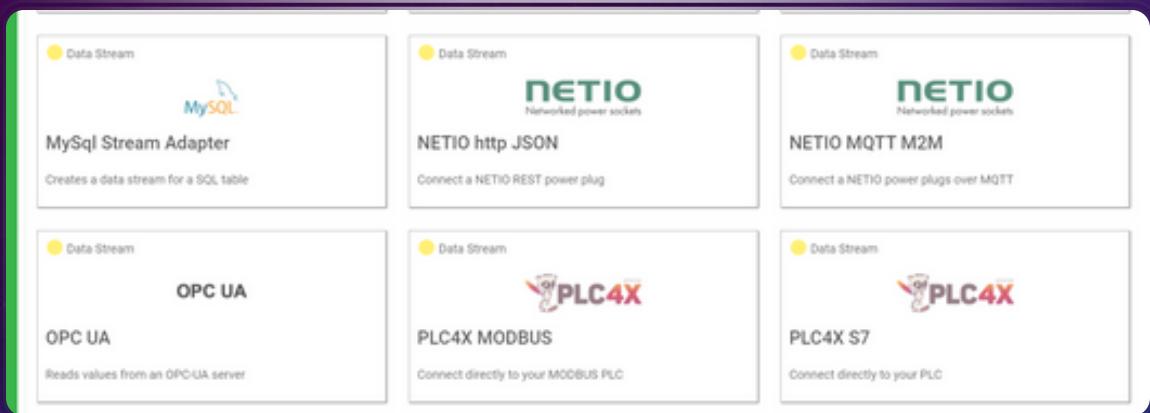
- Nudi korisnički interfejs baziran na Web-u
- Lako kreiranje pipeline-a za procesiranje podataka
- Grafički editor

Inteligentno

- Model podataka uzima u obzir semantiku tokova podataka i setova
- Podrška za modeliranje
- Elementi se mogu povezati samo kada se njihovi zahtevi poklapaju



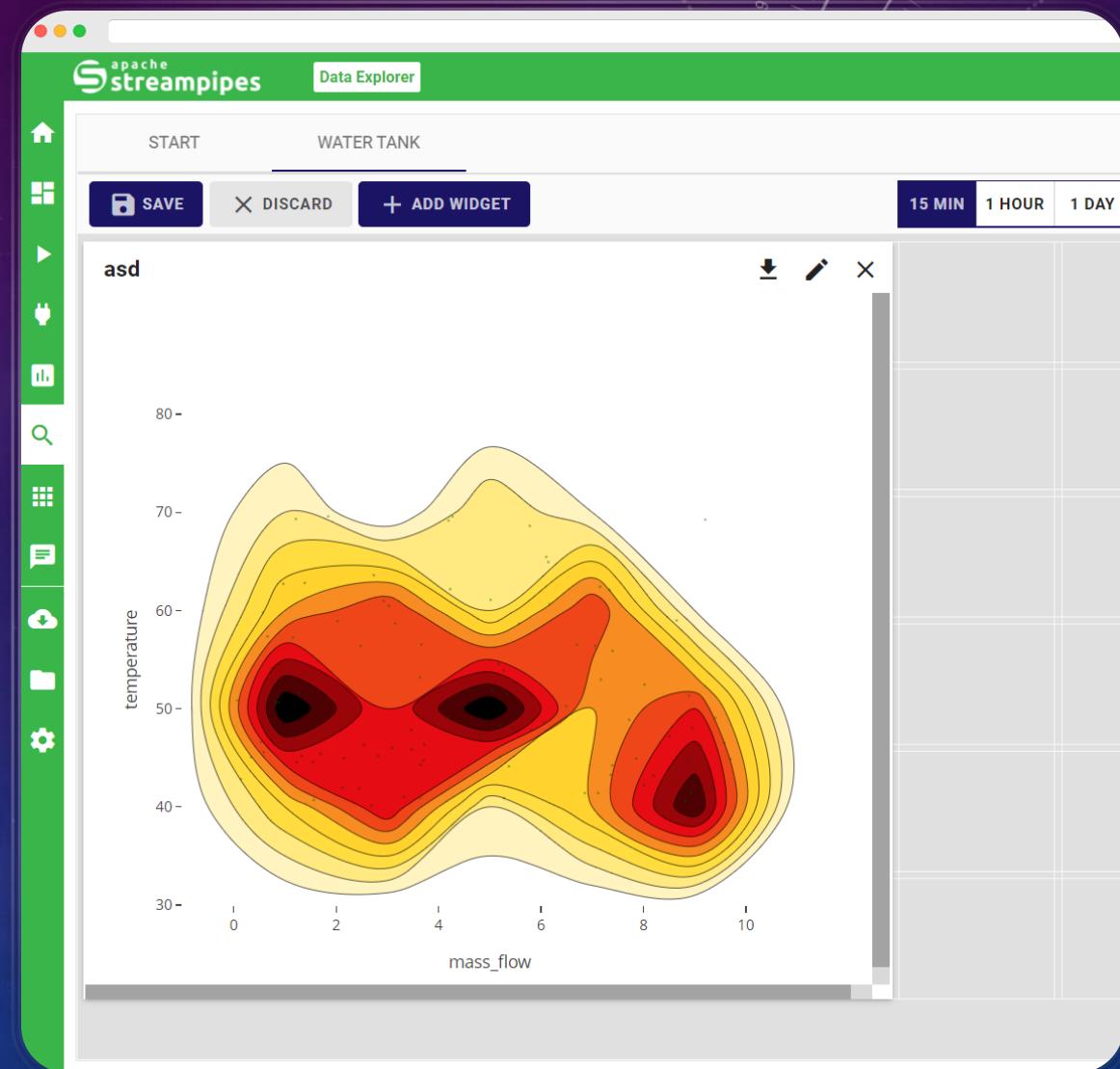
BRZO POVEZIVANJE IOT PODATAKA



- Podrška za setove i tokove podataka
- Generički adapteri
- Adapteri se lako konfigurišu sa korisničkog interfejsa
- Uključuje širok spektar popularnih protokola i tehnologija:
 - HTTP, REST, MQTT, Kafka, OPCUA, S7, Modbus, ROS
 - Web API-ji za ubacivanje podataka u Strempipes
 - Adapteri se mogu eksportovati i deliti

VIZUALIZACIJA PODATAKA U REALNOM VREMENU

- Dashboard u realnom vremenu
- Širok spektar dostupnih grafova
- Linijski grafovi, semafori, tabele...
- Data Explorer – brza vizualizacija i pregled vremenskih serijskih podataka
- Kreiranje upita u editoru bez koda



MANAGE ENDPOINTS

ALL

DATA SETS

DATA STREAMS

DATA PROCESSORS

INSTALL SELECTED

UNINSTALL SELECTED



Data Processor

UNINSTALL ...

Boolean Counter

Increases a counter on each change of a boolean value

Data Processor

UNINSTALL ...

Boolean Filter

Keeps events with the selected value

Data Processor

UNINSTALL ...

Boolean Logical Operator

Performs a logical boolean operation b/w selected fields

Data Processor

UNINSTALL ...

Boolean Timer

Measures how long a boolean measure

PROŠIRIVO

- Mogućnosti Strempipes-a su lako proširive korišćenjem ugrađenog SDK-a
- Elementi pipeline-a su enkapsulirani kao samostalni mikroservisi
- Lako dodavanje novih elemenata kroz Pipeline Element Marketplace
- Ukoliko su potrebni skroz novi elementi, mogu se kreirati uz pomoć dostupnog SDK-a i Maven-a

OSTALE ODLIKE

- Rukovođenje korisnicima i njihovim dozvolama
- Kreiranje korisnika, grupa
- Manipulacija vidljivošću resursa (adAPTERA, pipeline-a)
- Ugrađeni konfigurabilni sistem za mejlove
- Praćenje "zdravlja" pipeline-a i restartovanje u slučaju problema

The screenshot shows the Apache StreamPipes configuration interface. On the left, there's a sidebar with various icons. The main area has tabs for GENERAL, DATALAKE, MAIL, MESSAGING, and PIPELINE. Under the GENERAL tab, there are sections for User Accounts, Service Accounts, and Groups. The User Accounts section shows existing accounts like 'admin@stempipes.apache.org'. An 'Add user' dialog is open on the right, titled 'Add user'. It has a 'Basics' tab with fields for Email*, Full Name*, Password, Initial password*, Repeat password*, Groups, Roles (checkboxes for Admin, App User, etc.), and an Account tab. Buttons for SAVE and CANCEL are at the bottom.

Apache StreamPipes Configuration

GENERAL DATALAKE MAIL MESSAGING PIPELINE

User Accounts
Add and edit user accounts

+ NEW USER

Email
admin@stempipes.apache.org

Service Accounts
Add and edit service accounts

+ NEW SERVICE ACCOUNT

Username
sp-service-client

Groups
Manage user groups

+ NEW USER GROUP

Group name

Add user

Basics

Email *

Full Name *

Password

Initial password *

Repeat password *

Groups

Roles

Admin

App User

Dashboard User

Dashboard Admin

Data Explorer User

Data Explorer Admin

Connect Admin

Pipeline User

Pipeline Admin

Account

SAVE CANCEL

ARHITEKTURA STREAMPIPE-A

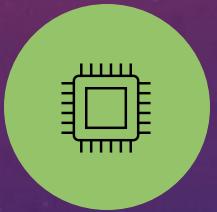
GRADIVNI ELEMENTI



Adapter



Data Set/Stream

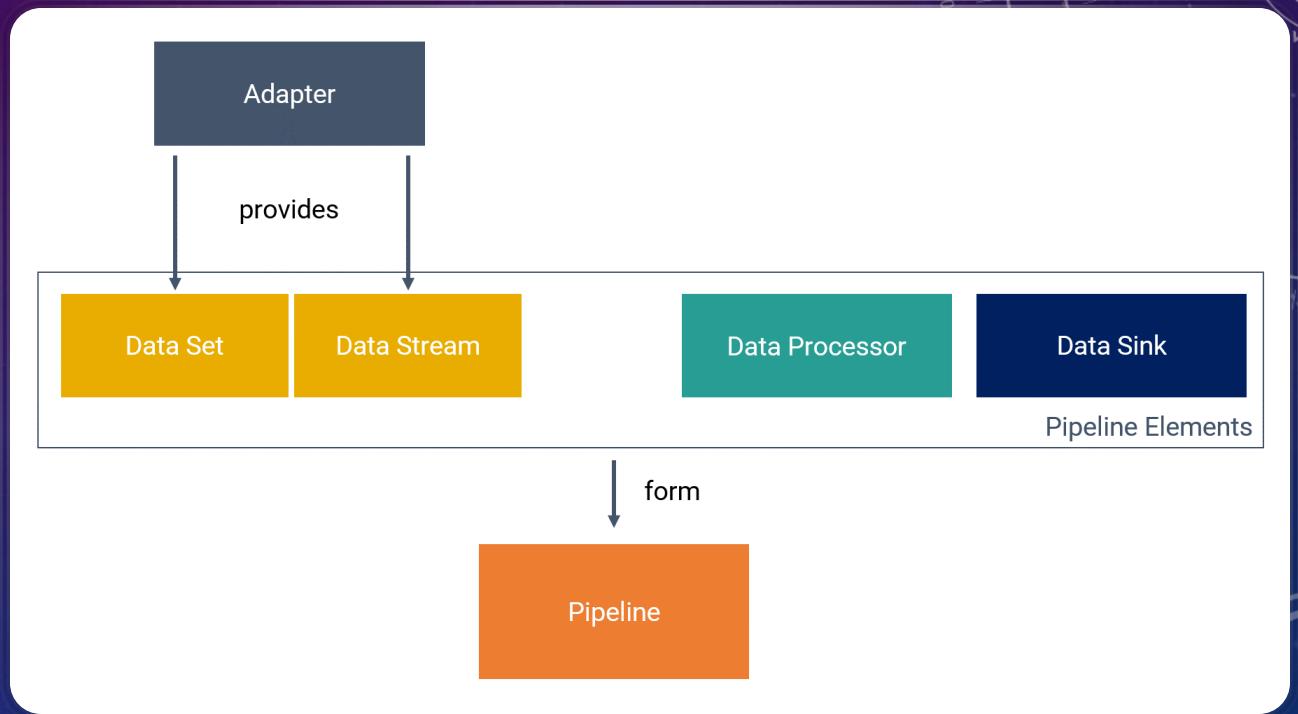
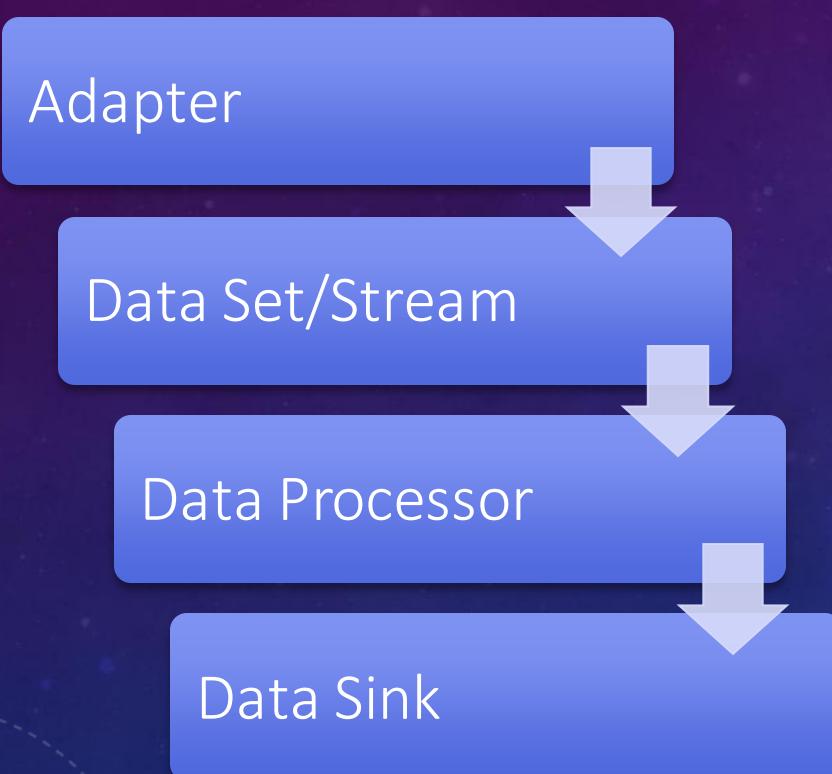


Data Processor



Data Sink

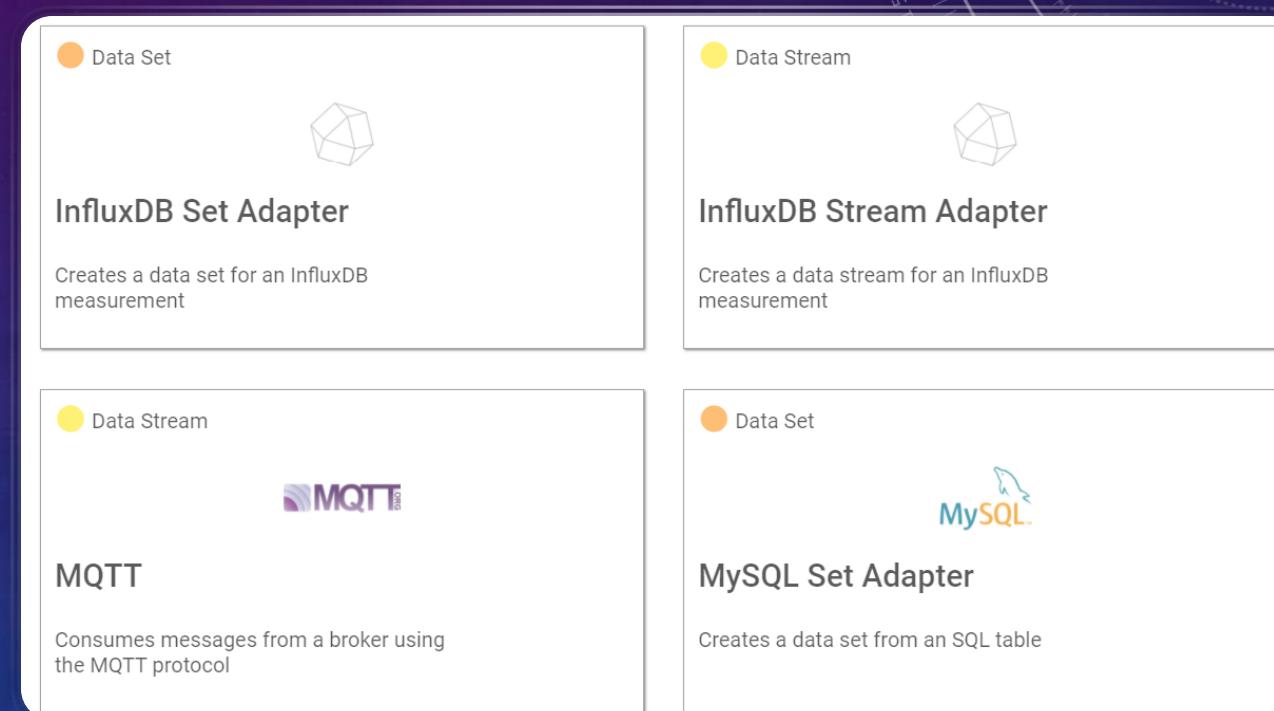
KAKO SU POVEZANI



Data Set/Stream, Processor i Sink formiraju Pipeline

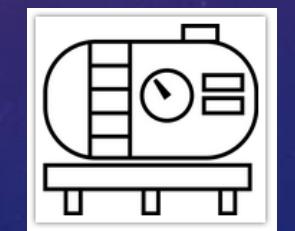
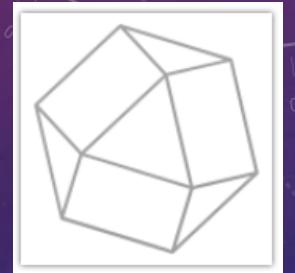
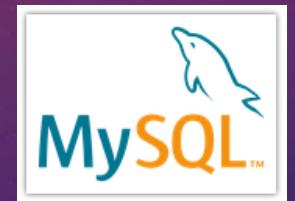
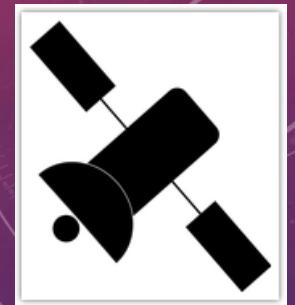
ADAPTER

- Povezuje se na bilo koji eksterni izvor podataka i prosleđuje primljene evente internom Streampipes sistemu
- Izlaz iz adaptera dostupan je vidu dva primarna bloka, Data Set i Data Stream
- Adapteri mogu biti kreirani putem Streampipes Connect, modul za povezivanje izvora podataka, ili može biti definisan korišćenjem SDK-a



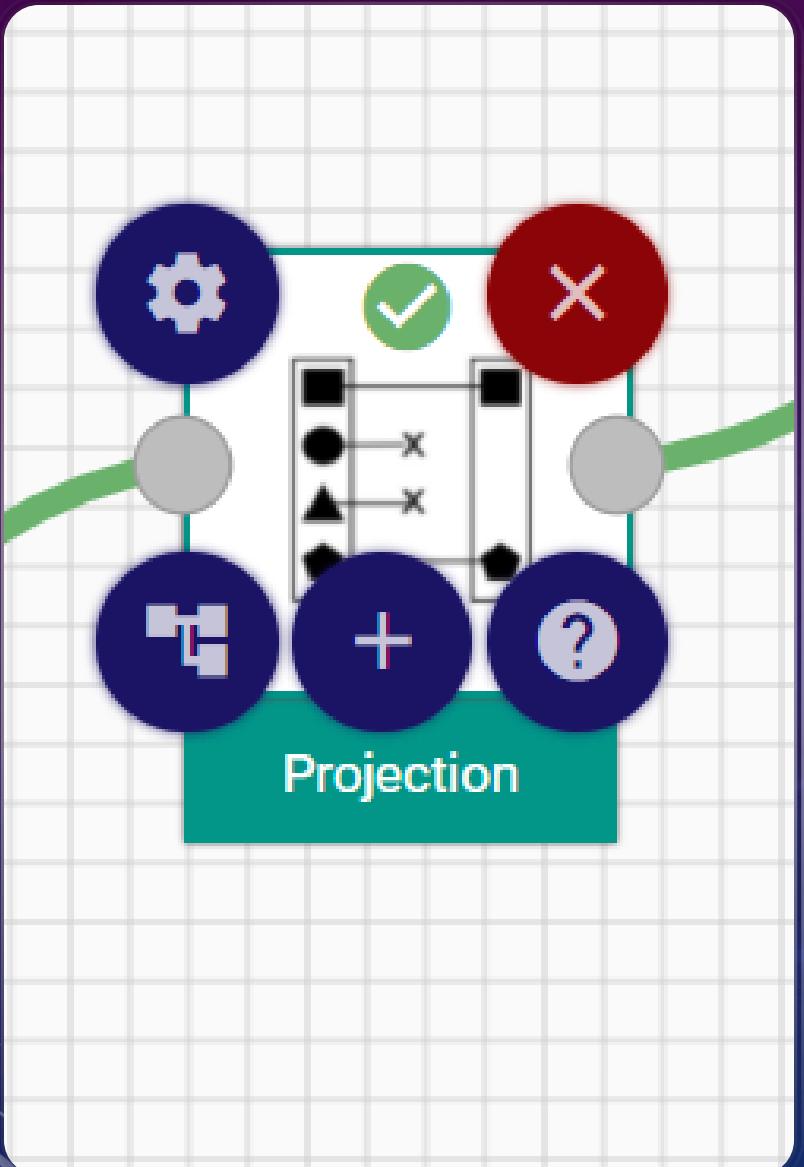
POSTOJEĆI ADAPTERI

- Apache Kafka – preuzima poruke sa Apache Kafka Broker-a
- File Stream – kontinualno preuzima podatke iz fajla
- HTTP Server – kontinualno pribavlja event-e sa HTTP REST endpoint-a
- ISS Location – prikazuje trenutnu poziciju Internacionalne Svetmirske Stanice (ISS)
- Image Upload – postavi zip fajl sa slikama i kreiraj jedan event za svaku sliku
- InfluxDB – kreira Data Set/Stream za InfluxDB
- MQTT – preuzima poruke sa broker-a korišćenjem MQTT protokola
- MySQL Adapter – kreira Data Set/Stream iz SQL tabele
- Machine Data Simulator – simulira podatke sa senzora mašine (pritisak, protok)
- ROS Bridge – povezuje robote koje pokreće ROS



DATA SET/STREAM

- Predstavljaju primarni izvor za rad sa event-ovima
- Stream je uređena sekvenca evenata
- Event se sastoji od jedne ili više obzervacionih vrednosti i dodatnih metapodataka
- Struktura (tj. šema) eventa nekog data stream-a ili set-a, čuva se u internom semantičkom registru Streampipes-a
- Za razliku od stream-a, setovi su ograničeni, imaju fiksan kraj, i bivaju konstantno reproducovani od strane sistema, od početka do kraja, kada se iskoriste kao deo pipeline-a



DATA PROCESORI

- Transformišu jedan ili više ulaznih tokova podataka u izlazni tok podataka
- Transformacije mogu biti jednostavne, na primer filtriranje bazirano na predefinisanom pravilu
- Mogu biti i kompleksne, kao što je primenjivanje algoritama baziranih na pravilima ili učenju
- Procesor se može primeniti nad bilo kojim tokom podataka koji odgovara zahtevima ulaza
- Većina procesora može biti konfigurisana da obezbedi korisnički-definisane parametre
- Definiše zahteve koje tok mora da ispunji, kao što su minimalne osobine koje event mora dostaviti
- Mogu čuvati stanje ili biti stateless

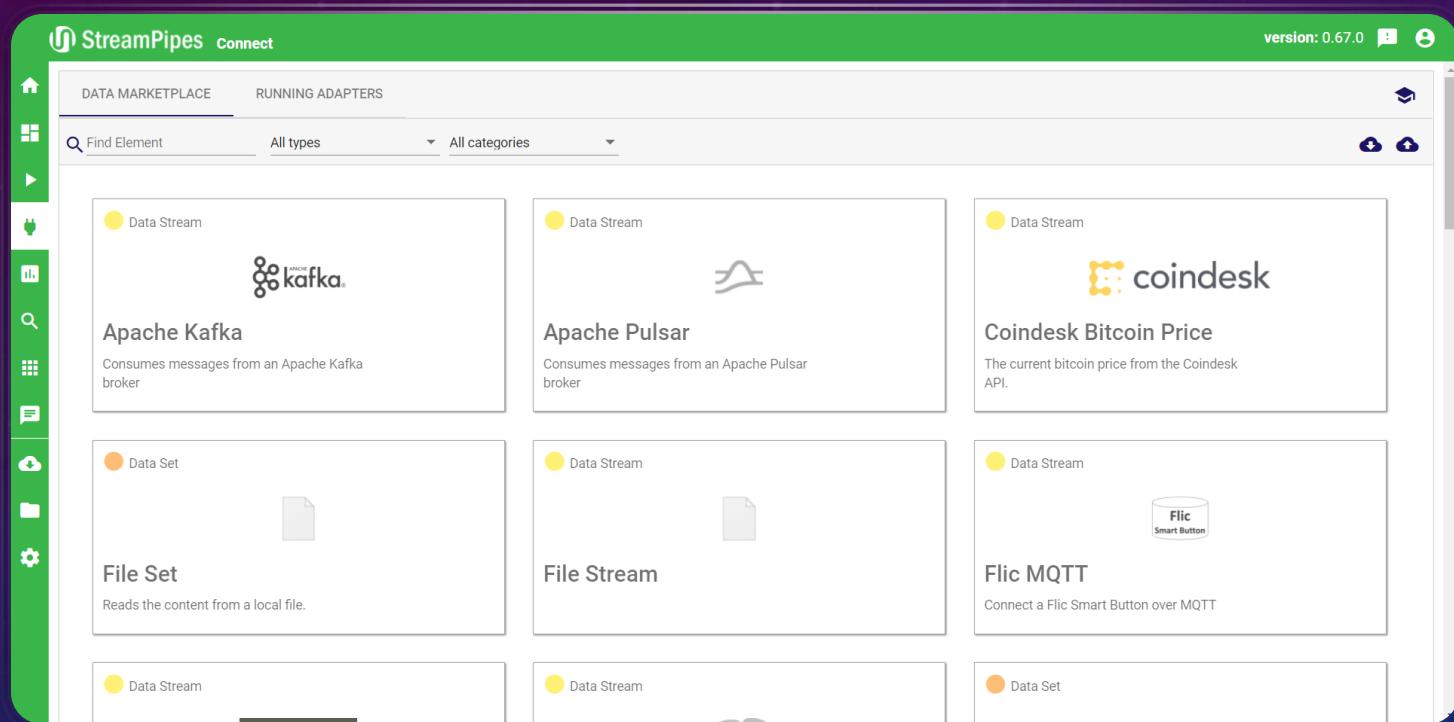
POSTOJEĆI DATA PROCESORI

- Agregacija - izvršava funkcije agregacije
- And – proverava da li se neki event dešava zajedno sa drugim eventom za zadato vreme
- Boolean Counter – broji koliko je puta bool promenljiva promenila vrednost
- Chunker - vrši segmentaciju datih tokena
- Count Array – uzima listu, računa veličinu i dodaje rezultat na kraj evenata
- Event Rate - vraća broj evenata po sekundi
- Field Mapper - računa novo polje od jedno ili više starih polja i računa njihovu heš vrednost
- Language Detection – detektuje jezik ulaznog teksta
- Projection - vraća samo odabrane vrednosti sa ulaza
- State Buffer – baferuje vrednost senzora, dokle god se stanje ne menja



MODULI APACHE STREAMPIPES SISTEMA

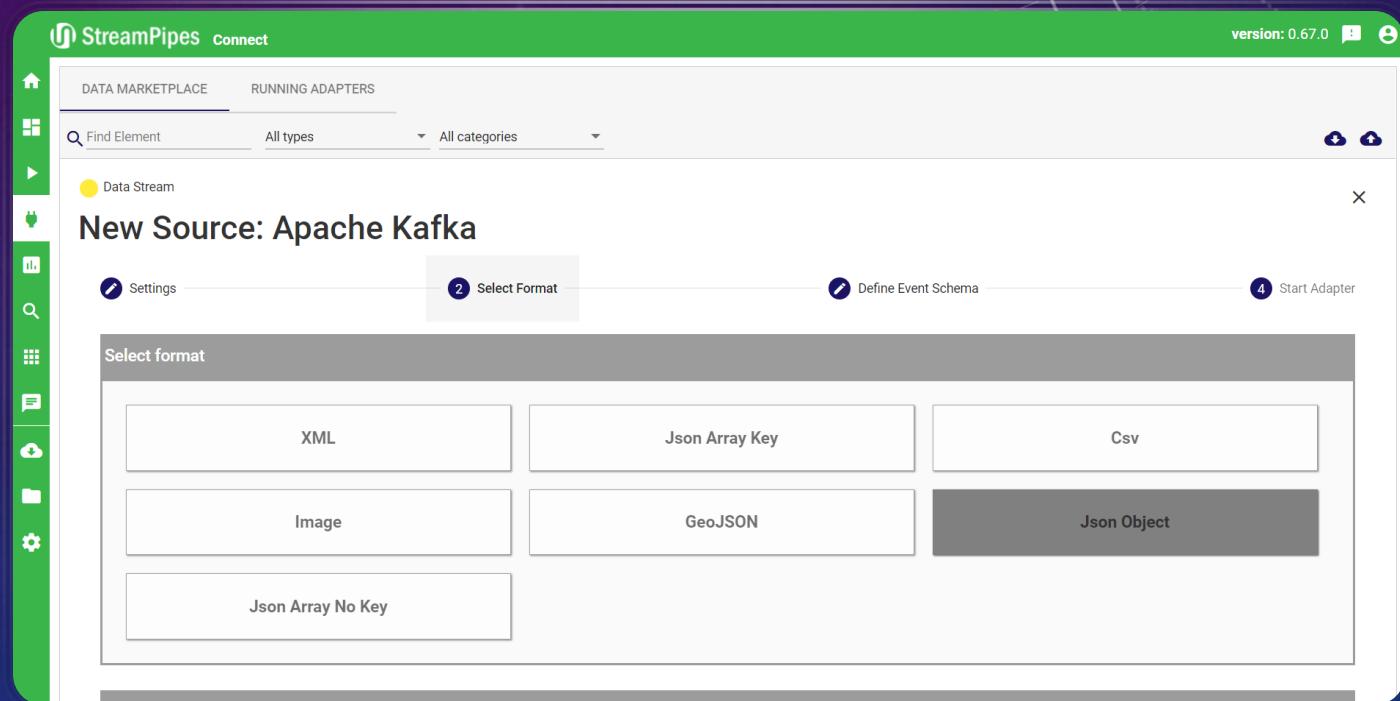
STREAMPIPES CONNECT



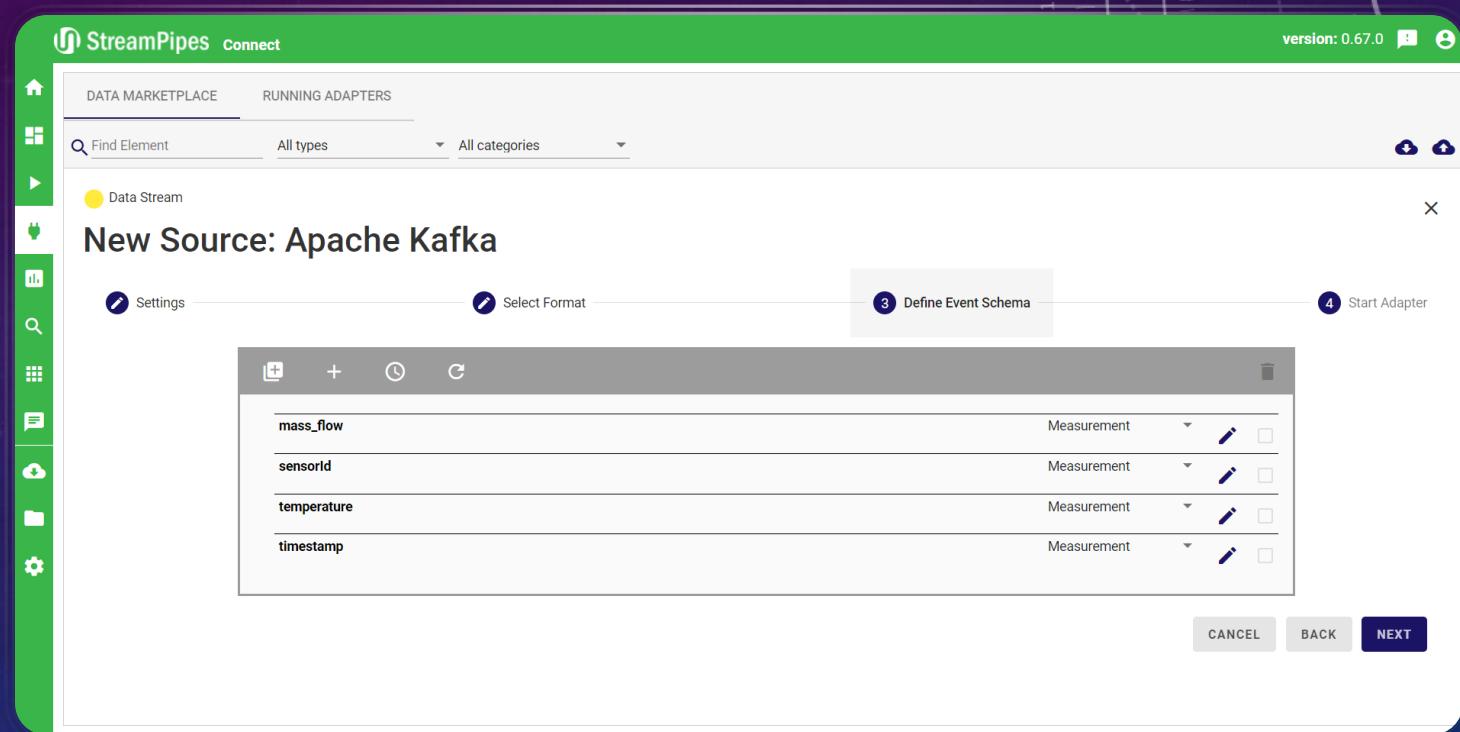
- Moduo za povezivanje eksternih izvora podataka
- Nudi veliki izbor adaptera za komunikacione protokole i neke specifične senzore
- Nudi i načine za preprocesiranje podataka bez upotrebe pipeline-a
- Pruža interfejs ka marketu adaptera trenutno instaliranih u Apache Streampipes

SPECIFIKACIJA FORMATA

- Sledeći korak u konfiguraciji
- Samo za generičke adapttere, koji podržavaju različite formate poruka koje se prenose
- Na primer, message broker-i koji mogu konzumirati poruke u JSON i u binarnom formatu
- Trenutno podržani formati uključuju: XML, razne JSON reprezentacije, slike i CSV
- U zavisnosti od izabranog formata, usleđuju dodatne konfiguracije specifične za taj format

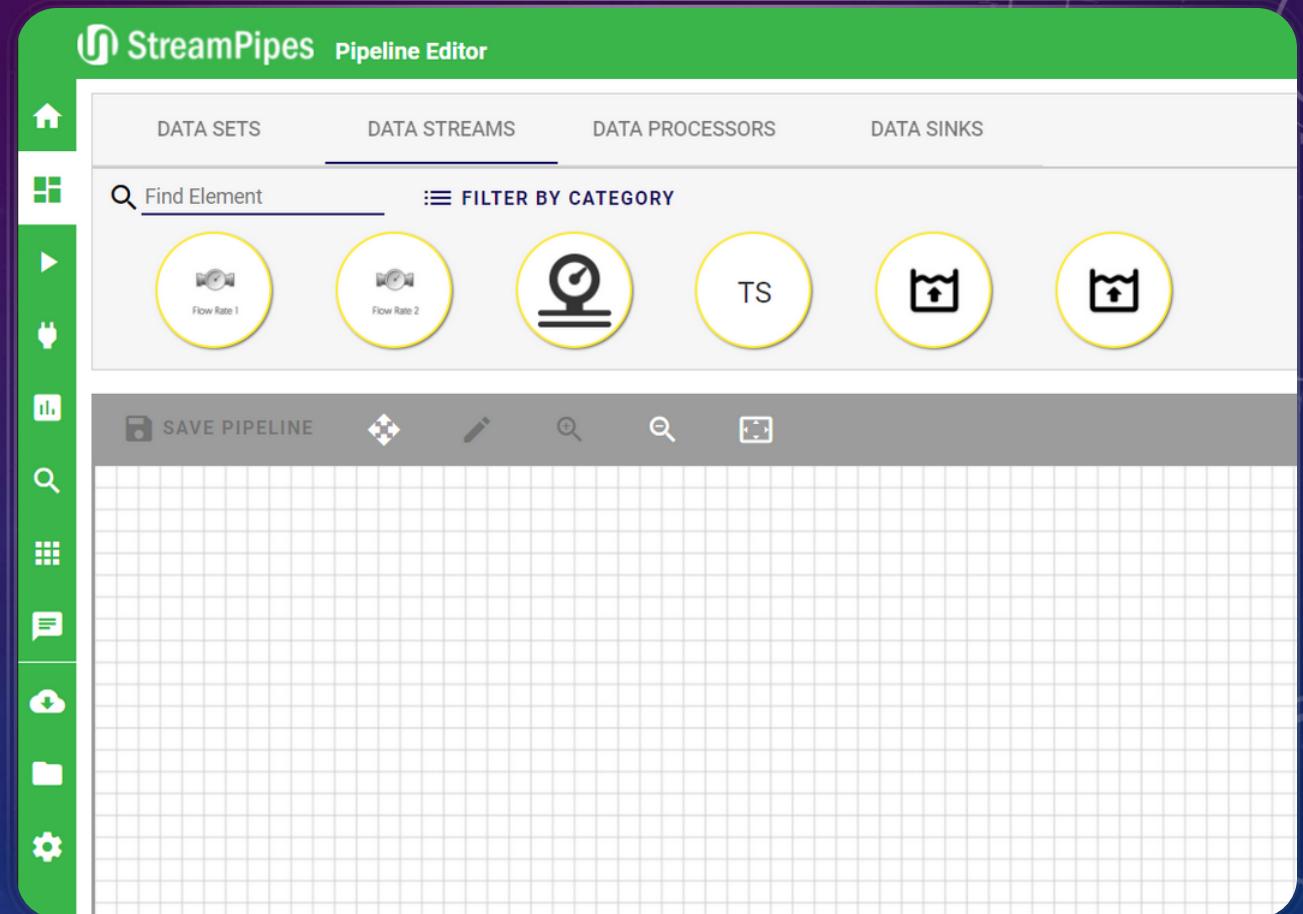


- Dodavanje ugnježdenog property-ja: modificuje strukturu eventa tako što kreira ugnježdenu strukturu
- Dodavanje statičke vrednosti: novo polje sa statičkom vrednošću
- Dodavanje timestamp-a: dodaje trenutni timestamp na svaki dolazni event
- Refresh: osvežava šemu
- Brisanje polja
- Oblast delovanja polja: za svako polje može se definisati jedinica, dimenzija ili heder, koji se kasnije koriste za konfiguriranje elemenata pipeline-a



PIPELINE EDITOR

- Moduo za rukovođenje pipeline-ima
- Kreiranje pipeline-a
- Konfiguracija
- Čuvanje izmena



PIPELINE EDITOR

Pipeline element

Flow Rate 2

Flow Rate 2
Simulates a water flow rate sensor

FIELDS **VALUES** **DOCUMENTATION**

Field Names

Field Name	Description	Runtime Name	Type
Timestamp	The current timestamp value	timestamp	Number
Sensor ID	The ID of the sensor	sensorid	Text
Mass Flow	Denotes the current mass flow in the sensor	mass_flow	Number
Volume Flow	Denotes the current volume flow	volume_flow	Number
Density	Denotes the current density of the fluid	density	Number
Fluid Temperature	Denotes the current temperature of the fluid	fluid_temperature	Number
Sensor Fault Flags	Any fault flags of the sensors	sensor_fault_flags	Number

CLOSE

Njegova konfiguracija

Customize Trend

Show documentation Show only recommended settings

Value to Observe
Specifies the value that should be monitored.
Value to Observe
mass_flow

Increase/Decrease
Specifies the type of operation the processor should perform.
 Decrease
 Increase

Percentage of Increase/Decrease
Specifies the increase in percent (e.g., 100 indicates an increase by 100 percent within the specified time window.)

Time Window Length (Seconds)
Specifies the size of the time window in seconds.
Time Window Length (Seconds) *
Specifies the size of the time window in seconds.

Select output

Flow Rate 1

Timestamps

SAVE **CANCEL**

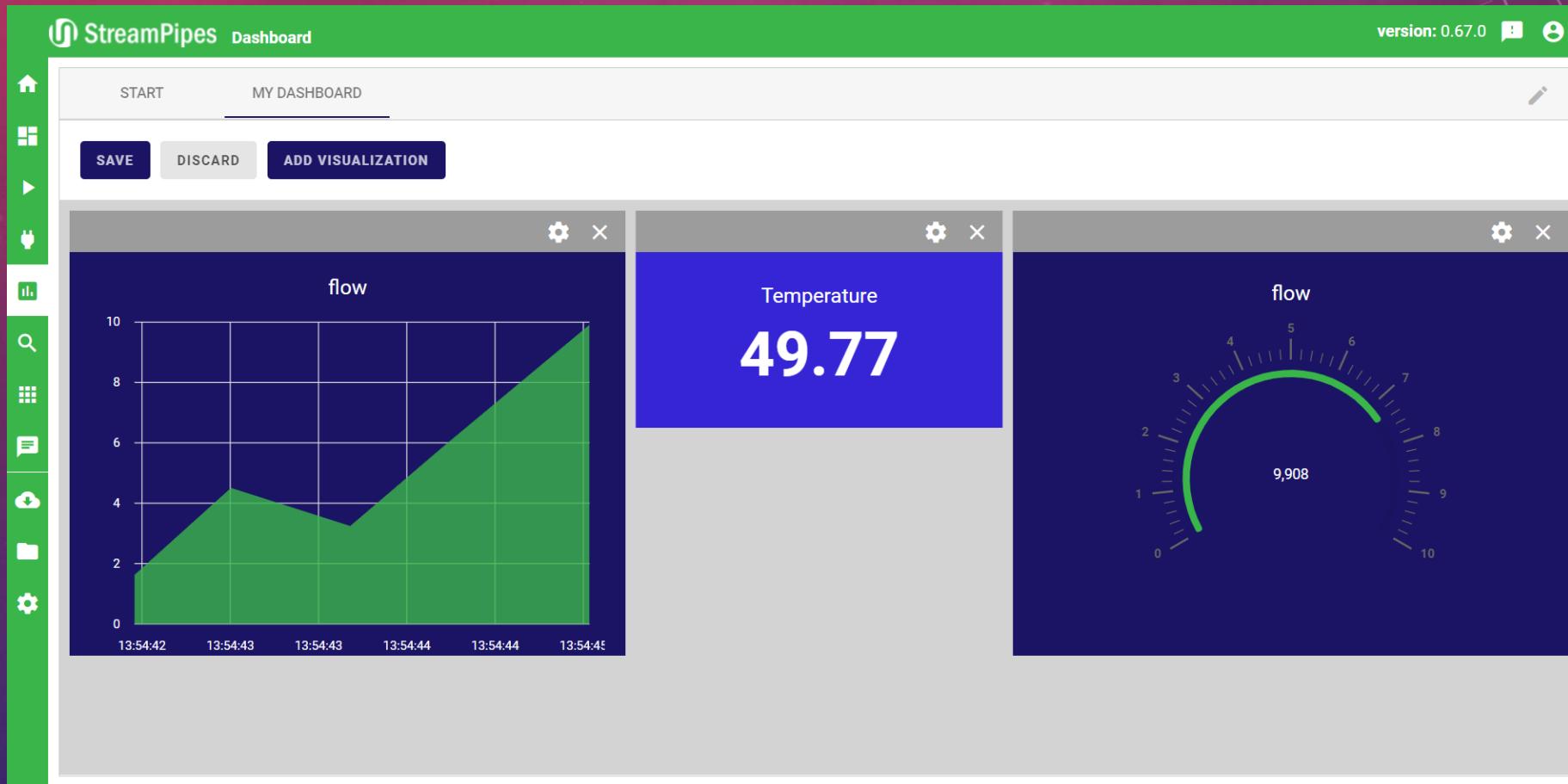
PIPELINE MENADŽER

- Daje pregled svih pipeline-a i omogućava osnovne akcije nad njima: pokretanje, zaustavljanje pipeline-a, prikazivanje detalja, modifikovanje, brisanje, kategorizaciju

The screenshot shows a web-based pipeline management interface. At the top, there are tabs for "MANAGE CATEGORIES" and "ALL PIPELINES". Below these are two buttons: "START ALL PIPELINES" and "STOP ALL PIPELINES", with "STOP ALL PIPELINES" currently highlighted. A section titled "My pipelines" displays a single entry: "flow" (status: "Start", created: "29.04.2021 11:28"). The interface includes standard table controls like search, edit, and delete icons. At the bottom right, there are pagination controls showing "Items per page: 20" and "1 - 1 of 1".

The screenshot shows a detailed view of a pipeline's status. A green header bar reads "Pipeline Status". Below it, a green success message says "✓ Pipeline flow successfully started" with a "HIDE DETAILS" link. Underneath, a list of active components is shown with green checkmarks and URLs: "Trend at URL http://pipeline-elements-all-jvm:8090/sepa/org.apache.streampipes.processors.siddhi.increase ✓", "Threshold Detector at URL http://pipeline-elements-all-jvm:8090/sepa/org.apache.streampipes.processors.filters.jvm.threshold ✓", "Dashboard Sink at URL http://pipeline-elements-all-jvm:8090/sec/org.apache.streampipes.sinks.internal.jvm.dashboard ✓", and "Data Lake at URL http://pipeline-elements-all-jvm:8090/sec/org.apache.streampipes.sinks.internal.jvm.datalake ✓".

PRIMER DASHBOARD-A



NOTIFIKACIJE

- Modulo za notifikacije kreira interna obaveštenja
- Može ga koristiti svaki pipeline koji uključuje data sink "Notification"
- Notifikaciona poruka može sadržati placeholder-e za vrednosti koje će biti prikazane u runtime-u

The screenshot shows a user interface titled "MY NOTIFICATIONS". On the left, there is a list item with a blue circular icon containing "MFN" and the text "My first notification Alarm". On the right, there is a detailed view of the same notification. The title "My first notification" and subtitle "Alarm" are at the top. Below this, two dark blue callout boxes show messages from May 2, 2021, at 7:48:15 PM and 7:48:16 PM. The first message reads: "This is an alarm of notification 1. The mass flow is high 6.6392". The second message reads: "This is an alarm of notification 1. The mass flow is high 0.187".

MY NOTIFICATIONS

MFN My first notification
Alarm

My first notification
Alarm

May 2, 2021 7:48:15 PM

This is an alarm of notification 1. The mass flow is high 6.6392

May 2, 2021 7:48:16 PM

This is an alarm of notification 1. The mass flow is high 0.187

STREAMPIPS CONFIGURATION

MAIN

```
public SpServiceDefinition provideServiceDefinition() {
    return SpServiceDefinitionBuilder.create( serviceGroup: "org.stefan_ana",
        serviceName: "SP elements",
        serviceDescription: "StreamPipes elements which are developed by Stefan&Ana", defaultPort: 8090)
        // DATA SOURCES
        .registerPipelineElement(new DataCollector())
        // DATA PROCESSORS
        .registerPipelineElement(new DataCleaner())
        .registerPipelineElement(new ComplexEventProcessing())

        .registerMessagingFormats(
            new JsonDataFormatFactory(),
            new CborDataFormatFactory(),
            new SmileDataFormatFactory(),
            new FstDataFormatFactory())
        .registerMessagingProtocols(
            new SpKafkaProtocolFactory(),
            new SpJmsProtocolFactory(),
            new SpMqttProtocolFactory())
        .addConfig(ConfigKeys.HOST, System.getenv( name: "SERVICE_HOST"), description: "Data processor host")
        .addConfig(ConfigKeys.PORT, System.getenv( name: "SERVICE_PORT"), description: "Data processor port")
        .addConfig(ConfigKeys.SERVICE_NAME, defaultValue: "Pipeline Elements", description: "Data processor service name")
        .addConfig(ConfigKeys.KAFKA_HOST, System.getenv( name: "KAFKA_HOST"), description: "Hostname for backend service for kafka")
        .addConfig(ConfigKeys.KAFKA_PORT, System.getenv( name: "KAFKA_PORT"), description: "Port for backend service for kafka")
        .addConfig(ConfigKeys.CONNECT_CONTAINER_WORKER_HOST, System.getenv( name: "SERVICE_HOST"), description: "The hostname of the connect container")
        .addConfig(ConfigKeys.CONNECT_CONTAINER_WORKER_PORT, System.getenv( name: "SERVICE_PORT"), description: "The port of the connect container")
        .addConfig(ConfigKeys.BACKEND_HOST, System.getenv( name: "SP_BACKEND_HOST") == null ? System.getenv( name: "SP_BACKEND_HOST")
            : "backend", description: "The host of the backend to register the worker")
        .addConfig(ConfigKeys.BACKEND_PORT, defaultValue: 8030, description: "The port of the backend to register the worker")
        .build();
}
```

ENVIRONMENT VARIABLES

```
# Those parameters are used by IntelliJ to set the default consul parameters for development  
# SP  
# Used when executed outside of Docker container (Overrides SERVICE_HOST and SERVICE_PORT)  
  
#SP_PORT=8030 # port of this extension running from IDE  
SP_DEBUG=true #overwrites both kafka settings from localhost to ENV  
CONSUL_LOCATION=localhost  
SP_HOST=192.168.0.32 # using IP to register to consul with this one, so it will be available to  
  
# Kafka  
KAFKA_HOST=localhost  
KAFKA_PORT=9094
```

CONSUL

localhost:8500/ui/dc1/services

dc1 Services Nodes Key/Value ACL Intentions Help Settings

Services 3 total

Service	Health Status	Service Type	Actions
consul	Healthy	Consul	View
core	Healthy	Core	View
ext	Healthy	Ext	View

CONSUL EXTENSIONS

dc1 Services Nodes Key/Value ACL Intentions Help Settings

< All Services ext

Instances Intentions Tags

Search Search Across Health Status Unhealthy to Healthy

Service	Address	Port	Tags
connect-adapter-iiot-gcykAi	consul-one	http://172.31.0.8:8001	spgroup:connect-adapter-iiot, sys:pe, adapter:org.apache.streampipes.connect.iiot.adapters.ros, adapter:org.apache.streampipes.connect.iiot.adapters.ros
connect-adapter-YAAED8	consul-one	http://172.31.0.11:8001	spgroup:connect-adapter, sys:pe, adapter:org.apache.streampipes.connect.adapters.simulator.randomdataset, adapter:org.apache.streampipes.connect.adapters.simulator.randomdataset
org-apache-streampipes-pe-all-flink-4OFVZp	consul-one	http://172.31.0.13:8090	spgroup:org-apache-streampipes-pe-all-flink, dprocessor:org.apache.streampipes.processors.pattern-detection.flink.sequence, dprocessor:org.apache.streampipes.processors.pattern-detection.flink.sequence
org-apache-streampipes-pe-all-jvm-2KhJmL	consul-one	http://172.31.0.10:8090	spgroup:org-apache-streampipes-pe-all-jvm, dprocessor:org.apache.streampipes.processors.transformation.jvm.booloperator.logical, dprocessor:org.apache.streampipes.processors.transformation.jvm.booloperator.logical
org-apache-streampipes-sources-simulator-watertank-7HZmFL	consul-one	http://172.31.0.7:8090	spgroup:org-apache-streampipes-sources-simulator-watertank, dstream:org.apache.streampipes.sources.simulator.flowrate2, dstream:org.apache.streampipes.sources.simulator.flowrate2
org.stefan_ana-HhxtYq	consul-one	http://192.168.0.32:8090	spgroup:org.stefan_ana, dprocessor:org.stefan_ana.pe.processor.cleaner.DataCleaner, dstream:org.stefan_ana.pe.source.data.collector

STREAMPIPES

INDUSTRY MONITORING AND ALERTING

PIPELINES

Home > Pipelines

+ NEW PIPELINE

▶ START ALL PIPELINES

■ STOP ALL PIPELINES

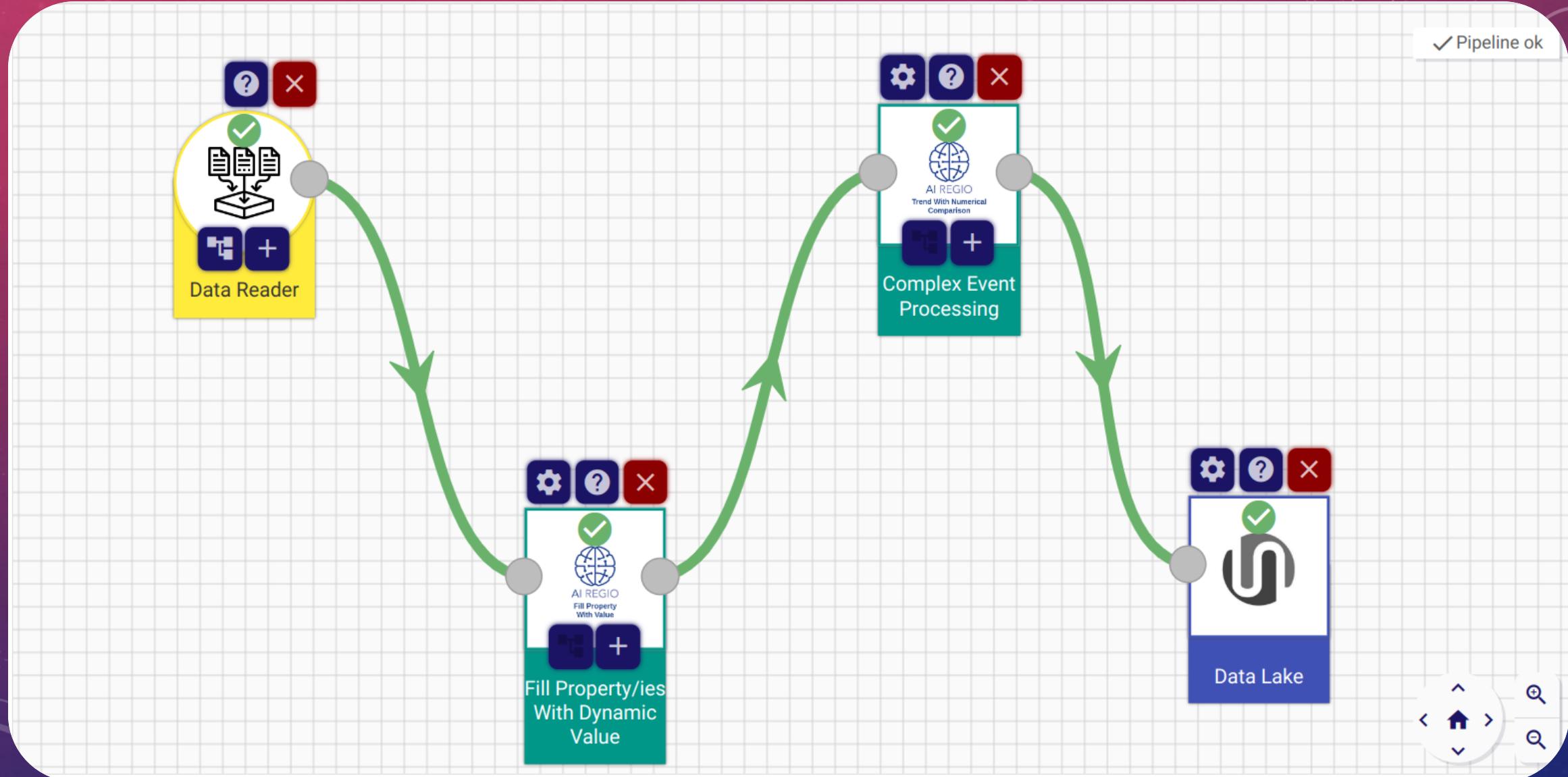


My pipelines

Status	Start	Name	Created	Action
	▶	Monitoring and alerting	10.01.2023 21:31	

Items per page: 20 ▾ 1 - 1 of 1 < >

PIPELINE



PROCESSOR LIFECYCLE

```
public class WarningChecker extends StreamPipesDataProcessor {  
    @Override  
    public DataProcessorDescription declareModel() { return null; }  
  
    @Override  
    public void onInvocation(ProcessorParams parameters, SpOutputCollector spOutputCollector,  
                            EventProcessorRuntimeContext runtimeContext) throws SpRuntimeException {  
    }  
  
    @Override  
    public void onEvent(Event event, SpOutputCollector collector) throws SpRuntimeException {  
    }  
  
    1 usage  
    @Override  
    public void onDetach() throws SpRuntimeException {  
    }
```

DECLARE MODEL

```
override
public DataProcessorDescription declareModel() {
    return ProcessingElementBuilder
        .create(ID)
        .withAssets(Assets.DOCUMENTATION, Assets.ICON)
        .withLocales(Locales.EN)
        .category(DataProcessorType.AGGREGATE)
        .requiredStream(StreamRequirementsBuilder
            .create()
            .requiredPropertyWithUnaryMapping(
                EpRequirements.numberReq(),
                Labels.withId(PROPERTY_KEY),
                PropertyScope.NONE)
            .build())
        .requiredIntegerParameter(Labels.withId(WINDOW_LENGTH_KEY), defaultValue: 24)
        .requiredIntegerParameter(Labels.withId(WINDOW_LENGTH_KEY_CRITICAL), defaultValue: 6)
        .requiredSingleValueSelection(
            Labels.withId(OPTION_OPERATOR_KEY),
            Options.from(...optionLabel: "<", "<=", ">", ">=", "==" , "!="))
        .requiredIntegerParameter((Labels.withId(COMPARISON_VALUE_KEY)), min: 1, max: 600, step: 1)
        .requiredTextParameter(Labels.withId(OUTPUT_NAME_KEY), multiLine: false, placeholdersSupported: false)
        .outputStrategy(OutputStrategies.customTransformation())
        .build();
}
```

DECLARE MODEL (UI CONFIG)

Customize Complex Event Processing

Window length

Length of the trend being detected

Window length *

1000

Window of critical time (in minutes)

Last n minutes time for tracking critical warnings

Window of critical time (in minutes) *

3

Comparison operator

Comparison operator that is used in trend detection

- <
- <=
- >
- >=
- ==
- !=

Comparison value



Output property name

Name of the output property which values are either "True" or "False"

Output property name *

newprop

ON INVOCATION

```
@Override  
public void onInvocation(ProcessorParams parameters, SpOutputCollector spOutputCollector, EventProcessorRuntimeContext run  
    propertyName = parameters.extractor().mappingPropertyValue(PROPERTY_KEY);  
    windowLength = parameters.extractor().singleValueParameter(WINDOW_LENGTH_KEY, Integer.class);  
    operator = parameters.extractor().selectedSingleValue(OPTIONAL_OPERATOR_KEY, String.class);  
    comparisonValue = parameters.extractor().singleValueParameter(COMPARISON_VALUE_KEY, Integer.class);  
    outputPropertyName = parameters.extractor().singleValueParameter(OUTPUT_NAME_KEY, String.class);  
    windowLengthCritical = parameters.extractor().singleValueParameter(WINDOW_LENGTH_KEY_CRITICAL, Integer.class);  
  
    siddhiManager = new SiddhiManager();  
    siddhiAppRuntime1 = siddhiManager.createSiddhiAppRuntime(createQuery1());  
    siddhiAppRuntime2 = siddhiManager.createSiddhiAppRuntime(createQuery2());
```

ON EVENT AND ON DETACH

```
override
public void onEvent(Event event, SpOutputCollector collector) throws SpRuntimeException {
    ArrayList<HashMap> res = (ArrayList<HashMap>) event.getRaw().get("s0::data_array");
    for(int i =0; i< res.size(); i++) {
        try {
            inputHandler1.send(new Object[]{res.get(i).get("temperature"), res.get(i).get("timestamp")});
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    }
    logger.info("Siddhi: Data sent to first stream: (" + Date.from(Instant.now()) + ")");
}
```

1 usage

```
@Override
public void onDetach() throws SpRuntimeException {
    siddhiAppRuntime1.shutdown();
    siddhiAppRuntime2.shutdown();
    siddhiManager.shutdown();
```

COMPLEX EVENT PROCESSING (CEP)

- Definicija: CEP je metoda analiziranja i reagovanja na složene, high-volume stream-ove podataka u realnom vremenu.
- Ključni koncepti: CEP uključuje prepoznavanje obrazaca i veza u stream-ovima podataka i korišćenje tih informacija za pokretanje akcija ili generisanje upozorenja.
- Primene: CEP se koristi u različitim industrijama, kao što su finansije, telekomunikacije, transport i druge. Primeri uključuju detekciju prevara, praćenje performansi mreže i praćenje nivoa inventara u realnom vremenu.
- Prednosti: CEP omogućava organizacijama da brzo prepoznaju i reaguju na važne događaje, što može dovesti do povećane efikasnosti, boljeg donošenja odluka i smanjenja rizika.

SIDDHI ENGINE

- Definicija: Siddhi je open-source CEP engine napisan u jeziku Java i može se pokretati na različitim platformama.
- Funkcionalnosti: Siddhi podržava širok spektar operacija obrade streamova podataka, uključujući filtriranje, praćenje obrazaca i agregaciju. Takođe pruža podršku za SQL-like jezik upita i proširivu arhitekturu plugin-a.
- Primene: Siddhi engine se koristi u mnogim aplikacijama u realnom vremenu, kao što su IoT, detekcija prevara, detekcija anomalija i druge.

QUERY EXAMPLE

```
private String createQuery1() {  
    String defineStream = "define stream WarningStream(value double, time long); ";  
  
    String from = "from " +  
        "WarningStream" + "#window.length(" + windowLength + ")\\n";  
    String select = "select ifThenElse(value" + operator + comparisonValue + ", true, false) as warning," +  
        "value as average, time as timestamp\\n";  
    String insertInto = "insert into WarningAlarmStream;";  
    return defineStream + from + select + insertInto;  
}  
}  
} // class
```

ASP.NET CORE WEB API MIKROSERVIS

ASP.NET CORE

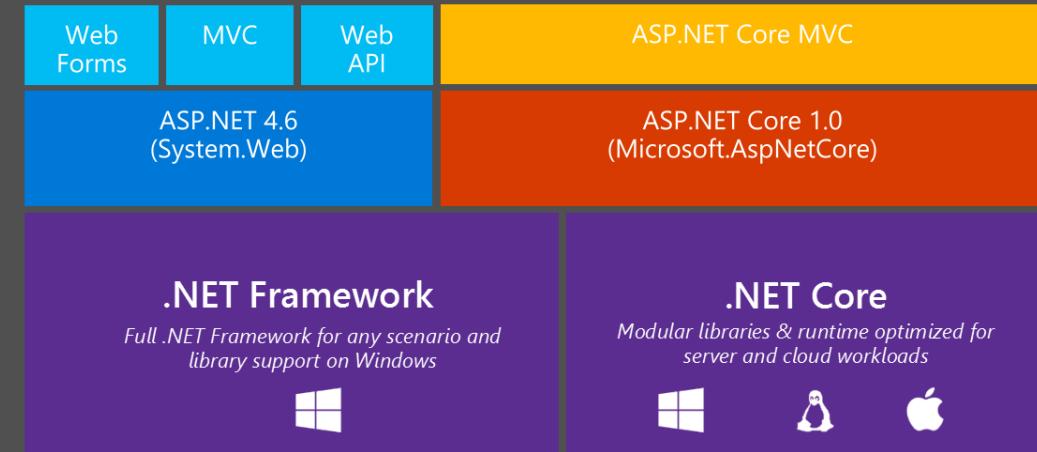
- Cross-platform i open-source okruženje
- Predstavlja redizajnirani ASP.NET 4.x
- Promene u arhitekturi koje su uvele veću modularnost okruženja
- Za kreiranje web aplikacija i servisa, IoT aplikacija, bekend servisa
- Podržava cloud deploy



KARAKTERISTIKE ASP.NET CORE-A

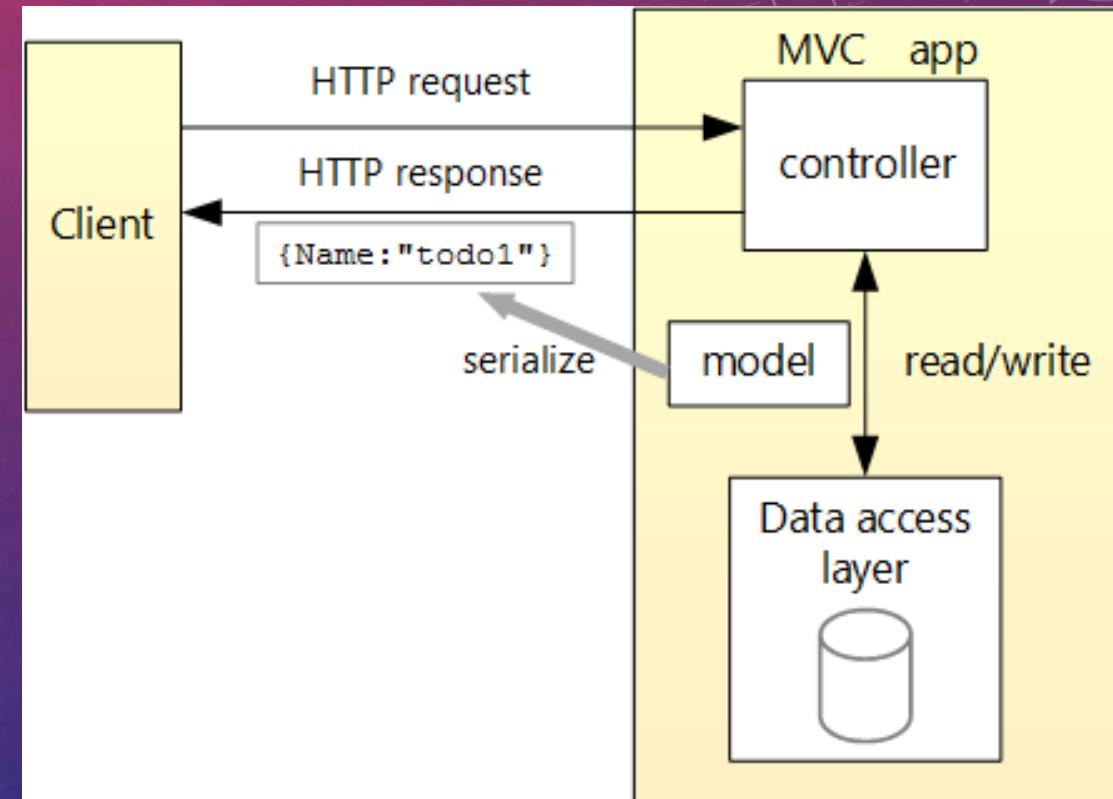
- Arhitektурно организовано за тестирање
- Razor странице олакшавају кодирање страница
- Blazor омогућава коришћење C#-а у броузер-у упоредо са JavaScript-ом
- Mogućnost развоја и покретања на Windows-у, macOS-у и Linux-у

ASP.NET Core in a Nutshell



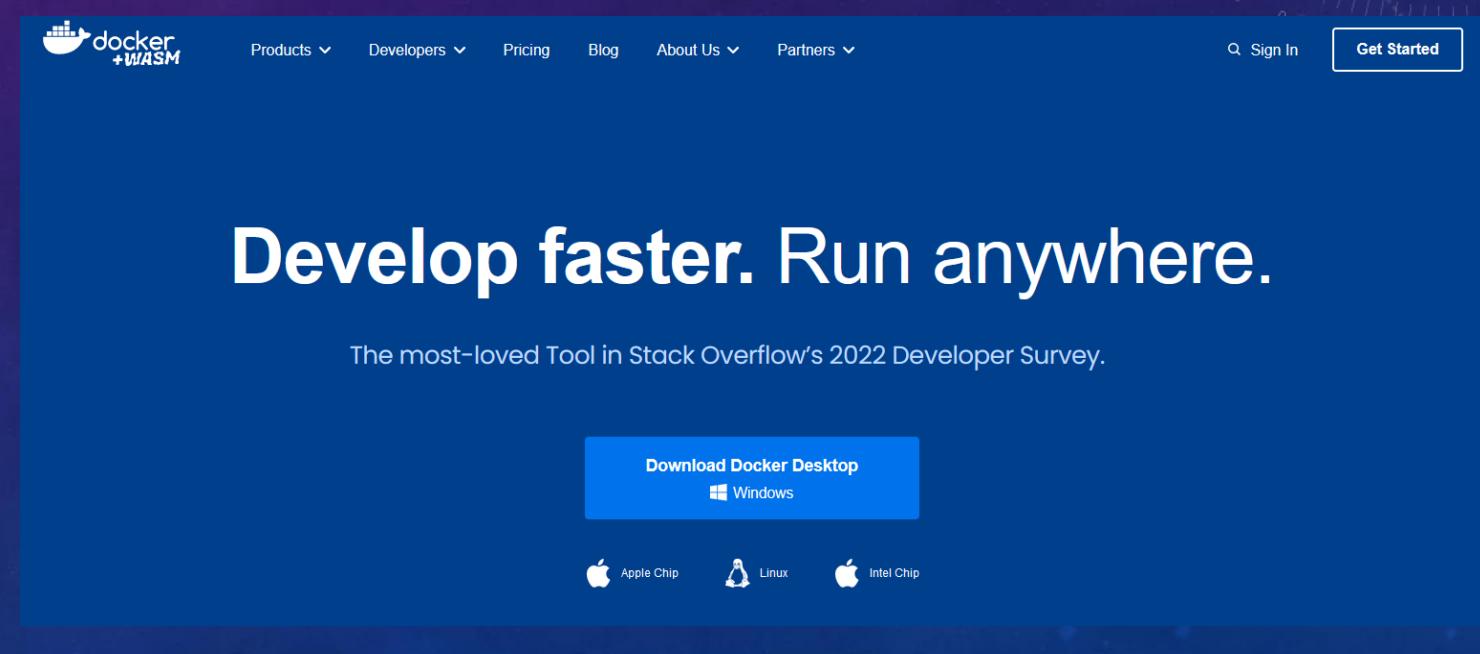
ASP.NET CORE WEB API

- Za potrebe projekta, kreirana je web API aplikacija
- Struktura kao na slici
- Klijent – Data procesor u Streampipes
- Šalje zahteve API-ju (uglavnom POST)
- API prima HTTP zahtev, deserijalizuje podatke
- Upisuje ih u bazu (InfluxDB 2.0)



POKRETANJE APLIKACIJE

- Aplikacija je kontejnerizovana i pokreće se u Docker-u
- Potrebno je instalirati Docker Engine
- To je moguće uraditi na više načina
- Najlakše je instalirati Docker Desktop



The screenshot shows the official Docker website. At the top, there's a navigation bar with links for Products, Developers, Pricing, Blog, About Us, and Partners. On the right side of the header are search, sign-in, and "Get Started" buttons. The main visual features a large blue whale logo carrying shipping containers, with the word "docker" written below it in a stylized green font. Below the logo, a prominent headline reads "Develop faster. Run anywhere." followed by a subtext stating "The most-loved Tool in Stack Overflow's 2022 Developer Survey." A blue call-to-action button says "Download Docker Desktop" with a "Windows" option underneath. At the bottom, there are icons for Apple Chip, Linux, and Intel Chip.

POKRETANJE APLIKACIJE

- Na osnovu dockerfile-a kreira se docker image
- Image fajl se pokreće zajedno sa ostalim image fajlovima za InfluxDB i Grafanu

```
1 # Stage 1
2 FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS build
3 WORKDIR /build
4 COPY . .
5 RUN dotnet restore
6 RUN dotnet publish -c Release -o /app
7
8 # Stage 2
9 FROM mcr.microsoft.com/dotnet/core/aspnet:3.1 AS final
10 WORKDIR /app
11 COPY --from=build /app .
12 EXPOSE 80
13 ENTRYPOINT ["dotnet", "Streampipes-API.dll"]
```

POKRETANJE APLIKACIJE

- docker-compose-yml se pokreće komandom "docker compose up" i podižu se sva tri kontejnera
- Obratiti pažnju na portove
- Kreirani su i volumes, tako da se podaci i izmene pamte i nakon brisanja kontejnera

```
1  version: "3.9"
2  services:
3    stremppipes:
4      build: .
5      ports:
6        - "8080:80"
7    influxdb:
8      image: influxdb:latest
9      container_name: influxdb2
10     volumes:
11       - influx-data:/var/lib/influxdb2:rw
12     # env_file:
13     #   - .env
14     #   entrypoint: ["./entrypoint.sh"]
15     ports:
16       - 8085:8086
17     restart: unless-stopped
18   grafana:
19     image: grafana/grafana:9.1.1
20     ports:
21       - 3000:3000
22     restart: unless-stopped
23     volumes:
24       - ./grafana/provisioning/datasources:/etc/grafana/provisioning/datasources
25       - grafana-data:/var/lib/grafana
26     volumes:
27       grafana-data:
28       influx-data:
```

BAZA

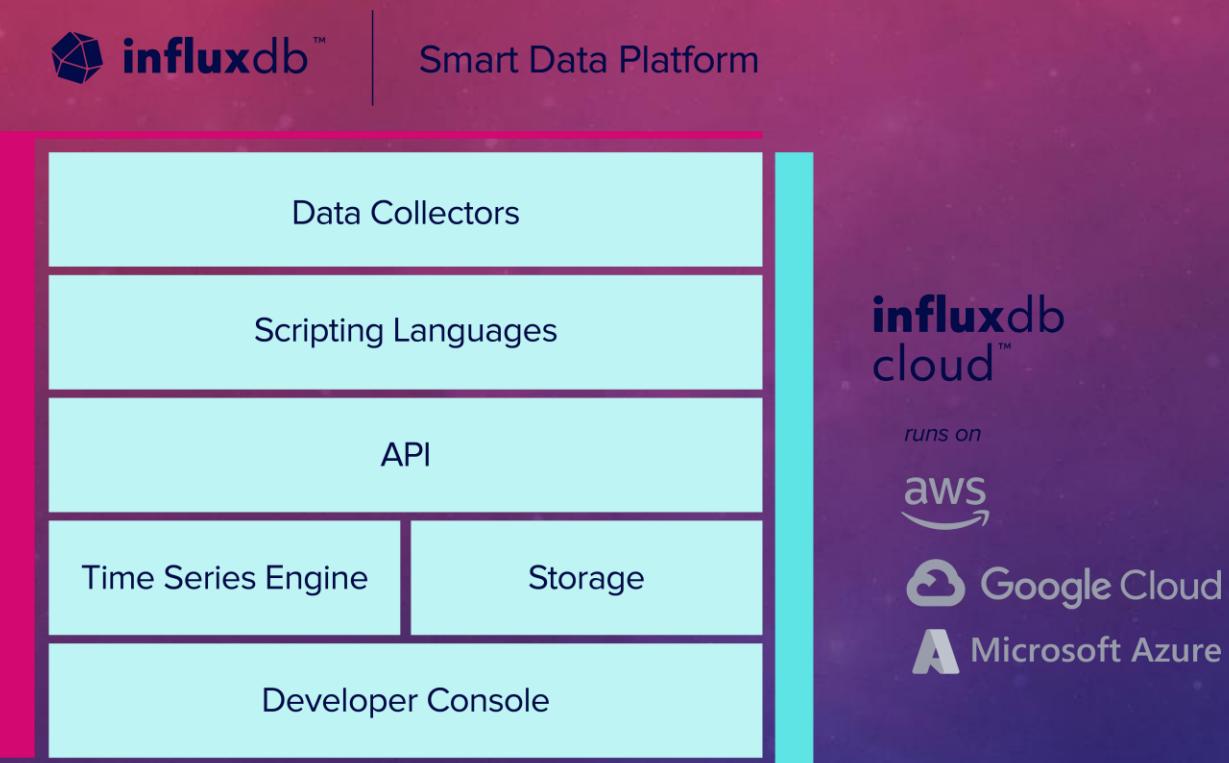
- Kao baza korišćen je InfluxDB 2.0
- To je platforma bazirana na vremenskim serijama podataka
- Poseduje moćan API za aplikacije u realnom vremenu
- Engine za vremenske serije sa visokim performansama



A screenshot of the InfluxDB user interface. On the left is a sidebar with icons for Home, User, Data, Explore, Boards, Tasks, Alerts, and Settings. A green star icon labeled "Upgrade Now" is at the bottom. The main area is titled "Getting Started" and contains three numbered steps: 1. "Load your data" (illustrated with a bucket icon), 2. "Build a dashboard" (illustrated with a chart icon), and 3. "Set up alerting" (illustrated with a hexagonal grid icon). Below these steps is a section titled "Some Handy Guides and Tutorials" with links to "Get Started with Flux", "Explore Metrics", "Build a Dashboard", and "Write a Task". On the right side, there's a "Recent Dashboards" search bar, a "System" section, "Useful Links" (Documentation, Community Forum, Feature Requests, Report a bug), and a "Version (d9dec89)" status bar.

INFLUXDB

influxdb
connected edge



- Bilduj jednom, a deploy na više okruženja i produkata
- Mogućnost pokretanja na cloud-u, edge-u, ili u lokalnom okruženju
- U ovom projektu, influxdb, kao i ostale komponente, je kontejnerizovan i pokreće se u docker-u

INFLUXDB DASHBOARD

Data Explorer

Graph CUSTOMIZE Local SAVE AS

Create a query. Have fun!

Query 1 +

View Raw Data CSV Past 15m SCRIPT EDITOR SUBMIT

FROM
Search for a bucket
test-bucket
_monitoring
_tasks
+ Create Bucket

Filter
_measurement 1
Search _measurement tag va
monitoringData

Filter
cleanData
Search keys
_field
Search _field tag values
cleanData
dirtyData
security

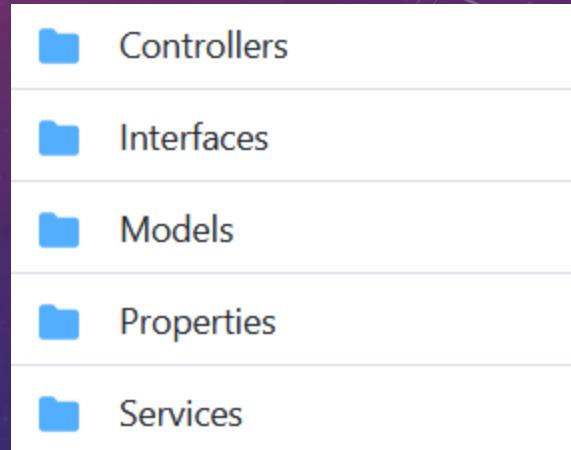
Filter
_field
Search _field tag values
outlier
pressure
temperature
value

WINDOW PERIOD
CUSTOM AUTO
auto (10s)
Fill missing values ?

AGGREGATE FUNCTION
CUSTOM AUTO
mean
median

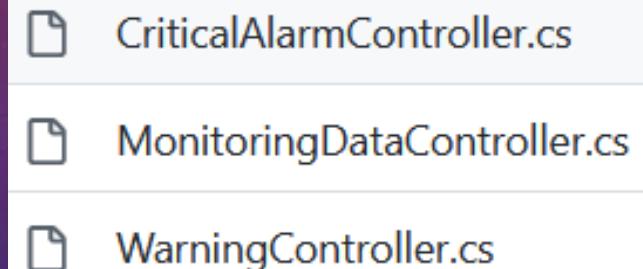
WEB API

- Struktura aplikacije prikazana na slici
- Kontroleri osluškuju i prihvataju zahteve
- Modeli oblikuju podatke tako da se lako pribavljuju iz zahteva i upisuju u bazu
- Servisi predstavljaju sponu između aplikacije i baze i obavljaju upis i čitanje



KONTROLERI

- Aplikacija ima tri osnovne funkcije:
upis vremenskih, serijskih podataka u bazu,
upis upozorenja u bazu i upis kritičnih alarma u bazu
- Za ove svrhe kreirana su tri kontrolera,
`MonitoringDataController`, `WarningController`,
`CriticalAlarmController`
- Svaki od njih poziva odgovarajuće metode InfluxDB servisa
za upis u bazu
- Servis je kontrolerima dostupan zahvaljujući ugrađenom
Dependency Injection-u



```
CriticalAlarmController.cs
MonitoringDataController.cs
WarningController.cs
```

MonitoringDataController

```
[HttpPost]
0 references
public ActionResult<MonitoringData> Create(MonitoringDataRequest request)
{
    foreach (MonitoringData data in request.MonitoringDataList)
    {
        try
        {
            _influxService.Write(write =>
            {
                var point = PointData.Measurement("monitoringData")
                    .Tag(request.IsClean ? "cleanData" : "dirtyData", request.IsClean ? "cleanData" : "dirtyData")
                    .Field("pressure", data.Pressure)
                    .Field("temperature", data.Temperature)
                    .Timestamp(data.Timestamp, WritePrecision.Ms);

                write.WritePoint(point, "test-bucket", "organization");
            });
        }
        catch (Exception e)
        {
            Console.WriteLine(e.ToString());
        }
    }

    return Ok();
}
```

Warning/Alarm Controller

```
[HttpPost]
0 references
public ActionResult<MonitoringData> Create(Warning data)
{
    _influxService.Write(write =>
    {
        var point = PointData.Measurement("monitoringData")
            .Tag("security", "warning")
            .Field("value", data.Value)
            .Field("outlier", data.Outlier)
            .Timestamp(data.Timestamp, WritePrecision.Ms);

        write.WritePoint(point, "test-bucket", "organization");
    });

    return Ok();
}
```

```
3 references
public void Write(Action<WriteApi> action)
{
    using var client = InfluxDBClientFactory.Create("http://host.docker.internal:8085", _token);
    using var write = client.GetWriteApi();
    Console.WriteLine(client.HealthAsync().Result);
    action(write);
}

1 reference
public async Task<T> QueryAsync<T>(Func<QueryApi, Task<T>> action)
{
    using var client = InfluxDBClientFactory.Create("http://host.docker.internal:8085", _token);
    var query = client.GetQueryApi();
    return await action(query);
}
```

INFLUXDB SERVIS

SERVIS KLASA PRUŽA FUNKCIONALNOSTI UPISA I ČITANJA IZ INFLUX BAZE

MODELI

- Modeli omogućavaju normalan tok aplikacije i razmenu podataka
- Warning i CriticalAlarm modeli sadrže podatke o vrednosti koja je izazvala alarm, kao i o vremenu kada se desio
- MonitoringData predstavlja podatke koji se prate; u ovom konkretnom slučaju to su pritisak i temperatura, i njihove vremenske markice
- Request i Response modeli su tu da omoguće komunikaciju sa Strempipes delom aplikacije

 CriticalAlarm.cs
 MonitoringData.cs
 MonitoringDataRequest.cs
 MonitoringDataResponse.cs
 StrempipesDatabaseSettings.cs
 Warning.cs

MONITORING DATA | WARNING | REQUEST

```
14 references
public class MonitoringData
{
    [BsonId]
    [BsonRepresentation(BsonType.ObjectId)]
    5 references
    public string Id { get; set; }

    [JsonPropertyName("timestamp")]
    [Column(IsTimestamp = true)]

    1 reference
    public long Timestamp { get; set; }

    [JsonPropertyName("temperature")]
    [Column("temperature")]
    1 reference
    public double? Temperature { get; set; }

    [JsonPropertyName("pressure")]
    [Column("pressure")]
    1 reference
    public double? Pressure { get; set; }
}
```

```
1 reference
public class Warning
{
    [JsonPropertyName("warning")]
    1 reference
    public int Value { get; set; }

    [JsonPropertyName("timestamp")]
    1 reference
    public long Timestamp { get; set; }

    [JsonPropertyName("value")]
    1 reference
    public double Outlier { get; set; }
}
```

```
1 reference
public class MonitoringDataRequest
{
    [JsonPropertyName("data_array")]
    1 reference
    public List<MonitoringData> MonitoringDataList { get; set; }

    [JsonPropertyName("timestamp")]
    0 references
    public long Timestamp { get; set; }

    [JsonPropertyName("isClean")]
    2 references
    public bool IsClean { get; set; }
}
```

GRAFANA

- Grafana je alat za vizuelizaciju podataka
- Postoje Dashboards, u okviru kojih može biti više panela
- Jeden panel prikazuje podatke na jedan, izabrani način



Grafana

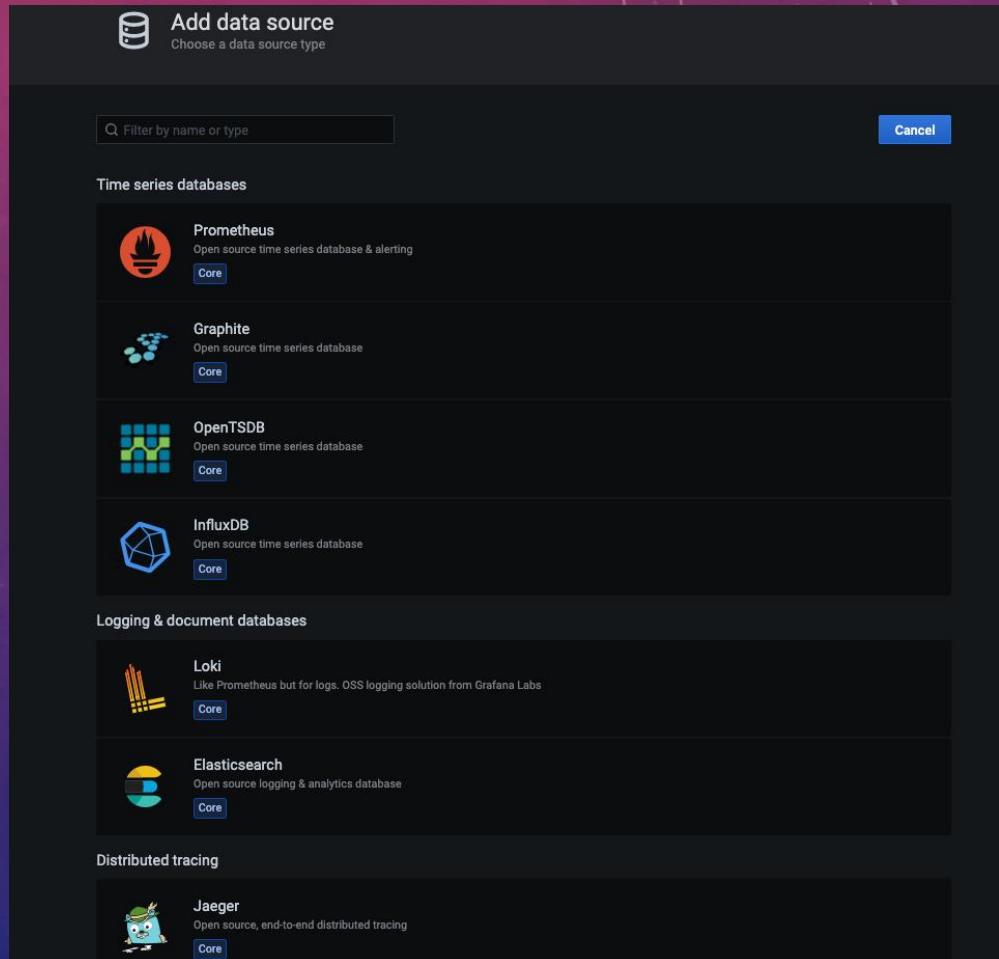
GRAFANA

- Postoji mnogo različitih tipova prikaza podataka
- Graf, tabela, stubići, linijski prikaz, brzinometar, i mnogi drugi



GRAFANA

- Kao datasource u Grafani izabran je InfluxDB i kreiran je konektor
- Ovime su Grafana i InfluxDB povezani i Grafana redovno uzima podatke iz Influx-a i prikazuje ih
- Kreirani odgovarajući dashboard i paneli
- Za ispravno pribavljanje podataka, potrebno je napisati Flux upit koji filtrira i preuzima podatke iz baze



GRAFANA PODEŠAVANJE KONEKTORA

Name ⓘ InfluxDB Default

Query Language
Flux

 Support for Flux in Grafana is currently in beta
Please report any issues to:
<https://github.com/grafana/grafana/issues>

HTTP

URL	<input type="text" value="http://influxdb:8086"/>
Access	Server (default) <input type="button" value="Help >"/>
Allowed cookies	New tag (enter key to add)
Timeout	<input type="text" value="Timeout in seconds"/>

Auth

Basic auth <input checked="" type="checkbox"/>	With Credentials <input checked="" type="checkbox"/>
TLS Client Auth <input type="checkbox"/>	With CA Cert <input checked="" type="checkbox"/>
Skip TLS Verify <input type="checkbox"/>	
Forward OAuth Identity <input type="checkbox"/>	

Basic Auth Details

User	stefan
Password	configured

Custom HTTP Headers

+ Add header

InfluxDB Details

Organization	organization
Token	configured
Default Bucket	test-bucket
Min time interval	10s
Max series	1000

GRAFANA PANEL CONFIG

← Monitoring and alerting / Edit Panel ⚙️ Discard Save Apply

Table view Fill Actual Last 5 minutes ⟳

Time series

After cleaning

temperature cleanData

Query 1 Transform 0 Alert 0

Data source InfluxDB ⌚ Query options MD = auto = 941 Interval = 200ms Query inspector

A (InfluxDB)

```
1 from(bucket: "test-bucket")
2   |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3   |> filter(fn: (r) => r["_measurement"] == "monitoringData")
4   |> filter(fn: (r) => r["cleanData"] == "cleanData")
5   |> filter(fn: (r) => r["_field"] == "temperature")
6   |> sort()
7   |> yield(name: "sort")
```

Search options All Overrides

Panel options

Title After cleaning

Description

Transparent background

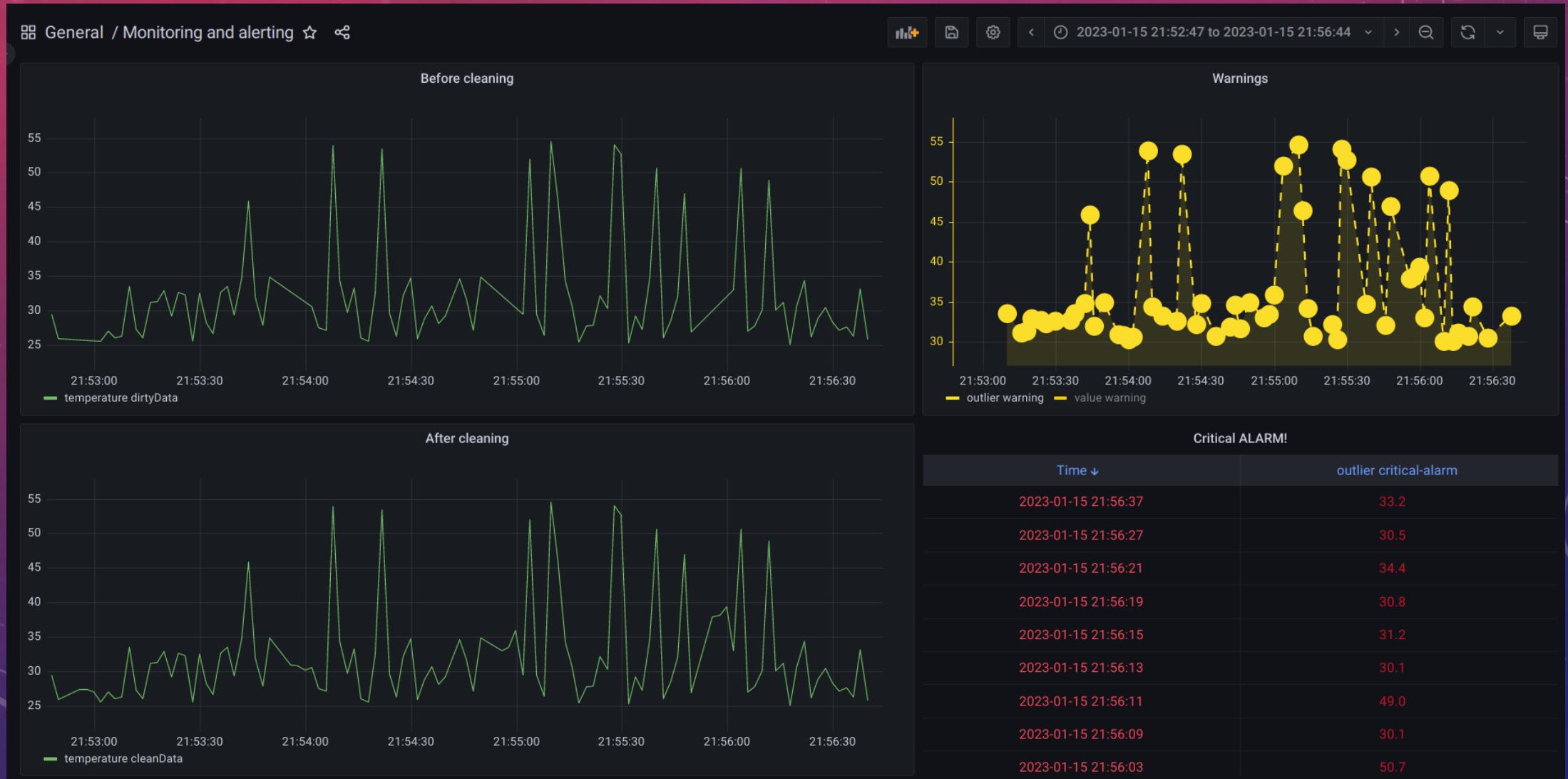
Panel links Repeat options

Tooltip

Tooltip mode Single All Hidden

Legend Visibility

GRAFANA DASHBOARD



HVALA NA PAŽNJI!