Introduction to AsyncAPI for Apache Kafka

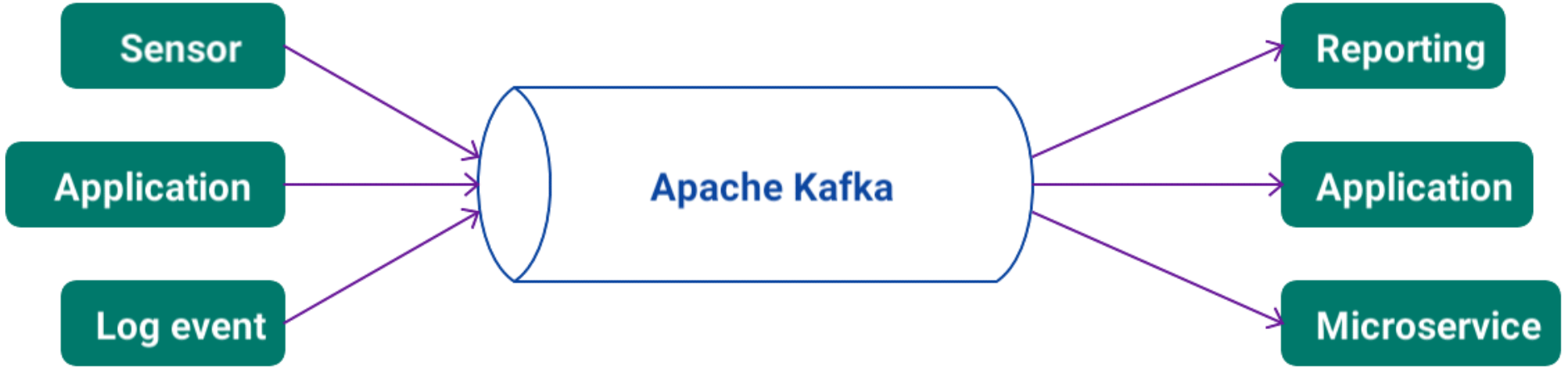
Lorna Mitchell, Aiven

Meet Apache Kafka

"Apache Kafka is an open-source distributed event streaming platform" - https://kafka.apache.org

- Designed for data streaming
- Real-time data for finance and industry
- Very scalable to handle large datasets
- Distributed log platform, each channel is a "topic" and has "partitions"

Kafka Architectures



Producers send data, Consumers receive it.

AsyncAPI

Open standard for describing event-driven and data streaming systems.

AsyncAPI

- Website: https://asyncapi.com
- Open YAML/JSON standard describes message-based systems
- Supports Apache Kafka, MQTT, AMQP, WS ....
- Tools to generate docs, code and more

Why use AsyncAPI?

Describe your event-driven systems in a clear and reusable way

- Design-first and clear contracts between systems
- Clearly track changes in a text-based description
- Seamless integrations with other systems
- Enclose existing payload descriptions (CloudEvents, Avro) within AsyncAPI

Generated Docs

SUB `door-sensor`

Door sensors (external and internal)

Open/closed state information from the doors.

Accepts the following message:

Door Sensor Reading door-sensor-data

Door sensor data

Payload **Object**

location **String**

state

Enum: "open" "closed"

Additional properties are allowed.

sensor

Examples

door-sensor-data

Payload

Example #1

```
{
  "location": "Car park",
  "state": "open"
}
```

Example #2

```
{
  "location": "Roof-level fire exit",
  "state": "closed"
}
```

AsyncAPI Descriptions

AsyncAPI Structure

Top-level elements:

- `asyncapi` and `id`
- `info`
- `servers`
- `channels`
- `tags`
- `components`

Info Section

Valuable metadata is held in `info`.

```yaml
info:
  title: Thingum Industries Sensors
  description: Keeping the factory and all the machines running nicely
  version: 1.0.0
  contact:
    name: Lorna
    email: lornajane@aiven.io
    url: https://github.com/aiven/thingum-industries
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
```

Channels Section

Main operations are described here

```yaml
channels:
  door-sensor:
    description: Door sensors (external and internal)
    subscribe:
      operationId: DoorSensor
      description: Open/closed state information from the doors.
      tags:
        - name: sensor
      bindings:
        kafka:
          clientId:
            type: string
      message:
        $ref: '#/components/messages/DoorData'
```

`$ref` Reusable Content

Refer to content in the `components` section

```
message:
  $ref: '#/components/messages/DoorData'
```

Useful for reuse and readability.

We can also refer to other files:

```
message:
  $ref: 'doors-publish.yaml#/components/messages/DoorData'
```

Components Section

A collection of reusable components

```yaml
components:
  messages:
    DoorData:
      name: door-sensor-data
      title: Door Sensor Reading
      description: Door sensor data
      payload:
        type: object
        properties:
          location:
            type: string
          state:
            enum: ["open", "closed"]
```

AsyncAPI Tools

AsyncAPI Playground

Playground: https://playground.asyncapi.io/

AsyncAPI 2.0.0 playgroun

https://playground.asyncapi.io 133%

AsyncAPI playground

Type the URL of the AsyncAPI document to import... Import document

HTML Markdown

```
1 asyncapi: 2.1.0
2 info:
3   title: Account Service
4   version: 1.0.0
5   description: This service is in charge of
  processing user signups
6 channels:
7   user/signedup:
8     subscribe:
9       message:
10        $ref: '#/components/messages/UserSignedUp'
11 components:
12   messages:
13     UserSignedUp:
14       payload:
15         type: object
16         properties:
17           displayName:
18             type: string
19             description: Name of the user
20           email:
```

Account Service 1.0.0

This service is in charge of processing user signups

Operations

SUB `user/signedup`

Accepts the following message:

UserSignedUp

Payload > Object

Use Your Editor

doors-publish.yaml - thingum-industries - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

THINGUM-INDUSTRIES

OUTLINE

- asyncapi 2.0.0
- id urn:com.github.a...
- {} info
  - title Thingum Ind...
  - description Keep...
  - version 1.0.0
  - {} contact
    - name Lorna
    - email lornajane...
    - url https://githu...
  - {} license
    - name Apache 2.0
    - url http://www.a...
  - {} servers
    - {} production

TIMELINE

CODETOUR

doors-publish.yaml > {} servers > {} production > [ ] security

```yaml
  development:
    url: localhost:9092
    protocol: kafka
    protocolVersion: '1.0.0'

channels:
  door-sensor:
    description: Door sensors (external and internal)
    publish:
      operationId: DoorSensor
      description: Open/closed state information from the doors.
      tags:
        - name: sensor
      bindings:
        kafka:
          clientId:
            type: string
      message:
        $ref: '#/components/messages/DoorData'

components:
  messages:
    DoorData:
```

main* | Jira: Lorna Mitchell: | No active issue | Go 1.15.6 | 8 1 | Live Share | Git Graph | -- NORMAL -- | Go Update Available

Generate Documentation

Generator: https://www.asyncapi.com/generator

SUB `door-sensor`

Door sensors (external and internal)

Open/closed state information from the doors.

Accepts the following message:

Door Sensor Reading door-sensor-data

Door sensor data

Payload **Object**

location **String**

state

Enum: "open" "closed"

Additional properties are allowed.

sensor

Examples

door-sensor-data

Payload

Example #1

```
{
  "location": "Car park",
  "state": "open"
}
```

Example #2

```
{
  "location": "Roof-level fire exit",
  "state": "closed"
}
```

Generate Code

The same generator can do more than HTML output.

Generate code in Java, Python or NodeJS - or add your own template.

```
ag asyncapi.yaml @asyncapi/nodejs-template \
    -o nodejs -p server=production

npm install

NODE_ENV=production npm start
```

Tools Landscape

- Playground: https://playground.asyncapi.io
- Generator: https://www.asyncapi.com/generator
- Microcks: https://microcks.io/
- Spectral: https://stoplight.io/open-source/spectral/
- VSCode: https://github.com/asyncapi/vs-asyncapi-preview

AsyncAPI and You

Open standards, seamless integrations

Resources

Apache Kafka: https://kafka.apache.com

AsyncAPI: https://asyncapi.com

Aiven: https://aiven.io - try the free trial

Examples: https://github.com/aiven/thingum-industries

Microcks: https://microcks.io/

API Specifications Conference (Sep 28-29)

AsyncAPI Hackathon (Oct 1-31) Conference (Nov 16-18)
https://conference.asyncapi.com