



KubeCon

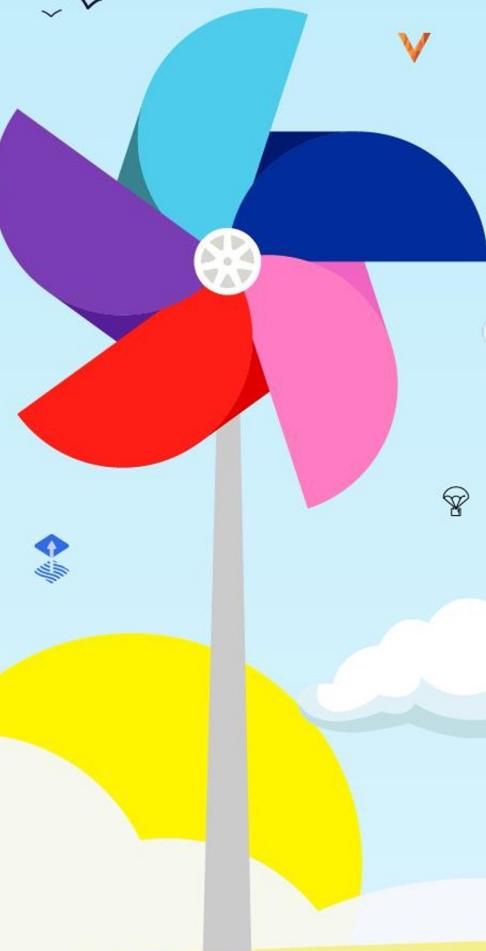


CloudNativeCon

Europe 2023



TiKV



Code of Conduct

Remember the **Golden Rule**: Treat others as you would want to be treated - with kindness and respect

Scan the QR code to access and review the **CNCF Code of Conduct**:



Virtual Audience Closed Captioning

Closed captioning for the virtual audience is available during each session through [Wordly](#).  The [Wordly](#) functionality can be found under the “Translation” tab on the session page.

[Wordly](#) will default to English. If another language is needed, simply click the dropdown at the bottom of the “Translation” tab and choose from one of 26+ languages available so you don’t miss a beat from our presenters.

*Note: Closed captioning is ONLY available during the scheduled live sessions and will not be available for the recordings on-demand within the virtual conference platform.

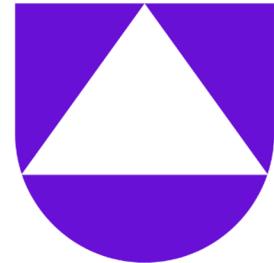
Session Q+A

- Virtual attendees may submit questions to speakers through the CNCF Slack channel: **#2-Kubecon-sessions**
- Please create a thread and tag the speaker(s) with questions about their talk.
- Questions will be answered by the speaker and/or other community members after the session concludes.

Sponsor Shout-Out!

Thank you to our Session Recording
Sponsor:

uptycs





KubeCon



CloudNativeCon

Europe 2023

Enjoy the Session!





KubeCon



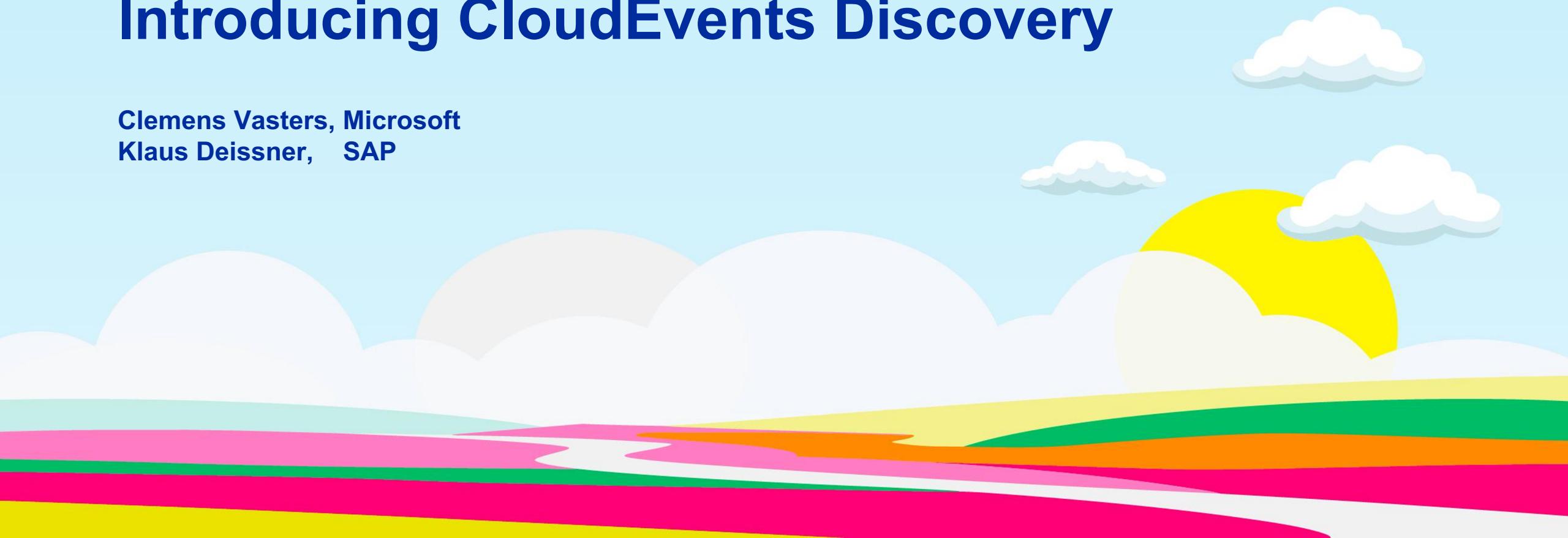
CloudNativeCon

Europe 2023

Introducing CloudEvents Discovery

Clemens Vasters, Microsoft

Klaus Deissner, SAP



What Happened so Far...

CloudEvents...

...specification was **started in CNCF's Serverless Working Group in (late) 2017**

...project approved as **sandbox-level project in May 2018**

...specification **v1.0 got released in October 2019**, incubation-level

Meeting of the Serverless Working Group At **KubeCon CloudNativeCon 2019 in San Diego**

⇒ Decision to **work on event discovery and subscriptions next**

Since then

- A lot of improvements and additional protocol bindings for the spec (today v1.02)
- SDK work
- Draft specs: CE SQL, **Registry**, Pagination and Subscriptions

What is Discovery?

What events are used in a specific context?

⇒ ***Event definition registry***

What do they look like?

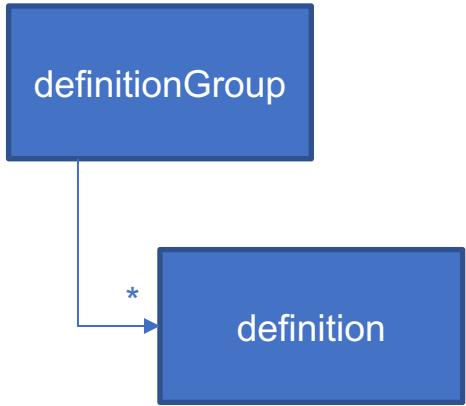
⇒ ***Schema registry***

Where can I consume or produce them?

⇒ ***Endpoint registry***



Event definitions



Groups of event definitions

Other message formats possible:

- AMQP
- MQTT
- Protobuf
- HTTP
- ...

```
{
  "id": "Contoso.CRM.Events.CustomerNoteAdded",
  "description": "A note has been added to a customer",
  "format": "CloudEvents/1.0",
  "metadata": {
    "attributes": {...}
  },
  "schemaurl": "#/schemaGroups/Contoso.CRM.Events/schemas/customerNoteAddedEventData"
}
```

Describing CloudEvent attributes

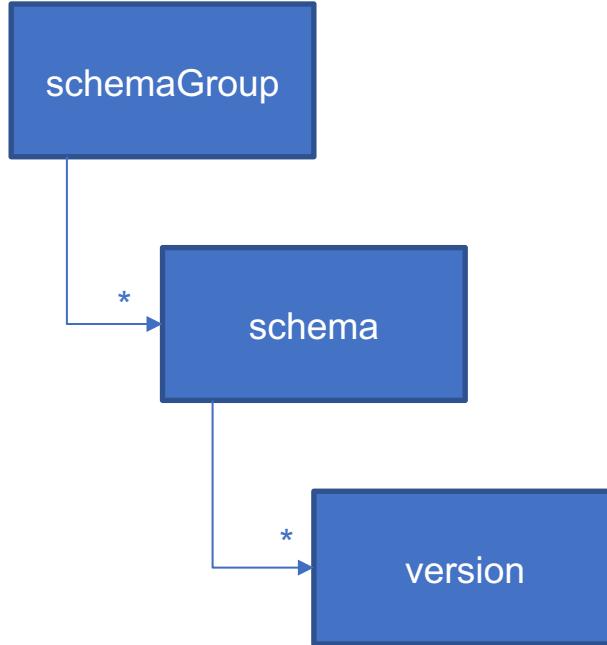
```
"attributes": {  
    "id": {  
        "type": "string",  
        "required": true  
    },  
    "type": {  
        "type": "string",  
        "value": "Contoso.CRM.Events.CustomerNoteAdded",  
        "required": true  
    },  
    "source": {  
        "type": "uritemplate",  
        "value": "/crm/customers/{region}",  
        "required": true  
    },  
    "subject" : {  
        "type":"uritemplate",  
        "value": "urn:uuid:{customer}",  
        "description":"Customer ID",  
        "required":true  
    }  
}
```

Additional constraints for context attributes

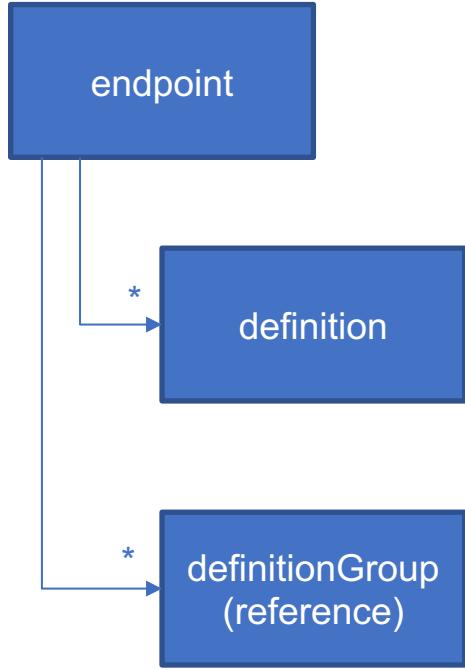
Mark attributes as *required*

Provide a more accurate type like *uri* or *uritemplate*

Schema registry



- Collect schema information for payloads in groups
- Store multiple versions of a schema
- Schema definitions are widely used beyond messages and events
- Independent from a specific schema definition language



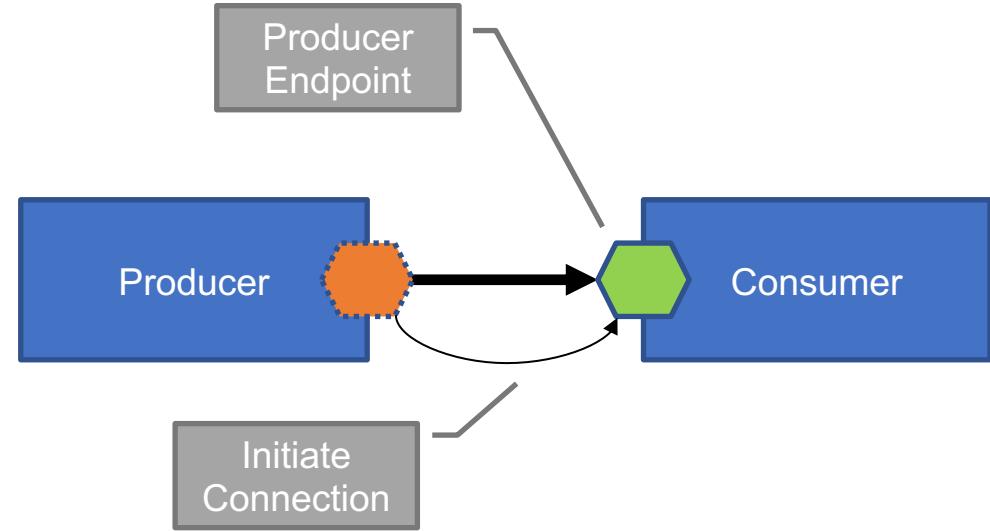
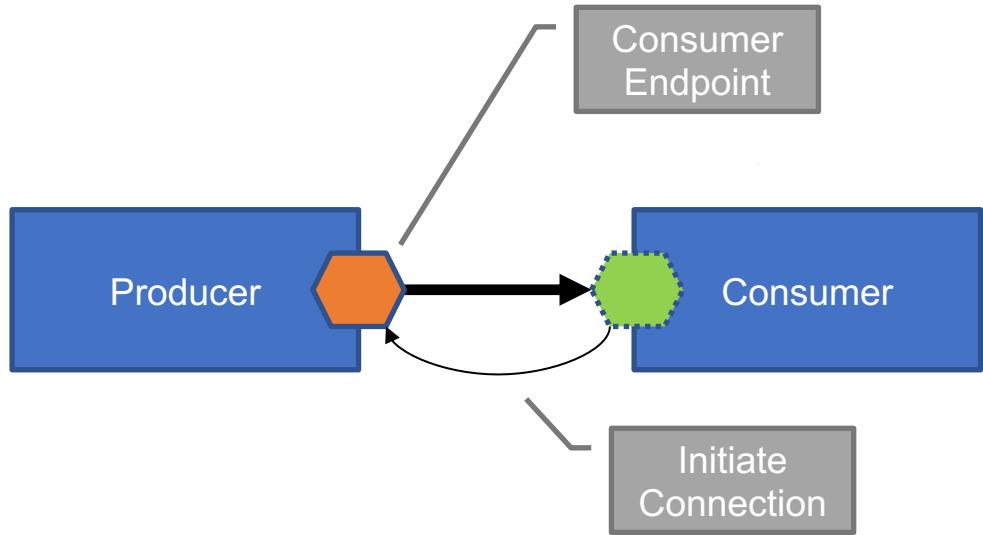
Superset of **definitionGroup**

Additional attributes

- **usage**: producer | consumer | subscriber
- **channel**: Correlate multiple endpoints of the same channel
- **definitionGroups**: References to existing definition groups
- **config**: Connectivity configuration

```
"Contoso.CRM.Eventing.Http": {
    "id": "Contoso.CRM.Eventing.Http",
    "usage": "producer",
    "config": {
        "protocol": "HTTP",
        "strict": false,
        "endpoints": [
            "https://erpsystem.com/events"
        ]
    },
    "channel": "my/topic",
    "definitionGroups": [
        "#/definitionGroups/Contoso.CRM.Events"
    ],
    "format": "CloudEvents/1.0"
}
```

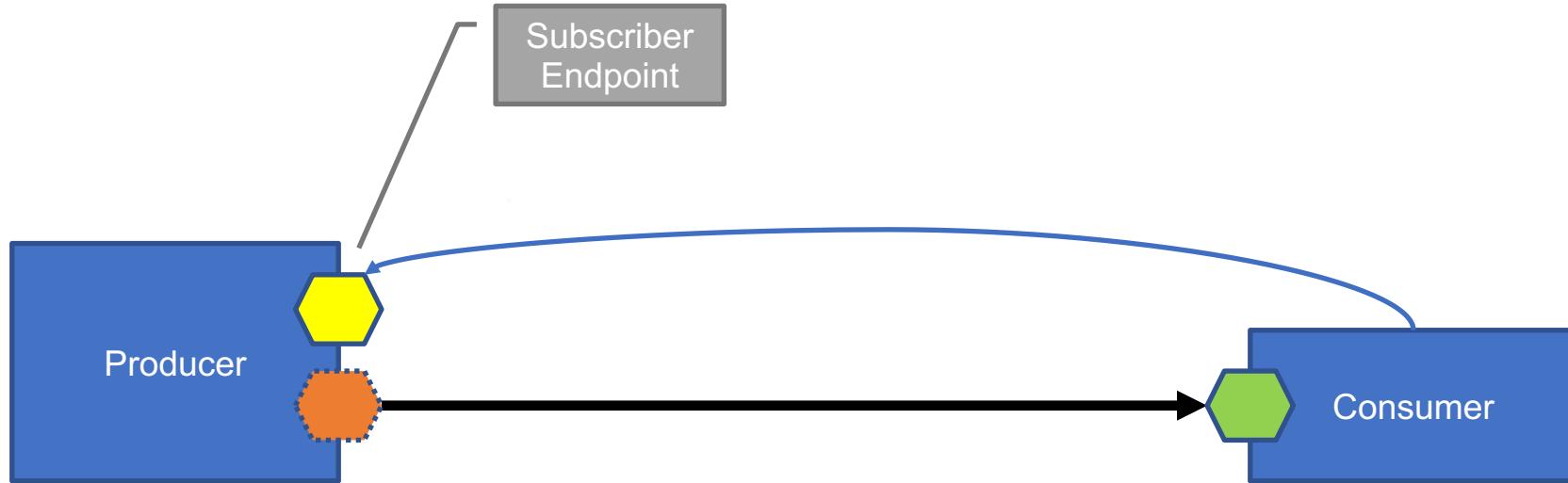
Producer and consumer endpoints



- Event consumers **consume events from these endpoints**
- Sometimes referred to as *Pull Model*
- Examples
 - Message consumption using MQTT or AMQP
 - Polling using HTTP (GET)

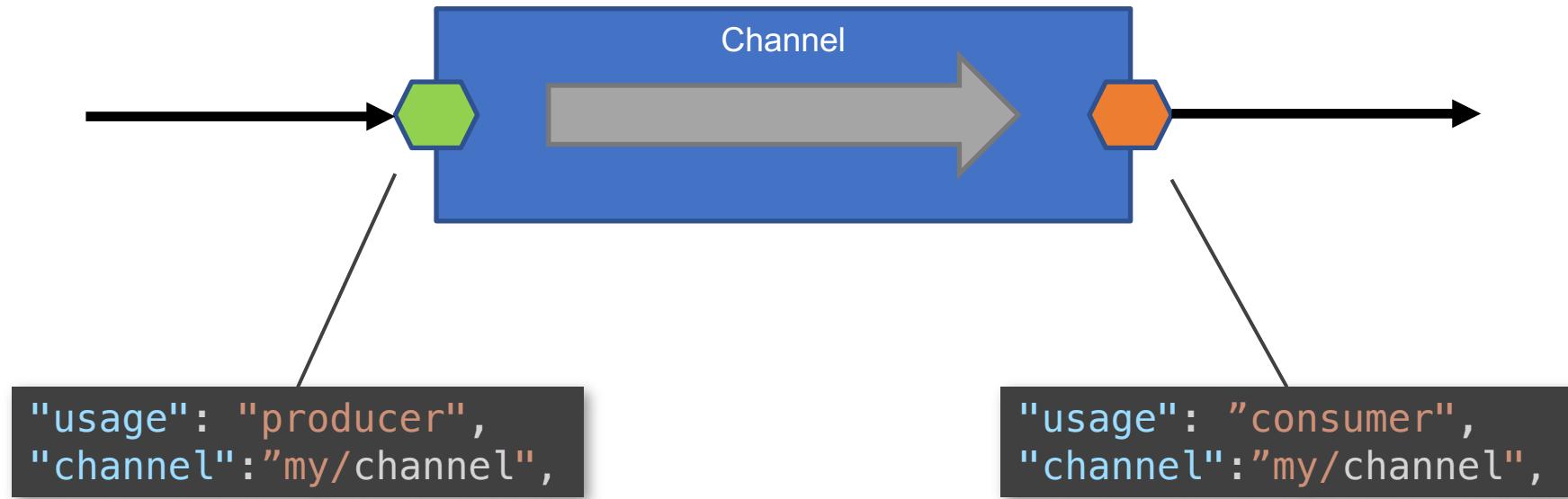
- Event **producers send events to these endpoints**
- Sometimes referred to as *Push Model*
- Examples
 - Message publishing to an MQTT or AMQP topic
 - Posting events to an HTTP web hook

Subscriber endpoints

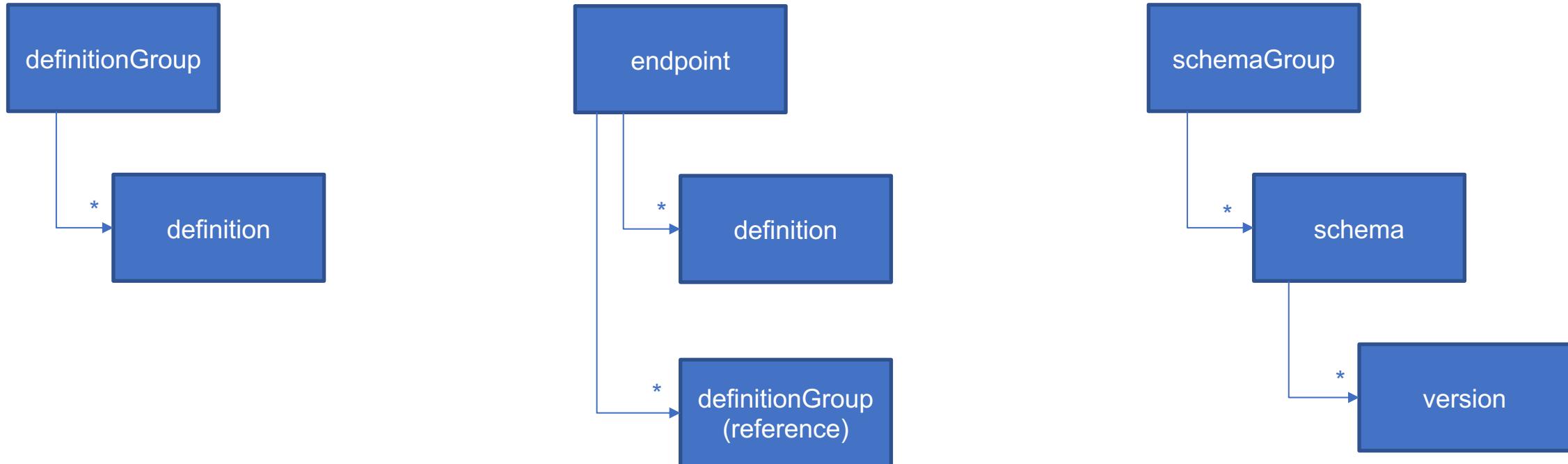


- A subscription endpoint **exposes a subscription API**
(<https://github.com/cloudevents/spec/blob/main/subscriptions/spec.md>)
- A subscription defines **a set of events** that shall be sent **to a specified producer endpoint**

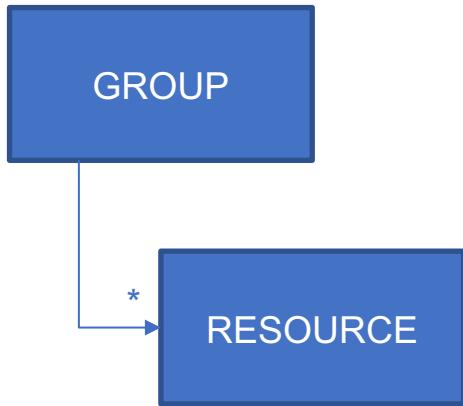
Channels



What do these have in common?



A Registry for Groups of Resources



API Access

/model	# Manage the registry model
/	# Show all Groups
/GROUPs	# Manage a Group Type
/GROUPs/gID	# Manage a Group
/GROUPs/gID/RESOURCES	# Manage a Resource Type
/GROUPs/gID/RESOURCES/rID	# Manage the latest Resource version
/GROUPs/gID/RESOURCES/rID?meta	# Metadata about the latest Resource version
/GROUPs/gID/RESOURCES/rID/versions	# Show version strings for a Resource
/GROUPs/gID/RESOURCES/rID/versions/vID	# Manage a Resource version
/GROUPs/gID/RESOURCES/rID/versions/vID?meta	# Metadata about a version

The New Repo on the Block

The screenshot shows the GitHub repository page for `xregistry`. The repository is titled "xRegistry" and is described as "The Extensible Registry". It has a Twitter handle @xregistryio. The navigation bar includes links for Overview (selected), Repositories (2), Projects, Packages, Teams (1), and People (3). Below the navigation, there's a section for "Popular repositories" featuring `xregistry.github.io` and `spec`. Both are marked as "Public".

xRegistry
The Extensible Registry
@xregistryio

Overview Repositories 2 Projects Packages Teams 1 People 3

Popular repositories

`xregistry.github.io` Public
xRegistry website

`spec` Public
xRegistry related specifications
1

Start Small...

All in one .cereg file

Managed with your source code

Inline everything

- Schema inside event definition
- Event definition inside endpoint

With infrastructure support

- Just schemas and event definitions
- Let infrastructure create endpoint definition



...Or Do Something More Advanced

Host a central registry in your organization

Define events and schemas for re-use

- Governance
- Versioning

Interoperability with other organizations

- Departments inside one company
- Vendors
- Customers

Federated discovery



Disclaimer

- Everything is **work in progress**
- Changes may and will happen

Opportunity: More people bring more ...

- ... use cases
- ... perspectives

Challenges

- Define CloudEvents for xRegistry
- Enrichment or overloading for event definitions?
- Endpoints that generically enrich events (example: partitionkey)
- Validate concepts **proof-of-concept implementations**



KubeCon



CloudNativeCon

Europe 2023

Questions?

Please provide your feedback

