KubeCon CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

CloudEvents and Beyond!

Doug Davis @ Microsoft
Co-Chair CloudEvents / Serverless WG

Agenda

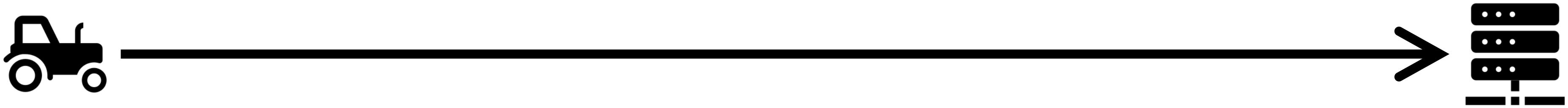
KubeCon CloudNativeCon North America 2022

CloudEvents – a quick review and update

Beyond CloudEvents

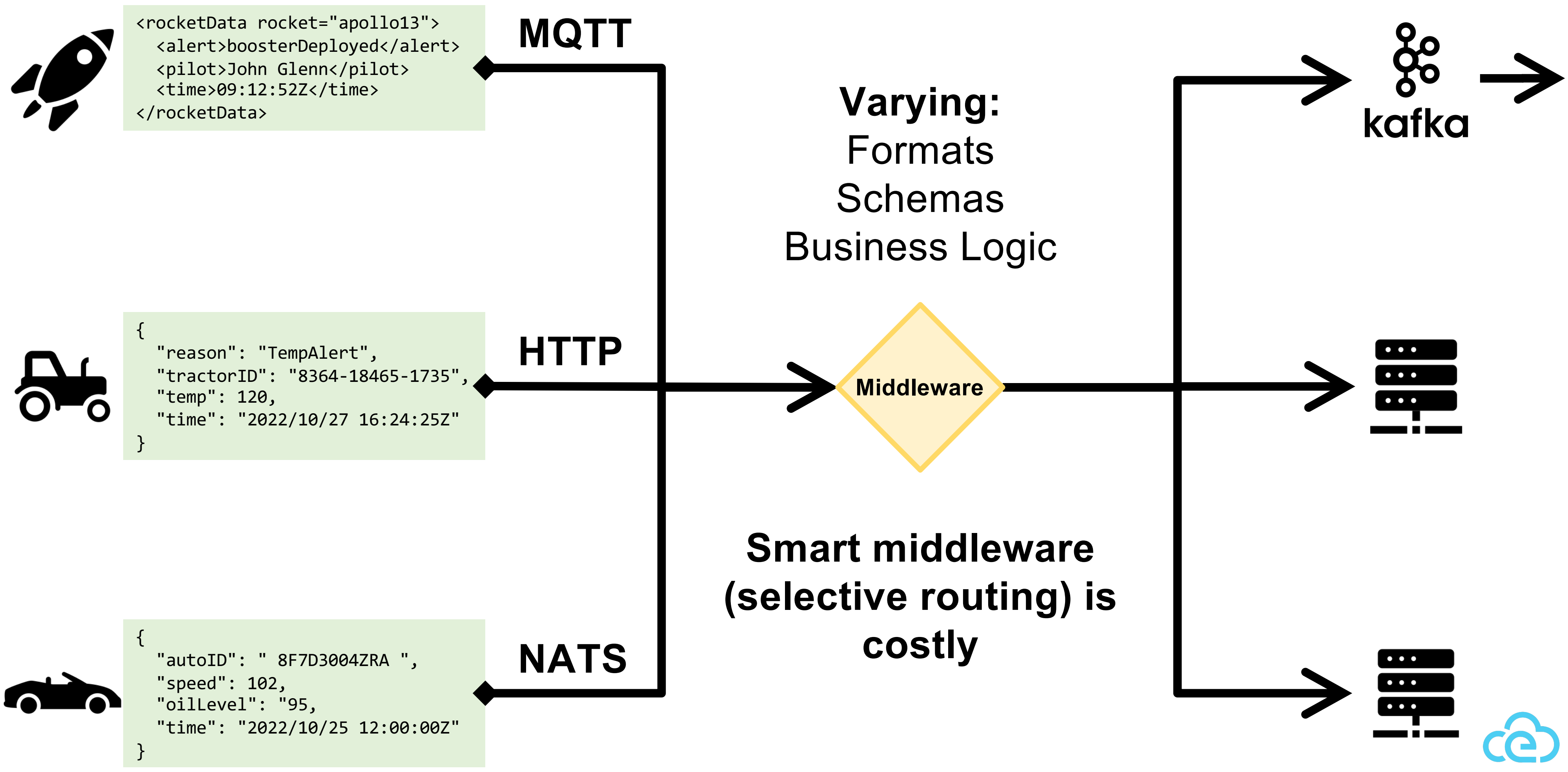
Events are Everywhere

```
{
  "reason": "TempAlert",
  "tractorID": "8364-18465-1735",
  "temp": 120,
  "time": "2022/10/27 16:24:25Z"
}
```



Known:
Producer
Transport = **Easy**
Schema
Format

Events are Everywhere



```
<rocketData rocket="apollo13">
  <alert>boosterDeployed</alert>
  <pilot>John Glenn</pilot>
  <time>09:12:52Z</time>
</rocketData>
```

```
{
  "reason": "TempAlert",
  "tractorID": "8364-18465-1735",
  "temp": 120,
  "time": "2022/10/27 16:24:25Z"
}
```

```
{
  "autoID": " 8F7D3004ZRA ",
  "speed": 102,
  "oilLevel": "95,
  "time": "2022/10/25 12:00:00Z"
}
```

Varying:
Formats
Schemas
Business Logic

Smart middleware (selective routing) is costly

CloudEvents

CloudEvents

- Defines common metadata for events
- Defines where to find that metadata in the messages

To aid in the delivery of events

No need to parse or understand the business logic just to route

CloudEvents is NOT trying to define a COMMON EVENT FORMAT!

CloudEvents in Action

HTTP - Binary

```
POST /events HTTP/1.0
Host: example.com
Content-Type: application/json
ce-specversion: 1.0
ce-type: com.bigco.newItem
ce-source: http://bigco.com/repo
ce-id: 610b6dd4-c85d-417b-b58f-3771e532

{
  "action": "newItem",
  "itemID": "93"
}
```

HTTP - Structured

```
POST /events HTTP/1.0
Host: example.com
Content-Type: application/cloudevents+json

{
  "specversion": "1.0",
  "type": "com.bigco.newItem",
  "source": "http://bigco.com/repo",
  "id": "610b6dd4-c85d-417b-b58f-3771e532",
  "datacontenttype": "application/json",
  "data": {
    "action": "newItem",
    "itemID": "93"
  }
}
```

CloudEvents Context Attributes

Required

- specversion (CE spec version)
- id
- source
- type

Optional (in spec)

- subject
- time
- datacontenttype
- dataschema

Extensions (no official standing)

- dataref
- distributed-tracing
  - traceparent
  - tracestate
- partitionkey
- sampledrate
- sequence
- severity
  - severitytext
  - severitynumber

CloudEvents Status

Deliverables

- CloudEvents specification v1.0.2 (Feb 2022)
- Transport Bindings (HTTP, AMQP, MQTT, NATS, Kafka, WebSockets)
- Encoding Formats (JSON, AVRO, Protobuf, XML*, Cbor*) (* Draft)
- SDKs (CSharp, Go, Java, Javascript, PHP, PowerShell, Python, Ruby, Rust)
- Primer

What's next?

- Customer feedback...
  - More proposed extensions
- **Additional community pain points...**

Eventing Lifecycle

CloudEvents helps with "routing" of events

What about the rest of the lifecycle?

- **Discovery** of Event Producer Endpoints
  - Consumption / subscription mechanism
- **Discovery** of Event Consumer Endpoints
  - How to push to an Event consumer
- **Discovery** of Event types, formats, attributes, ...
  - Discover supported Endpoints
  - For Event validation, automation/tooling, code generation, ...

This isn't just about events, but messages in general

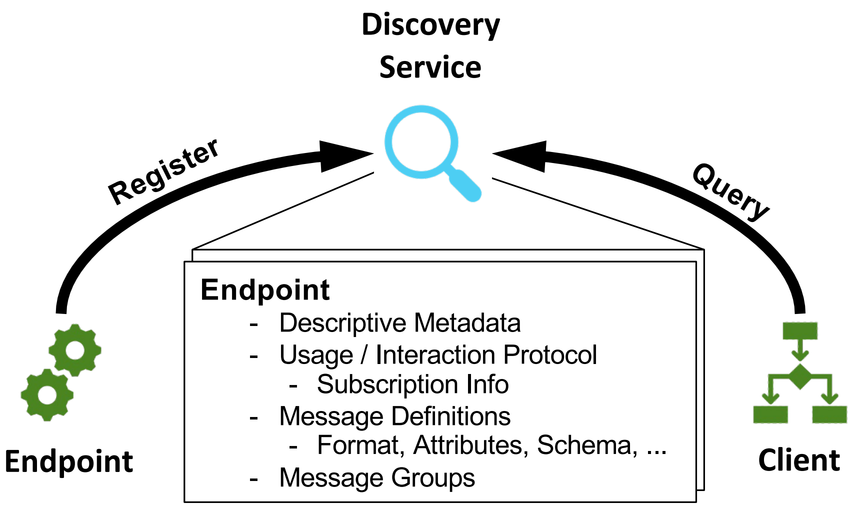
Discovery Service

Endpoint

- Sends or Receives messages (events)

Client needs to discover:

- Endpoints
  - Message definitions (format, schema, ...)
  - Groups of messages
- Client "usage" of the Endpoint
  - Consumer vs Producer
  - Location
  - Interaction protocol
- Subscription mechanism + options



Endpoint
- Descriptive Metadata
- Usage / Interaction Protocol
  - Subscription Info
- Message Definitions
  - Format, Attributes, Schema, ...
- Message Groups

Discovery Service - Sample "Endpoint"

```
{
  "id", "name", "self", "description",

  "usage": "subscriber", # or "consumer", "producer"
  "config": {
    "protocol": "http",
    "endpoints": "https://mydiscovery.com/subMgr"
  },

  "groups": [ "s3-events" ],
  "definitions": {
    "s3-obj-created": { ... },
    "s3-obj-updated": { ... },
    "s3-obj-deleted": { ... },

    "obj-replicated": { ... },
    "obj-synced": { ... }
  }
}
```

Definition

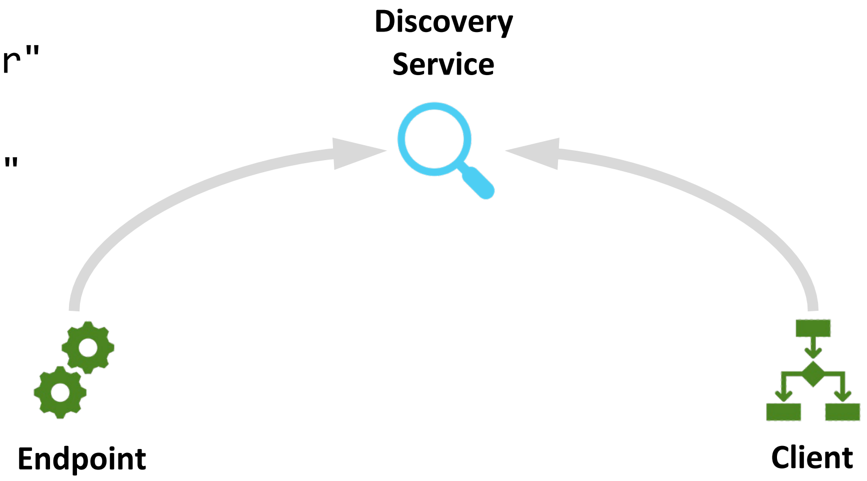
```
{
  "id": "name", "self", "description",

  "ownergroup": "/groups/s3-events",
  "format": "cloudevents/1.0",
  "metadata": {
    "attributes": {
      "type": {
        "required": true,
        "type": "string",
        "value": "com.aws.s3.Created",
      },

      ... other CE attributes ...
    }
  }
}
```

Discovery Service - Usage

```
{
  ...
  "usage": "subscriber",   # or "consumer", "producer"
  "config": {
    "protocol": "http",
    "endpoints": "https://mydiscovery.com/subManager"
  },
  ...
}
```



Discovery Service

Endpoint

Client

Consumer

- Pull messages from "config.endpoints"

Producer

- Send messages to "config.endpoints"

Subscriber

- Needs more...

Subscriptions

Discovery "Subscription mechanism"

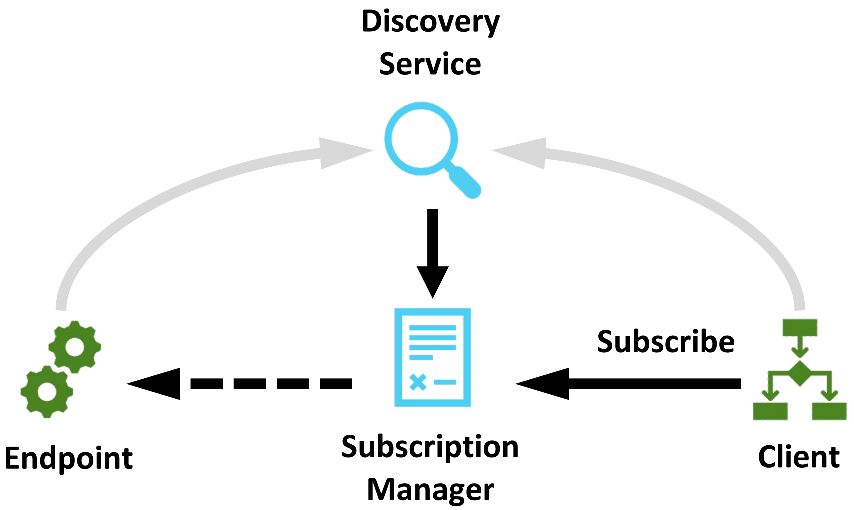
- How/where to subscribe
- Options - e.g. is filtering supported?
- Push vs Pull & transfer mechanism

Subscriptions API

- Defines "config.protocol" + options
- Defines mechanism if needed (e.g. HTTP)

Subscription Manager

- May be the Producer or act on its behalf



```
"usage": "subscriber",
"config": {
  "protocol": "http",
  "endpoints": "https://mydiscovery.com/subManager"
  ... subscription options ...
},
```

The Last Mile

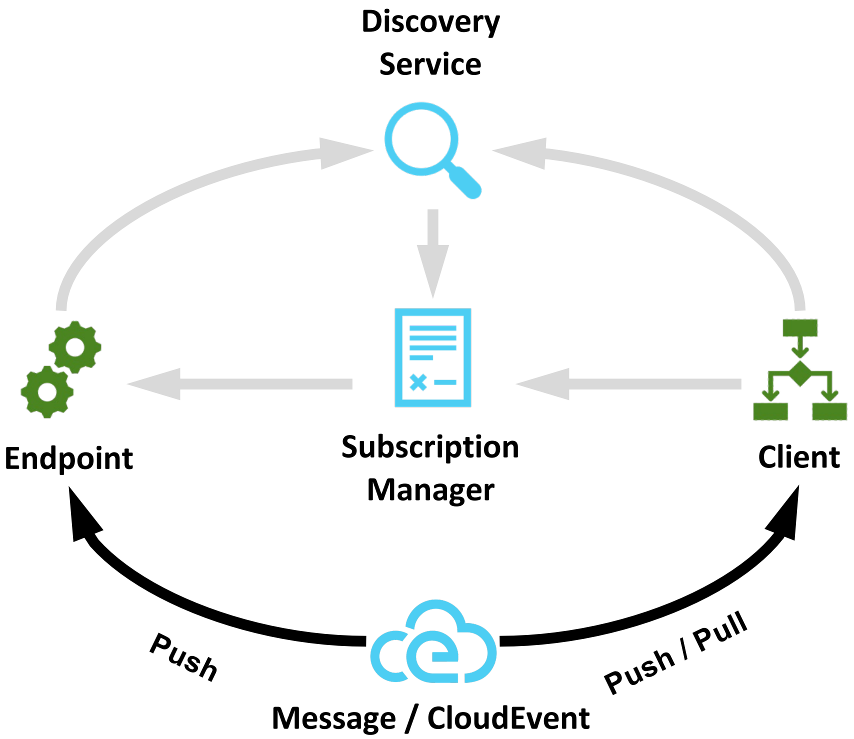
Message Delivery

- Transfer of Messages
- Validation of message
  - Metadata from Discovery Service
  - Including Message Schema

Message Schema

- Validation
- Serialization / Deserialization
- Tooling - e.g. code generation

No standard for storing/managing schemas

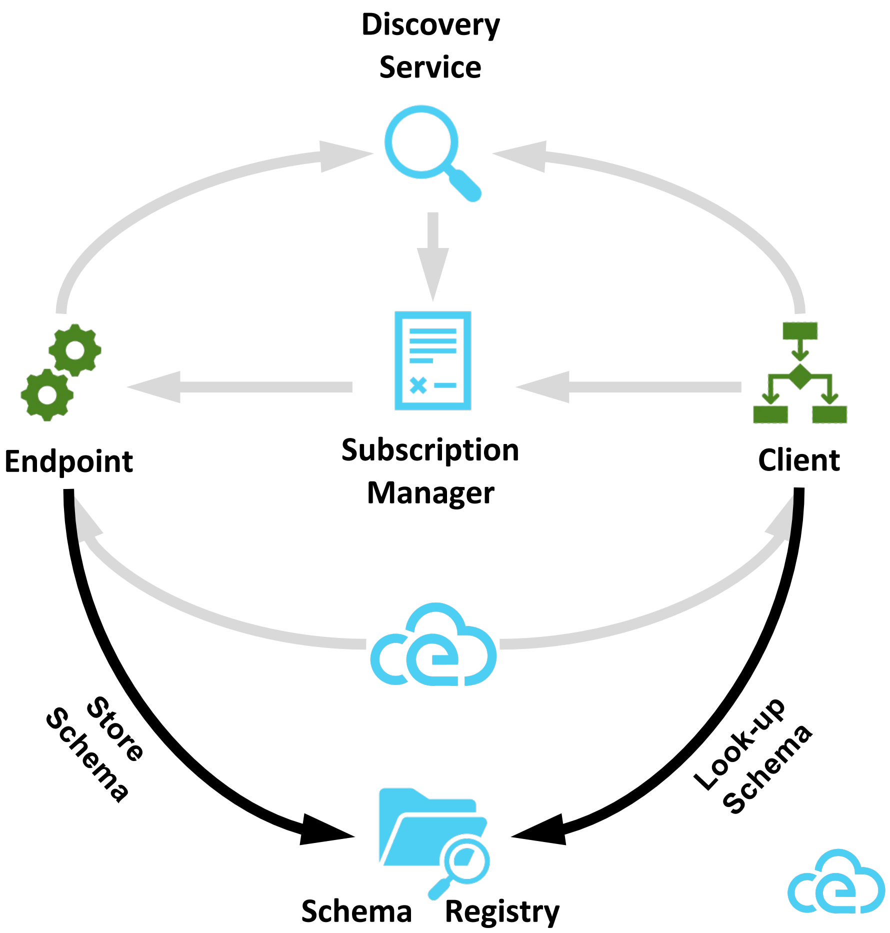


Schema Registry

Goals:

- Keep it simple
- Protocol neutral
- Scenario/schema/messaging neutral

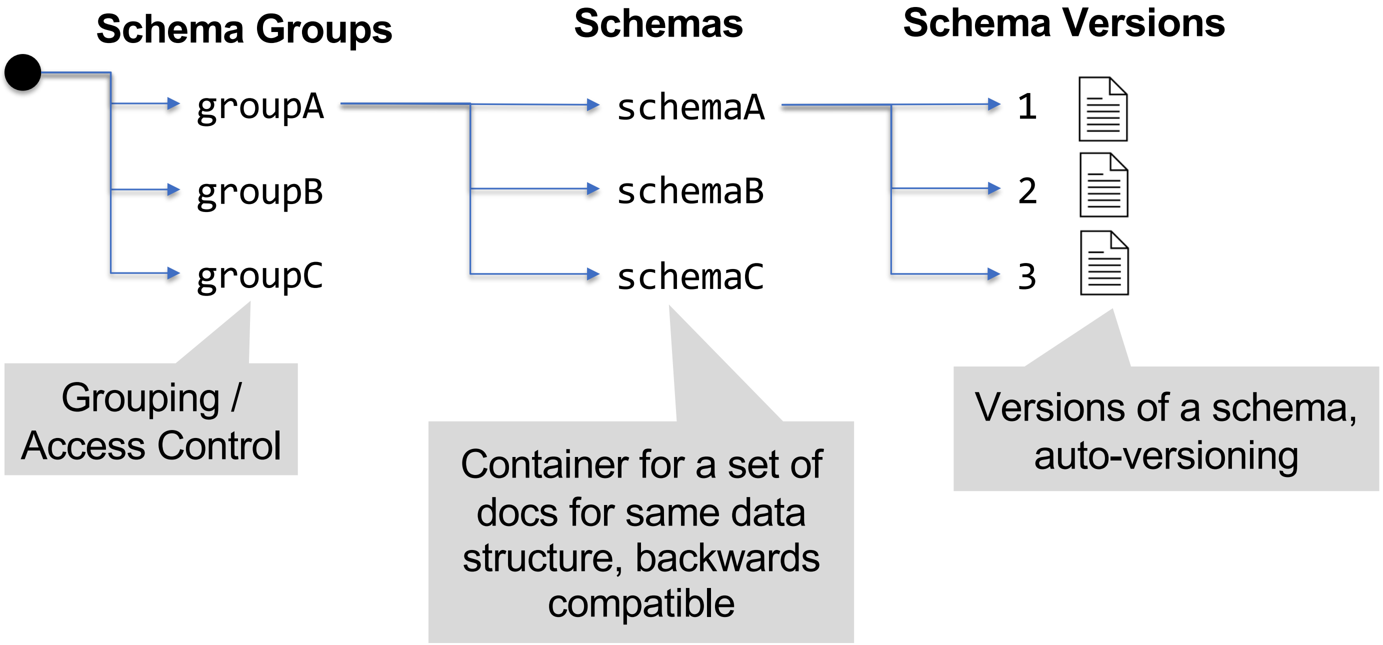
In HTTP terms: just GET/PUT



Schema Registry

Structure:

http://schemas.example.com/mygroup/schemas/myschema/versions/2



Schema Groups

groupA

groupB

groupC

Schemas

schemaA

schemaB

schemaC

Schema Versions

1

2

3

Grouping / Access Control

Container for a set of docs for same data structure, backwards compatible

Versions of a schema, auto-versioning

Summary

New Specs:

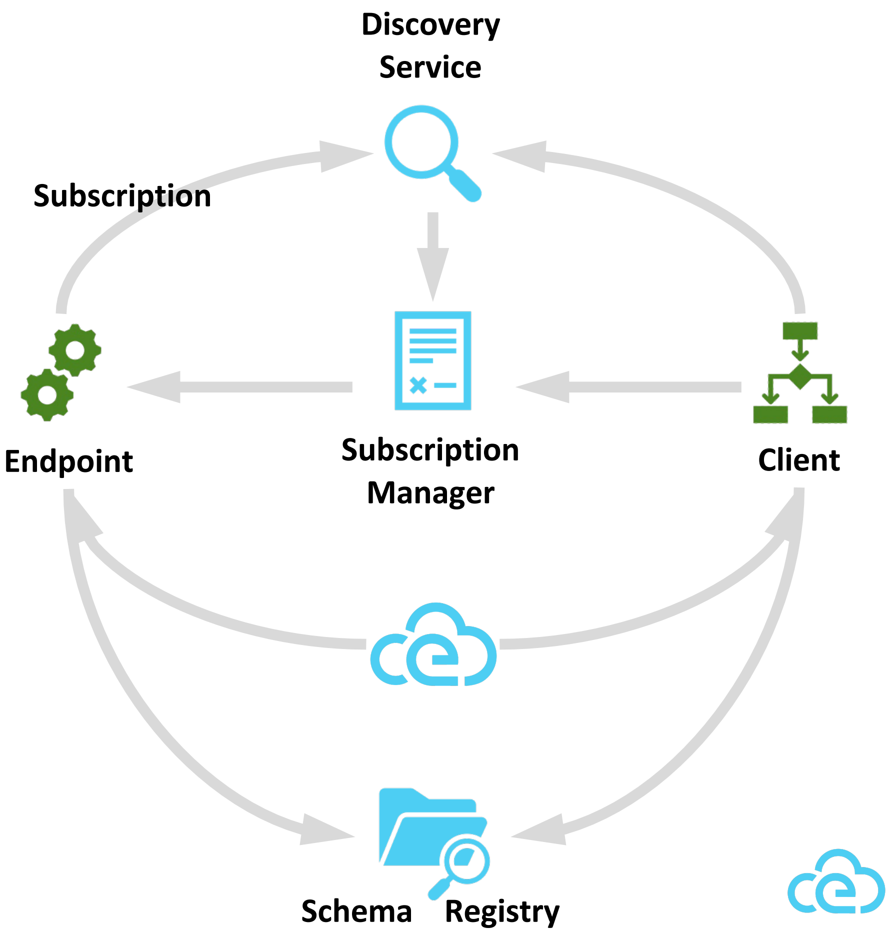
- Discovery API
- Subscriptions API
- Schema Registry

Goals:

- Improve interop
- Enable automation
  - Runtime generation & validation
  - Dev-time tooling

Status: WIP - release candidates by EOY?

- Looking for end-user feedback!!



Q&A - Thank You!

More information

- CloudEvents: https://cloudevents.io
- Specifications repo: https://github.com/cloudevents/spec
- Weekly calls: Thursdays @ 12pm ET (see repo for dial-in info & slack info)

Need more information?

- Doug Davis - dug at microsoft.com | @duginabox

Please scan the QR Code to
leave feedback on this session

Please scan the QR Code above to
leave feedback on this session