Where we are with Discovery

Klaus Deißner

WG-Serverless Meeting, 2021/06/24

Purpose of CloudEvents Discovery

- Consumers find out…
  - …what they can subscribe to
    - What sources are available?
    - What event types do they provide?
  - …how they can subscribe
    - Subscription endpoint
    - Supported filter dialects
  - …how they can consume
    - Protocols
- Intermediaries replicate discovery information
- Producers register with intermediaries to *announce* what they produce

Resource Model

Service Metadata

```
{
  "id": "[a globally unique UUID]",
  "epoch": "[discovery entry epoch value]",
  "name": "[unique name for this services]",
  "url": "[unique URL to this service]",
  "description": "[human string]", ?
  "docsurl": "[URL reference for human documentation]", ?
  "specversions": [ "[ce-specversion value]" + ],
```

Subscription Information

```
  "subscriptionurl": "[URL to which the Subscribe request will be sent]",
  "subscriptionconfig": { ?
    "[key]": "[type]", *
  },
  "subscrioptiondialects": [ "[dialect]" ], ?
  "authscope": "[string]", ?
  "protocols": [ "[string]" + ],
```

Events

```
  "events": [ ?
    { *
      "type": "[ce-type value]",
      "description": "[human string]", ?
      "datacontenttype": "[ce-datacontenttype value]", ?
      "dataschema": "[ce-dataschema URI]", ?
      "dataschematype": "[string per RFC 2046]", ?
      "dataschemacontent": "[schema]", ?
      "sourcetemplate": "[URI template per RFC 6570, level 1]", ?
      "extensions": [ ?
        { *
          "name": "[CE context attribute name]",
          "type": "[CE type string]",
          "specurl": "[URL to specification defining the extension]" ?
        }
      ]
    }
  ]
}
```

- Single Resource
- De-normalized (#630)
  - **Event type** not a separate resource
    - No re-use of event type definitions across services
    - Relation to static catalogs?
  - **Subscription information**
    - Attached to Service, but might be different in intermediaries
    - Mixed in by intermediaries?
    - What does this mean for epoch?

Service

- *A "service" represents the entity which manages one or more event sources and is associated with producers that are responsible for the generation of events. (spec)*
- Terminology discussion:
  - Source, producer, provider… Service? (#620)
- Questions
  - What does this mean for different architectures and tenancy models?
    - Regions, availability zones, clusters, namespaces…
    - Service vs. service instance
    - **Extreme case:** Each tenant of a service may be different

Identifying a Service

- **`id`**
  - UUID, assigned by the Discovery Endpoint
  - MUST NOT change
- **`name`**
  - *This value MUST be unique (case insensitive) within the scope of this Discovery Endpoint*
- **`url`**
  - *This value MUST be usable in subsequent requests, by authorized clients, to retrieve this Service entity.*
- **`epoch`**
  - *The only requirement is that the value MUST always increase each time the Service Entry is updated*

Discovery API (GET)

- GET
  - /services
  - /services/{id}
  - /services?name={name} (#682 Add support for partial name matching)

Management API

POST	PUT	DELETE
/services	/services/{id}	/services/{id}
/services?import	/services/{id}?import	

- POST works according to all-or-nothing principle
- “import” keeps service id, takes epoch into account
- Question: does PUT and ?import fulfill the idempotency requirements of PUT?

Status / Topics

- Issues
- Scenarios
  - Extend idea of `sourcetemplate` to other attributes?
- “Usability”
- Relation to other standards like OpenAPI and AsyncAPI

Brainstorming

- Discovery events instead of OR in addition to management API (#829)
- Apply concepts from schema registry
  - URIs
  - Authority
  - Replication
- Different protocol bindings (simple for discovery events)
- Service or “package” document
  - Documents are flexible
  - Can be stored together with code