52north
exploring horizons

REST INTERFACE FOR PUB/SUB

Design and Practical Examples

Matthes Rieke

AGILE 2017 Workshop “Event-based Dissemination and Processing of Geospatial Information”
Wageningen, 2017-05-09

OVERVIEW

1. PubSub Concepts
2. Mapping Concepts to REST
3. Data Model
4. Demo
5. Next Steps and Outlook

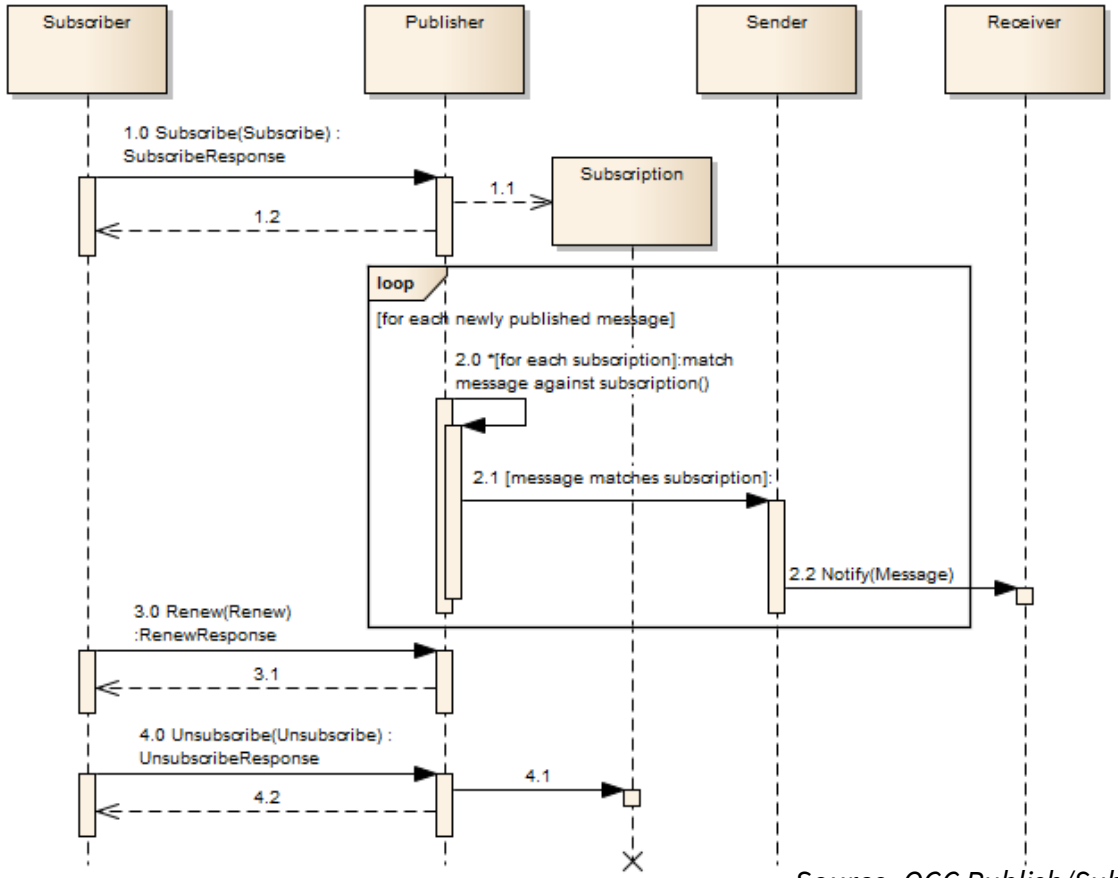
MOTIVATION

- OGC published in August 2016
  - “Publish/Subscribe Interface Standard 1.0 – Core”
  - “Publish/Subscribe Interface Standard 1.0 – SOAP Protocol Binding Extension”
- No REST binding specified yet
  - ***Clients***, in particular browser- and mobile-based, ***will not implement a heavyweight SOAP binding***
- OGC (i.e. within Testbed-12) identified basic concepts on how to approach REST bindings for OGC web services

OWS COMMON AND REST

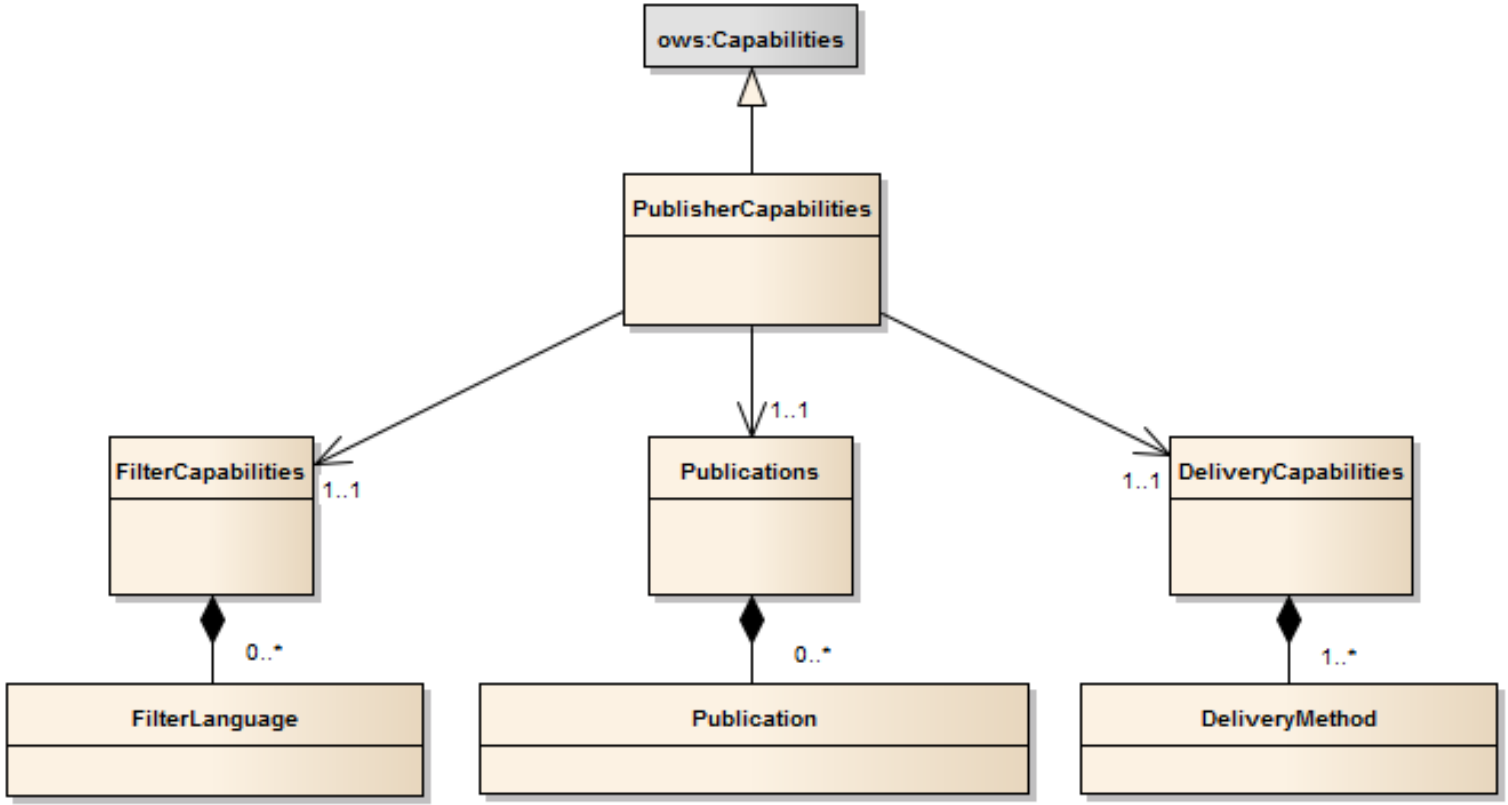
- Every OGC Web Service inherits from OWS Common
  - In particular, a *Capabilities document has to be provided*
    - *e.g. provided resources, operations, provider information*
- Some concepts within the OWS Capabilities are not applicable for REST APIs
- *OGC 16-035 Testbed-12 REST Architecture Engineering Report* states
  - "… while the first information [on provided resources] can be embedded in the Contents section of a Capabilities document, … in case of WPS ***the decision was made to not provide the OperationsMetadata section***."
  - "Recommendation 11: The OGC should specify how an additional API description can be provided that defines how HTTP methods can be applied…"
- Design decision for our PubSub REST *implementation*: do not provide a capabilities document for the time being, rely on common concepts (e.g. OpenAPI/Swagger for API description)

PUBSUB ARCHITECTURE AND CONCEPTS



Source: *OGC Publish/Subscribe Interface Standard 1.0 – Core*

PUBSUB ARCHITECTURE AND CONCEPTS



Source: *OGC Publish/Subscribe Interface Standard 1.0 – Core*

MAPPING CONCEPTS TO REST

- **Publications**
  - `<base-url>/publications/`
  - `<base-url>/publications/:publicationId`
- **DeliveryCapabilities**
  - `<base-url>/deliveryMethods/`
  - `<base-url>/deliveryMethods/:deliveryMethodId`
- **FilterCapabilities**
  - `<base-url>/templates/`
  - `<base-url>/templates/:templateId`

MAPPING CONCEPTS TO REST

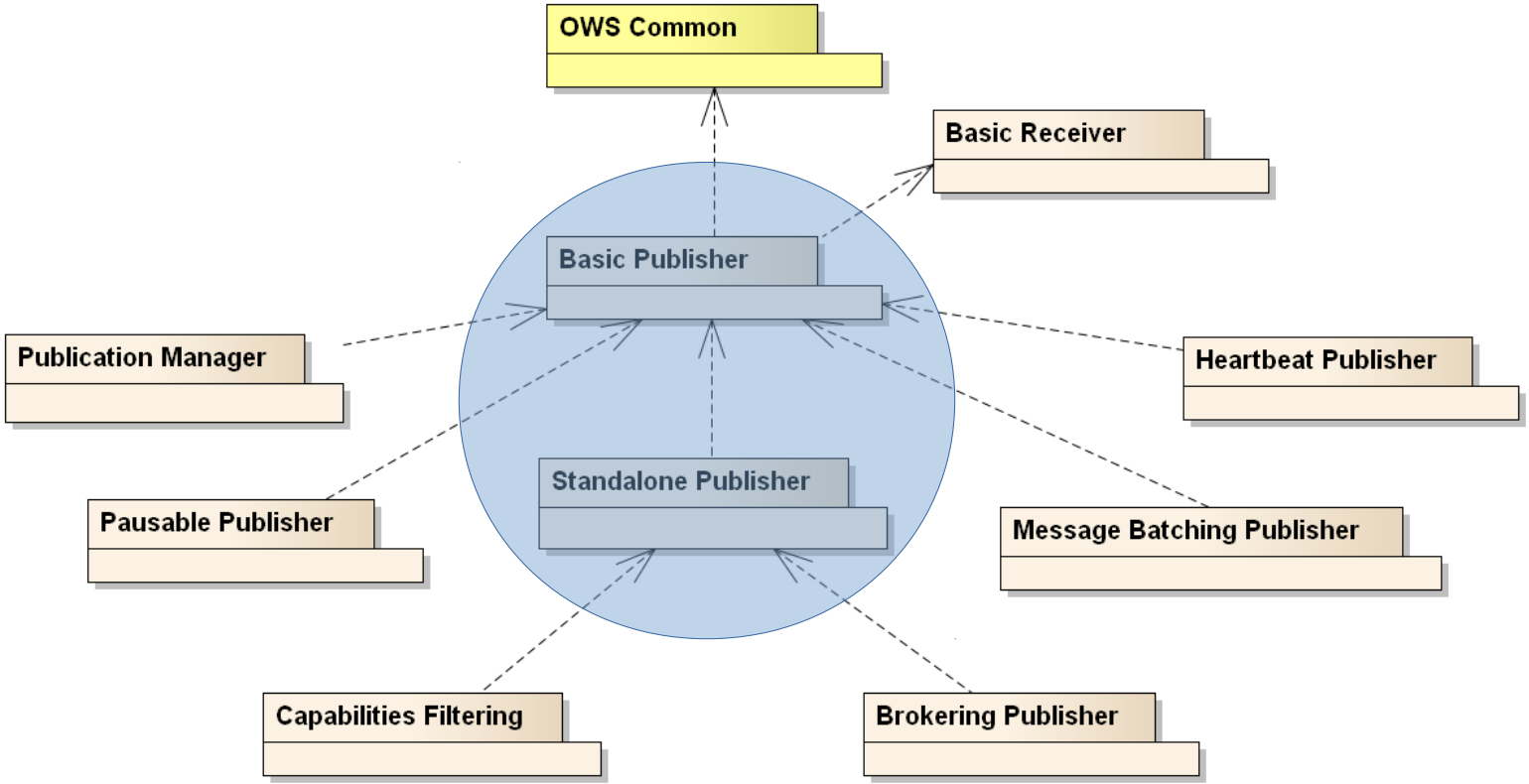
Additional Resources

- **Event endpoint**
  - Provides access to data that matched a subscription
  - `<base-url>/events/`
  - `<base-url>/events/:eventId`
  - `<base-url>/events/?subscription=:subscriptionId`
- **Subscription endpoint**
  - `<base-url>/subscriptions/`
  - `<base-url>/subscriptions/:subscriptionId`

MAPPING - INTERFACE METHODS

- `GetCapabilities` operation
  - Not available yet – maybe `HTTP GET` to the root resource
- `Subscription` management operations
  - **`Subscribe`**: `HTTP POST` against `<base-url>/subscriptions/`
  - **`Unsubscribe`**: `HTTP DELETE` against `<base-url>/subscriptions/:subId`
  - **`Renew`**: `HTTP PUT` against `<base-url>/subscriptions/:subId`
    - adjusted `terminationTime` property
  - **`GetSubscription`**: `HTTP GET` against `<base-url>/subscriptions/:subId`

MAPPING CONCEPTS



Source: *OGC Publish/Subscribe Interface Standard 1.0 – Core*

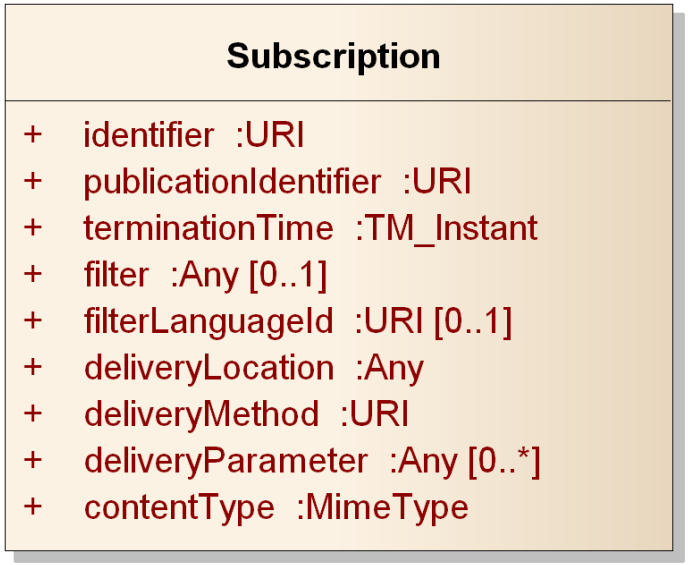
DATA MODEL

- Encoding of resources in JSON
- Currently follows a pragmatic approac
  - → atm not 100% spec compliant

```
{
  "label":"Point Polygon Subscription",
  "publicationId":"pip-pub",
  "template":{
     "id":"pointInPolygon",
     "parameters":{
        "coordinates":{
           "value":"60.0 60.0"
        }
     }
  },
  "deliveryMethods":[{
     "id":"email",
     "parameters":{
        "to":{
           "value":"test@test.de"
        }
     }
  }],
  "enabled":true,
  "terminationTime":"2030-06-19T13:22:08.248+02:00"
}
```

```
{
  "id": "email",
  "label": "Email delivery",
  "description": "Email delivery",
  "parameters": {
    "cc": {
      "type": "text",
      "label": "cc",
      "optional": false
    },
    "subject": {
      "type": "text",
      "label": "subject",
      "defaultValue": "[Notification] Rule matched",
      "optional": false
    },
    "to": {
      "type": "text",
      "label": "to",
      "optional": false
    }
  }
}
```

DATA MODEL




```
{
  "label":"Point Polygon Subscription",
  "publicationId":"pip-pub",
  "template":{
     "id":"pointInPolygon",
     "parameters":{
        "coordinates":{
           "value":"60.0 60.0"
        }
     }
  },
  "deliveryMethods":[{
     "id":"email",
     "parameters":{
        "to":{
           "value":"test@test.de"
        }
     }
  }],
  "enabled":true,
  "terminationTime":"2030-06-19T13:22:08.248+02:00"
}
```


Source: *OGC Publish/Subscribe Interface Standard 1.0 – Core*

EXAMPLES AND DEMO

- Wupperverband Eventing REST API
  - Quick API demo
  - Quick facts
    - timeseries (SWE) = publication (PubSub)
    - Authentication/authorization – user and group management
- WaterInnEU research project
  - Point in polygon spatial filter (→ Filtering Capabilities)
  - WebSocket delivery method

WUPPERVERBAND

für Wasser, Mensch und Umwelt

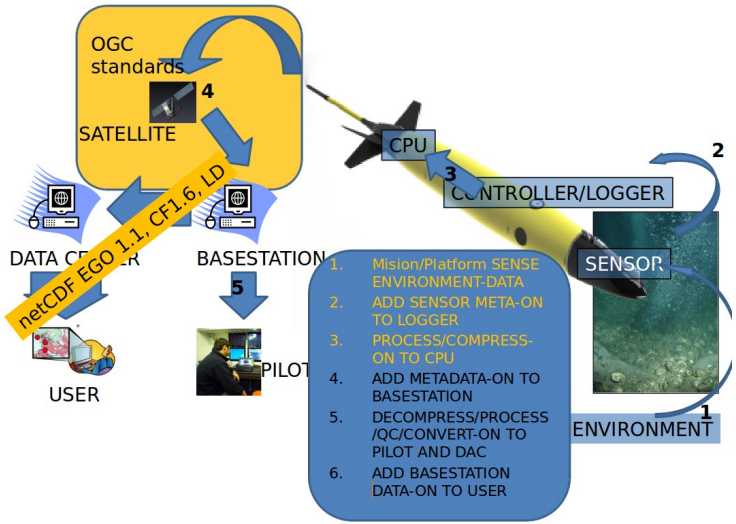
WaterInnEU

RESEARCH AGENDA

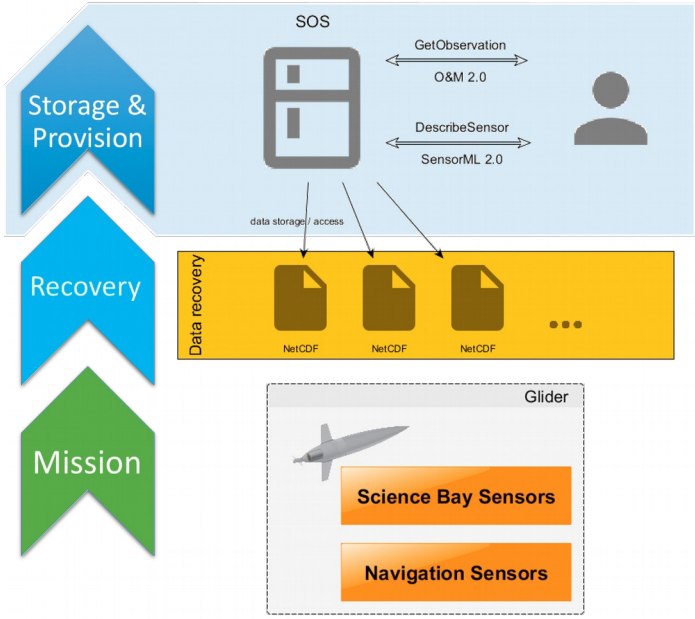
- Oceanology: Incorporate Pub/Sub into data management architecture of Deep Sea Gliders

  → **BRIDGES H2020**

BRIDGES
BRINGING TOGETHER RESEARCH AND INDUSTRY FOR THE DEVELOPMENT OF GLIDER ENVIRONMENTAL SERVICES



Source: *Hayes 2016*



Source: *BRIDGES D3.3 "Interface standards for applications of deep and ultra-deep glider"*

NEXT STEPS

- Approach on how to bring this into the OGC?
  - Ideas?
  - Alternative designs?
  - Template approach interesting?
- Outlook
  - Incorporate recommendations of O*GC 16-035 Testbed-12 REST Architecture Engineering Report*
  - General PubSub SWG work
    - Definition of DeliveryMethod profiles (e.g. MQTT, AMQP, WebSockets, …) → enables actual interoperability
    - Possible conceptual extension to include “Eventing” ~= creation of *higher level information* in contrast to *sub-setting of publications data stream*

52north
exploring horizons

THANKS!

Matthes Rieke

m.rieke@52north.org