Norwegian Meteorological Institute

Frost

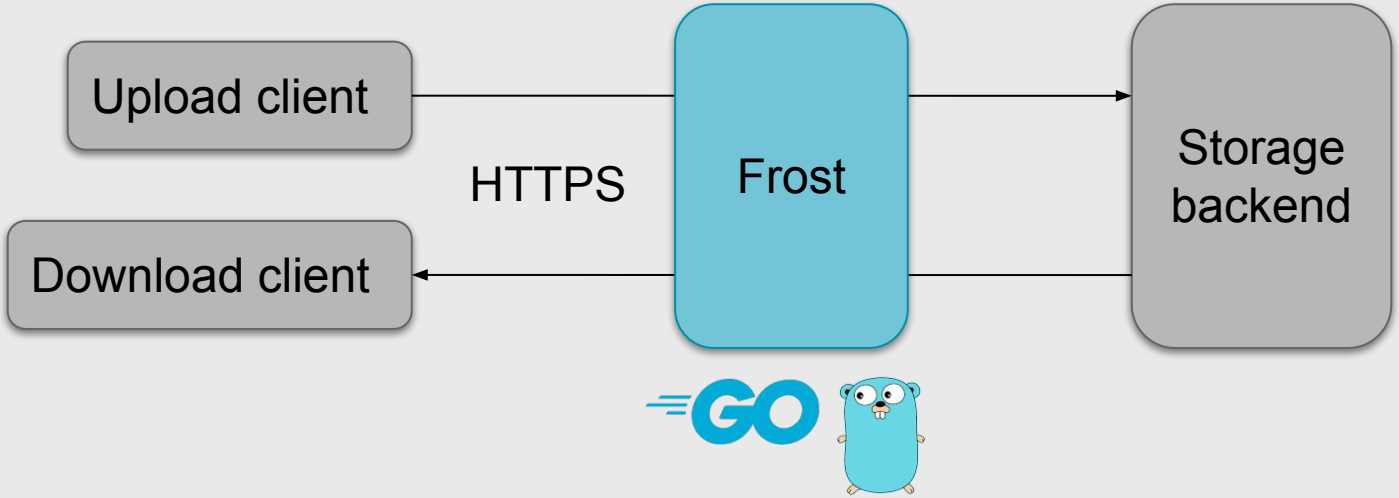
A REST API for point observations

2021-11-02 Jo Asplin

Scope

This presentation is about Frost *v1* (e.g. havvarsel-frost.met.no), not Frost *v0* (e.g. frost.met.no).

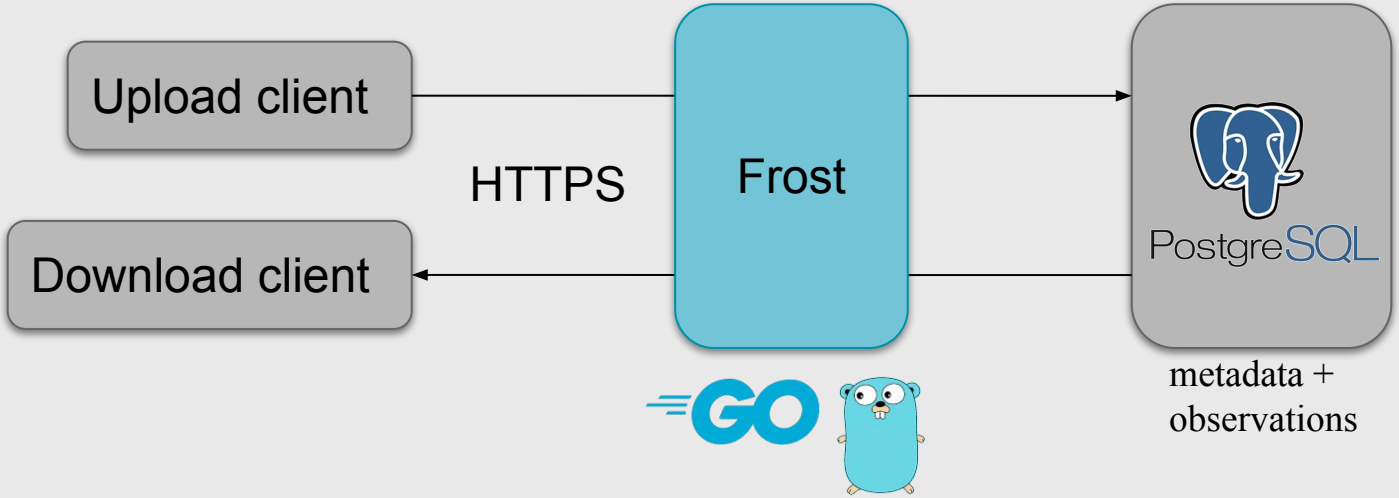
Overview



Norwegian Meteorological Institute

Storage backend - Option 1

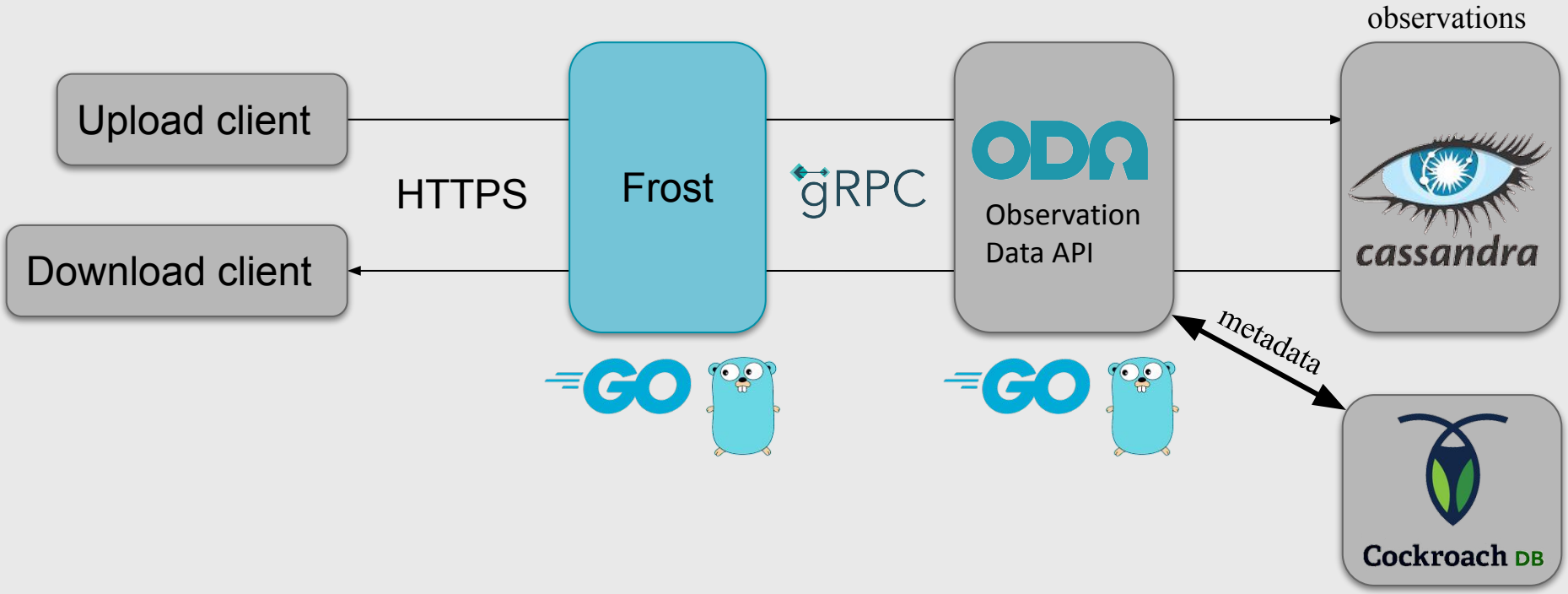
currently used for havvarsel-frost.met.no



Norwegian Meteorological Institute

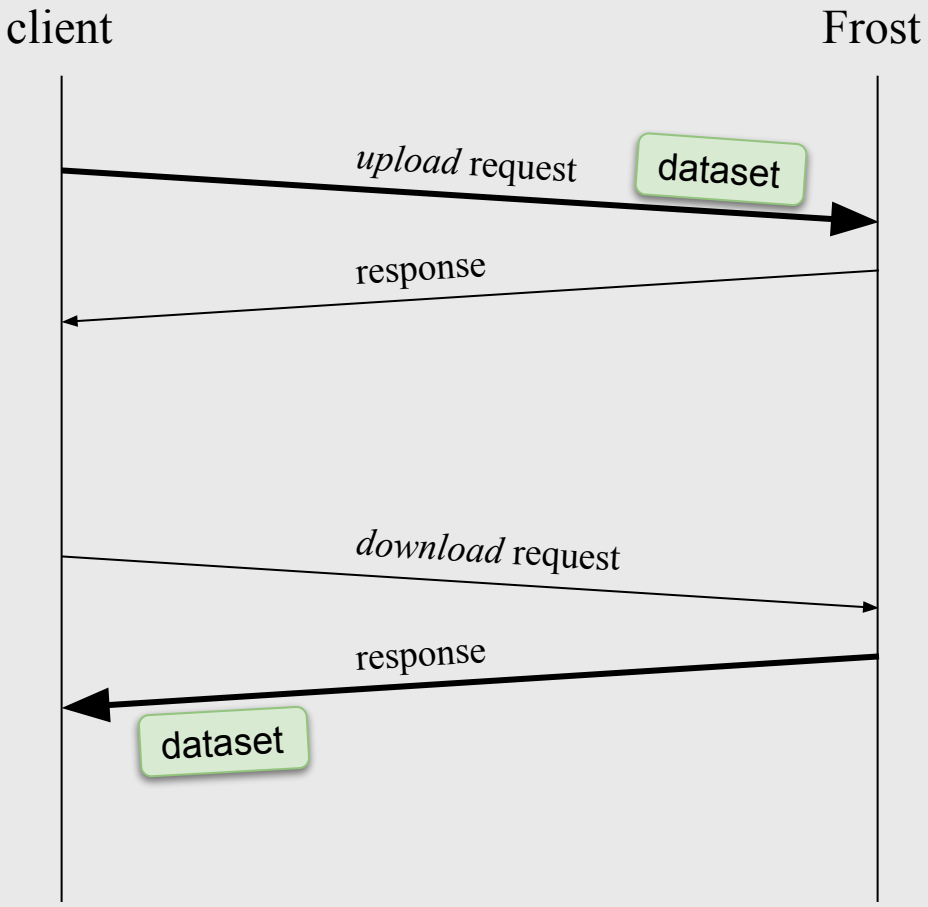
Storage backend - Option 2

higher capacity/performance - to be used
as MET's main point observation storage



Norwegian Meteorological Institute

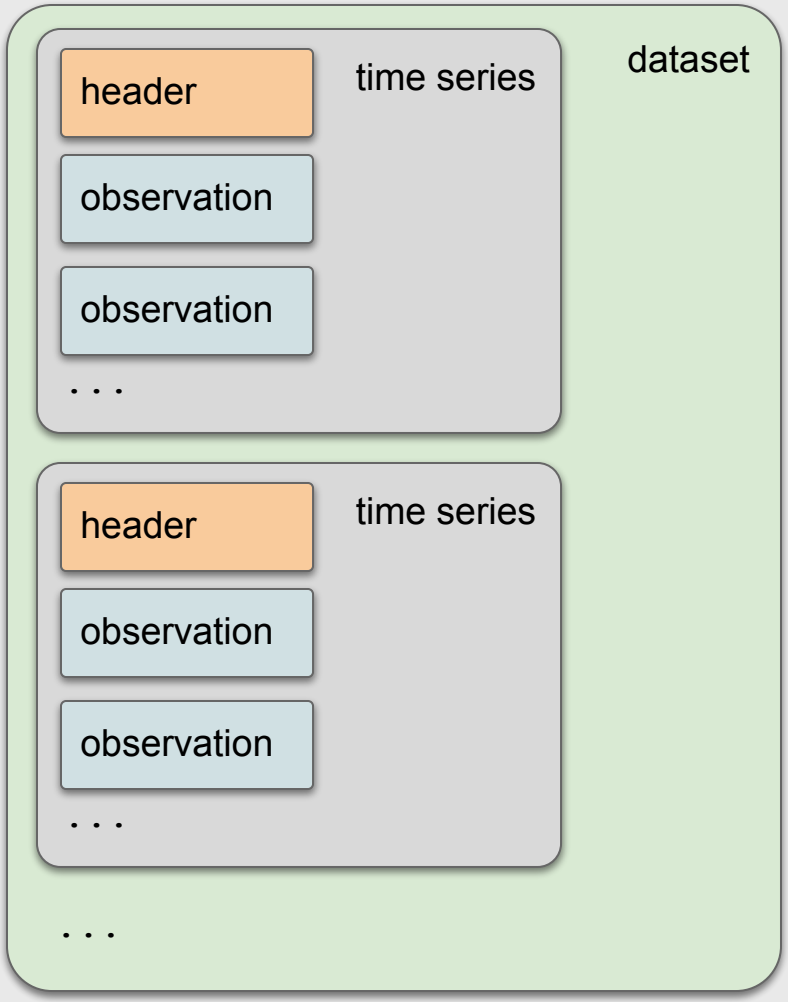
Request / response



- no session state kept in server, i.e. each request can be understood in isolation
- same dataset format for both upload and download
- HTTPS/POST for upload
- HTTPS/GET for download

Norwegian Meteorological Institute

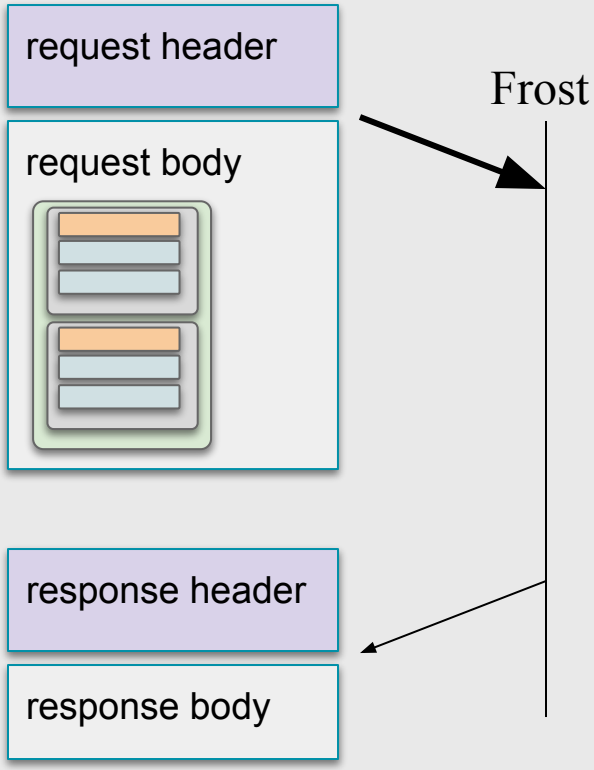
Overall dataset structure



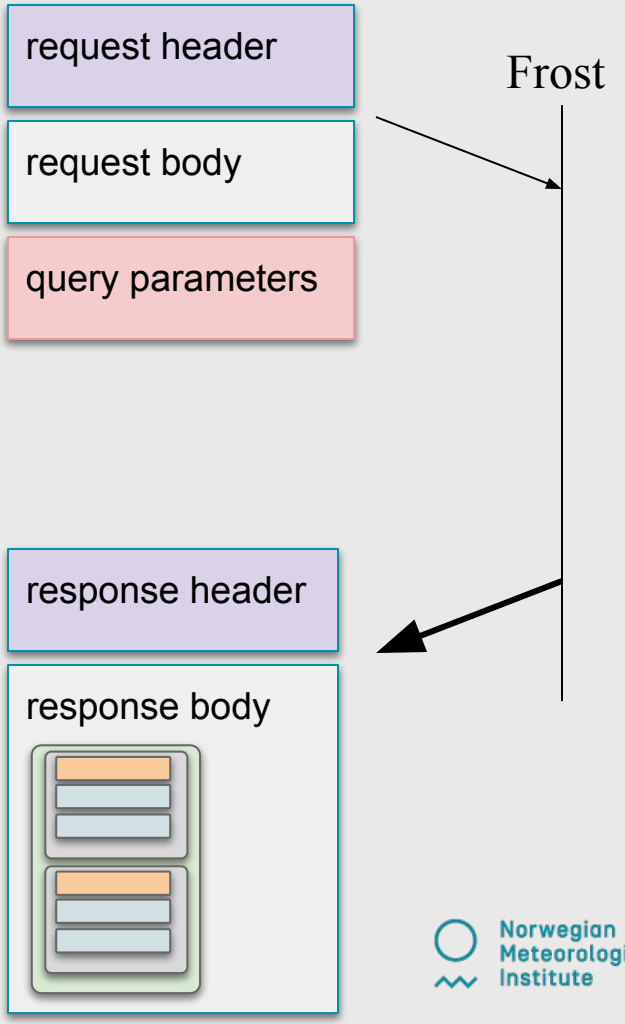
Norwegian Meteorological Institute

Datasets passed over HTTPS

Upload with HTTPS/POST requests:

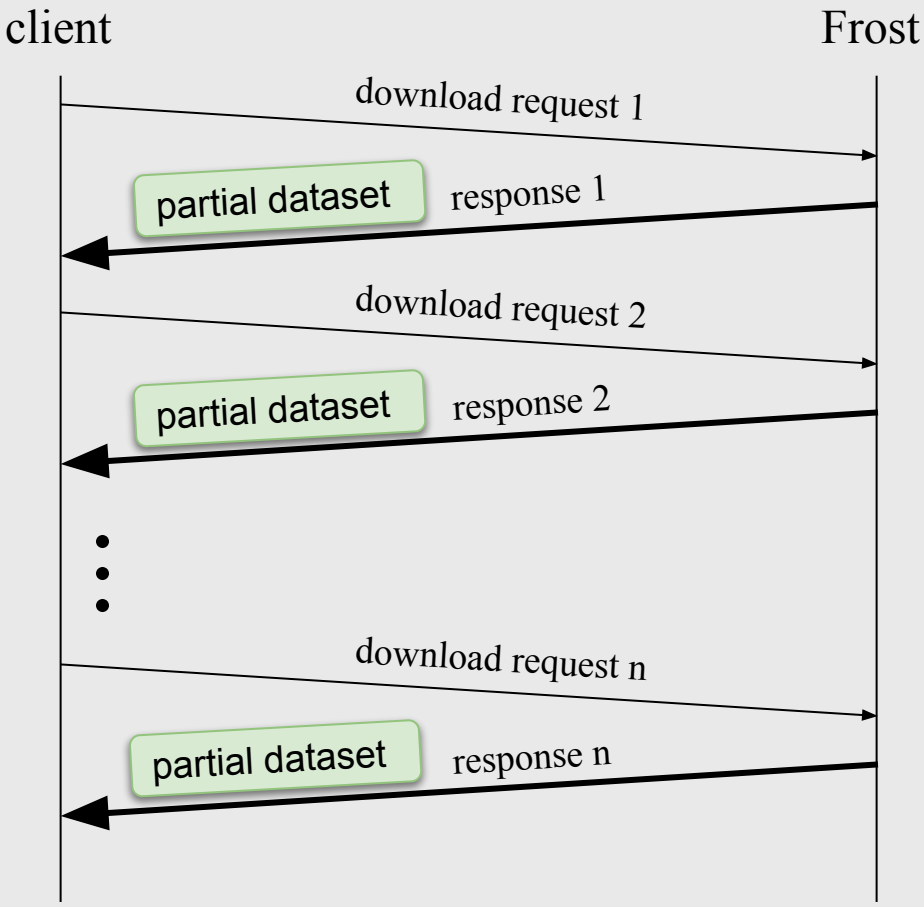


Download with HTTPS/GET requests:



Norwegian Meteorological Institute

Downloading a large dataset



A single request may result in a too large response (limit imposed by client or Frost).

Option 1: Partition explicitly by sending *n* different subrequests.

Option 2: Partition implicitly by using a special *pagination protocol*: send the same request *n* times and copy fields from response header *i* to request header *i*+1.

In any case the client must assemble subresponses into a total result.

Norwegian Meteorological Institute

Overall dataset format (JSON)

```
{
    “tstype”: “<time series type>”,
    “tseries”: [
        {<time series 1>},
        {<time series 2>},
        ...
    ]
}
```

Time series format

```
{
    “header”: {
        “id”: {<primary key of time series>},
        “extra”: {<additional header fields>}
    },
    “observations”: [
        {
            “time”: “<observation time (ISO 8601)>”,
            “body”: {<observation value
                      + additional metadata>}
        },
        ...
    ]
}
```


Norwegian Meteorological Institute

Example: badevann

```
{
    "tstype": "badevann",
    "tseries": [
        {
            "header": {
                "id": {
                    "source": "badetassen.no",
                    "buoyID": "20",
                    "parameter": "temperature"
                },
                "extra": {
                    "name": "Møllebukta",
                    "pos": {
                        "lat": "58.941010",
                        "lon": "5.670380"
                    }
                }
            },
            "observations": [ NEXT PAGE! ]
        }
    ]
}
```

Norwegian Meteorological Institute

Example: badevann (cont'd)


```
"observations": [
    {
        "time": "2021-10-31T10:25:30Z",
        "body": {
            "value": "10.3"
        }
    },
    {
        "time": "2021-10-31T12:25:36Z",
        "body": {
            "value": "10.1"
        }
    },
    ...
]
```


Norwegian Meteorological Institute

Example: glider


```
{
    "tstype": "glider",
    "tseries": [
        {
            "header": {
                "id": {
                    "source": "UIB-GI",
                    "gliderID": "5620625",
                    "parameter": "sea_water_temperature"
                },
                "extra": {
                    "name": "sg562"
                }
            },
            "observations": [ NEXT PAGE! ]
        }
    ]
}
```


Norwegian Meteorological Institute

Example: glider (cont'd)


```
"observations": [
    {
        "time": "2020-06-16T06:00:00Z",
        "body": {
            "pos": {
                "lat": 59.819879,
                "lon": 10.578601
            },
            "value": 12.34,
            "qc_flag": "9"
        }
    },
    ...
]
```


Norwegian Meteorological Institute

Example: vertical-profile

```
{
    "tstype": "vertical-profile",
    "tseries": [
        {
            "header": {
                "id": {
                    "instrument": "...",
                    "parameter": "..."
                },
                "extra": {
                    ...
                }
            },
            "observations": [  NEXT PAGE! ]
        }
    ]
}
```

Norwegian Meteorological Institute

Example: vertical-profile (cont'd)

```
"observations": [
    {
        "time": "2021-10-31T10:25:30Z",
        "body": {
            "pos": {
                "lat": "...",
                "lon": "..."
             },
             "depth": ["...", "...", ...],
             "value": ["...", "...", ...],
             "qc_flag": ["...", "...", ...]
        }
    },
    ...
]
```

Norwegian Meteorological Institute

Selecting time series and observations

EXAMPLE 1:

Select from time series type *badevann* all temperature observations from buoy id 38 in October 2021.

```
https://havvarsel-frost.met.no/api/v1/obs/badevann/
get?time=2021-10-01T00:00:00Z/2021-11-01T00:00:00Z&
incobs=true&buoyids=38
```

Selecting time series and observations (cont'd)

EXAMPLE 2:

Select from time series type *badevann* all temperature observations from buoy ids 38 and 20 in October 2021.

```
https://havvarsel-frost.met.no/api/v1/obs/badevann/
get?time=2021-10-01T00:00:00Z/2021-11-01T00:00:00Z&
incobs=true&buoyids=38,20
```

obs times buoy id 20:
2021-10-06T12:51:38
2021-10-06T13:21:39
2021-10-06T14:21:43
2021-10-06T14:51:45
...

obs times buoy id 38:
2021-10-11T12:54:34
2021-10-12T12:54:15
2021-10-12T18:54:14
2021-10-13T00:54:13
...

Selecting time series and observations (cont'd)

EXAMPLE 3:

Select from time series type *badevann* only headers (i.e. no observations) for buoy ids 38 and 20.

```
https://havvarsel-frost.met.no/api/v1/obs/badevann/
get?&incobs=false&buoyids=38,20
```

Selecting time series and observations (cont'd)

EXAMPLE 4:

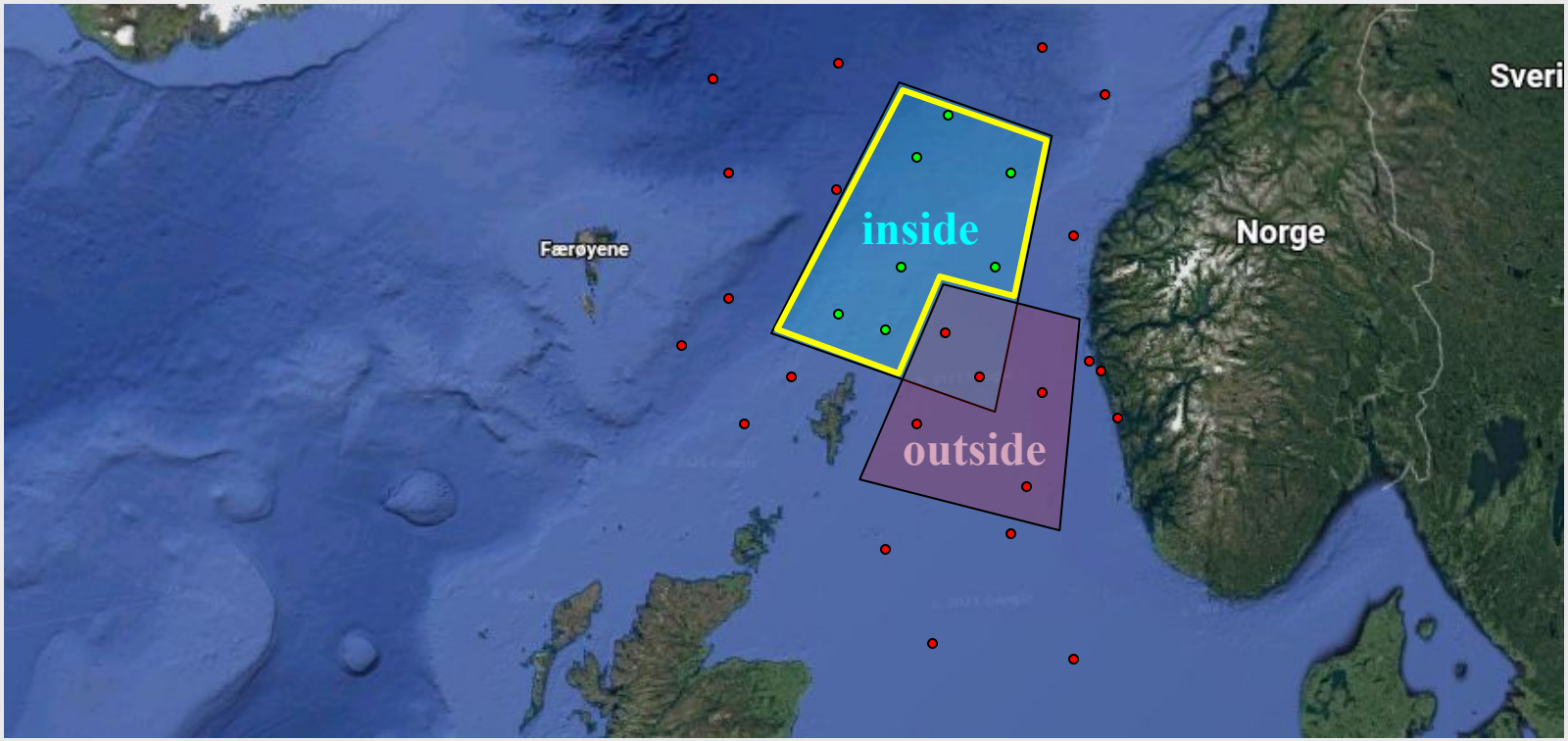
Select from time series type *badevann* only the latest available temperature observations from buoy ids 38 and 20.

```
https://havvarsel-frost.met.no/api/v1/obs/badevann/get?time=latest&latestmaxage=P15D&latestlimit=2&incobs=true&buoyids=38,20
```

Selecting time series and observations (cont'd)

EXAMPLE 5:

Select observations inside and outside specific geo regions (WORK IN PROGRESS!).



Selecting time series and observations (cont'd)

EXAMPLE 5:

Select observations inside and outside specific geo regions (WORK IN PROGRESS!).

```
https://havvarsel-frost.met.no/api/v1/obs/TSTYPE/get?
inside=[REGION,REGION,...]&outside=[REGION,REGION,...]
&...
```

REGION:

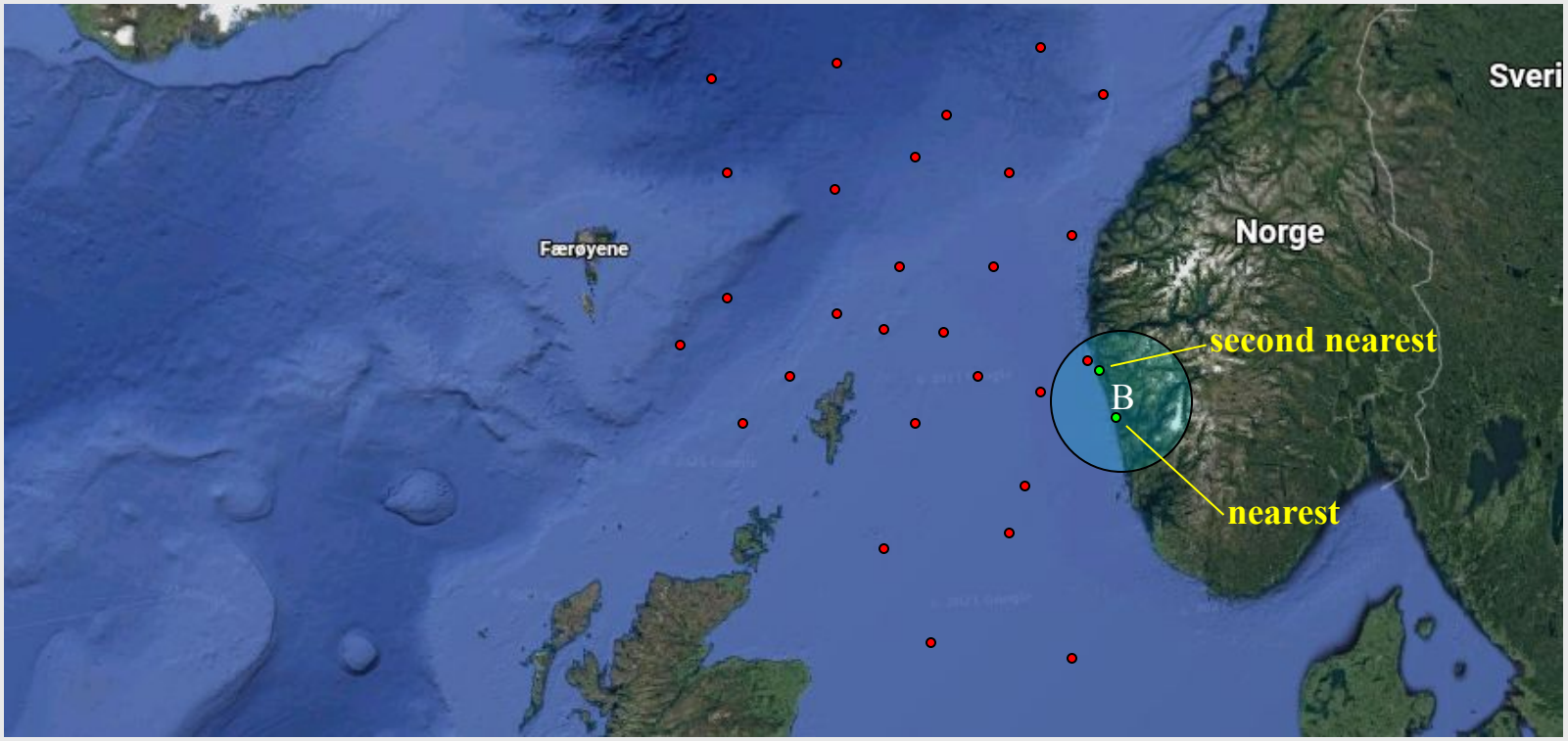
- polygon defined by 3 or more (lat,lon) points
- circle
- latitude range
- longitude range

all regions have a height range (MAMSL) so they effectively define *volumes*

Selecting time series and observations (cont'd)

EXAMPLE 6:

Select observations closest to specific geo positions (WORK IN PROGRESS!).



Selecting time series and observations (cont'd)

EXAMPLE 6:

Select observations closest to specific geo positions (WORK IN PROGRESS!).

```
https://havvarsel-frost.met.no/api/v1/obs/TSTYPE/get?
nearest=NEARESTSPEC&...
```

NEARESTSPEC:

- maxdist
- maxcount
- height range
- points: (lat1,lon1,height1), (lat2,lon2,height2), ...

Norwegian Meteorological Institute

Python example 1

Select from time series type *badevann* only headers (i.e. no observations) for buoy ids 38 and 20.

```
#!/usr/bin/env python3

import requests, json

r = requests.get(
    'https://havvarsel-frost.met.no/api/v1/obs/badevann/get?'
    'buoyids=38,20&incobs=false')

if r.status_code != 200:
    raise Exception('request failed; status code={}'.format(r.status_code))

print(json.dumps(r.json(), indent=4, separators=(',', ': '), ensure_ascii=False))
```

```
joa@pc4727:~$ python3 example1.py
{
    "data": {
    "tstype": "badevann",
    "tseries": [
        {
            "header": {
                "id": {
                    "buoyid": "20",
                    "parameter": "temperature",
                    "source": "badetassen.no"
                },
                "extra": {
                    "name": "Møllebukta",
                    "pos": {
                        "lat": "58.941010",
                        "lon": "5.670380"
                    }
                }
            },
            "observations": null
        },
```

• • •

```
• • •
        {
            "header": {
                "id": {
                    "buoyid": "38",
                    "parameter": "temperature",
                    "source": "badetassen.no"
                },
                "extra": {
                    "name": "Sjøsanden",
                    "pos": {
                        "lat": "58.018192",
                        "lon": "7.445817"
                    }
                }
            },
            "observations": null
        }
    ]
    }
}
```


Norwegian Meteorological Institute

Python example 2

Select from time series type *badevann* all temperature observations from buoy ids 38 and 20 in October 2021, and output each time series as (time, temperature) lines.

```
#!/usr/bin/env python3

import requests

r = requests.get(
    'https://havvarsel-frost.met.no/api/v1/obs/badevann/get?'
    'buoyids=38,20&incobs=true&time=2021-10-01T00:00:00Z/2021-11-01T00:00:00Z')

if r.status_code != 200:
    raise Exception('request failed; status code={}'.format(r.status_code))

for ts in r.json()['data']['tseries']:
    print('\nheader: {}'.format(ts['header']))
    print('observations:')
    for obs in ts['observations']:
        print('{},{}'.format(obs['time'], obs['body']['value']))
```

```
joa@pc4727:~$ python3 example2.py
header: {'id': {'buoyid': '20', 'parameter': 'temperature', 'source': 'badetassen.no'},
'extra': {'name': 'Møllebukta', 'pos': {'lat': '58.941010', 'lon': '5.670380'}}}
observations:
2021-10-06T12:51:38.245Z,11.9
2021-10-06T13:21:39.938Z,12.2

• • •
2021-10-31T13:55:38.248Z,10.2
2021-10-31T17:25:57.586Z,10.4
2021-10-31T21:56:05.418Z,10.5

header: {'id': {'buoyid': '38', 'parameter': 'temperature', 'source': 'badetassen.no'},
'extra': {'name': 'Sjøsanden', 'pos': {'lat': '58.018192', 'lon': '7.445817'}}}
observations:
2021-10-11T12:54:34.358Z,16.8
2021-10-12T12:54:15.415Z,17.0

• • •
2021-10-17T18:54:01.722Z,-0.9
2021-10-18T00:54:01.293Z,-2.7
2021-10-18T06:54:01.027Z,0.3
```

Security

- authentication
- authorization

implemented in Frost v0, still not complete for Frost v1 (a pragmatic solution for write/read access is implemented)

Performance

- caching
- rate limiting

implemented in Frost v0, still not done for Frost v1