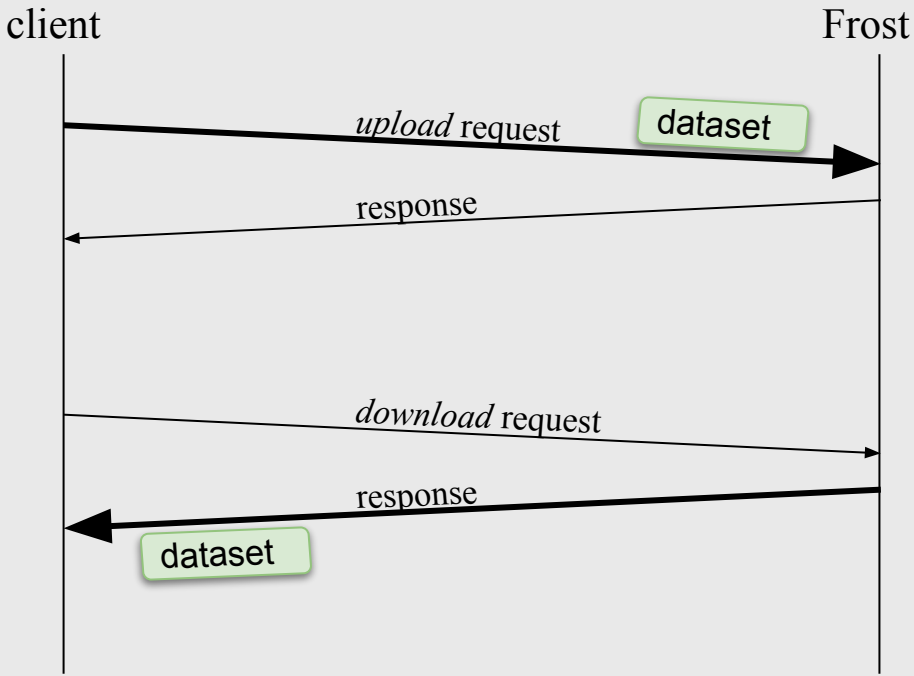
Sending observations to MET via Frost

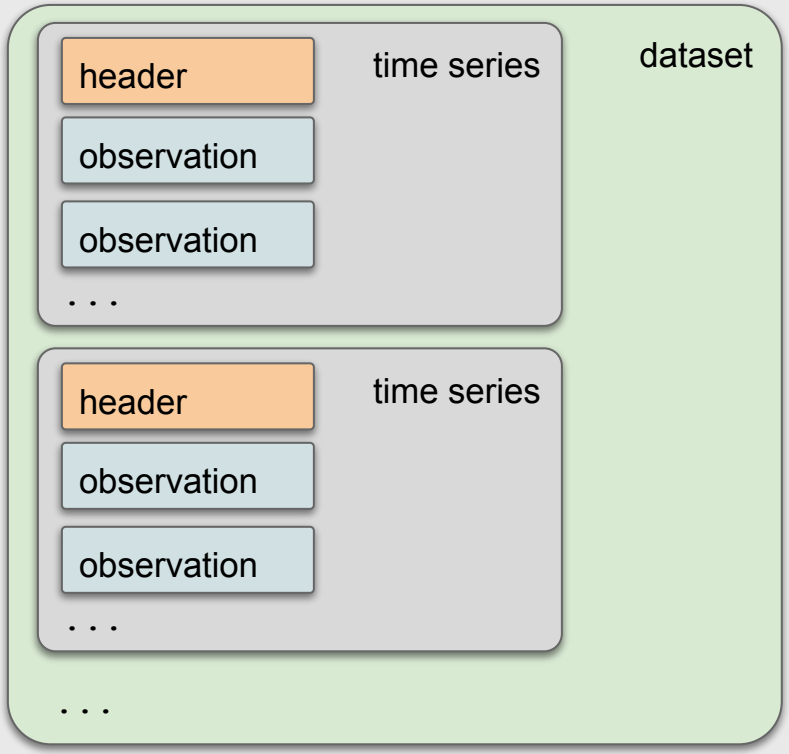
Request / response



- 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

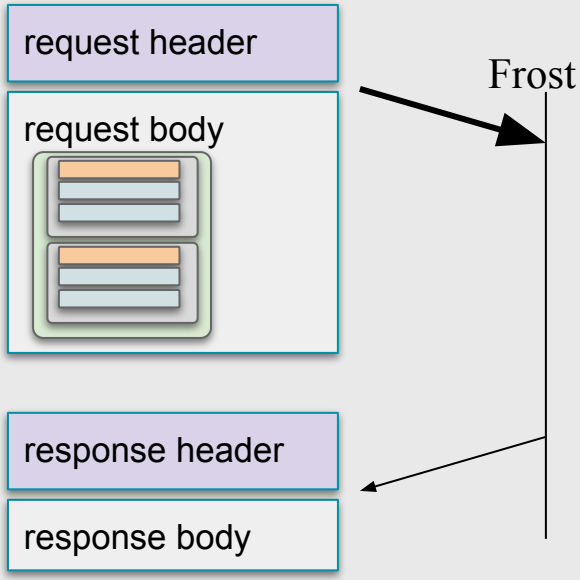
Overall dataset structure



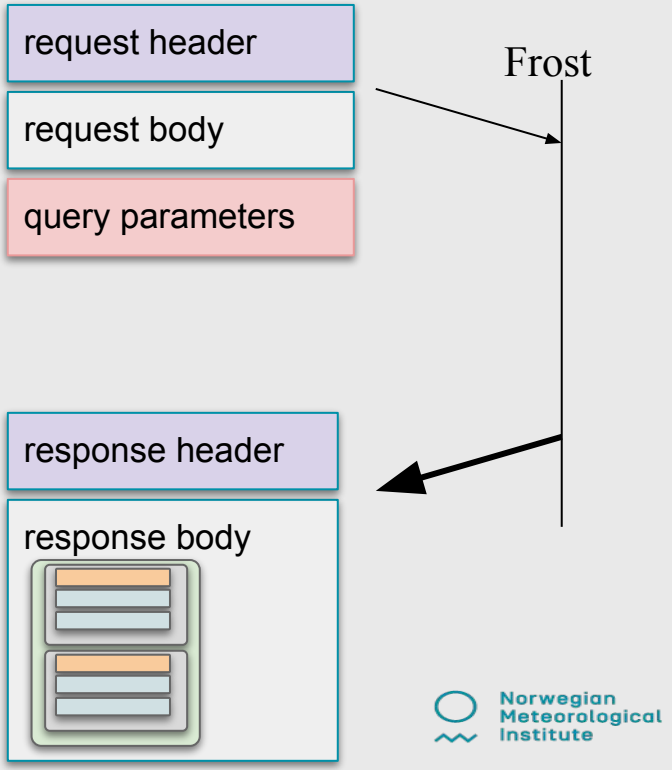
Norwegian Meteorological Institute

Datasets passed over HTTPS

Upload with HTTPS/POST requests:



Download with HTTPS/GET requests:



Norwegian Meteorological Institute

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>}
        },
        ...
    ]
}
```

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

Example: Avinor

```
{
    "tstype": "avinor-awos-xml",
    "tseries": [
        {
            "header": {        NOTE EMPTY HEADER!
                "id": {}
            },
            "observations": [  NEXT PAGE! ]
        }
    ]
}
```

Norwegian Meteorological Institute

Example: Avinor (cont'd)

```
"observations": [
    {
        "time": "2024-06-10T12:00:00Z", E.G. PUBLISH TIME
        "body": {
            "value": "<gzip'ed base64 encoded XML doc>"
        }
    },
    ...
]
```

```
<?xml version="1.0"encoding="utf-16"?>
<AwosSensorDataMessage ...>
<ambiHeader> ... </ambiHeader>
<sensorMeasurement> ... </sensorMeasurement>
<sensorMeasurement> ... </sensorMeasurement>
...
```

Norwegian Meteorological Institute

Avinor - issues

- Access via firewall whitelisting
  - MET needs client IP from Avinor
- Return values from Frost's /put endpoint
  - Avinor needs to handle these:
    - 200 Ok
    - 500 Internal Server Error
    - 503 Service Unavailable
    - 400 Bad Request
- Need XML schema + policy for version updates
- XML doc must be well-formed
  - attributes must be quoted etc.
-

Avinor - Python example

```
...
# read command-line args
frost_url_base, obs_file, obs_time = parse_args(sys.argv[1:])

# create dataset
obs_val = create_obs_val(obs_file)
obs = {
      'time': obs_time,
      'body': {
            'value': obs_val,
      }
}
dset = {
      'tstype': 'avinor-awos-xml',
      'tseries': [
            {
                  'header': {
                  'id': {}
                  },
                  'observations': [obs],
            }
      ],
}
```

Norwegian Meteorological Institute

```
# issue request

url = '{}/api/v1/obs/avinor-awos-xml/put'.format(frost_url_base)

response = requests.post(url, files={'dataset': json.dumps(dset)})

if response.status_code == 200:  # success, so dump response body to valid and pretty-printed JSON
      print(json.dumps(response.json(), indent=4))
else:  # dump error details
      print(f'status code: {response.status_code} ({response.reason})')
      print(f'response content: {response.content}')
```

Norwegian Meteorological Institute

Write authentication

- IP whitelisting
  - must write from these IPs, and that's it

- write tokens
  - can write from any IP, but must provide a write token

Ingest service for existing project

havvarsel-frost.met.no

Exploring Swagger UI (based on OpenAPI spec) for time series types *badevann*, *glider*, and *vertical-profile*.

Norwegian Meteorological Institute

General ingest service

frost-ingest.met.no (for example)

General ingestion from external sources, like Avinor, NMBU etc.

Under development!