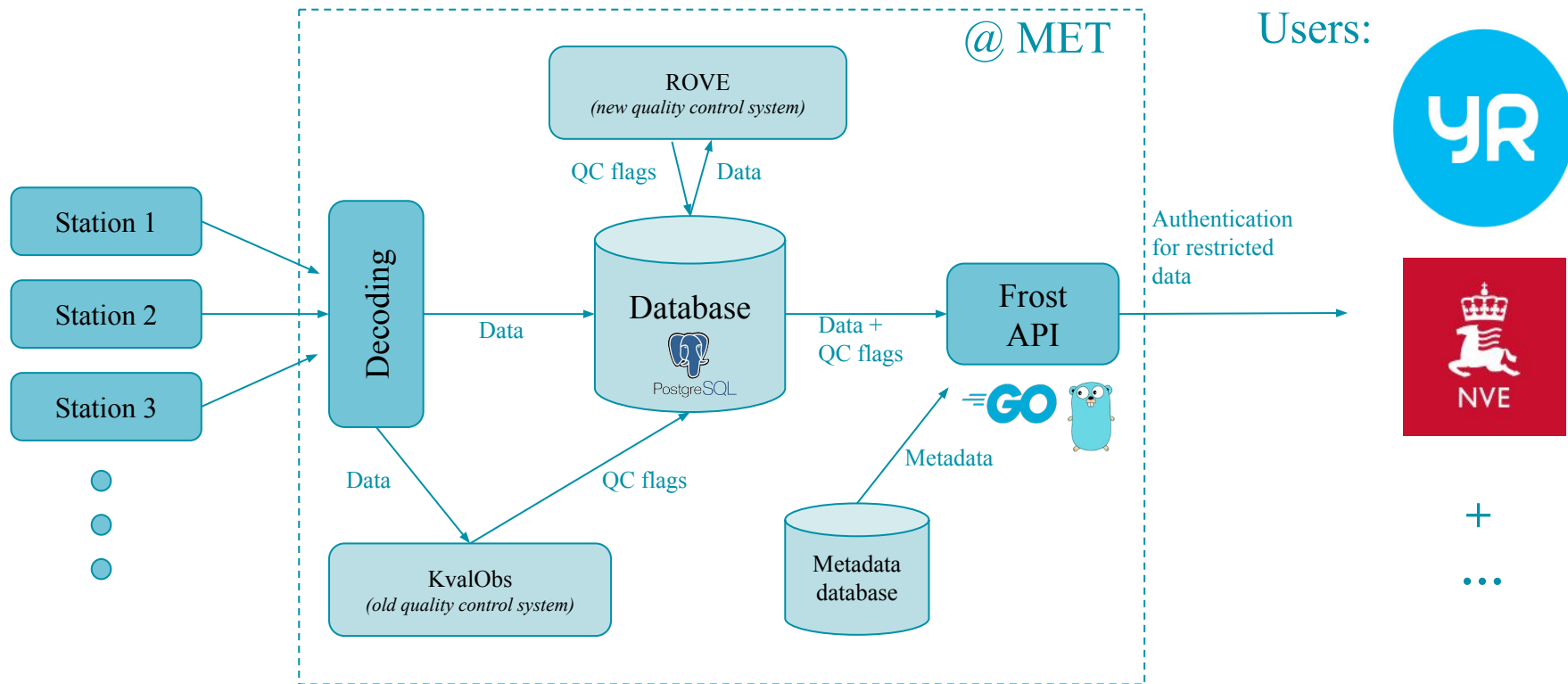# Renewal of Quality Control and Ground Observation Storage at MET Norway

Storage, Quality control, and public API

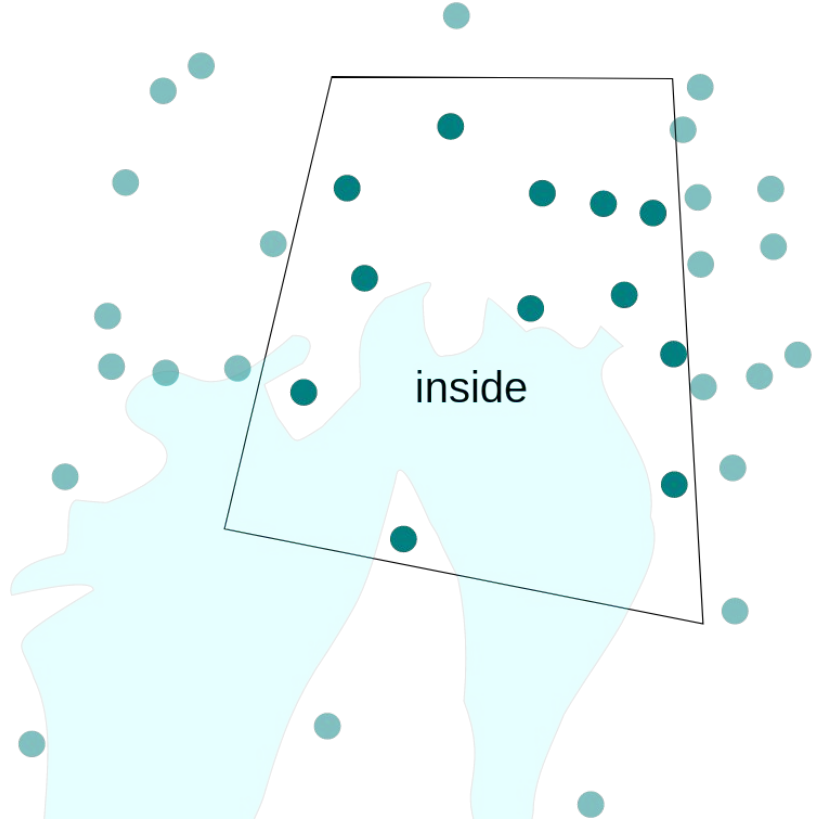05.11.2025

Data flow

Norwegian Meteorological Institute

Storage of data

- Ways people search/get data:
  - Climate: a time series back in time as far as it exists
  - Now: recent observations in an area

- Plan to better support the use case:

  Newest data over a given area
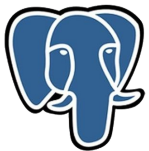   - Spatial QC
    - Yr

inside

Database

Live Atmospheric Readings Database: https://github.com/metno/lard

Database = an organized collection of data
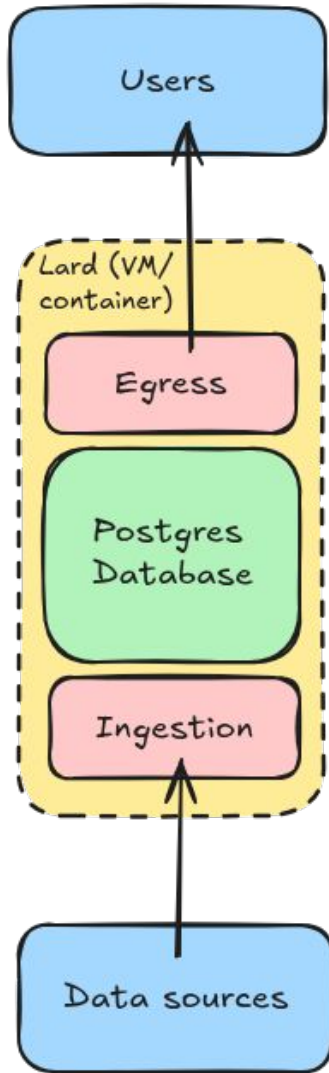
Replicated Postgres:
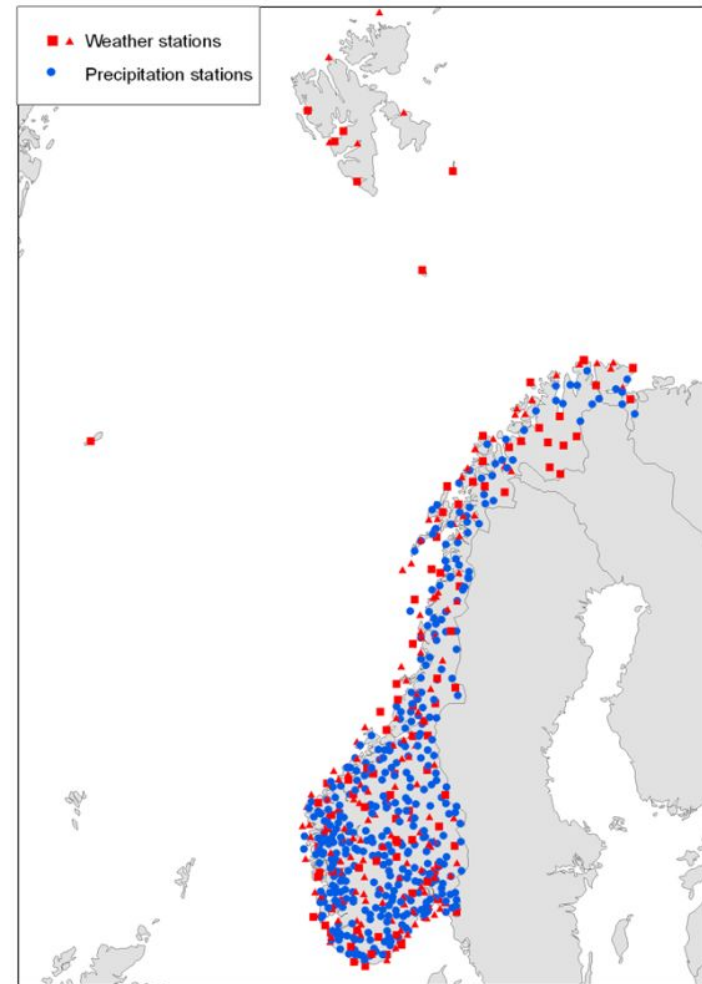- Can be scaled up/down (high availability)
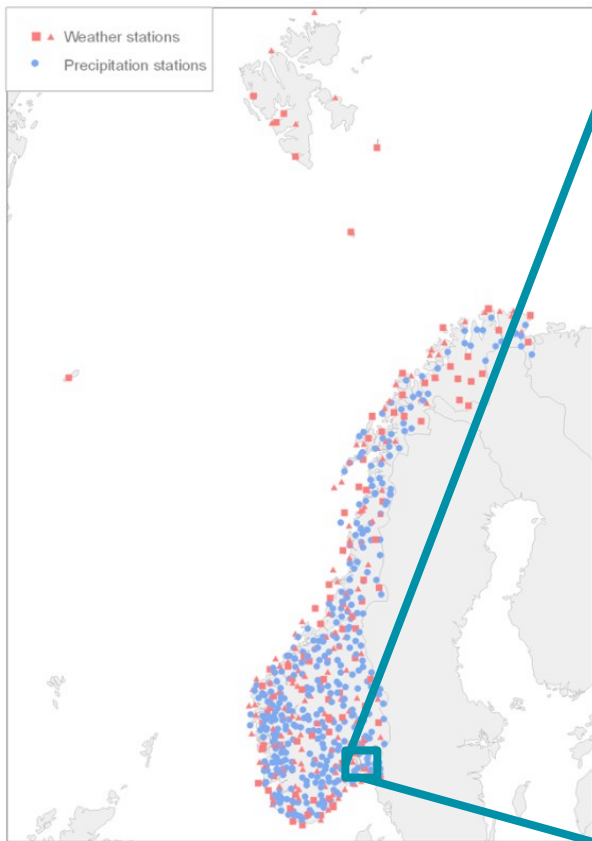
PostgreSQL

RUST

Stations in Norway

Approximately:
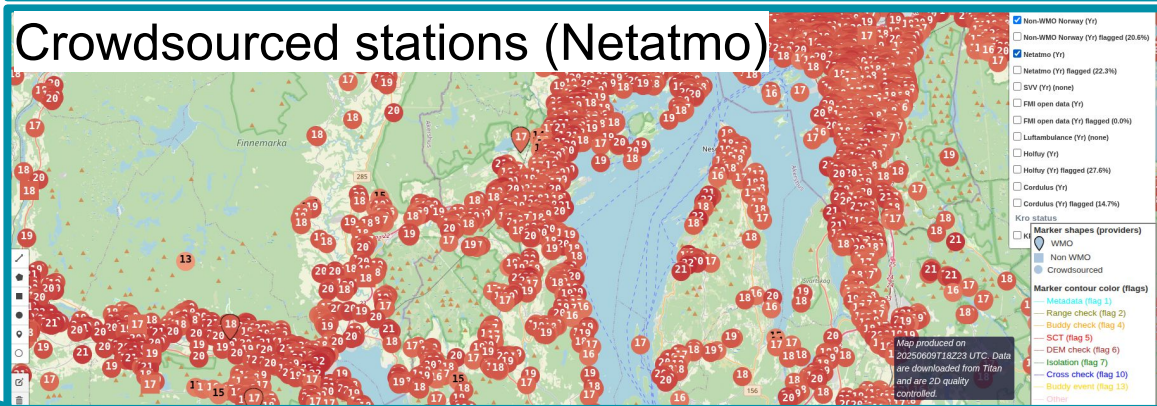- 400 MET stations
- 1000 external
- Many times that crowdsourced

Norwegian Meteorological Institute

Stations in Norway

MET stations + partner stations

Crowdsourced stations (Netatmo)

Trends in observations

We may have more data, new sensor types, more frequent

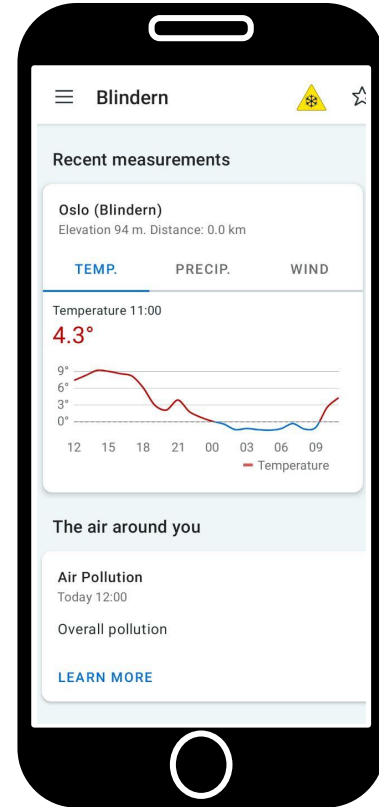We have new types of data

But it has more ways to fail
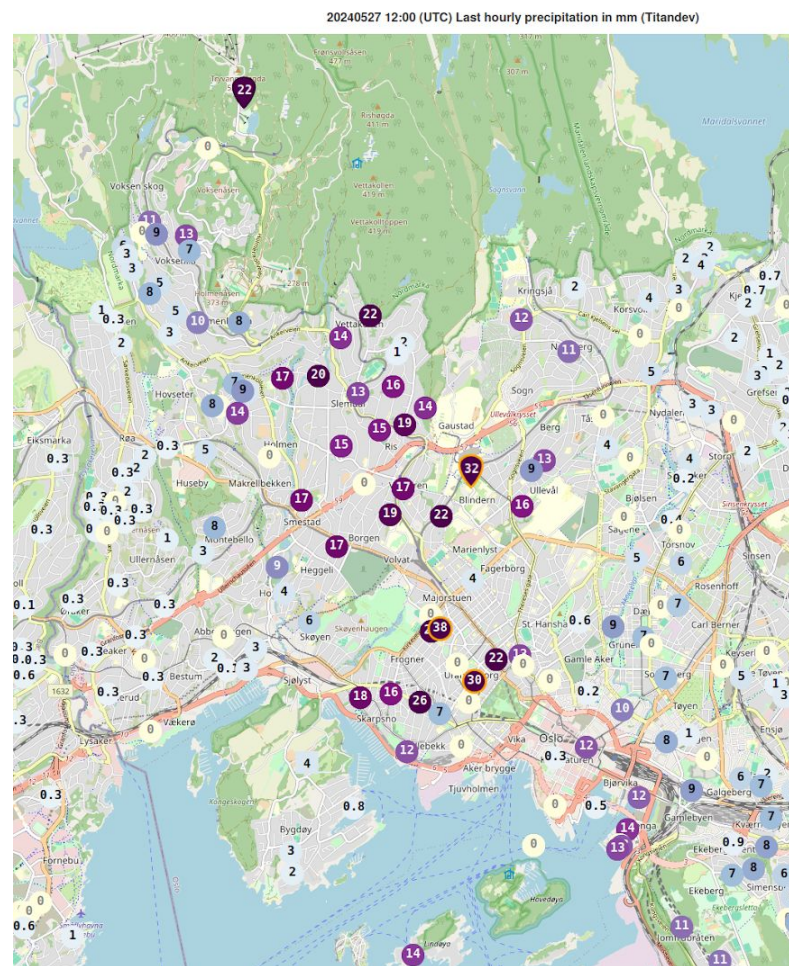
We need data in real-time for downstream applications
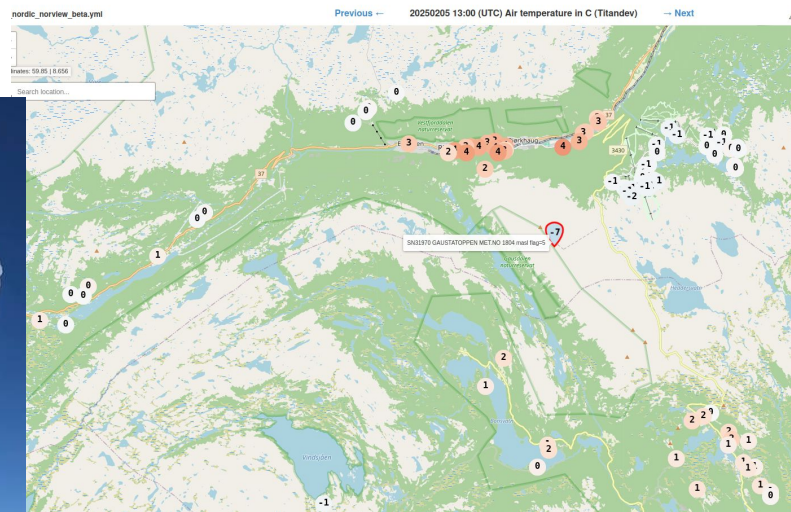
Why quality control?

Quality control

**Automatic quality control** needs algorithms that try to find problems with sensors, unlikely data values, or data that is inconsistent with other observations in the area
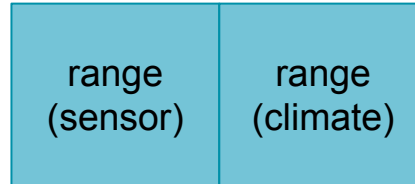
Quality control

- Have historically relied on algorithms that look at one individual time series

- Development of spatial tests (as amount of data sources increased)

One test alone may not tell the whole story, need to develop sensible pipelines…
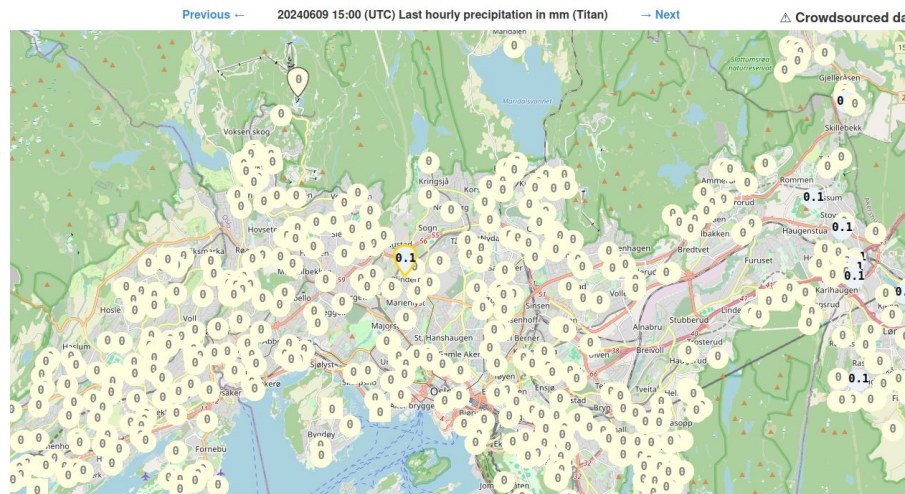
Example quality control pipelines

Single observation →

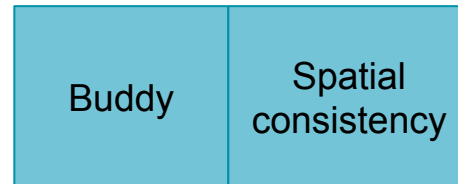| range (sensor)	range (climate)

Time series
(most recent observation + some previous observations) →

Step	Spike	Flatline
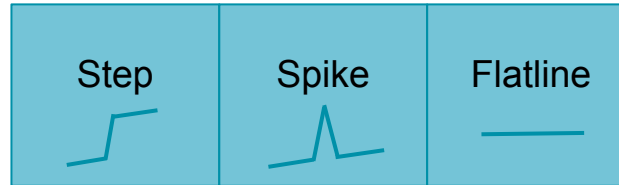
Spatial
(short time range for a defined area) →

Buddy	Spatial consistency
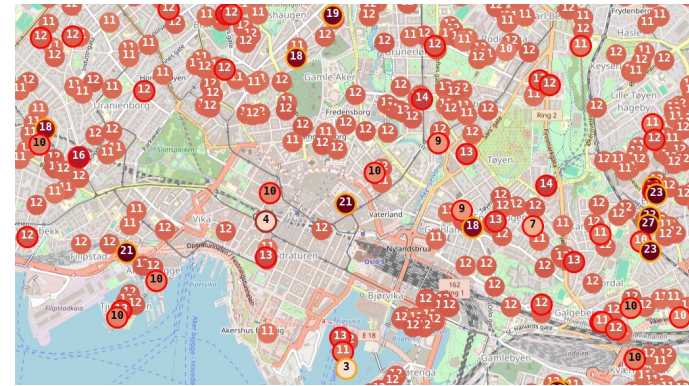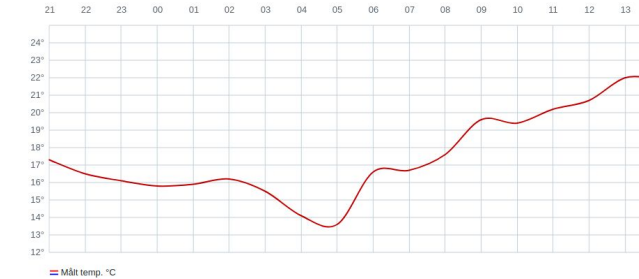
New quality control system

Real-time Observation Validation Engine: https://github.com/metno/rove

4 types of tests:

- **single:** tests run on a single data point

- **series:** tests run on a set of values from a time series

- **spatial:** tests run on an area with many observations

- **consistency:** compare between tests, compare to weather model, etc.

Code for the QC tests: https://github.com/intarga/olympian

Architecture

ROVE

Data coming in to MET Norway

Scheduler
Schedules QC tests routes data between the switch and harness

Triggers QC request

Ingestor
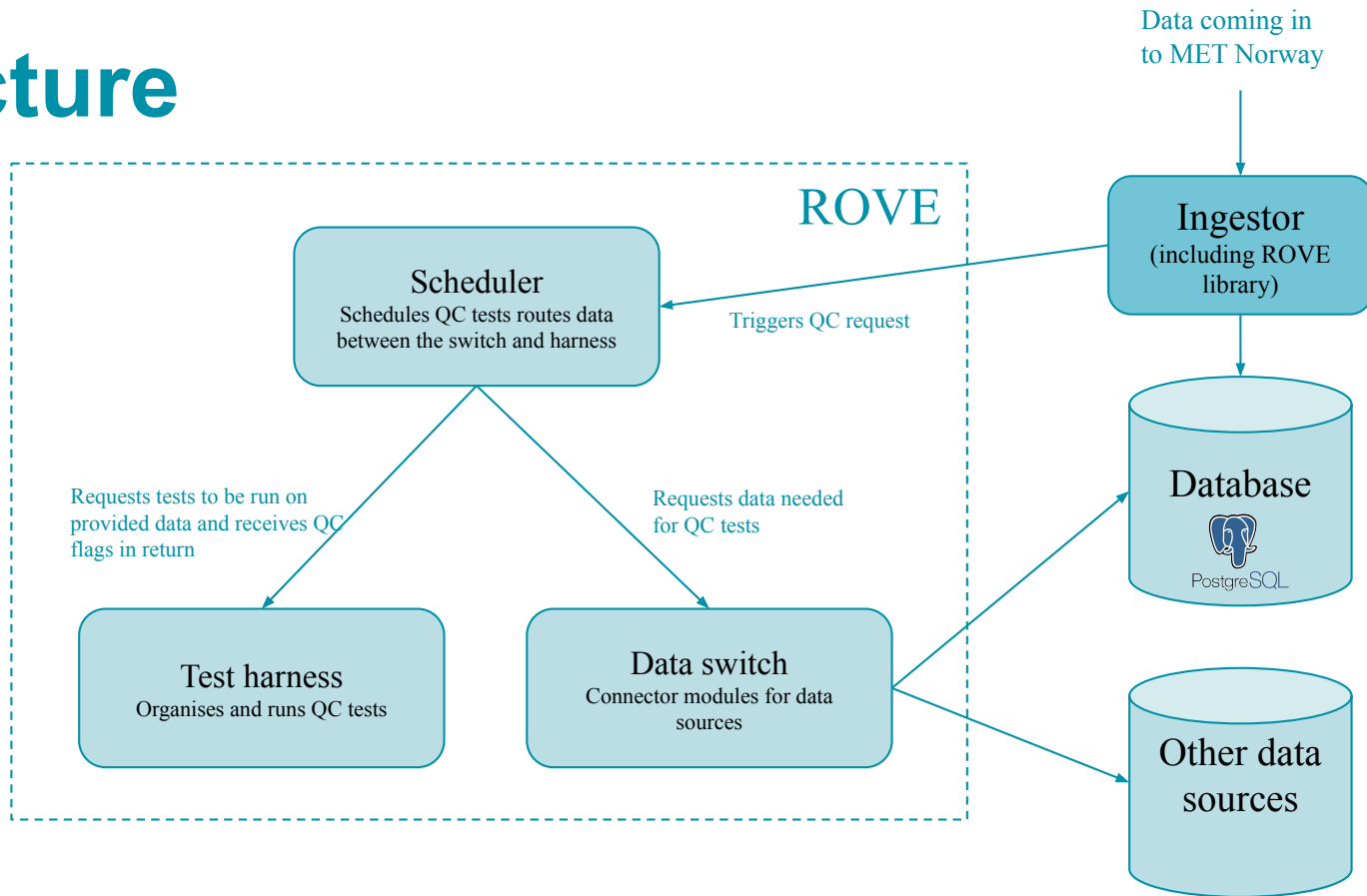(including ROVE library)

Requests tests to be run on provided data and receives QC flags in return

Requests data needed for QC tests

Test harness
Organises and runs QC tests

Data switch
Connector modules for data sources

Database
PostgreSQL

Other data sources

Louise Oram

Telephone: +47 459 50 275
E-mail: louise.oram@met.no

Norwegian
Meteorological
Institute

Other links

https://crates.io/crates/rove

https://github.com/metno/titanlib

API

FROST: https://frost-beta.met.no/

- For internal and external users
- CF standard names for weather elements (https://frost.met.no/elementtable)
    - air_temperature
    - max(air_temperature PT1H)
- Filtering by various metadata
    - for example nothing below a certain quality code
- Geosearch
    - find the nearest stations: nearest(POINT(10 59))

Norwegian Meteorological Institute