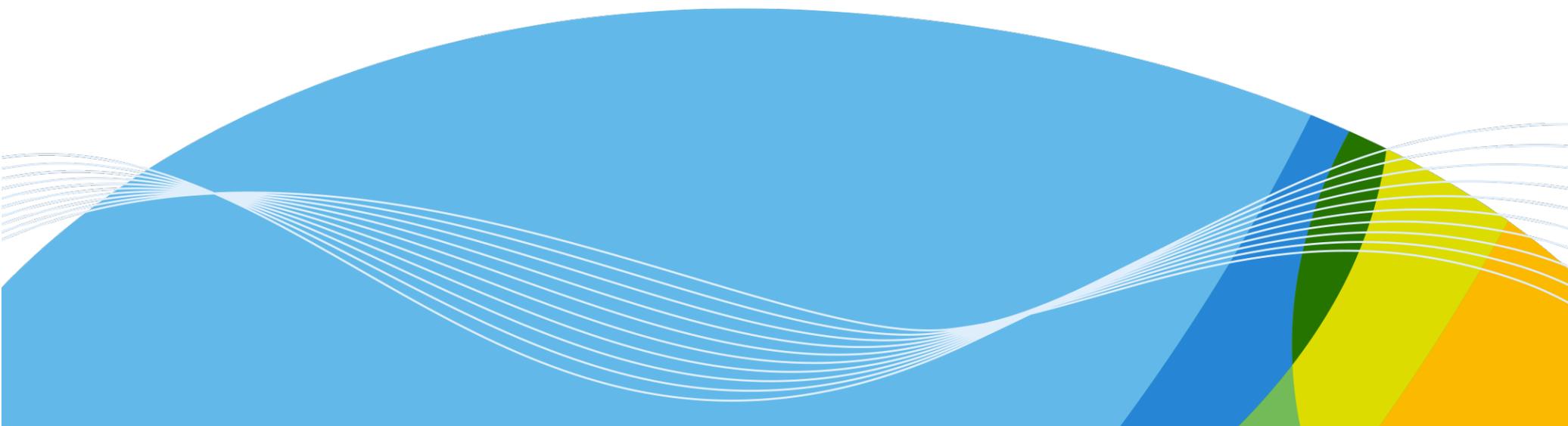




FINNISH METEOROLOGICAL INSTITUTE

MetOcean Data Server Brainstorm

Roope Tervo, Mikko Rauhala, Tarja Riihisaari, Mika Heiskanen, Tuomo Lauri, Santeri Oksman
Finnish Meteorological Institute





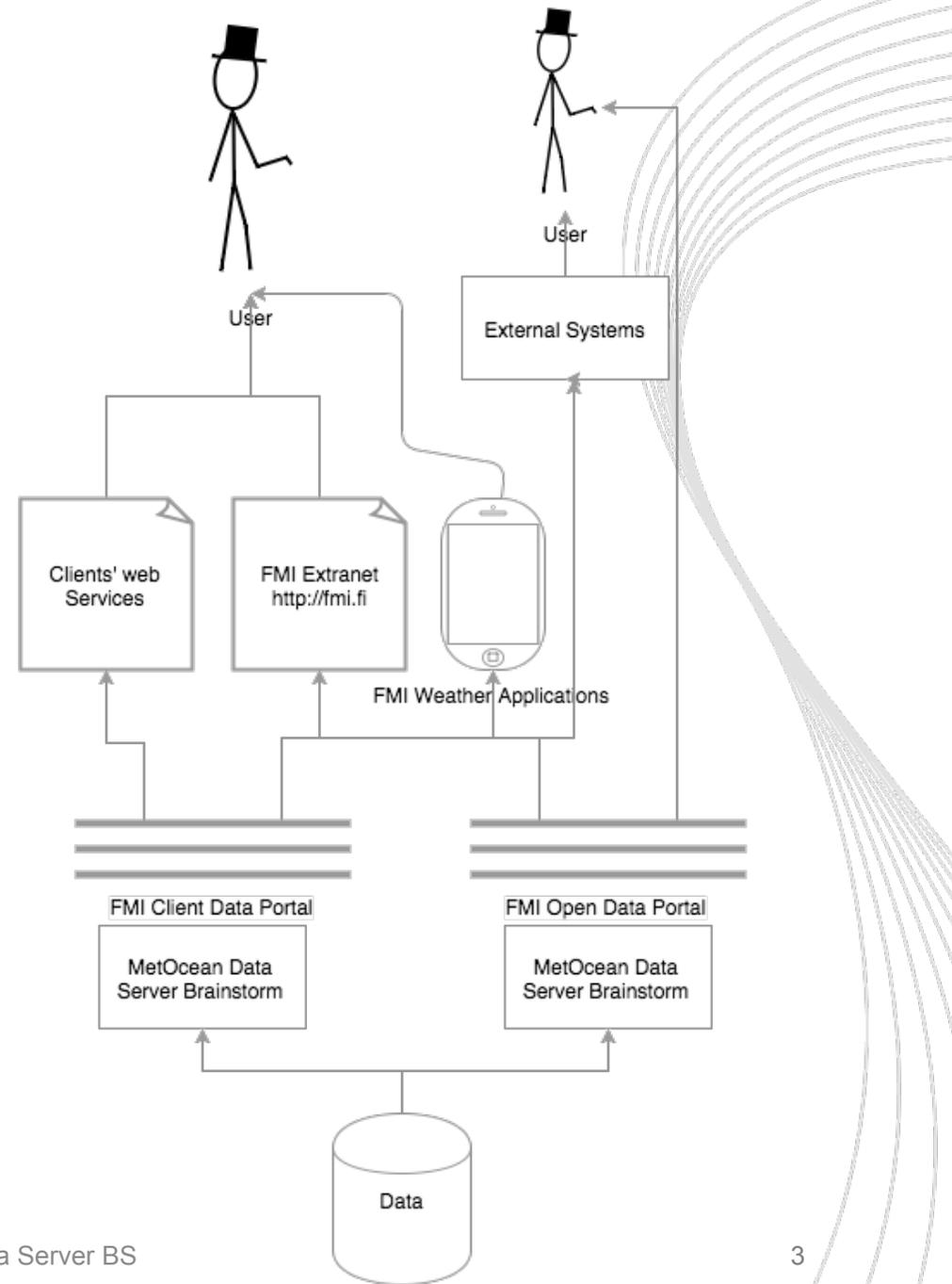
In a Nutshell

- Data and product server for MetOcean data
- Produced by FMI to meet requirements of web based services
- High capacity & availability
- Data is extracted and products generated always on-demand
- Several output formats
 - WMS, WFS 2.0
 - JSON, XML, ASCII, HTML, SERIAL
- INSPIRE Compliant
- Operative since 2008
 - FMI client services (since 2008)
 - Finnish Meteorological Institute (FMI) Open Data Portal (since 2013)



Usage

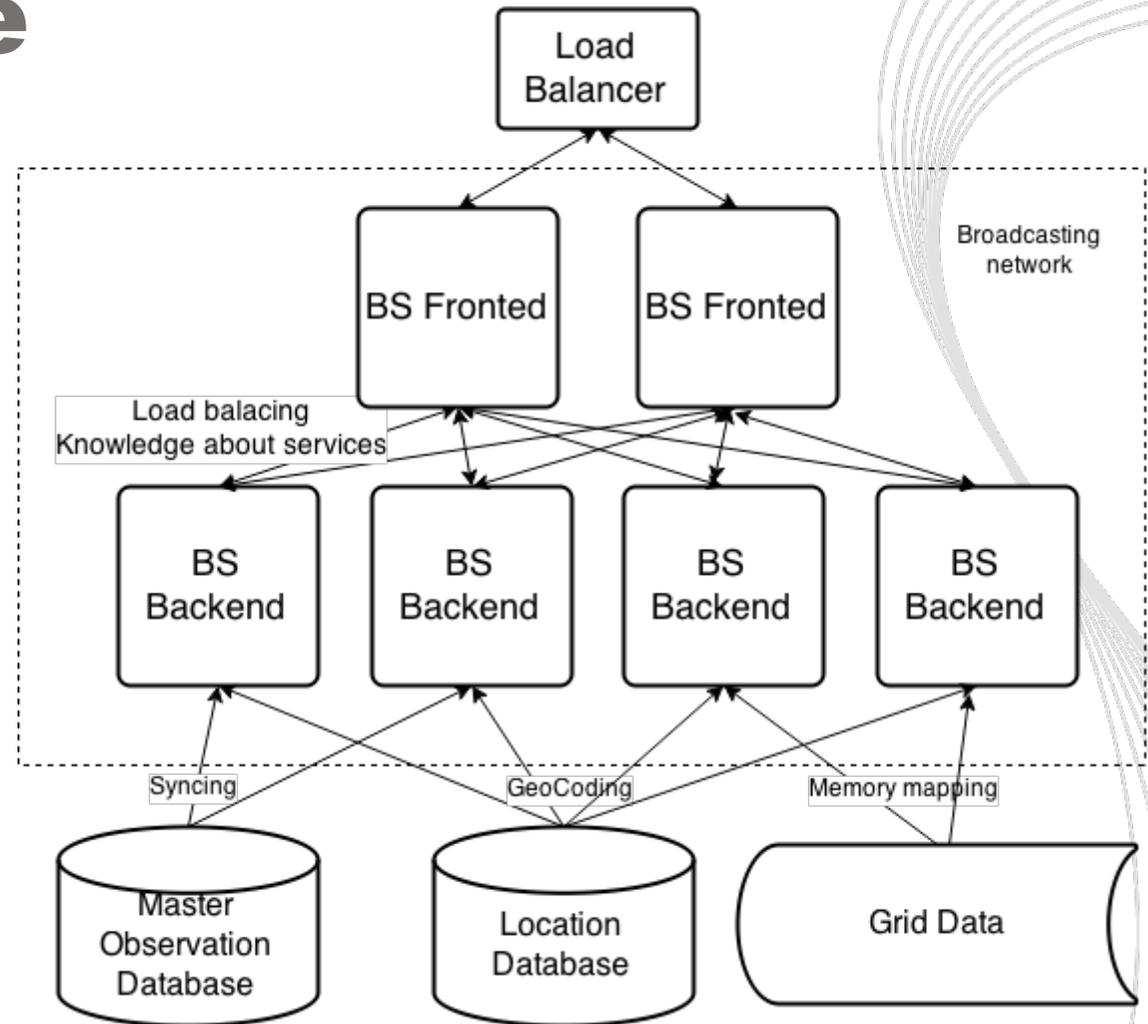
- Basis of FMI product generation





Architecture

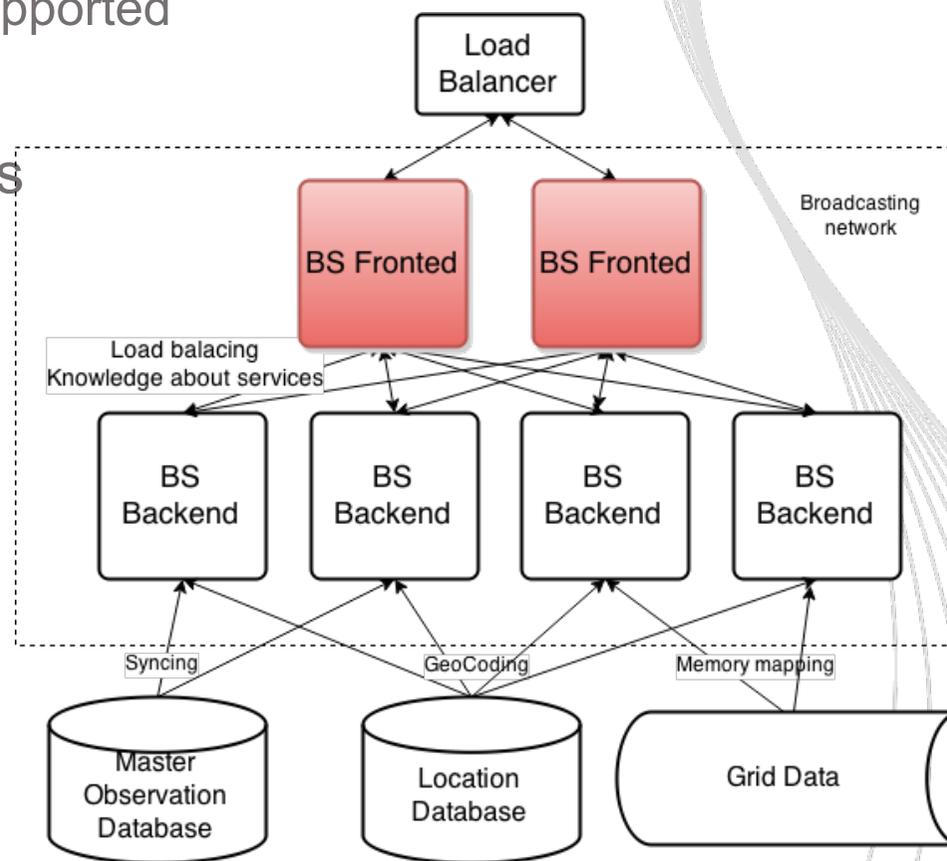
- Frontend
 - Load balancing with service discovery
 - Product cache
- Backend
 - Different backends may contain different services
- Plugin-based architecture
 - *Engines* provide shared access to the data
 - *Plugins* provide different APIs based on data provided by the engines





Frontend

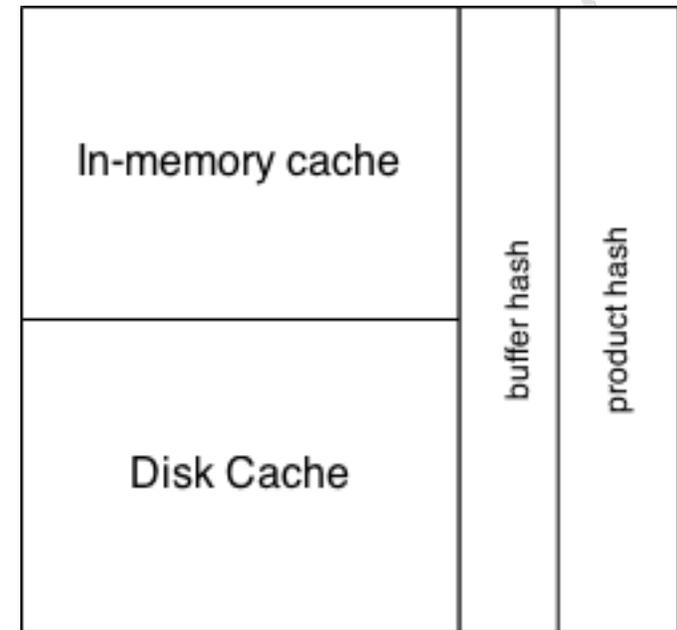
- Provides HTTP 1.1 server
 - Chunked, gzip and no encoding supported
 - No full standard compliance
- Monitors status of backend services
- Forwards requests to the backends with required services
 - Load balancing
- Different queues for fast and slow services
- Provides LRU product cache





Frontend cache

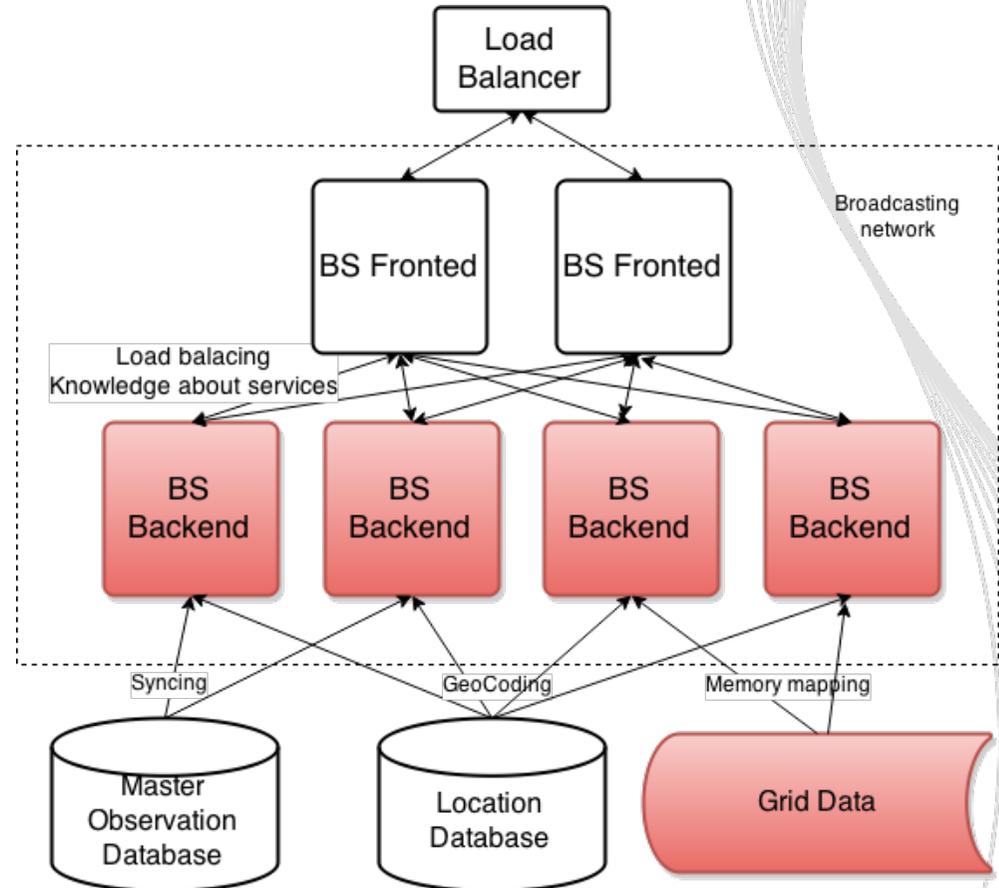
- Cache contains two phase mapping of product
 - ETags works as product hash
 - Consists of request url, product configuration and used data
 - If any of above changes, product is deprecated
 - ETags are mapped to buffer hash
 - Data is stored based on buffer hash
 - For example transparent radar images without rain are all mapped to as one cache entry
 - Significant effect to cache size
- LRU cache
 - Most used content in memory
 - Rarely used content on local disk





Data Sources qengine

- Provides access to grid data
- Supported data formats: querydata
 - Ready tools for converting other formats (grib, netcdf, hdf...) to querydata
- Memory-maps the data from NFS
- Supports both spatial and temporal interpolation and nearest point selection
 - Used method depends on the parameter
- Selects the best data source for requested region

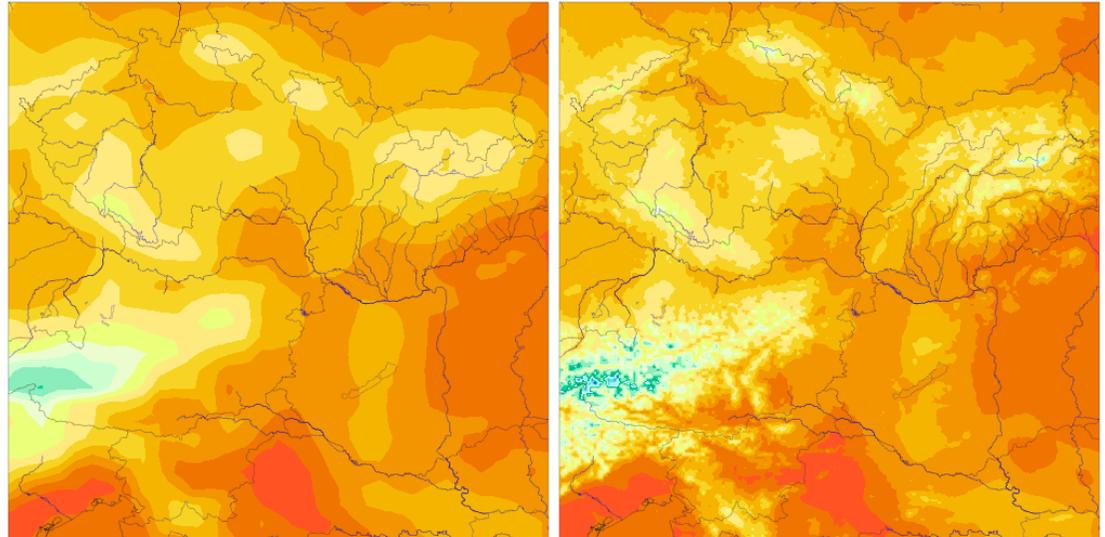




Data Sources

qengine

- Post-processing capabilities
 - Corrects the data based on accurate DEM (up to 30 meter resolution) and land/water information
 - Correlation done to temperature based on the difference between model and real topography
 - Land/water information used to give more weight on corresponding grid points in interpolation
 - Calculates derivative parameters
 - FeelsLike, sunset, day length...

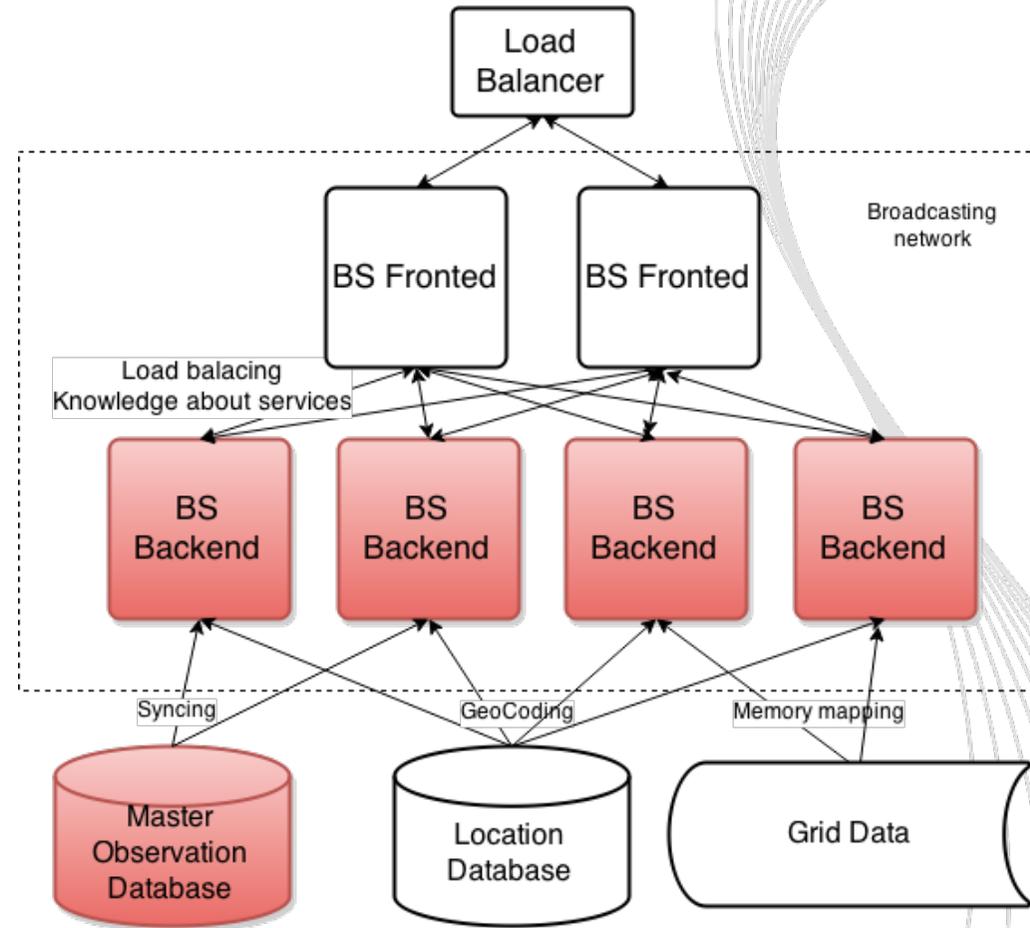




Data Sources

obsengine

- Provides access to observation data stored in SQL database
- Keeps local SpatiaLite database for latest data
- Sync with the master database
- Requests not available in the cache are forwarded to master database

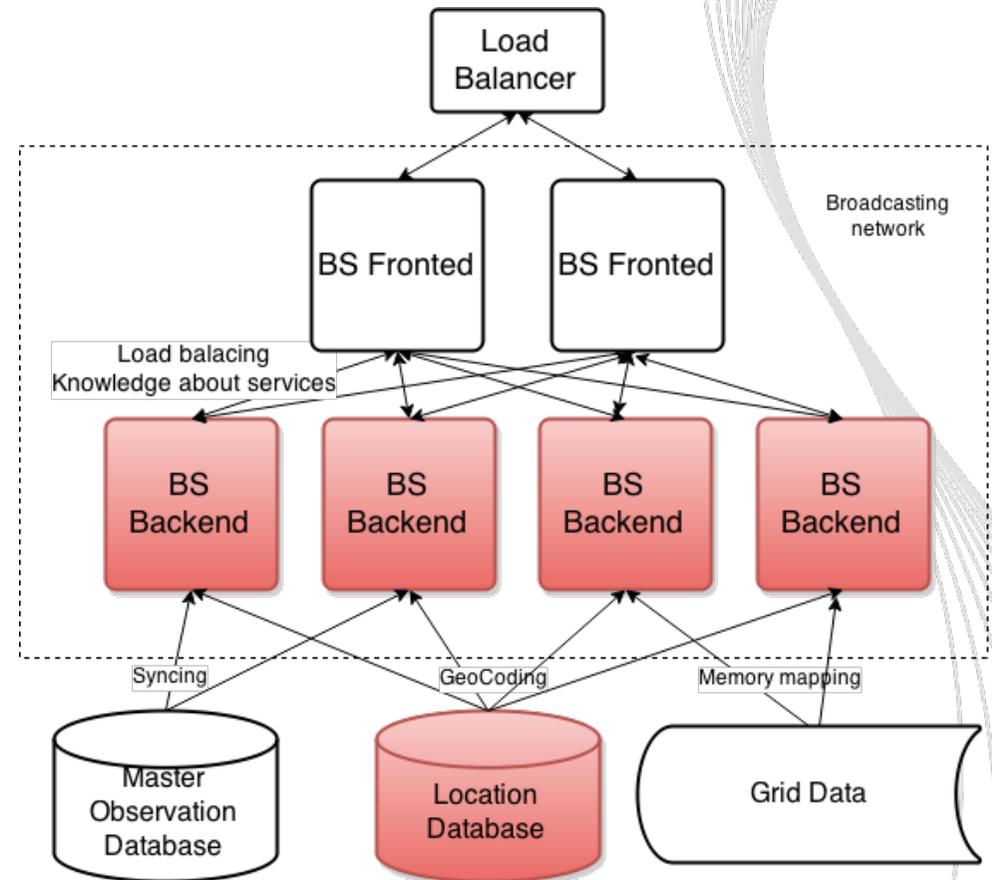




Data Sources

geoengine

- Provides geolocation services
 - Resolve location information for coordinates and vice versa
- Based on geonames.org
- PostGIS database
- Synced from geonames.org to local database once a day





WMS

- Version 1.3.0
- Supports png, svg
- Supports all proj.4 projections
- Supports for time and other custom dimensions
- Data sources: grid data and PostGIS
 - Contours coverages and isolines on-fly from grid data (qengine)
 - Draws PostGIS geometries based on database queries
- Generates SVG which are rendered to requested raster format
 - Supports rich effects of SVG rendering model
 - Styling with CSS
 - Rendering done with libsvg
- Configuration with json
 - Configuration variables can be overridden in requests (i.e. thresholds)



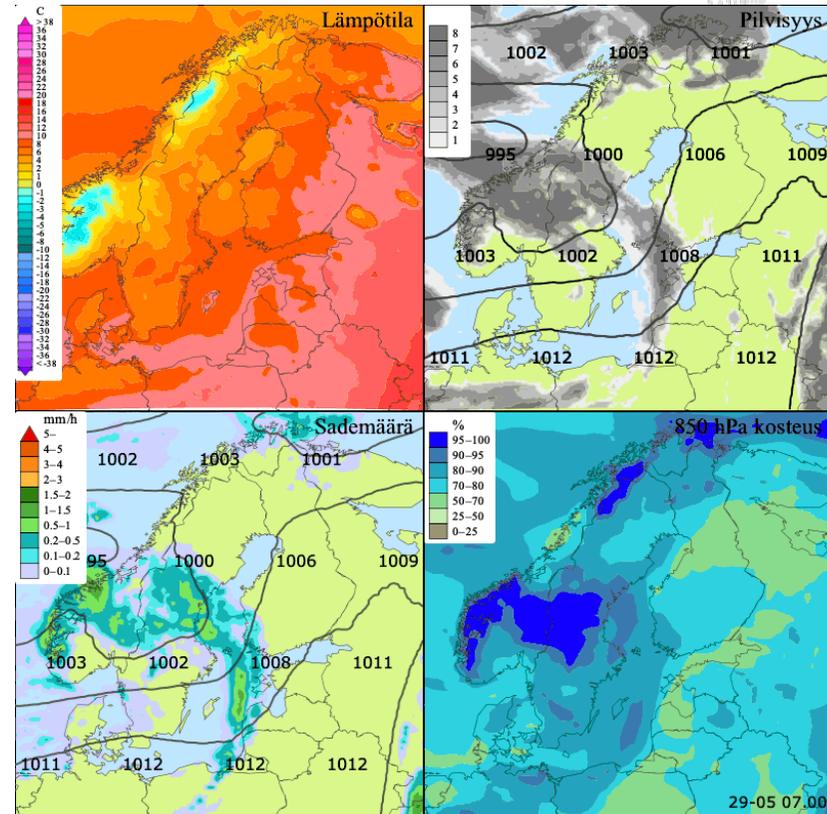
WMS

- Support for meteorological specific visualizations
 - Wind barbs, wind arrows, fronts...
- Support for thresholds
 - i.e. draw images only where temperature is between -2 and 2 degrees
 - Also possibility to mask content based on other parameter (i.e. draw precipitation areas only where wind gusts are over 8 m/s)
- Very efficient cache
 - Supports frontend product cache
 - Several LRU caches in backend
 - For contours, SVG products and ready rendered products



Dali

- Provides generic images from grid data
 - WMS is usage Dali plugin
 - Richer but non-standard API
- Support for several views in one image
 - i.e. same area with different times side by side
- Support all SVG effects
- Possible to center location based on location name





WFS

- Version 2.0.0 Basic Profile
- INSPIRE Compliant
- Support all proj.4 projections
- Capabilities: *ImplementsQuery, ImplementsAdHocQuery, ImplementsMinStandardFilter, ImplementsMinSpatialFilter, ImplementsMinTemporalFilter*
- Supports KVP- and XML-encodings
- Support for simple and complex features
- Data sources: grid data (qengine) and observation data (obsengine)
 - Point forecasts from grid data
 - Contours from grid data
 - Observations from obsengine
- Most of the data provided via stored queries to access good enough performance



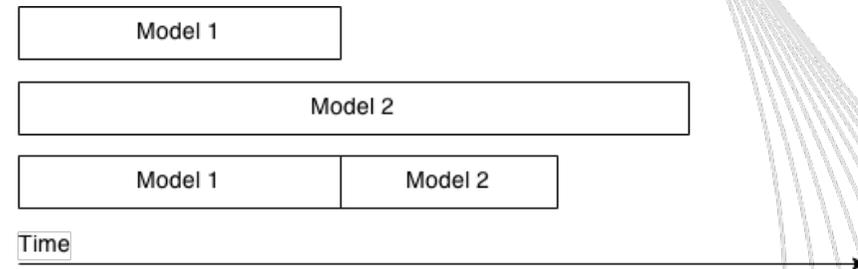
Download (WCS-like)

- Provides grid data (from qengine) as binary data
- Supported output formats: *GRIB1*, *GRIB2*, *NetCDF* and Querydata,
- Supports all proj.4 projections (depends on output format support)
- Supports slicing by
 - area (bbox)
 - elevation (pressure and/or model level)
 - time (start time, end time and origin time)
- Possibility to define grid resolution by
 - selecting every Nth grid point to x and y direction
 - grid size → data is interpolated to new grid points



TimeSeries

- Provides observations and point forecasts to the point
- Output formats: *JSON, XML, ASCII, HTML, SERIAL*
- Support for aggregate values over time and area
 - *Median, mean, min, max, sum, avg, integ, sdev, trend, change, count, percentage*
- Support for combining different models
 - Giving two models cause first one to be supplemented with second one after time range of first one ends
- One or more locations per query
 - Possible selection methods: *name, coordinates, set of locations, location id, bbox, path, WKT*





Textgen

- Provides automatically generated weather forecast texts based grid data
- Supported languages: *Finnish*, *Swedish* and *English*
- Area of interest can be given as area or point and radius
 - Location and area information fetched from PostGIS

Finland

Expected weather until Friday morning:

Rain in some places. Rain can be heavy until late evening. The minimum night temperature is 7...12 degrees, tomorrow the the maximum day is 15...20 degrees. Mainly south-westerly wind 5-10 m/s, that turns mainly west from the evening. The wind turns mainly south from Thursday afternoon.

Expected weather from Friday morning until Saturday morning:

Rain showers in some places. Showers can be heavy in the afternoon and evening. The maximum day temperature is 15...20 degrees, the minimum night 0...+11 degrees. South to south-westerly wind 4-9 m/s, at its strongest 12 m/s.



Q3 (WPS-like)

- Provides service to process the data and return output as data or image
- Input: LUA scripts
- Output formats
 - Matrix as text
 - Matrix as binary (querydata)
 - JSON
 - Contoured images: *svg, png, jpeg, pdf*

```
local param= T
local limit= 0

local r,err= HIR{ hybrid=true, params={param,Z,P} }
assert(r,err)

-- Iterate levels from down to up
-- Store height and pressure when >= 'limit' (last
will remain)

local m_Z= matrix() -- heights collected; originally
all 'nan'
local m_P= matrix() -- pressures collected
local m_v= matrix() -- value at such positions (not
needed)

for g in grids_by_level(r) do
  for pos,v in points(g[param]) do
    if v>=limit then
      m_Z[pos]= g.Z[pos]
      m_P[pos]= g.P[pos]
      m_v[pos]= v
    end
  end
end
return m_Z, m_P, m_v
```



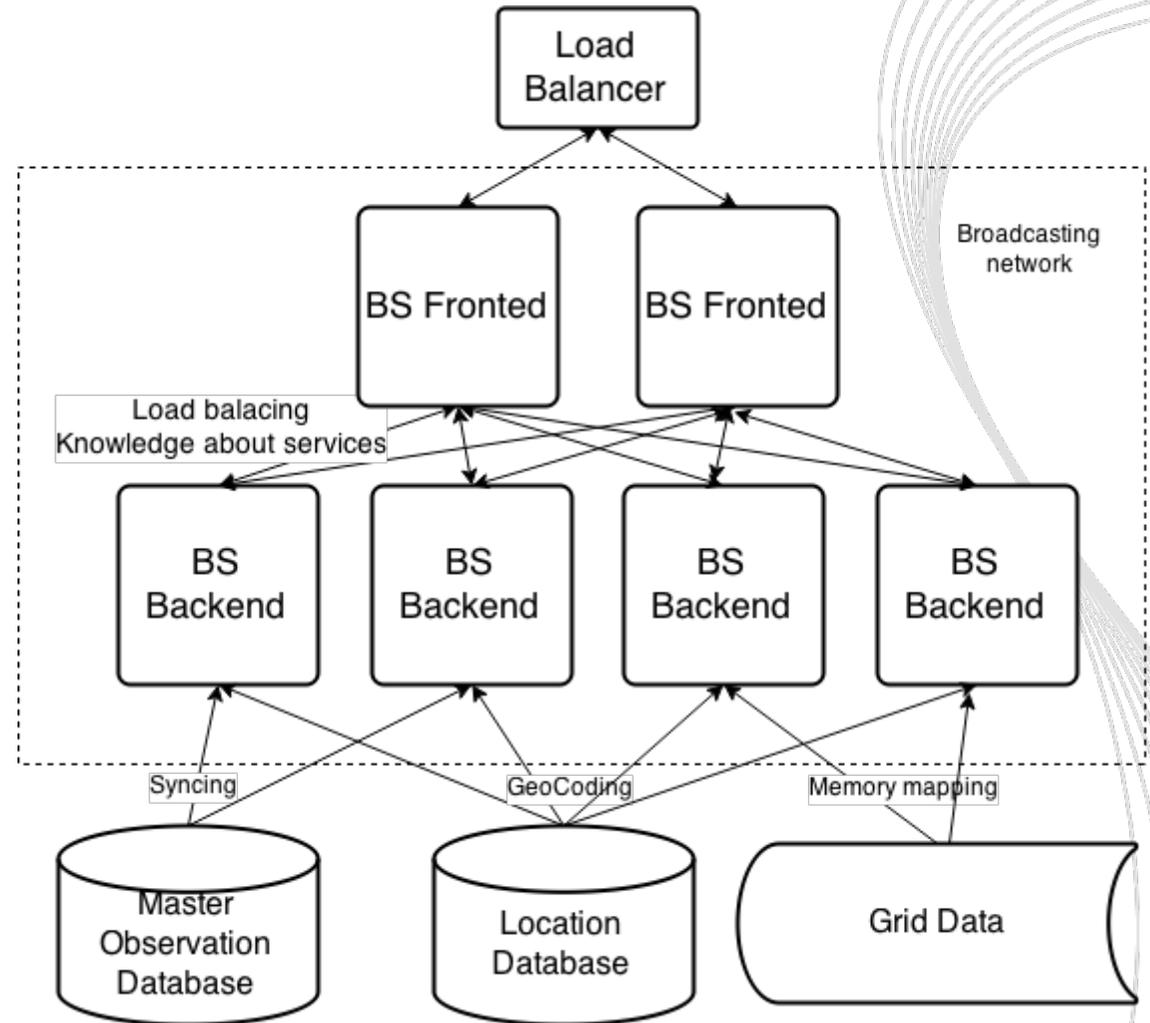
Other Plugins

- *Autocomplete plugin* provides autocomplete server for searching locations
- *Csection plugin* provides cross sections at given time from grid data (qengine)
- *Trajectory plugin* provides trajectory calculation for massless particles in current weather situation



FMI Setup

- 2 frontends
 - RAM: 256G
 - CPU: 24x 2.10GHz
 - OS: RHEL7
- 5 backends
 - RAM: 12G
 - CPU: 24x2.50GHz
 - OS: RHEL7
- Load Balancer
 - F5 BIG IP 11
- Databases
 - Master Observation Database: Oracle
 - Location Database: PostGIS 9.3





Performance

- FMI setup
 - > 30 000 000 req/day
 - Peaks over 650 req/s
 - Average response times varies depending on the request:
 - WFS: 140 ms/req, WMS: 130 ms/req, Timeseries: 30 ms/req, Autocomplete: 4 ms/req
 - Overall average: 70 ms/req
 - Over 99,95% availability
- No limits have been found yet
- Slow query queue easy to DoS due to possibility to large data requests and heavy operations
 - Don't affect to fast query queue



Roadmap for the future

