Visual Weather web services

Jozef Matula
Visualisation Team Leader
IBL Software Engineering

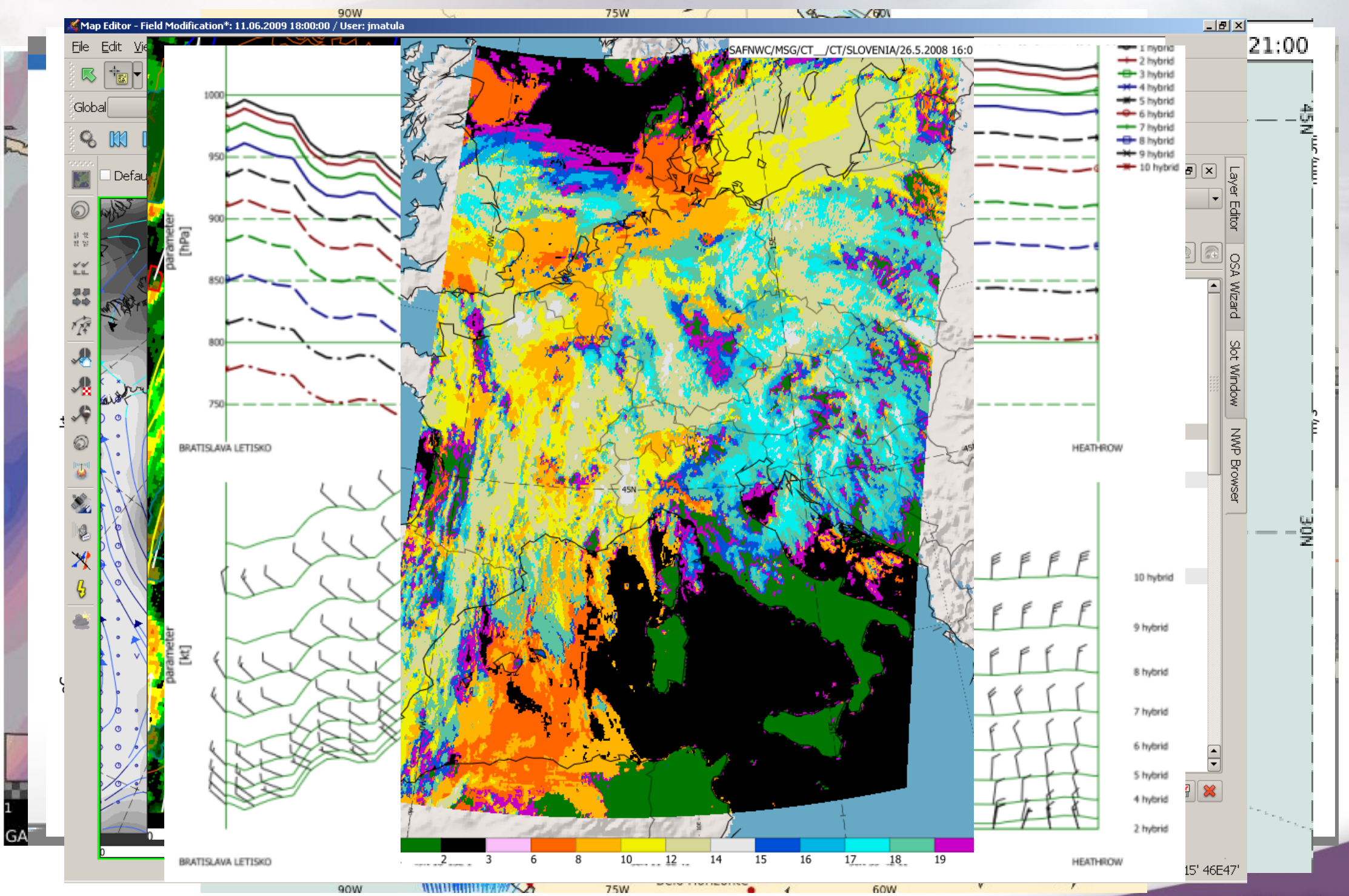
Outline

- Visual Weather in a nutshell.
- Path from Visual Weather (as meteorological workstation) to Web Server (server providing web services).
- What is Flexi Weather and Online Weather.
- There is always something what can be optimised.

Visual Weather in a nutshell

- Visual Weather is a meteorological workstation SW:
  - Met data processing and visualisation (observations, NWP, remote sensing, other GISs)
  - Interactive forecasting tools
  - Forecast production and workflow management
  - Batch production
  - Extensible with Python API
  - Web services (WMS, WCS, WFS, JMBL, bespoke in Python)
- From all-in-one laptop to enterprise client-server
- Highly configurable and integration-capable.

Various VW screenshots & products



Quick audience survey

- Who worked/works on development of a visualisation workstation?
- Who knows what is OGC?
- Who knows what is a Web Service?

FROM WORKSTATION TO PROVIDING WEB SERVICES

What a web services is

- Wikipedia: *A **Web service** is a method of communication between two electronic devices over a network.*

  [very precise explanation of imprecise thing]
- **Why are people building applications using them?**

1. For fun (experimenting what is possible).
2. Because it is a fashion.
3. There is a desire for web applications with no installation and maintenance costs.
4. To decuple the presentation vs. data/business logic tiers and allow “remote use”.
5. Customers want it.

What we have done

- Our 1st OGC Web Map Service server 6 years ago
  - Heavily using WMS dimensions
  - Custom extensions on the edge of standard
- OGC Web Coverage Service to access NWP data with NetCDF, GeoTIFF and GRIB outputs.
- OGC Web Feature Service:
  - For weather conceptual feature models (fronts, clouds, H/L, storm areas…)
  - For all kinds of in situ-data (observations, lightning, station locations…)
- Various bespoke Web Service (Python API, JSON)

Need for a client

- At certain stage we realised that for building and testing our web service capabilities we need:
  - Experimental application which understands all our protocols and can utilise them:

    Online Weather

  - Design & code patterns allowing mocking-up testing applets easily:

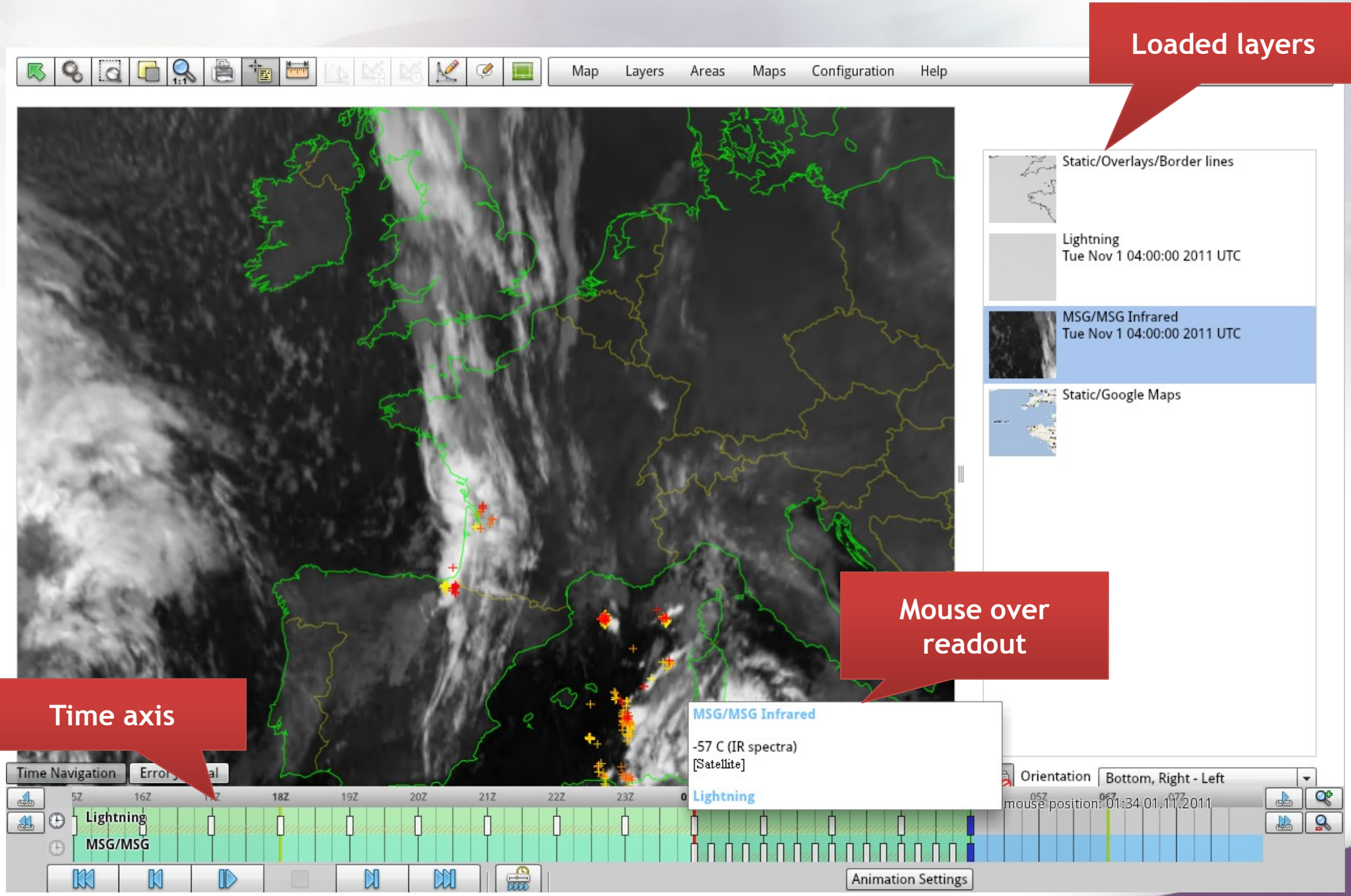
    Flexi Weather

- We chose the Adobe Flex Web 2.0 technology (uses Action Script language) as our client platform because of good browser-market share and comprehensive graphics support.

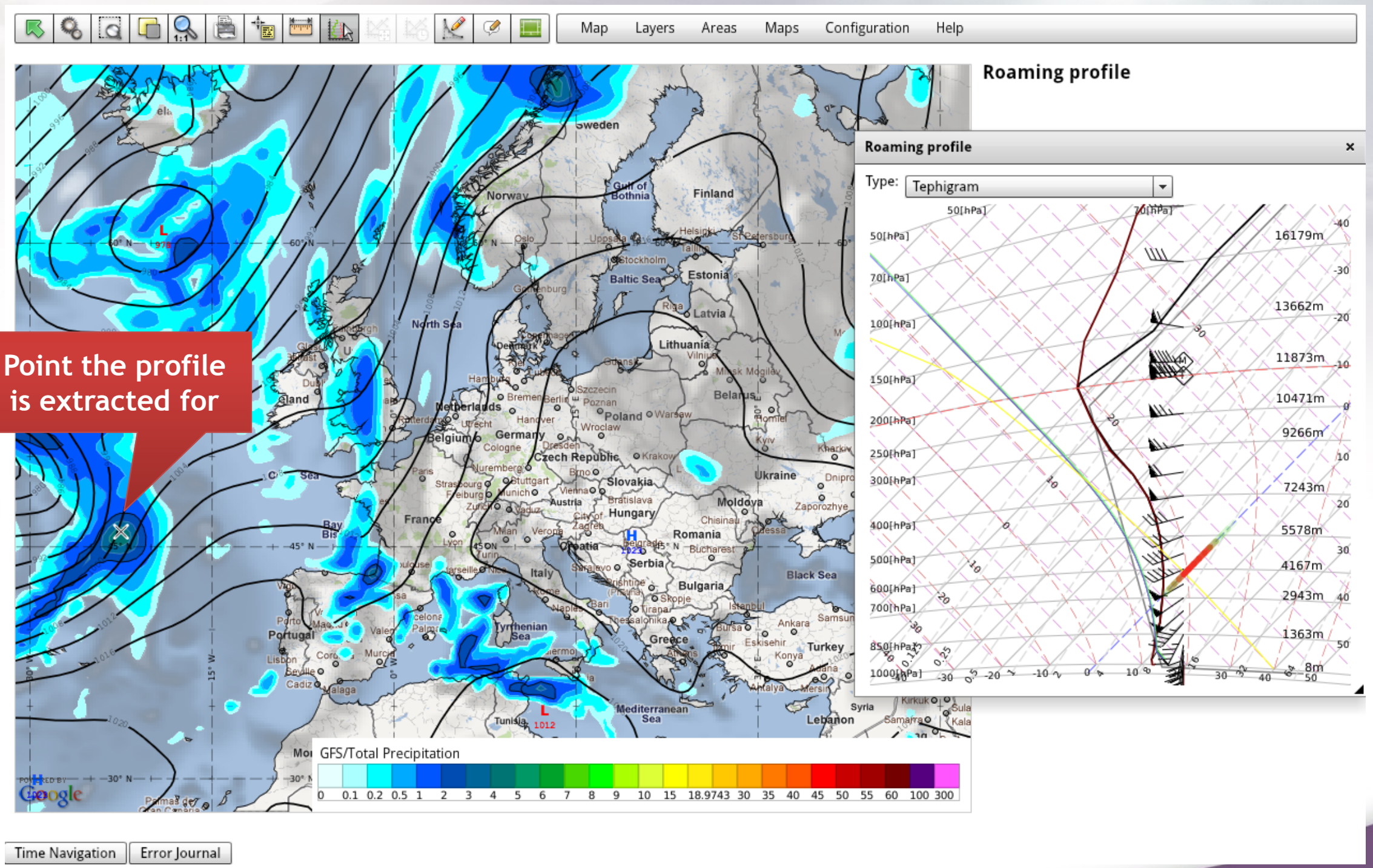
Online Weather

- Set of web applications which mimic their “elder & richer brother” applications in *Visual Weather*:
  - **Browsing Weather** – data visualisation and layering, animation, chart editing.
  - **Weather Monitor** – monitoring of observed properties reaching warning/critical thresholds.
  - **Message Editor** – forecast form entry production application with NWP/obs. first-guess support.
  - **Task Manager** – task and workflow management tool (a kind of forecaster’s ToDo list)
  - **Product Delivery Monitor** – application to track delivery of products (e-mails, faxes).

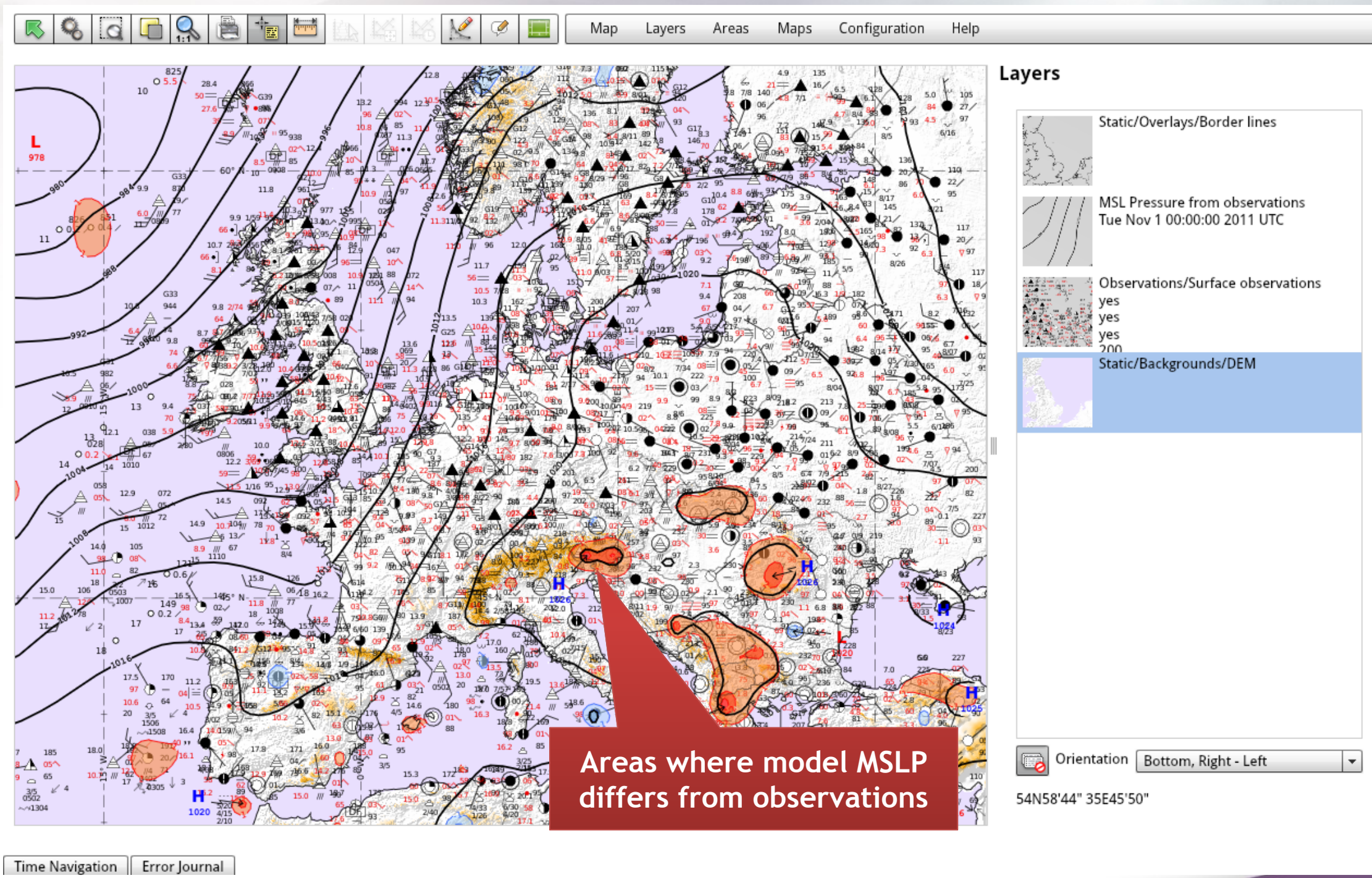
Browsing Weather - Satellite & Lightning



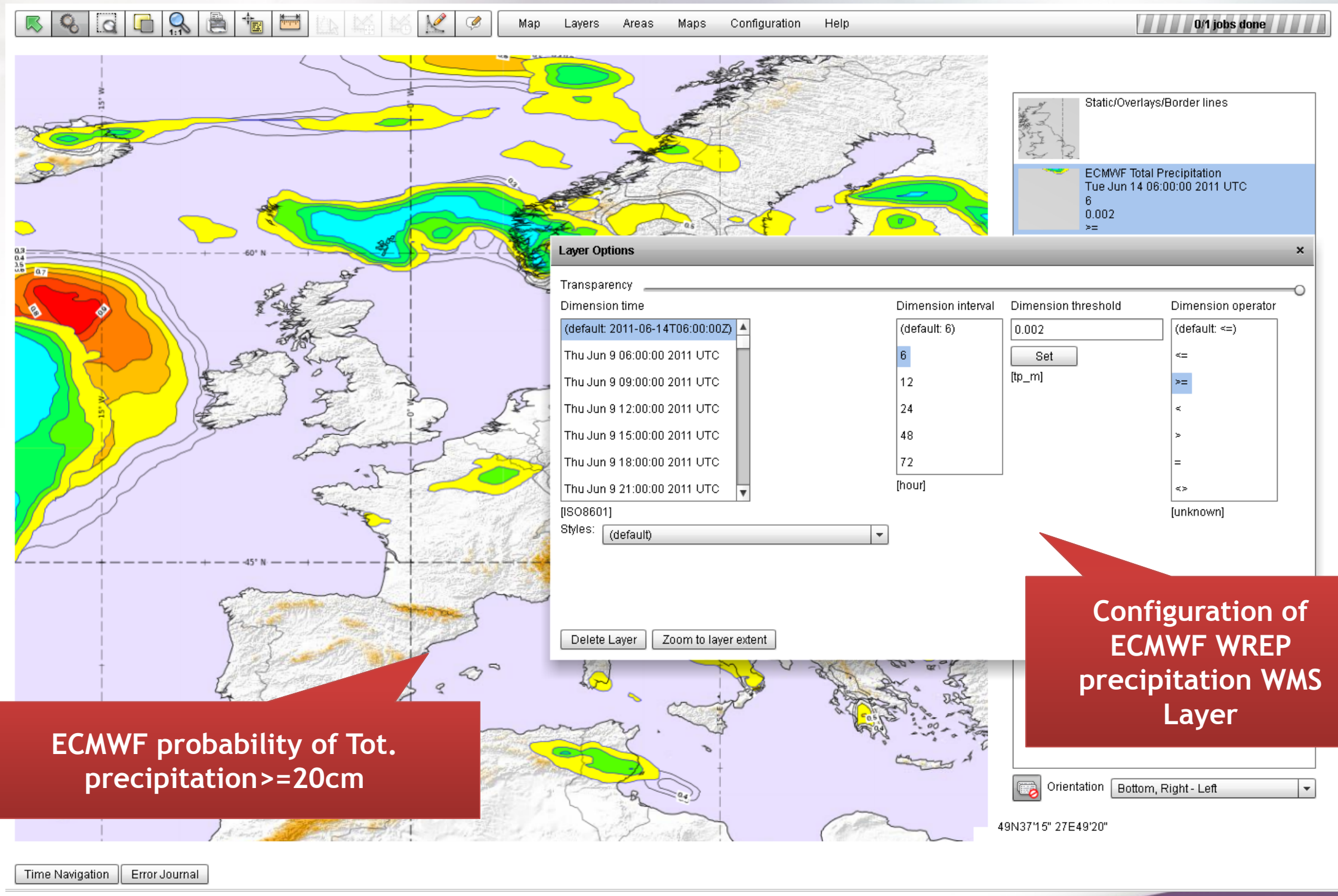
Browsing Weather -NWP vertical profiler



Browsing Weather (NWP Verification)



Browsing Weather - ECMWF web service



Online Weather - Task Manager

Working shift time range

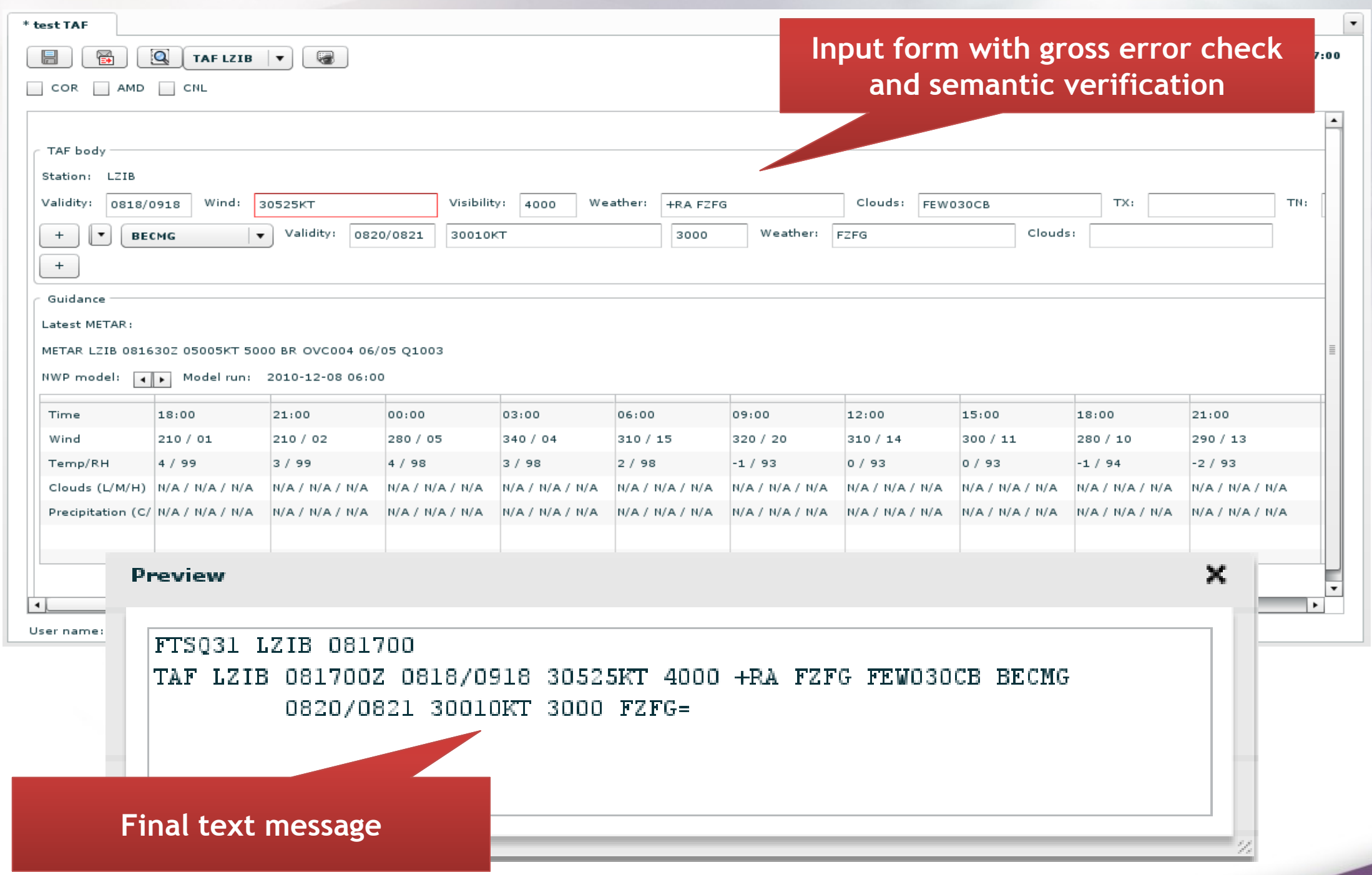
Choose roles | From: 20:00 | To: 08:00 | Brief mode | Show UTC time | Refresh

~~2011-10-31 21:00:00LT~~	Check data reception In the Operator Console look on all SADIS channels and verify that some data is being received recently.	Reopen ✔
~~2011-10-31 21:00:00LT~~	Create Low Level SIGWX chart Inspect the actual WAFC SIGWX forecasts and latest model data. In Feature Editor create the LAF product.	Edit
~~2011-10-31 21:30:00LT~~	Call Fire Police Call Fire Police +00123456911 and brief them about daily forecast and fire hazards.	Reopen ✔
~~2011-10-31 23:00:00LT~~	Check data reception In the Operator Console look on all SADIS channels and verify that some data is being received recently.	Reopen ✔
	Create High Level SIGWX chart In Feature Editor import the latest WAFC forecast and adust according to local need.	Edit
	…DIS channels and verify that some data is being received recently.	Reopen ✔
~~2011-11-01 01:00:00LT~~	…art Inspe… SIGWX forecasts and latest model data. In Feature Editor create the LAF product.	Edit
~~2011-11-01 01:00:00LT~~	Shift rotation … briefing from previous shift	Reopen ✔
Date: 2011-11-01 01:37:40LT		
2011-11-01 03:00:00LT	Check data reception In the Operator Console look on all SADIS channels…	Acknowledge
2011-11-01 05:00:00LT	Check data reception In the Operator Console look on all SADIS channels…	Acknowledge

Daily tasks, red line marks the current time

Available actions for each task

Online Weather - Forecast Entry



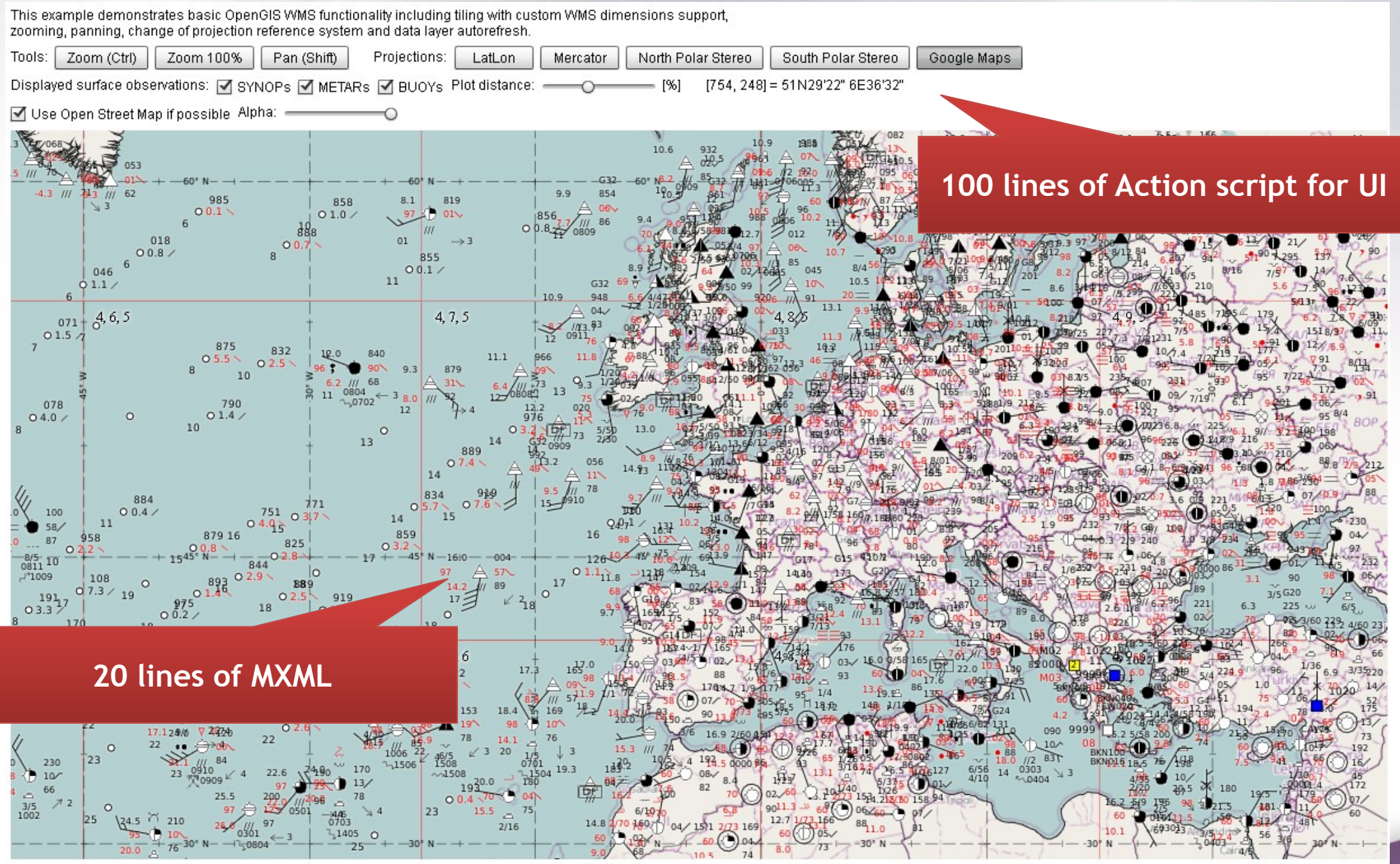
Flexi Weather

- *Geospatial mapping and data visualisation library for Adobe Flex with strong focus on the presentation of meteorological data.*

- Open Source available at:
  http://code.google.com/p/flexiweather

- Fixes problems of “Solid Earth GIS” libraries
- Supports WMS and tiling, layer superimposition, WFS, conceptual weather symbol rendering, …
- Supports portable devices (touch gestures)

Flexi Weather - Simple example

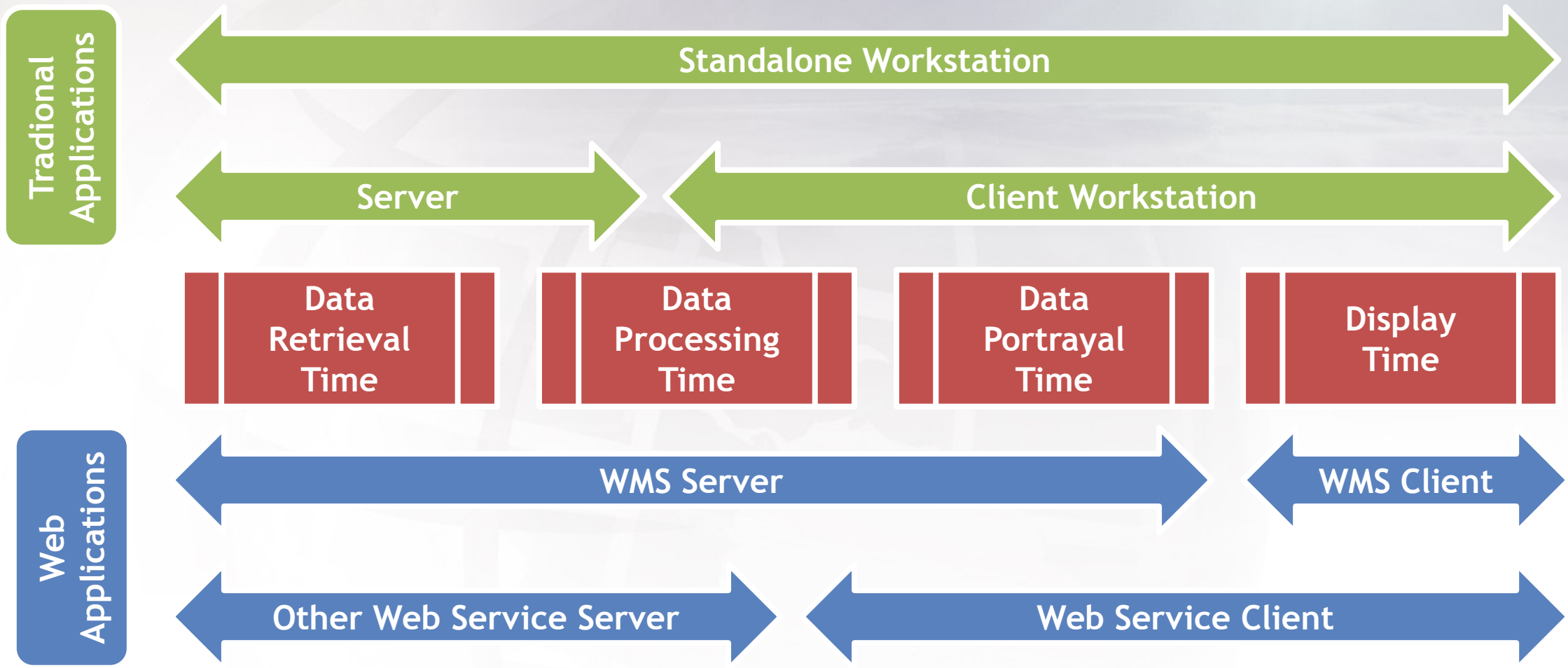


Flexi Weather & Online Weather on a tablet

THERE IS ALWAYS SOMETHING WHAT CAN BE OPTIMISED

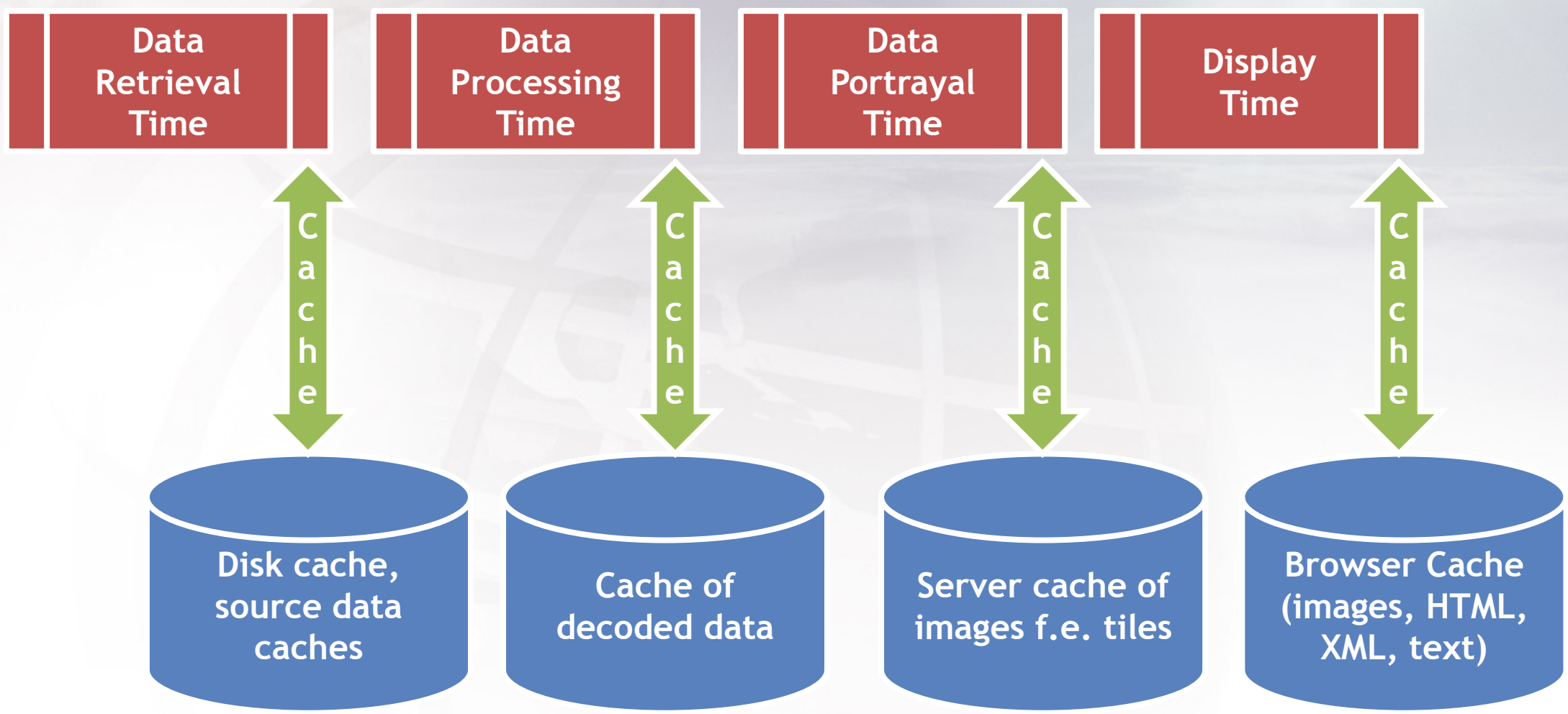
Performance, performance, performance...

- Data access time is composed of:



- "Page loading…" effect – it takes even 4-5 seconds, but it makes you feel better, because you see the progress.

Traditional improvement by caching



- Not everything can be cached
- Visual Weather has a memory cache of decoded data and rendered tiles.

Possibilities for improvement

Data Retrieval Time

- Disk caches/SSD disks/fast network are the key places for improvements
  Data structure improvement, if possible

Data Processing Time

- We utilise multi CPU core paralellisation and High Performance clustering (with facading load balancer)

Data Portrayal Time

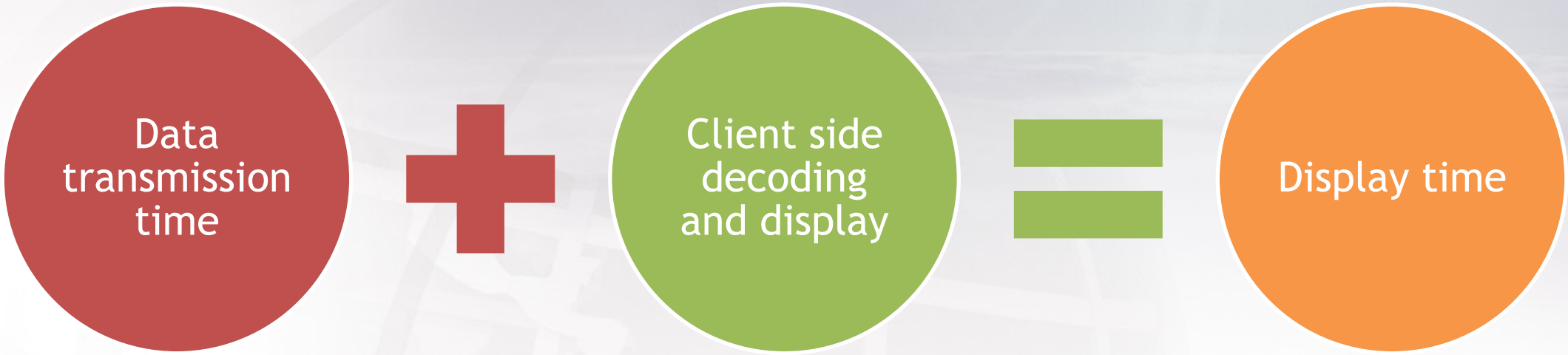
- Same as above
- Converting layer portrayal to tiles effectively and caching them.

Display Time

- Out of our control, but…
  …we had one use case which lead to surprising result…

Problems with low bandwidth users

- Low bandwidth usually implies high network latency



- Client side handling is insignificant
- Logically if data is smaller => will download faster => overall “feeling” would improve.
- Run-around time of multiple (for example 30 tiles) requests can be significant.
- **Images and XML - how can these be smaller?**

Compression of XML catalogs (Capabilities)

- Capabilities document of OGC are big XML files, very verbose having often ~several MB

- Luckily they can be well compressed with conventional algorithm ratios around 5:1.

- Compression could be explicit, but HTTP protocol allows transparent compression – quite rarely used feature, despite good support in browsers.

```
HTTP/1.1 200 OK
Content-Encoding: gzip
Content-Type: text/xml
```

How to make images smaller?

- When it comes to topic of which image formats browsers understand – one has only few real options – JPEG, GIF, PNG, BMP
  (SVG, JPEG2000 and others are purely supported).

- When dealing with transparency only PNG and GIF remain, while GIF is limited to simple 1-bit masking in comparison with PNG with real alpha channel.

- Surprise! There exists 8-bit PNG with alpha channel.

8-bit PNG, isn't it medieval technology?

- Again surprisingly it works in most of the browsers (where true color PNG works)
- Almost nobody can produce it ☺
- We tried and find out:
  - Pure compression time of 8-bit PNG is approximately only ¼ of the True Color ARGB PNG image.
  - Image size is typically 30-50% less then ARGB PNG.
  - Because Visual Weather rendering is True Color with anti-aliasing, most significant time of this is conversion to 8-bit paletted image.
  - Smaller = you can cache more.

Summary

- We have built and will continue building web services and web applications in parallel to traditional visualisation workstation – both OGC and bespoke – as needed.

- Optimising and scaling (caching, tiling) for (rendering) speed is only one side of the coin – reducing size of data being transmitted improves the response time particularly over WAN/low-bandwidth connections:
  - XML can be well compressed on HTTP level
  - 8-bit PNG is typically smaller than true color.

Thank you for attention!

Questions?