Invent weather map visualisation

A freely accessible weather-viewer using Google maps, tile-caching & WMS

Peter Trevelyan

EGOWS Annual Conference 2010

Met Office

Table of Contents

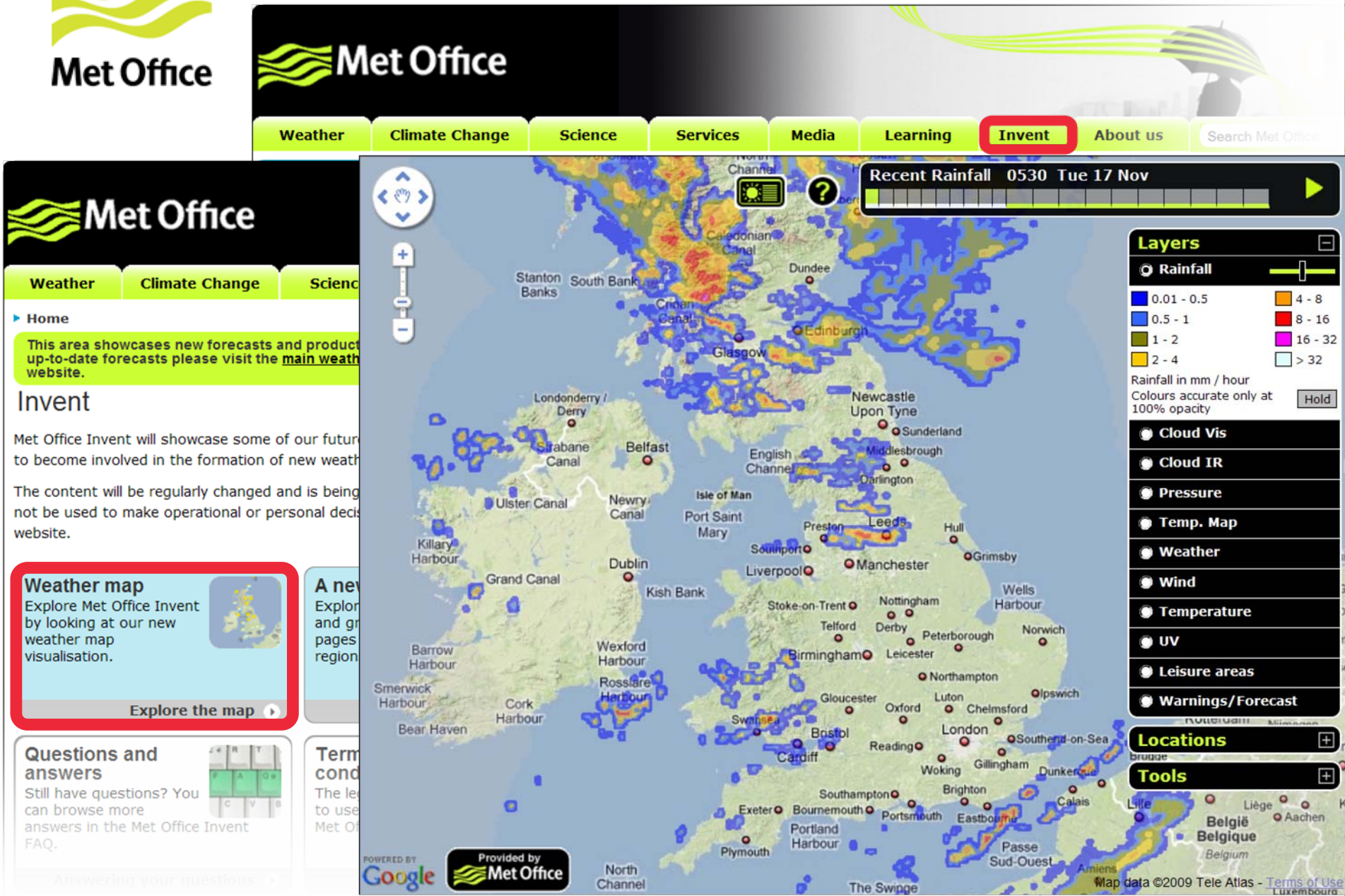
- What is “Invent”?
- Web Map Service
- Architecture for Scalability
- Conclusion

What is “Invent”?

What is “Invent”?

- Showcase for some of the Met Office future plans for presenting web-based weather forecasts, products and information
- Allows the General Public to become involved in the formation of new weather and climate change products, services or forecasts
- Essentially a beta version ➔ content can be regularly changed and continuously developed
- Today look at:
  Invent “Weather Map”
  ➔ JavaScript web client application accessing a Web Map Service

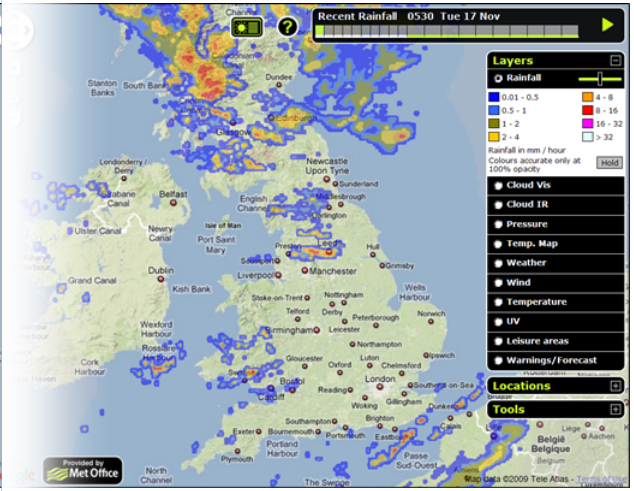
Where is Invent?

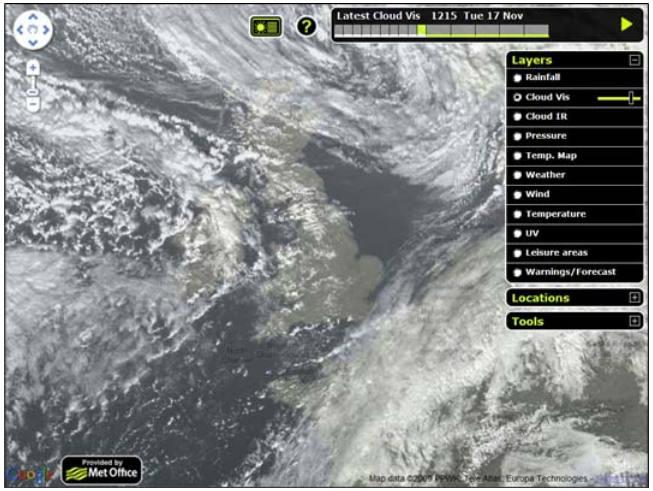


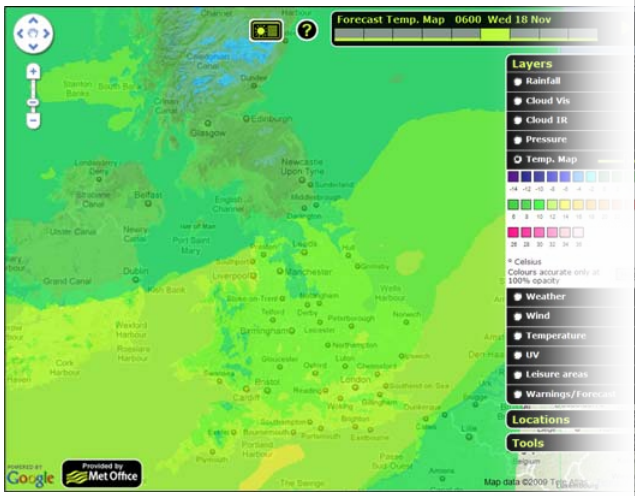
What can Invent Weather Map do? (1)

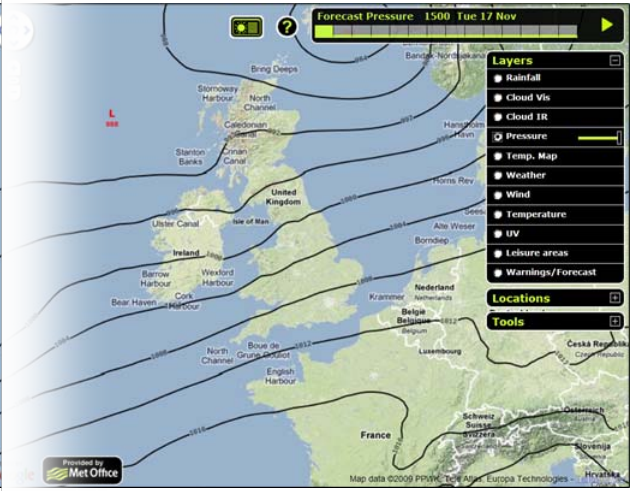
Different parameters

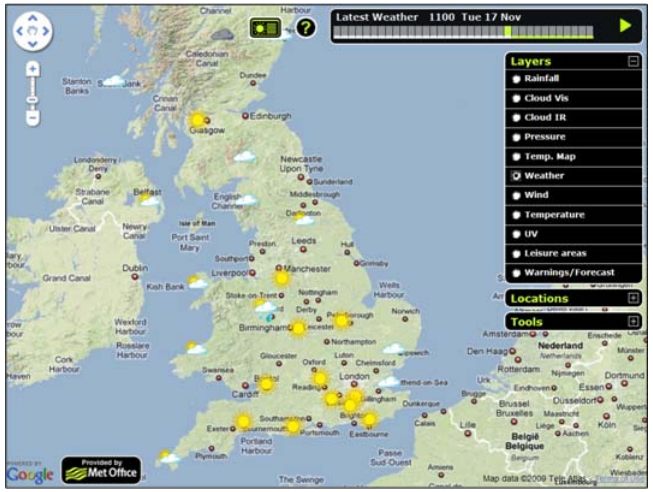
Different display styles

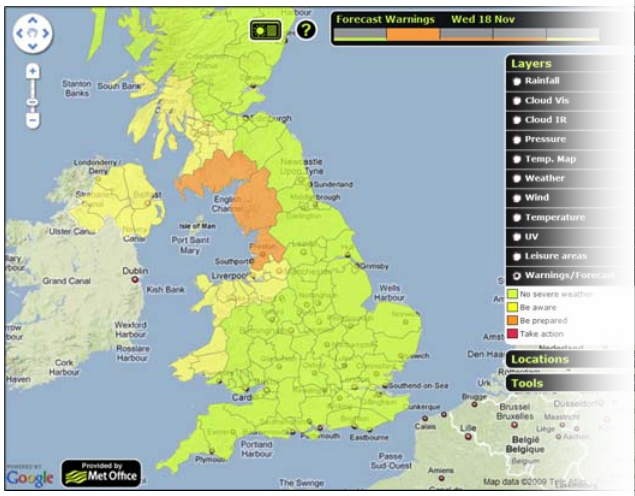




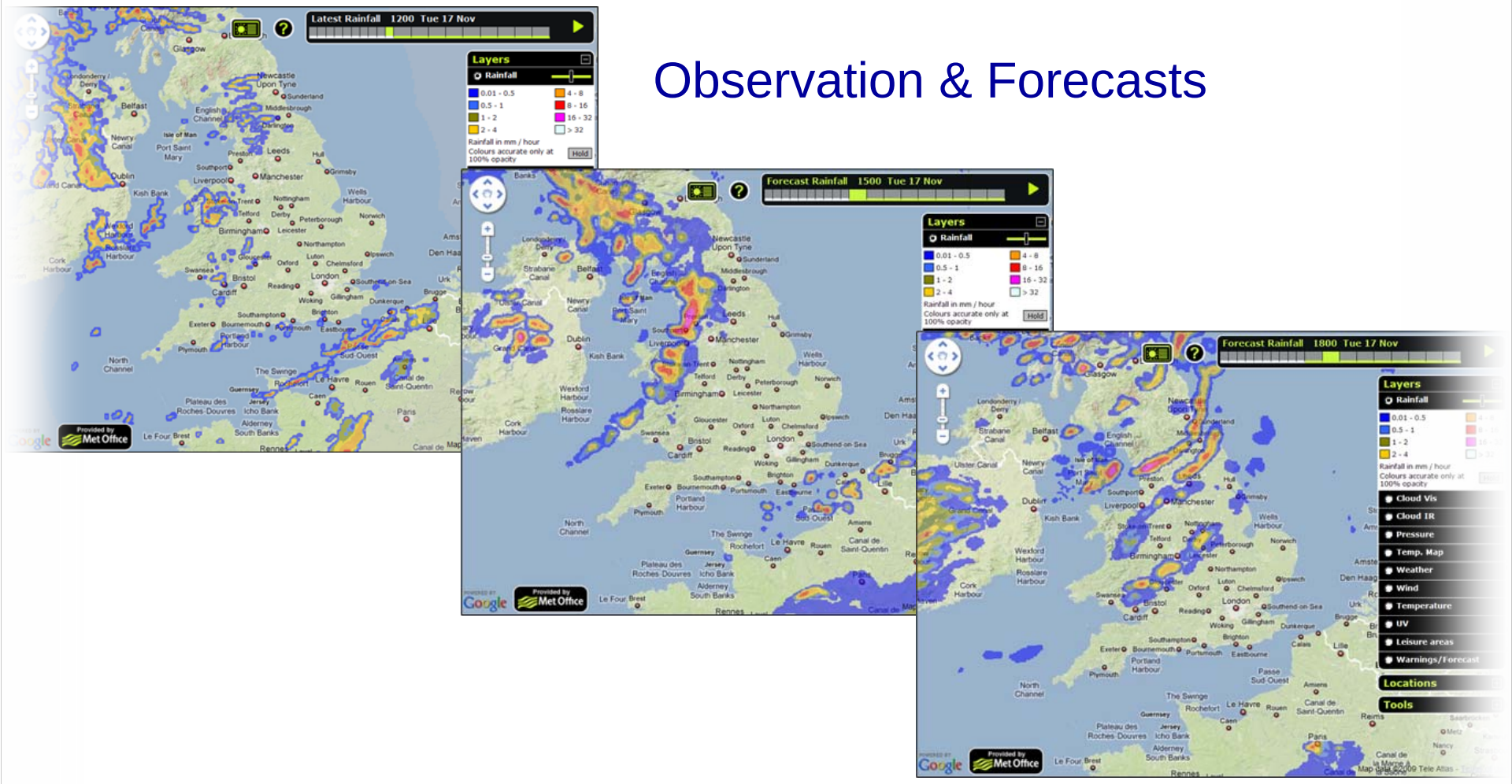








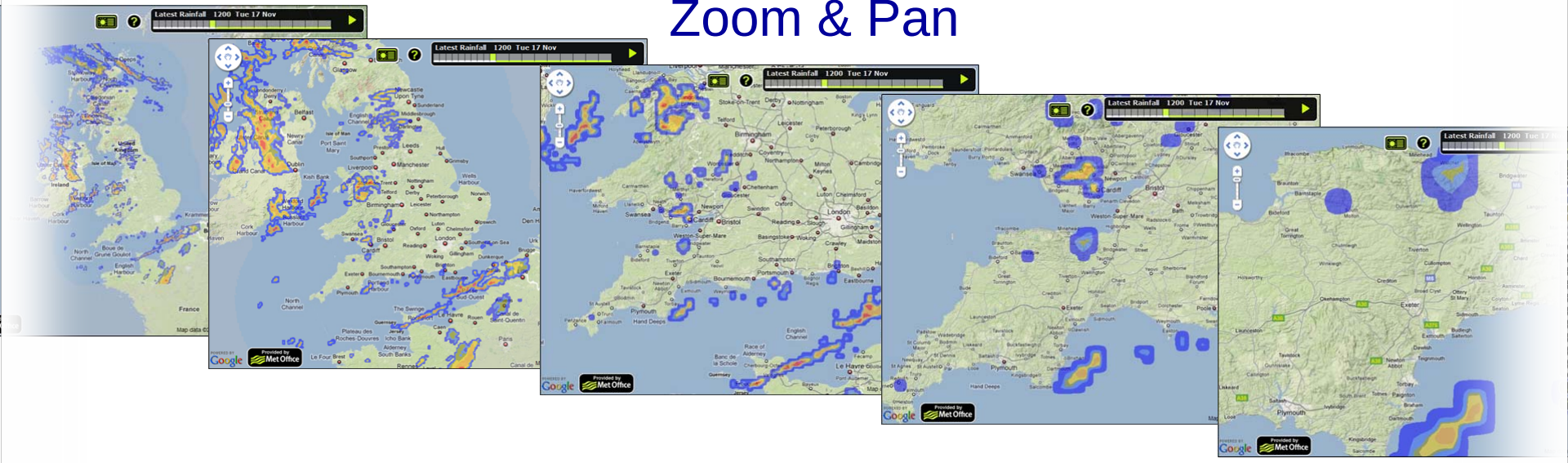
What can Invent Weather Map do? (2)



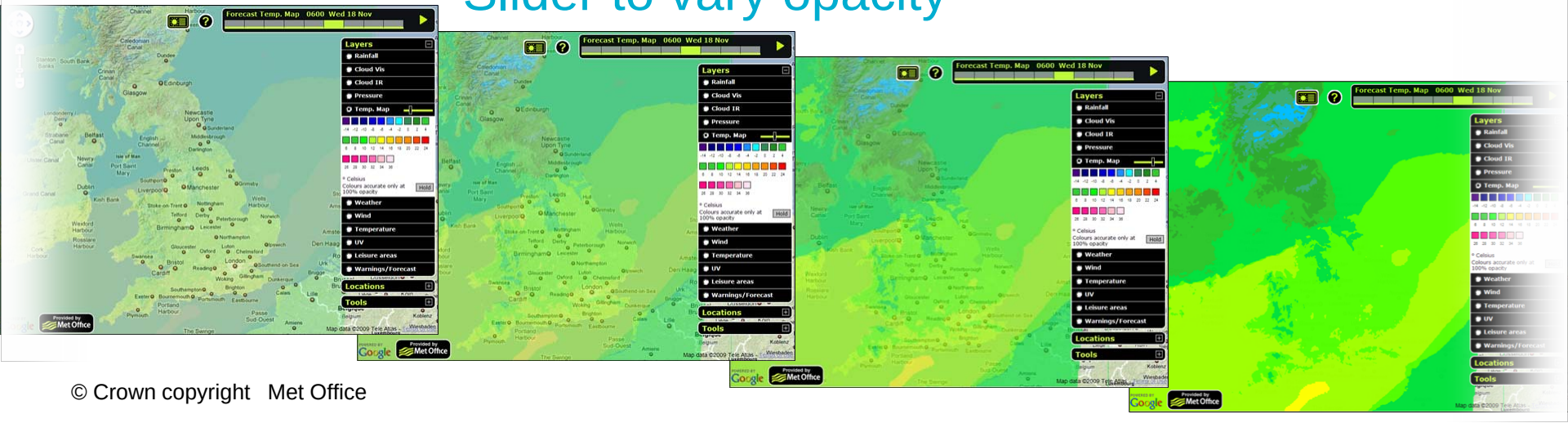
Observation & Forecasts

What can Invent Weather Map do? (3)

Zoom & Pan



Slider to vary opacity



Web Map Service

Web Map Service (1)

- Runs on the IBL's Visual Weather system
- Developed for the Met Office by IBL
- Based on WMS 1.3
- HTTP GET using Name-Value Pairs

Web Map Service (2)

- Common request parameters:
  - SERVICE = WMS
  - VERSION = 1.3.0
  - REQUEST = GetCapabilities / GetMap / GetFeatureInfo

Web Map Service (3)

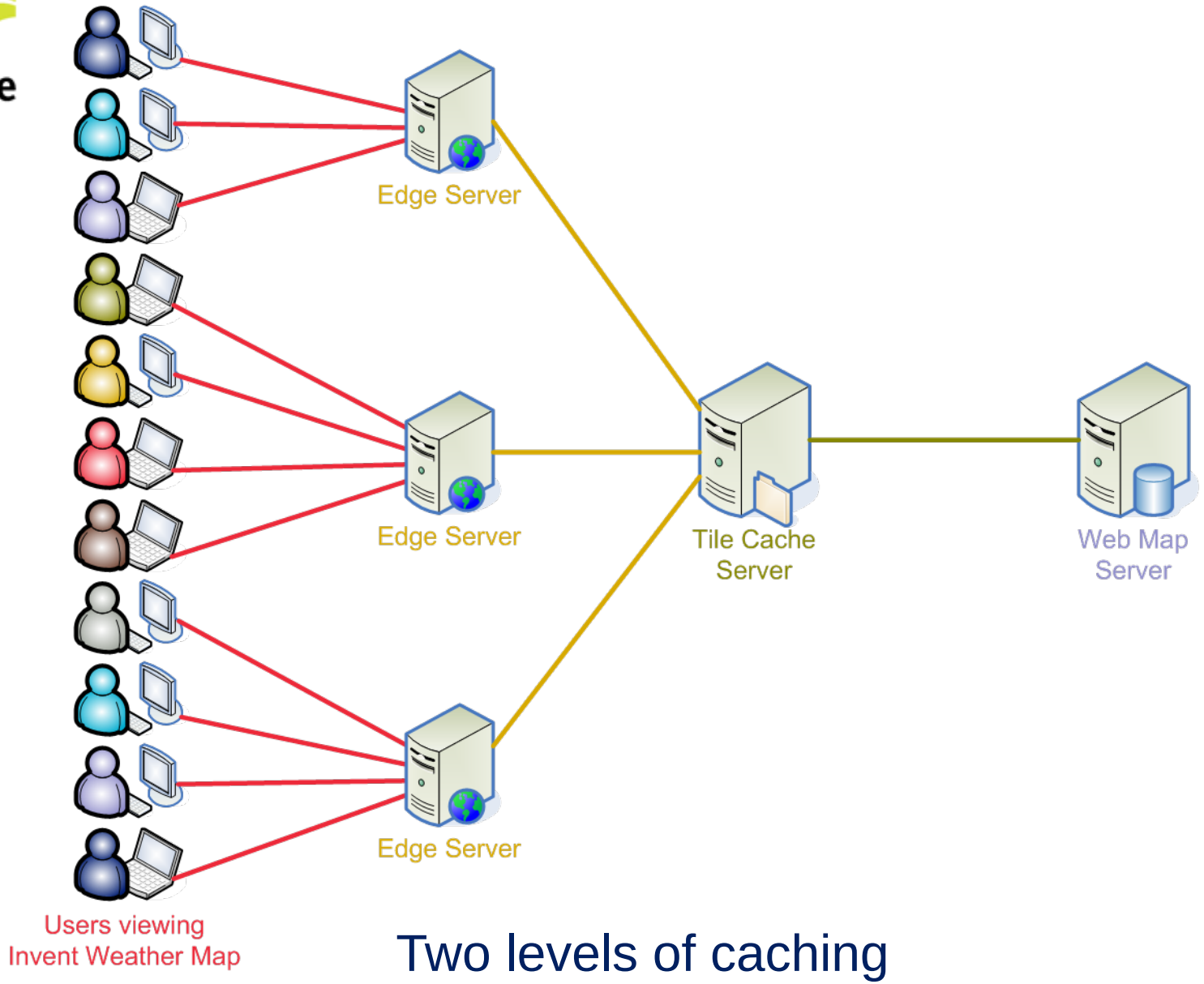
- GetMap request parameters:
  - LAYERS = layer_list
  - FORMAT = PNG (& GIF / JPEG / JPEG2 / TIFF / GeoTIFF)
  - CRS = namespace: id (CRS & EPSG)
  - BBOX = xmin,miny,minx,maxx (uses WMS 1.1 ordering)
  - WIDTH = output_width
  - HEIGHT = output_height
  - STYLE = (Usually defaulted)
  - TRANSPARENT = 0 / 1 (rather than TRUE/FALSE)
  - ELEVATION = pressure / height

Web Map Service (4)

- Date / time parameters:
  - Use ISO8601 except periods currently in form ‘+1’
- Observation data uses:
  - TIME = ‘valid time’
- Forecast data uses sample dimensions:
  - DIM_RUN = ‘analysis time’
  - DIM_FORECAST = ‘forecast period’ (+hours)

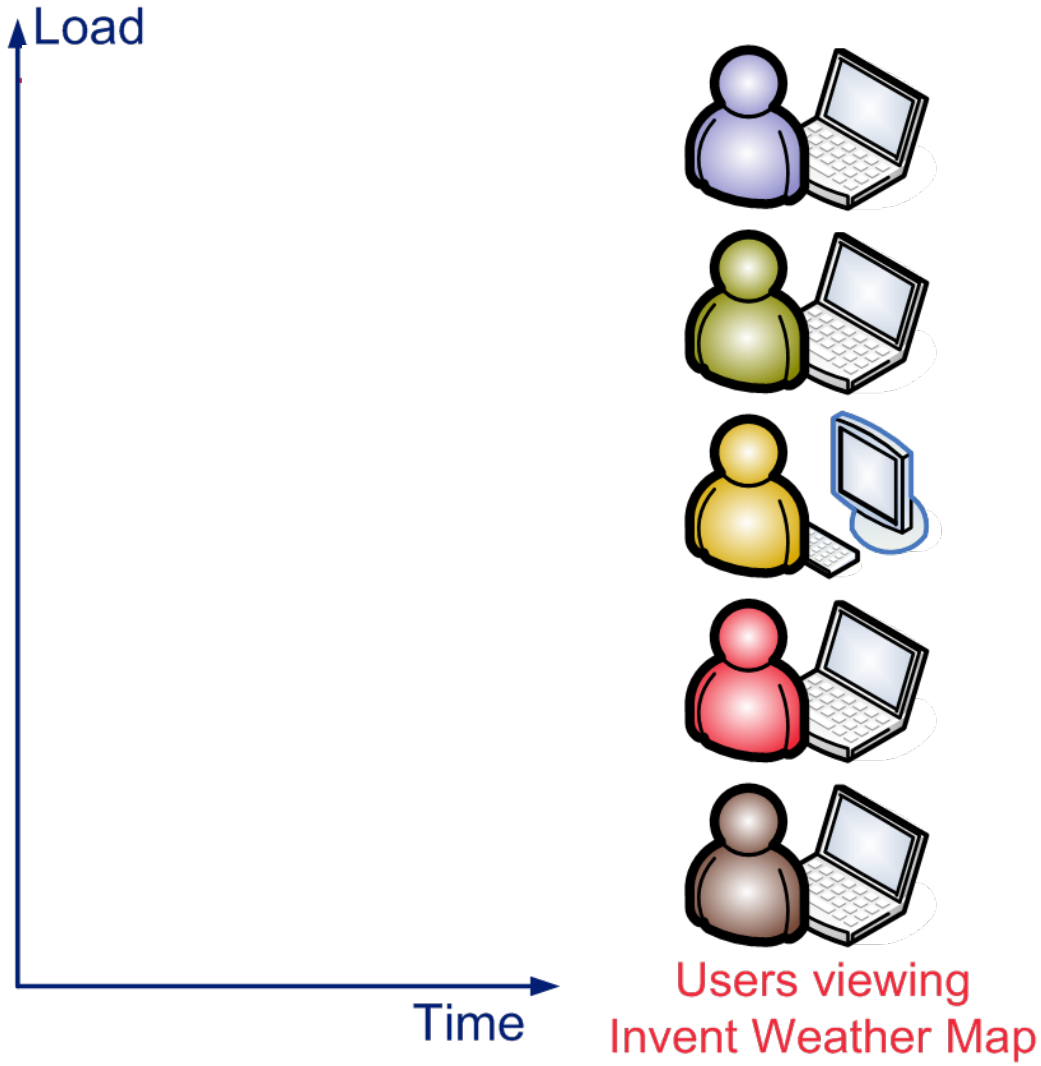
Architecture for Scalability

Architecture for Scalability



Two levels of caching

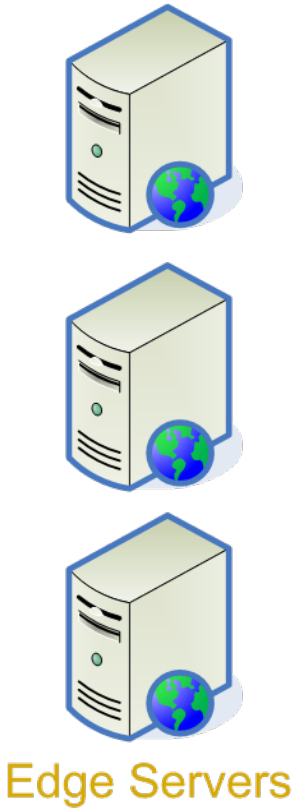
Support High User Load



Invent Weather Map:

- Uses Google Maps API
- Identifies which Tiles needed populate view port
- Requests tiles using X, Y, Z (zoom) coordinate system
- Uses Google Maps Tile API, a RESTful Web Service (just a URL)

Caching for Scalability (1)

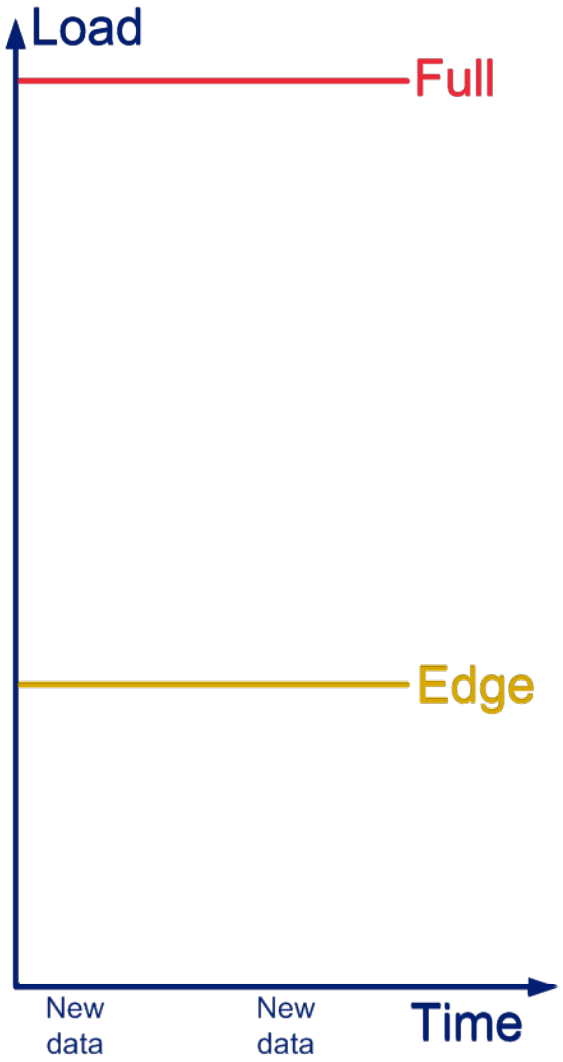


Edge Servers:

- Returns requested Tile to User, if it has a copy
- Otherwise, requests Tile from Tile Cache
- Caches returned Tile
- Returns requested Tile to User

- Provide Highly Scalable service (UK-centric)
- Externally-Hosted by Akamai

Caching for Scalability (2)

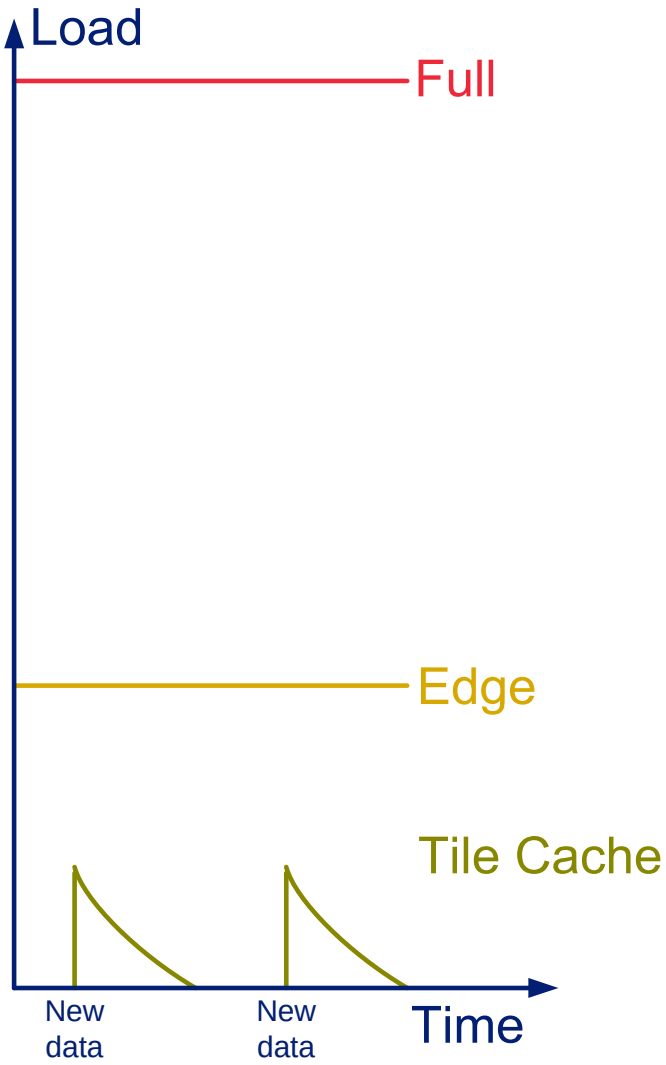


Tile Cache Server:

- Returns requested Tile to Edge Server if it has a copy
- Otherwise, calculates BBOX based on X,Y,Z
- Requests Tile using WMS
- Caches returned Tile
- Returns requested Tile to Edge Server

- Supports Scalable service
- Developed In-House

Caching for Scalability (3)



Web Map Server:

- Responds to WMS request for a Tile
- Tiles are 256 x 265
- Fixed set of Tiles for X,Y,Z
- Approach supports Efficient Caching

Conclusion

Conclusion

- “Invent” JavaScript web client application successfully deployed to showcase Met Office future plans for presenting web-based weather information
- Uses a Web Map Service implemented on IBL Visual Weather to deliver Tiles
- Architecture made Scalable through the use of two levels of Tile Caching

Acknowledgments

All of this work was carried out by others!

I am just presenting it.

Questions and answers