

[www.iblsoft.com](http://www.iblsoft.com)



## Prototyping WCS 2.0 extension for meteorological grid handling

Open   
Weather

Michal Weis

IBL

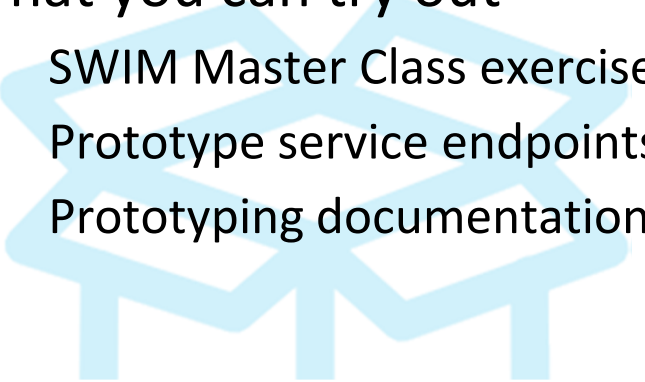
Pavol Novotný, PhD

Prototype developer, IBL

Jozef Matula

IBL

- What we achieved already
- Big Data and our related ambitions
- WCS 2.0 MetOcean extension prototype
  - Live demo
  - Idea failures and successes
  - (some) IBL extensions of the original “MetOcean Extension”
- What you can try out
  - SWIM Master Class exercise
  - Prototype service endpoints
  - Prototyping documentation



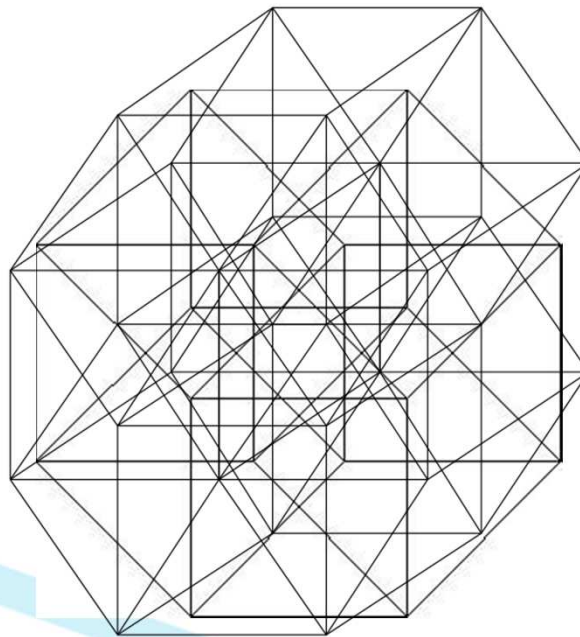
- Motivation: we are heading to data hungry future world
- How big the big data is really?
- We are on the dawn of weather data explosion - it is getting

## **BIG BIG BIG** (really **BIG**)

- number of major HPC upgrades at weather services
  - global models soon down to 10km, huge local models/EPS
- new generation of weather satellites
  - NPP, Himawari, GOES-R, MTG
- it is not possible to transfer all source data anymore, even manipulating within a single system gets impossible
  - need vertical & horizontal scalability, exploring other architectures
- Need for service centric approach, can't ship data

# Big Data Come in Different Dimensions

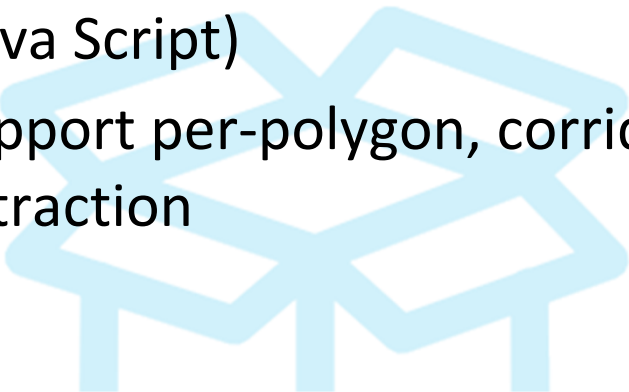
Most of this data is actually (somehow) gridded. Very obvious cartographic/raster data will come in simple 2D geographical grid



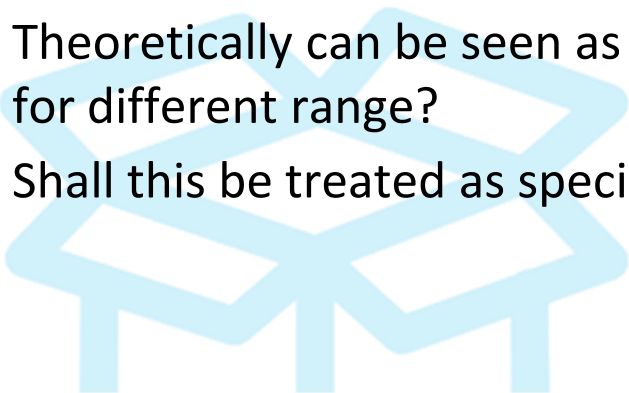
This is where we are just starting operating - at 5 dimensional Hypercube, and going even further.  
*(And by the way also in different shapes! Such as Icosahedral)*

# WCS 2.0 Prototype Ambitions

- Properly expose metadata about available gridded data
  - Correctly understand “model run” semantics (e.g every 6 hours there is new run  $\Rightarrow$  coverage available) instead of hiding it behind “reference something time”.
  - Understand: available parameters, units of measure, missing fields.
- Properly cope with isobaric height as “z” dimension and allow interpolation in this axis
  - ... and cope with other vertical level types too (e.g. gpm).
- Do as much as possible to on the server in order to allow extraction and processing of gridded data in a web application (Java Script)
- Support per-polygon, corridor (2D, 3D, 3.5D, 4D) data extraction



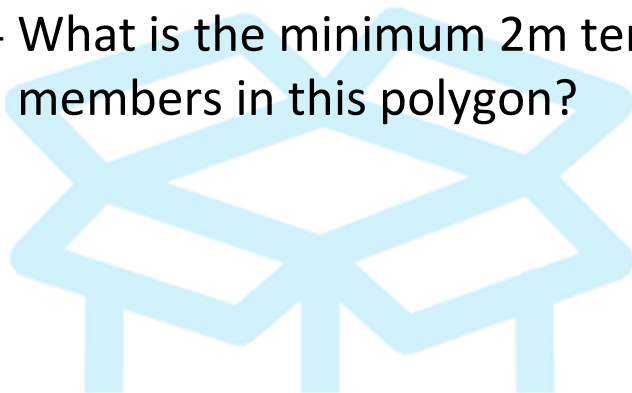
- Not suitable for REST-like “Permalink” approach. Requests contain various time varying parameters (e.g. absolute time range for trimming)
- Super-easy Denial of Service attacking.
- Ensemble dimension is not suitable for trimming (it is a discrete dimension), needs another approach instead
  - Does not make too much sense to trim range <5; 7> for to query only members 5, 6 and 7
- GRIB ranged layers e.g. “10-15cm under ground” are problem
  - Theoretically can be seen as GetCoverage Trim, but what if user asks for different range?
  - Shall this be treated as special “single value” vertical level type?





# Successes (Some Workaround Found)

- “Permalink” - one request forever:
  - Idea of “Latest run coverage” (kind of “best run”)
  - Slice and trim in “f” (forecast offset) dimension
- Denial of Service
  - Request time restriction, accessed grid count restriction, per-user policies
- Ensemble processing functions `GetCoverageStatistics`
  - Allowing probability thresholds, determining probability of event & PDF distribution
  - What is the minimum 2m temperature in this polygon?
  - What is the minimum 2m temperature from minimum of all ensemble members in this polygon?



# Prototype Service Endpoints

- Intro with examples of GetCoverage and GetCorridorCoverage requests (which you can try out already today):
  - <https://swim.iblsoft.com/wcs-intro/>
  - <https://swim.iblsoft.com/wxcube?SERVICE=WCS&REQUEST=GetCapabilities&VERSION=2.0.0>
  - <https://ogcie.iblsoft.com/metocean/wcs?SERVICE=WCS&REQUEST=GetCapabilities&VERSION=2.0.0>
  - <https://ogcie.iblsoft.com/wafc/kwbc?SERVICE=WCS&REQUEST=GetCapabilities&VERSION=2.0.0>

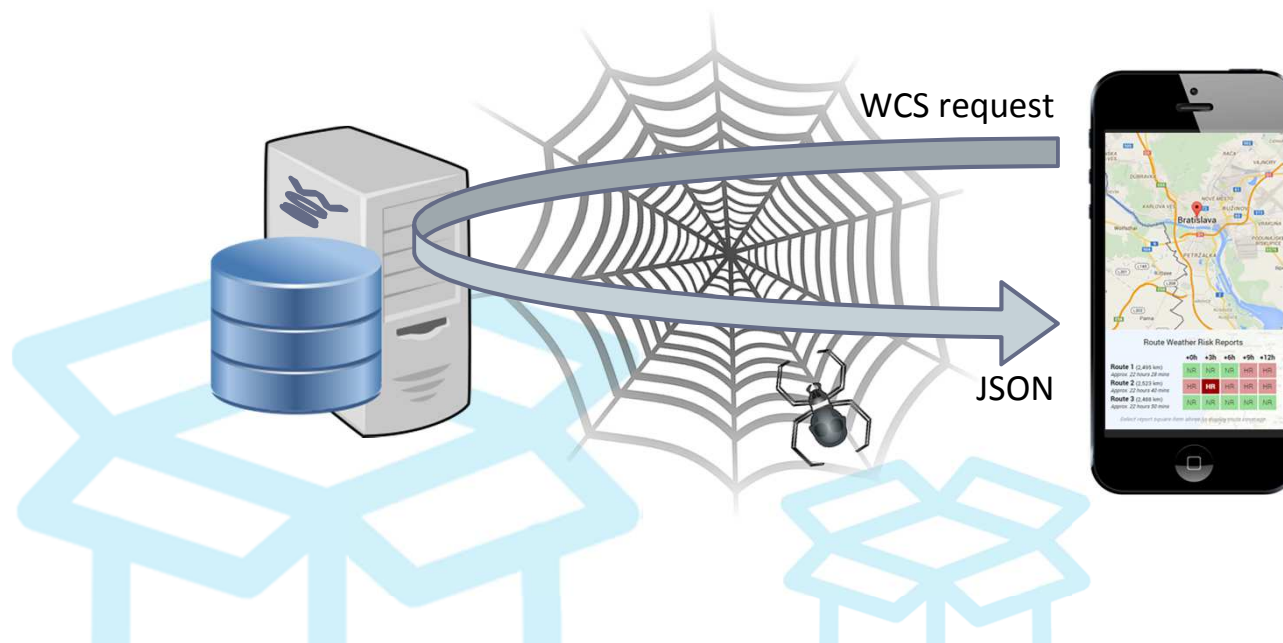




# Demo Apps Principle

So we have a powerful server - now we need to demonstrate what it is good for!

- Simple Java Script apps using Google Maps API as the background:
  - Server is dumb raw data provider
  - Application does the “clever decision” part (for now)



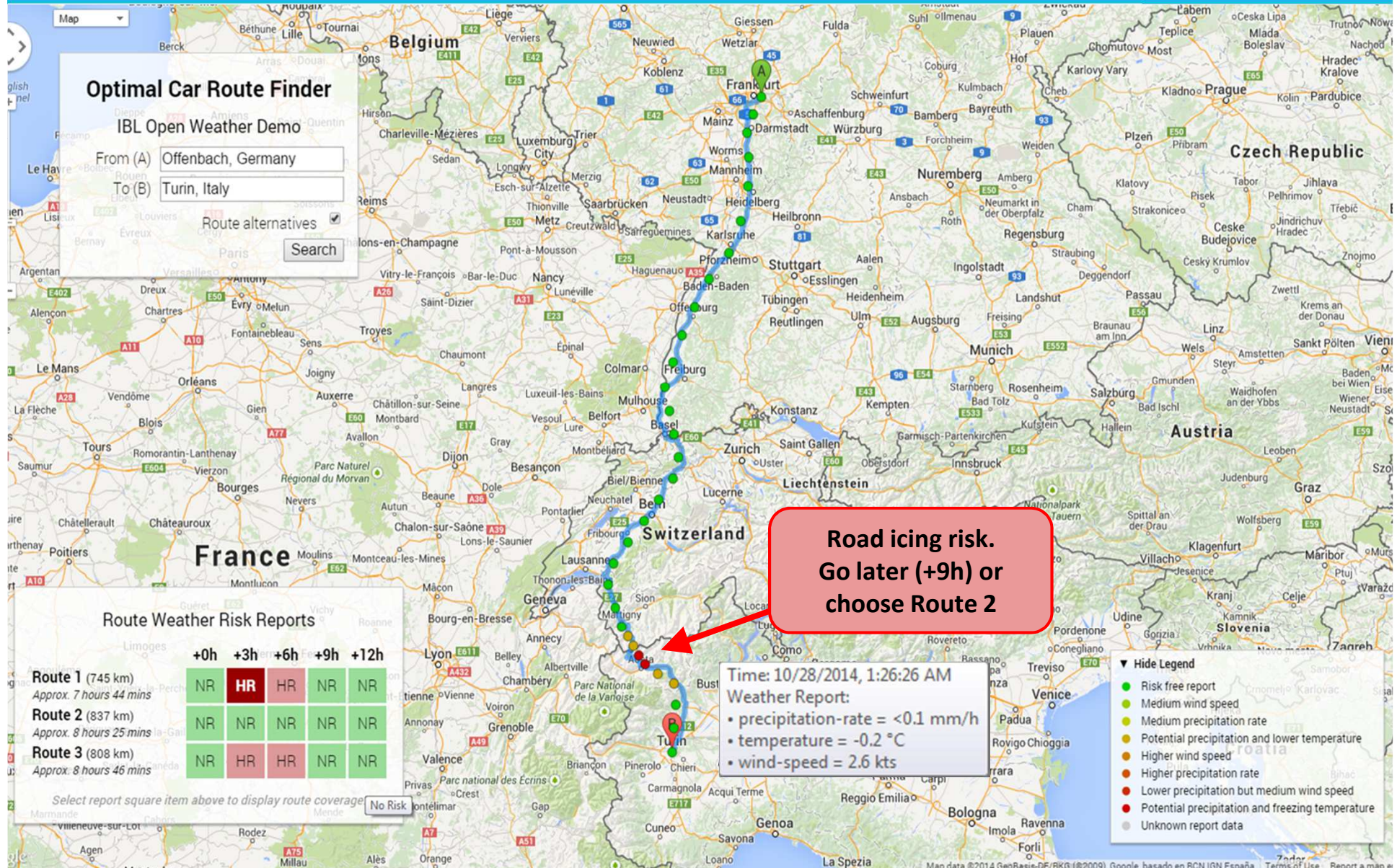
Examples that use WCS 2.0 MetOcean in somehow “practical way” for demonstration. See <https://ogcie.iblsoft.com/ow/>

- Optimal Car Route Finder
  - Optimal time for route travel by car depending on weather impacts.
- Optimal Flight Plan Finder
  - 3D+T Corridor extraction for optimal time of flight depending on weather impacts (turbulence, icing).
- Barbecue Time Opportunity
  - Best opportunity time window for barbecue depending on weather impacts.



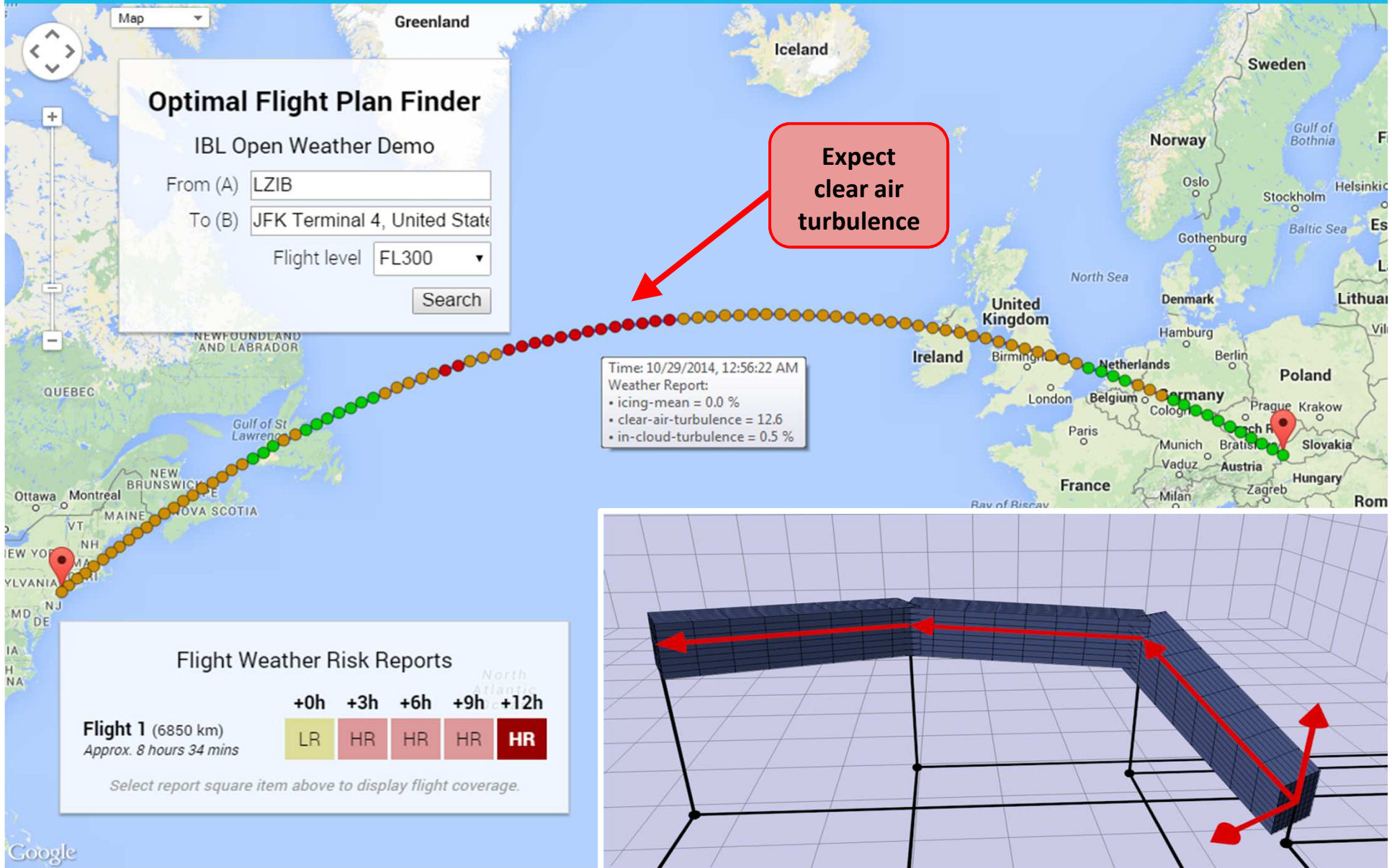


# Optimal Car Route Finder





# Optimal Flight Plan Finder





# Barbecue DST Window of Opportunity



Satellite

### Barbecue Time Opportunity

IBL Open Weather Demo

Address

Time from  to

Duration

### Place Weather Risk Reports

Frankfurter Street 135  
63067 Offenbach, Germany

	Today	Wed	Thu	Fri	Sat
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					

*Select report square item above to display place coverage.*

No barbecue in the next days

No surprise, sorry!

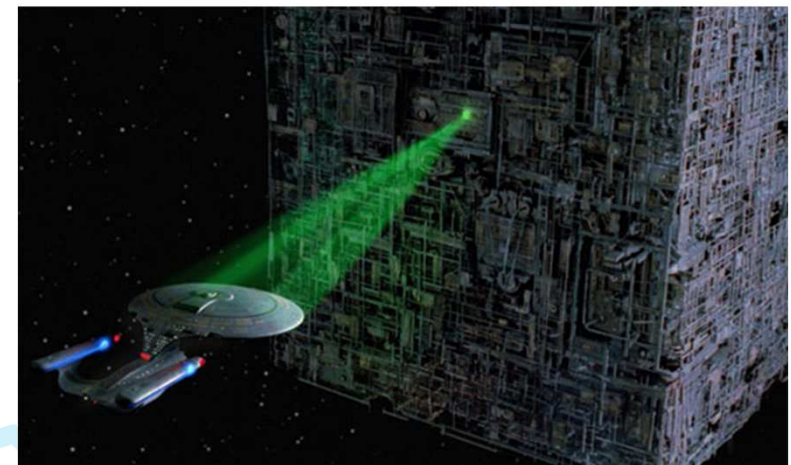
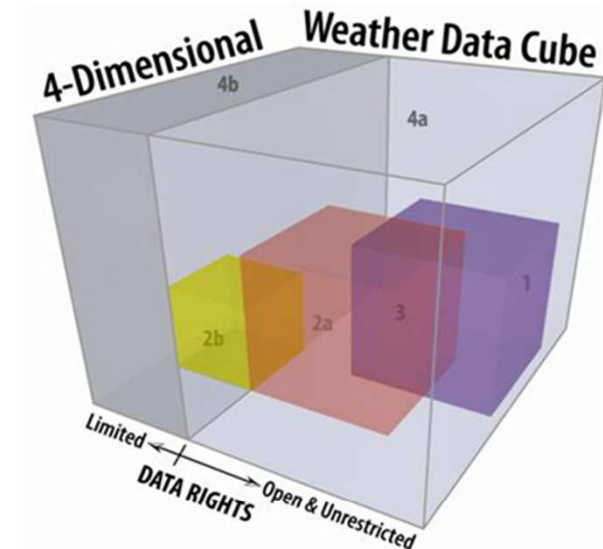
▼ Hide Legend

- Risk free report
- Medium wind speed
- Mild temperature
- Medium precipitation rate
- Higher wind speed
- Lower temperature
- Higher precipitation rate
- Unknown report data



# EuroControl SWIM Participation

- 4D Weather Cube buzz - have you ever seen it?
- Yet another ambition - Show that prototype can serve weather data as a WCS 2.0 - including:
  - very high resolution NWP models (500m resolution) around Vienna airport
  - global model
- Still a question how to blend different sources of data... It's client task(?)

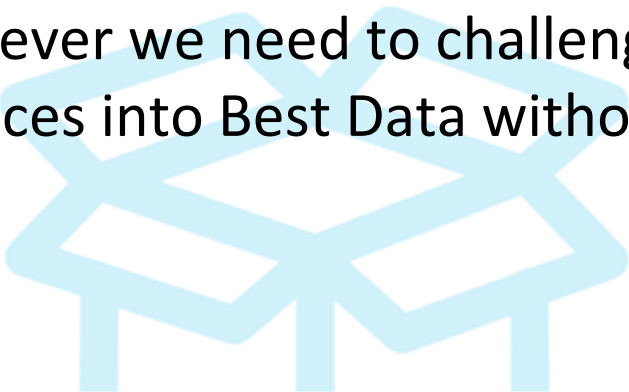


*Yes, we've seen Borg Cube, that's right!  
But not yet the Borg Hypercube...*



# From Big Data to Best Data (Challenges)

- Data blending of multiple nested gridded models for single trajectory: “Best Data” principle
- Although not visible, already simply done in BBQ TDA in client
- Performing multi-domain blending (also with maintaining PV) is doable task at server side - since client should not worry about it
- However we need to challenge blending of multiple Big Data services into Best Data without major traffic between them!



# Further Challenges...

- Traditional OGC architecture - one *service for data retrieval* - WCS another *service for data processing* - WPS/WCPS - cannot scale with data volume growth. Whatever you consider fast:
  - ADD: How long it takes to encode/decode/transfer data?  
*Forget XML, GML interchange*
  - MULTIPLY: For large data volumes, users share LAN bandwidth  
*10GBit & 10 users = 1GBit per user = 100MB transfer takes 1 second.*
  - MULTIPLY: What if user needs to evaluate alternative routes?  
*Balance between multiple coverage queries vs. one query for larger volume surrounding all routes (if possible)*
- Even bigger data reduction is needed “on the service side”
  - Need to “bring problem to data” (fascading WPS)
  - WPS/WCPS working on “local data” is a direction?
  - Customer algorithms running from App Store
  - Need to operate on cross-domain cross-server



- This is not a “Best Practice”
  - This is just a prototype to gather Real Practice
  - However it is already suitable for operational deployment (e.g. QoS and user policies)
- You can try it out
  - **Human to Machine** mode - try out demo apps  
<https://ogcie.iblsoft.com/ow/>  
next time when you fly, go by car for a long trip in winter or plan a barbecue.
  - **Machine to Machine** - read documentation, build WCS to queries:
    - NCEP GFS global coverage model
    - NCEP Washington WAFC aviation model
- Integration of WCS within other (our) products (data delivery, catalog)



Thank you

Open   
Weather

Your questions are welcome!

