

LEXIS & HiDALGO

Enabling Complex Big Data Workflows across HPC and Cloud Centres

J. Hawkes, M. Vuckovic, E. Danovaro, N. Manubens, C. Iacopino, S. Siemen, B. Raoult, T. Quintino

ECMWF Virtual workshop: Weather and climate in the cloud 2021



© ECMWF February 10, 2021



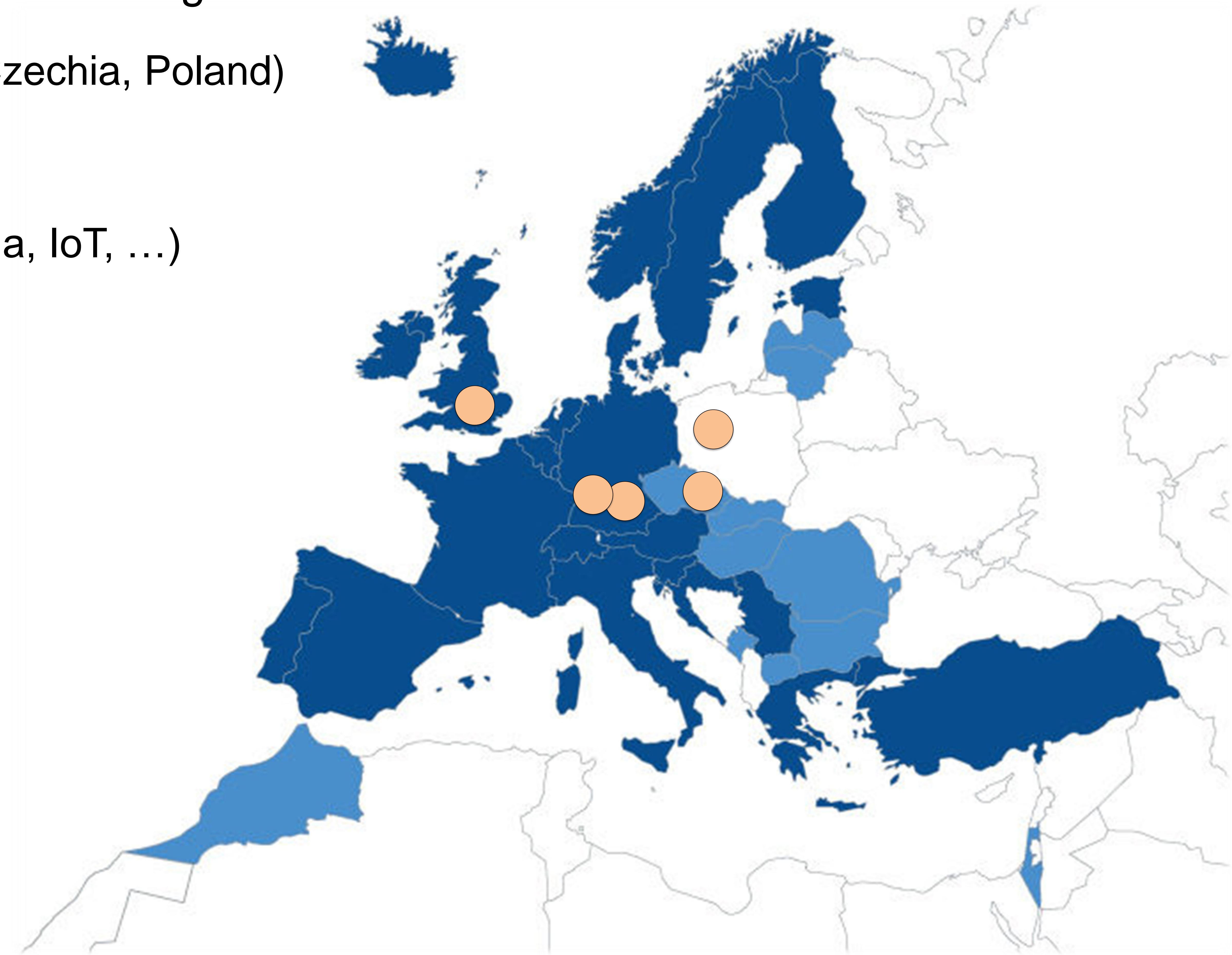
Introduction

- LEXIS & HiDALGO both construct complex, socio-economic modelling workflows

- Distributed across multiple computing sites (UK, Germany, Czechia, Poland)
- Distributed across multiple platforms (HPC, HPDA, Cloud)
- Combining data from multiple sources (Weather, Social Media, IoT, ...)

- ECMWF's role

- Upstream global weather forecasts on ECMWF HPC
- In-situ data pre-processing of meteorological data
- A facilitator of meteorological data transfer between centres
- Convergence of HPC & Cloud for meteorological data



LEXIS

HiDALGO

LEXIS

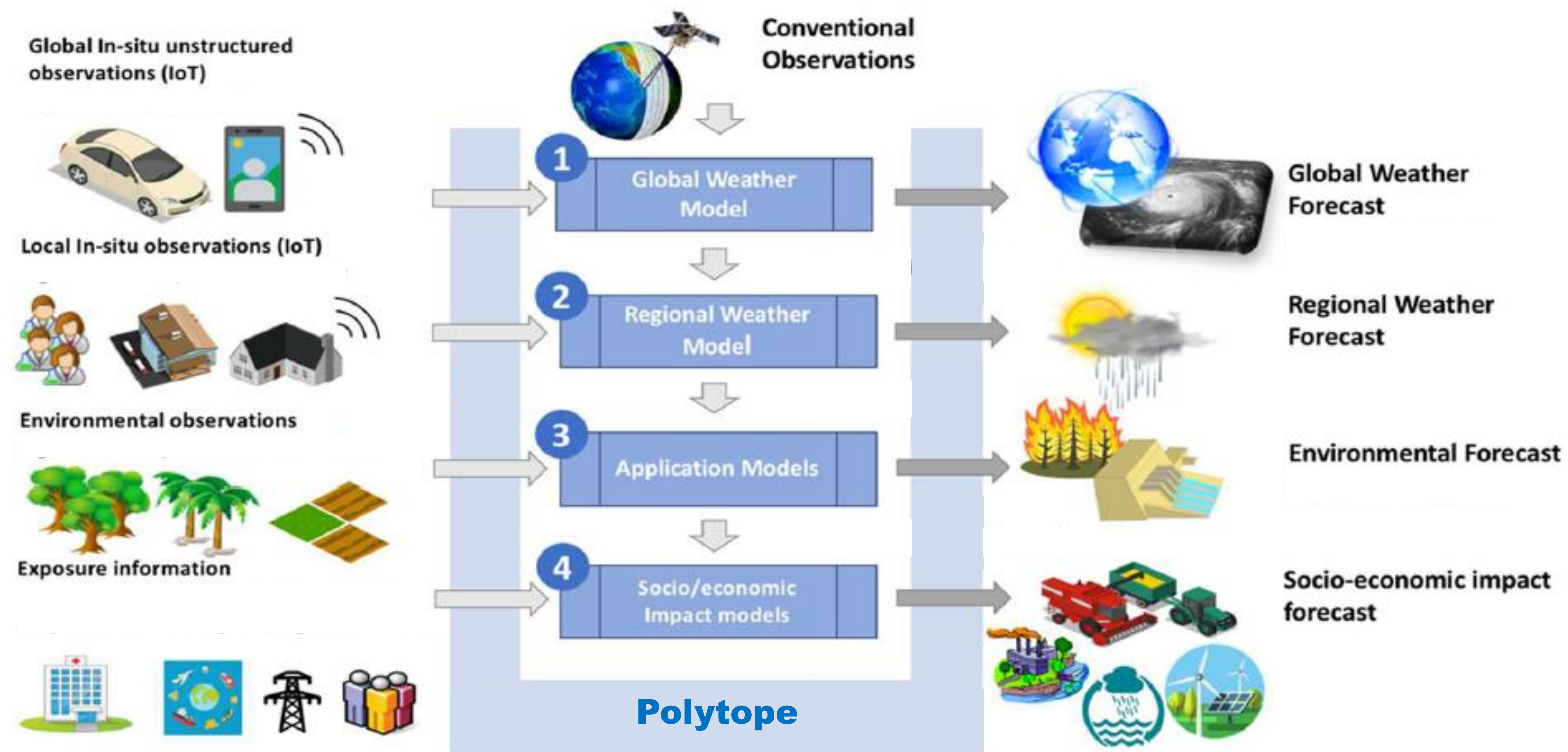


- LEXIS consists of three socio-economic pilots
 - Aeronautics
 - Catastrophe Alert systems
 - **Weather & Climate**
- Emphasis on distributed computing
 - Global weather model on ECMWF HPC
 - Regional weather models at LRZ (Germany), IT4I (Czechia) and CIMA (Italy) HPCs
 - Application models on LRZ (Germany) and IT4I (Czechia) cloud platforms
 - Socio-economic impact models on LRZ (Germany), IT4I (Czechia) cloud & other external resources
- Supports agricultural modelling, air pollution modelling and extreme rainfall forecasting



LEXIS

- ECMWF is building the LEXIS Weather and Climate Data API
 - Enabling interchange of meteorological data between multiple layers of the workflow
 - A distributed RESTful service providing access to in-situ or remote data at ECMWF, LRZ and IT4I
 - Built on existing technologies (FDB) and new developments (Polytope)



HiDALGO



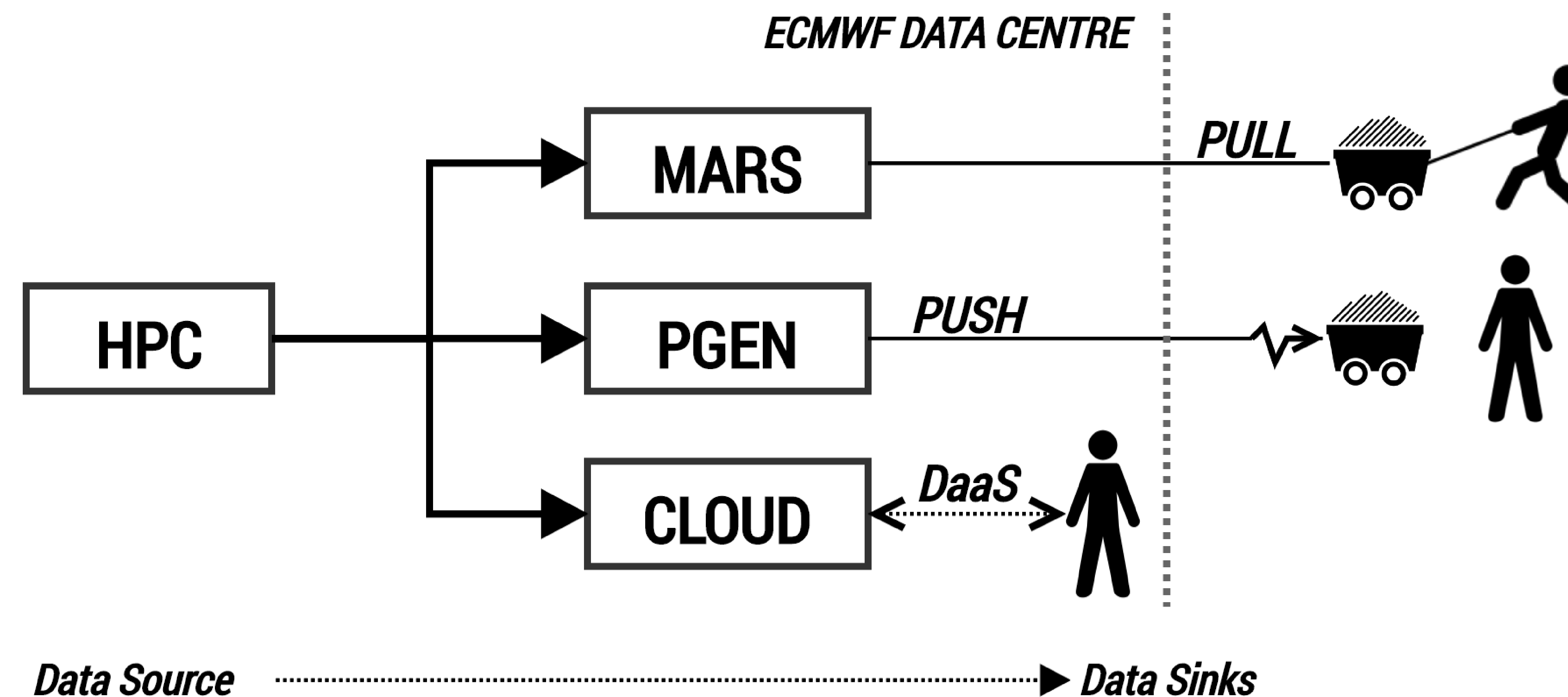
- HiDALGO consists of four socio-economic pilots
 - **Human migration modelling (e.g. in regions of conflict)**
 - **Urban air pollution modelling**
 - Social network modelling
 - Epidemic modelling (regional COVID-19 modelling)
- Emphasis on distributed modelling-on-demand
 - Workflows are user-driven, often for emergent situations
 - Workflows are data-centric and multi-disciplinary
 - ECMWFs global weather data must be available **dynamically and on-demand**
 - Data should be pre-processed in-situ (on European Weather Cloud) where necessary



HiDALGO

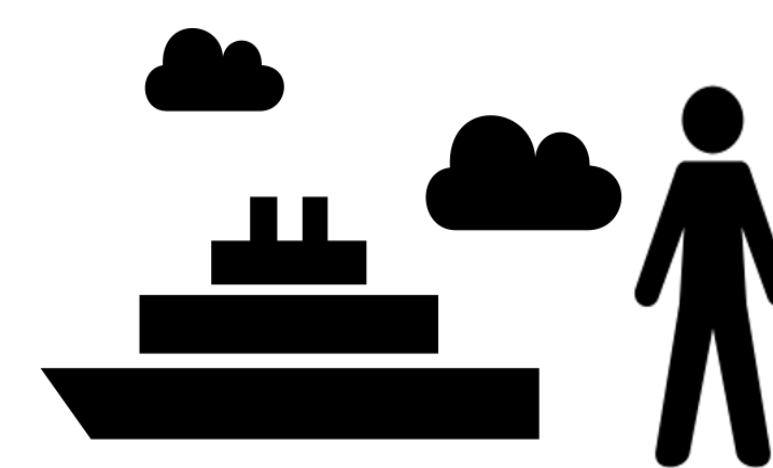


- ECMWF is building a meteorological data-as-a-service API for HiDALGO
 - Provide on-demand, dynamic access to real-time forecast data
 - Enabling direct, managed access to HPC data storage (FDB)
 - Accessible via European Weather Cloud or externally
 - Built on existing technologies (FDB) and new developments (Polytope)



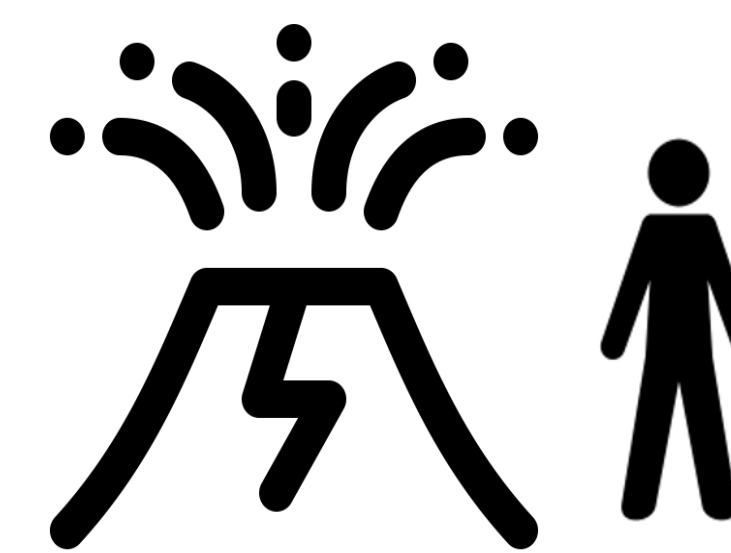
Our Ambition

- Raw data output and disseminated products are increasing in size
 - 5km ensemble will produce ~1PiB of raw data
- LEXIS & HiDALGO explore **efficient ways to access this data** directly from the HPC
- The European Weather Cloud provides a platform to **bring new and existing users to this data**
- Can we build an API that allows **more effective** access to this data?



User A

Forecast along a spatio-temporal path for transport optimization



User B

Dynamic, real-time, on-demand wind data for particle dispersion



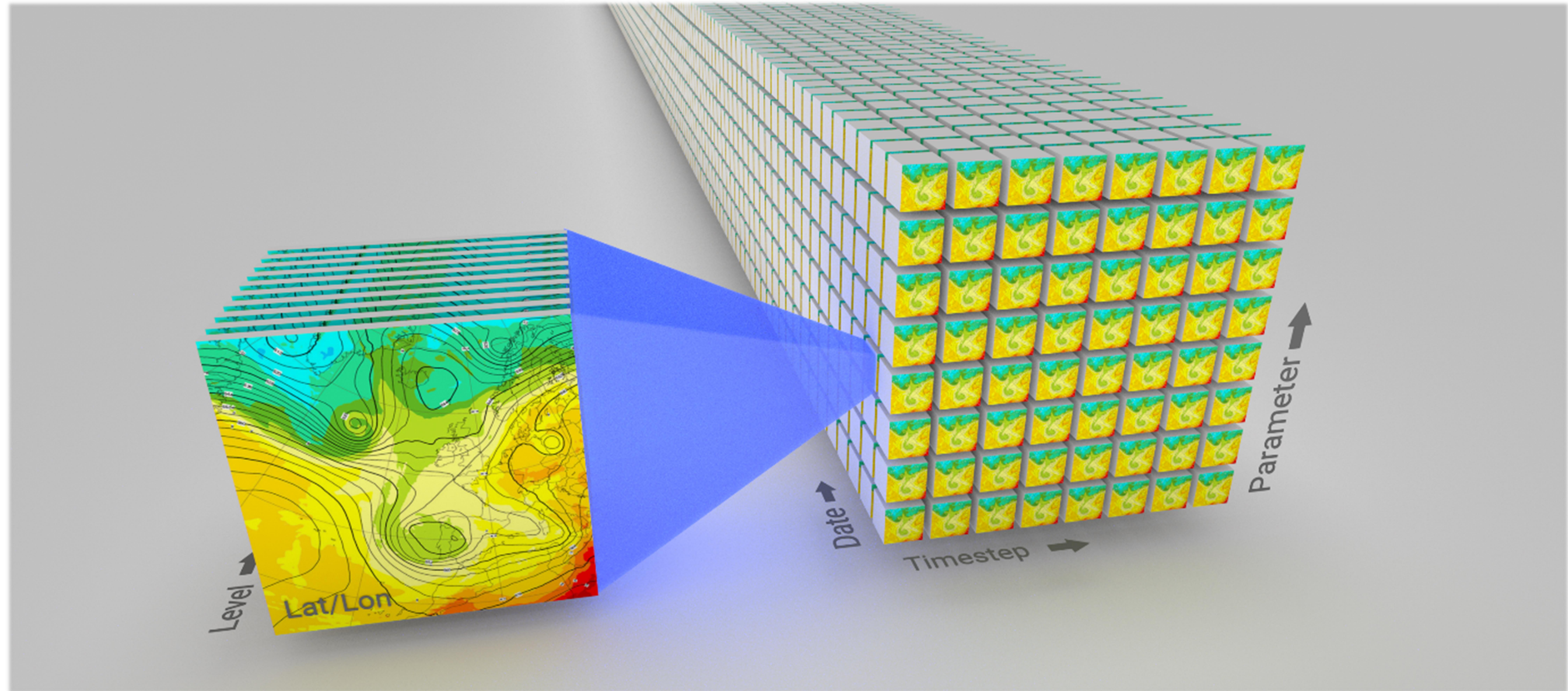
User C

Machine learning (e.g. for tropical cyclone prediction) has unknown data access patterns



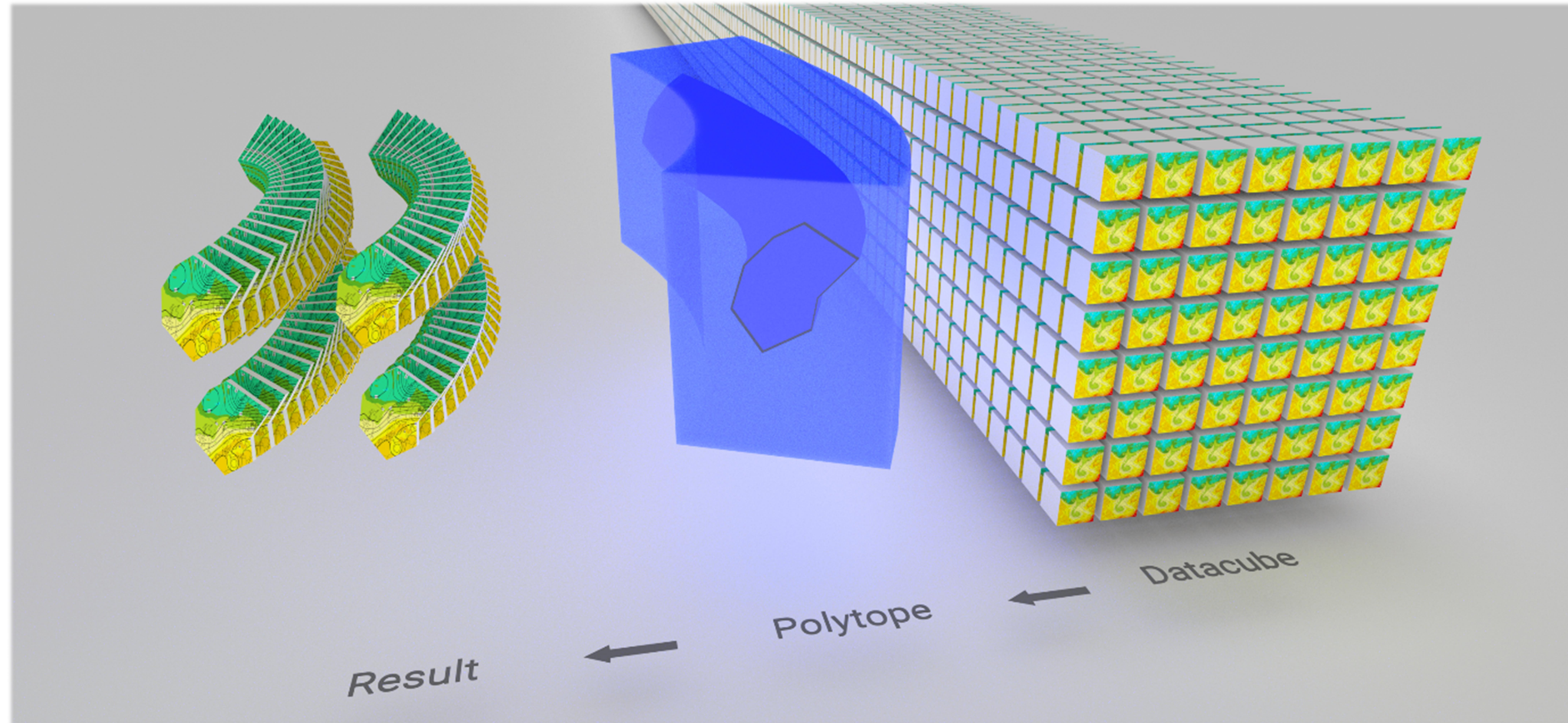
Polytope Concept

- Our meteorological data is a 6-dimensional hypercube
- Currently subset this data using orthogonal ranges (e.g. timestep = 1 to 100)



Polytope Concept

- Instead, we can subset our datacube using 6D polytopes (n-dimensional polygons)
- Backend (FDB) can leverage NVRAM to efficiently read non-sequential data



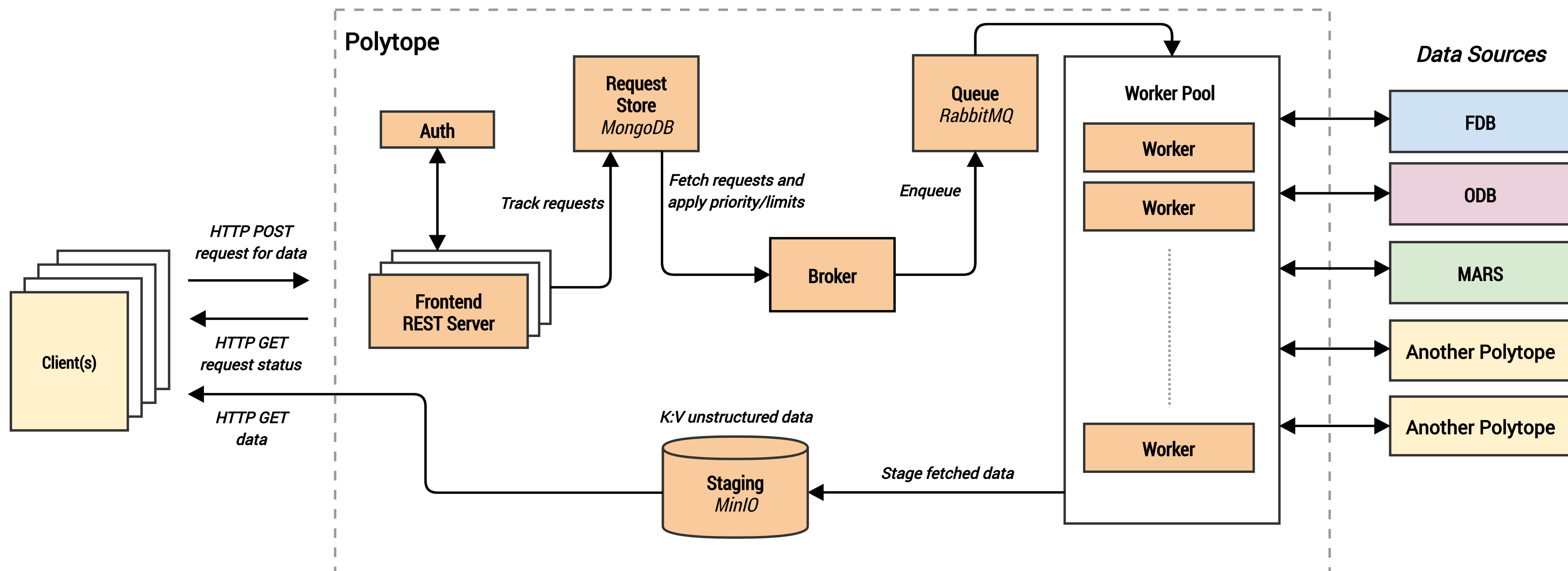
Polytope Concept

- Polytopes can be used to represent:
 - Trajectories or corridors
 - Areas and volumes
 - Points
 - Spatio-temporal paths
 - Vertical profiles
 - ... and much more
- An API that exposes this kind of access would be more efficient and more user-friendly:
 - For downstream workflows (as in LEXIS and HiDALGO)
 - For European Weather Cloud users



Polytope Design

- **Polytope** software provides data-as-a-service:
 - Connects to FDB, MARS, as well as conventional stores
 - Opensource and deployable anywhere (ad-hoc, Kubernetes, Docker-Swarm)
 - Can connect to other Polytopes to create a federated data service between data centres (as in LEXIS)
 - Will provide an interface for polytope-style requests



Polytope Design

- **Polytope** is a RESTful service with a dedicated python client:

```
client = polytope.api.Client(user_email='abc', user_key='xyz')

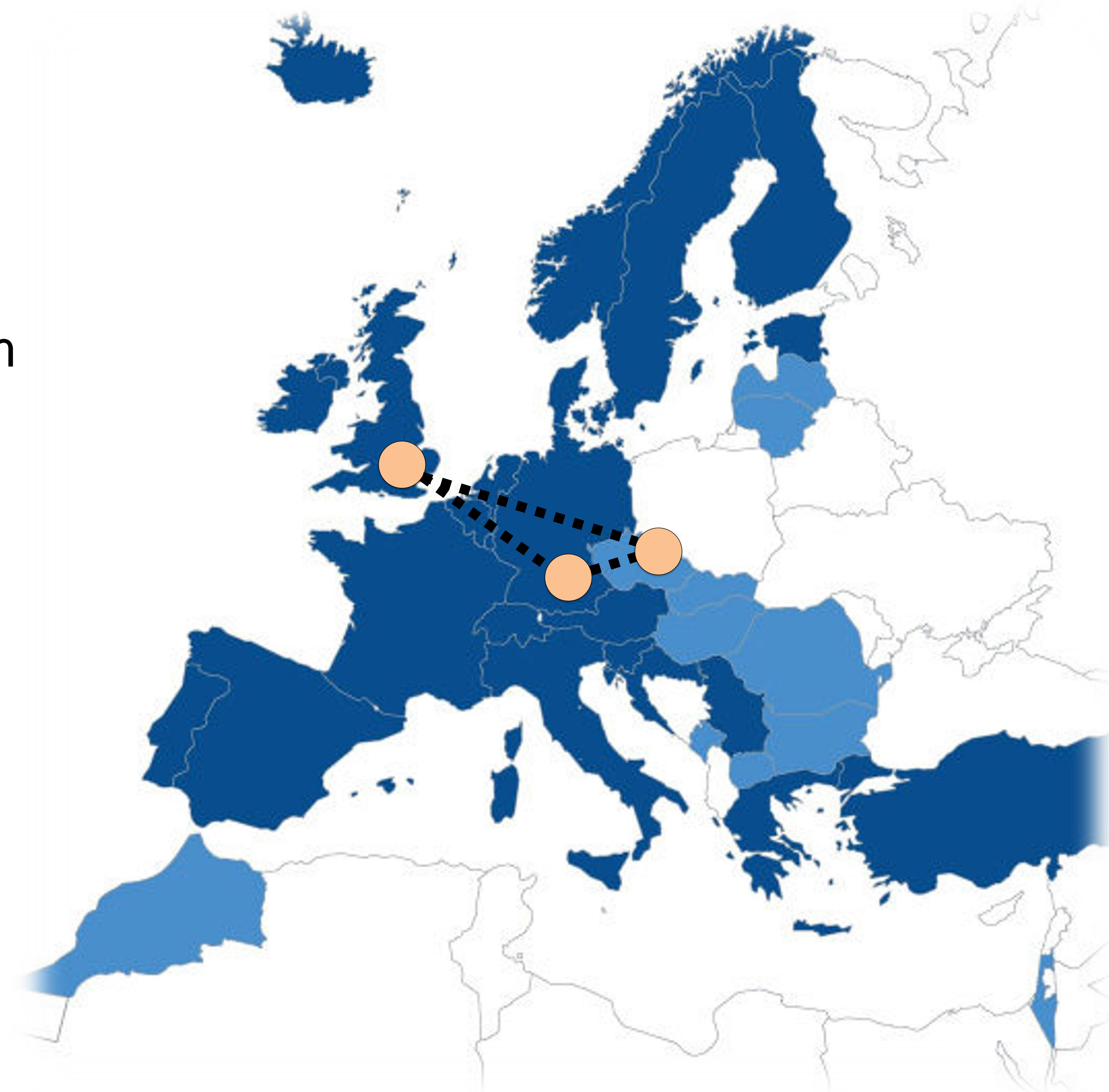
request = {
    'stream' : 'oper',
    'levtype' : 'sfc',
    'param' : '165.128/166.128/167.128',
    'step' : '0',
    'date' : '2015-01-18/to/2015-01-19',
    'type' : 'fc',
    'class' : 'ei'
}

files = client.retrieve('ecmwf-mars', request)
arr = xarray.open_dataset(files[0], engine='cfgrib')
```

* High-level interface for polytope-style requests coming soon!

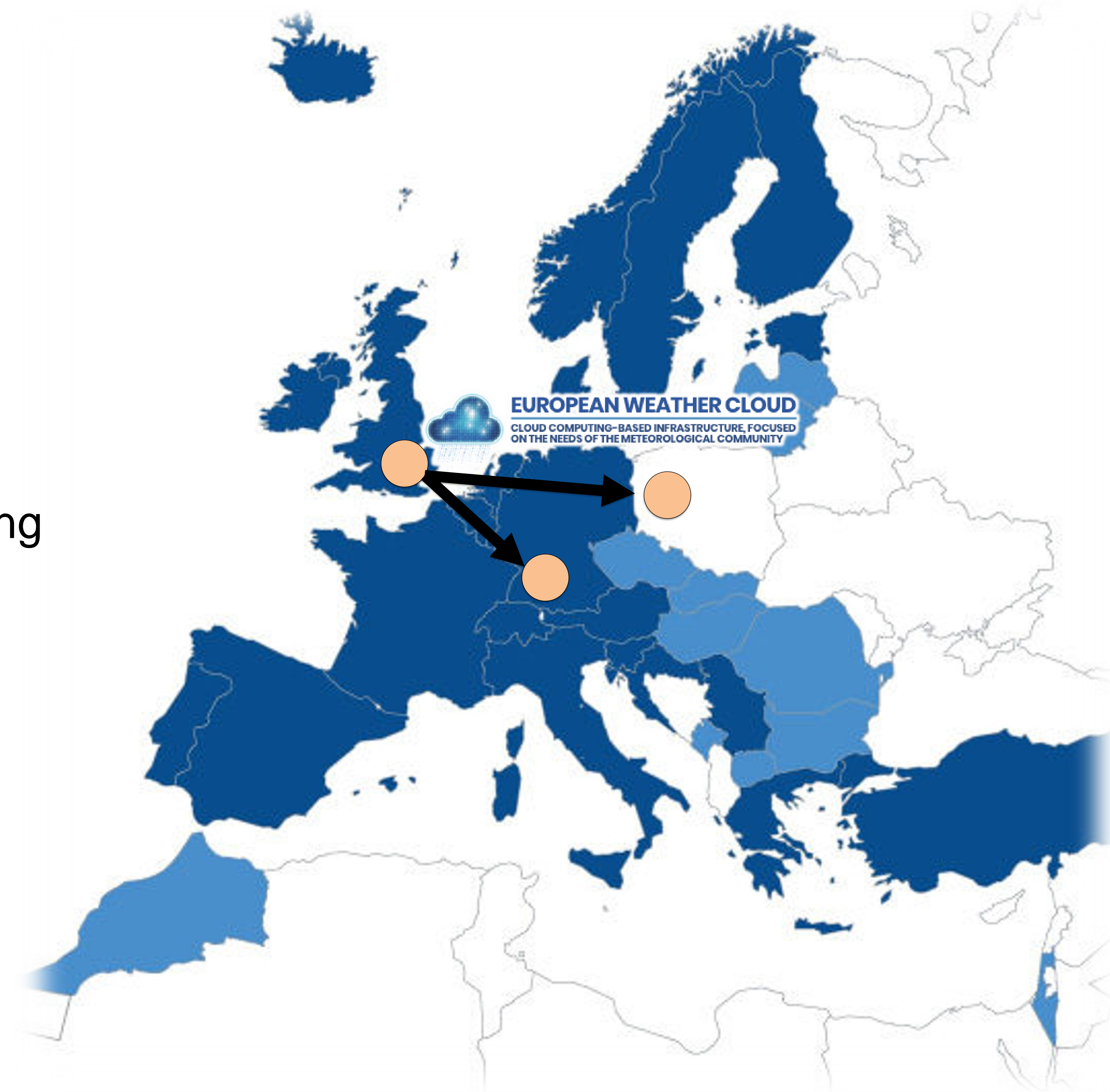
Polytope in LEXIS

- Currently deployed at ECMWF using Kubernetes
 - Connected to MARS (archive data) and HPC
- Currently deployed at LRZ (Germany) using docker-swarm
 - Connected to a local FDB to be used by regional models
 - Serving LRZ's supercomputer and cloud
- Soon to be deployed at IT4I (Czechia)
 - Connected to a Ceph FDB hosted at IT4I
 - Serving IT4I's supercomputer and cloud
- Will allow data access from anywhere, granting polytope-style feature extraction in-situ with the data



Polytope in HiDALGO

- Currently deployed at ECMWF using Kubernetes
 - Connected to MARS (archive data) and HPC
- Will be triggered on-demand by HiDALGO orchestrator
 - To deliver data to HPC & HPDA sites
 - To deliver data to European Weather Cloud for pre-processing



Conclusion

- LEXIS and HiDALGO end in Q4 2021
- They provide proof-of-concept for the convergence of HPC & Cloud for inter-disciplinary workflows
- Our developments on FDB and Polytope will continue
 - Building a data-as-a-service system for the European Weather Cloud
 - Building a powerful polytope-style feature extraction API for our real-time data



James.Hawkes@ecmwf.int