PLANET OS

How To Create the Google for Earth Data

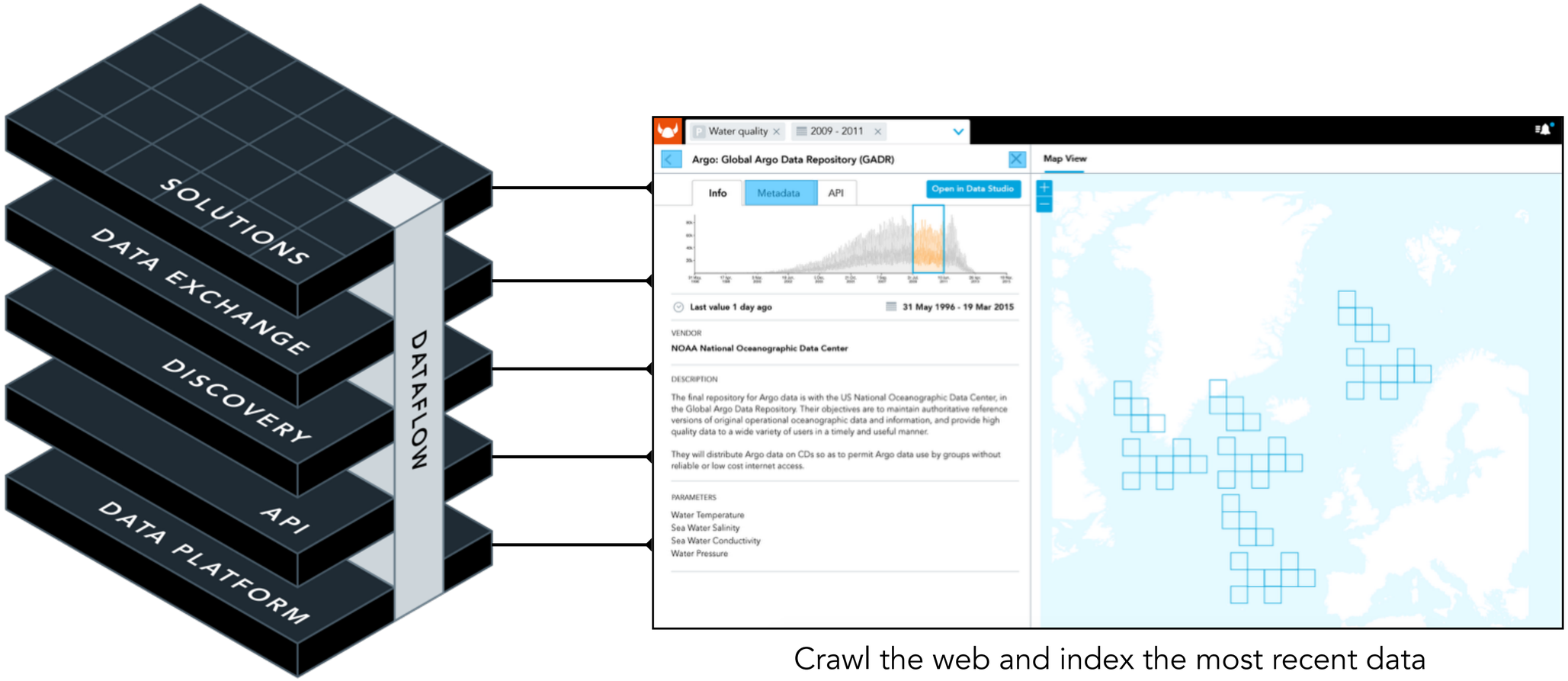
in the example of NOAA Big Data Project

Rainer Sternfeld

CEO & co-founder

Finding and accessing the right data is really hard

Planet OS Data Discovery makes it easy



Crawl the web and index the most recent data without moving the data itself until you need it

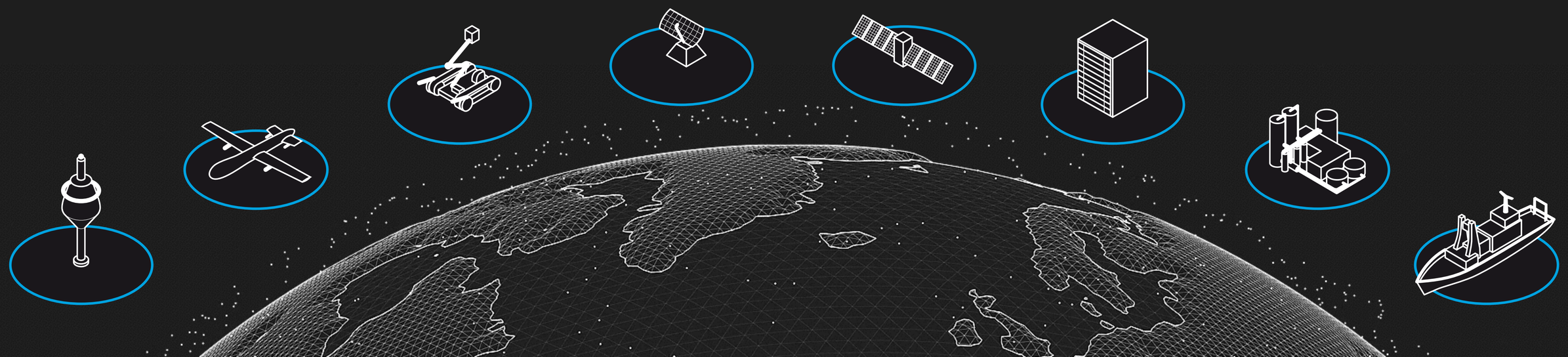
PLANET OS Cloud Platform for Industrial Sensor Networks

Sensor Data Discovery & Exchange

Search, Exploration
Visualisation, Analytics
Data Management
Marketplace
APIs

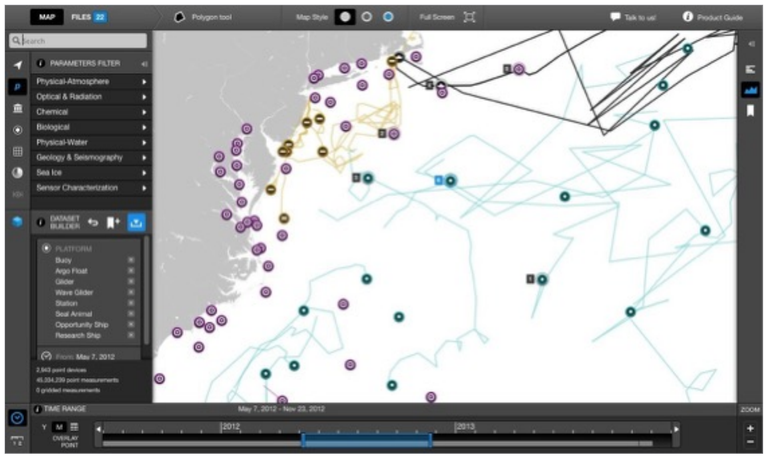
Building with:

Scala, Akka
RabbitMQ, Kafka
HDFS, HBase, Elasticsearch
Spark, Spark Streaming
GIS Libraries (raster, geometry)
Compressed array formats

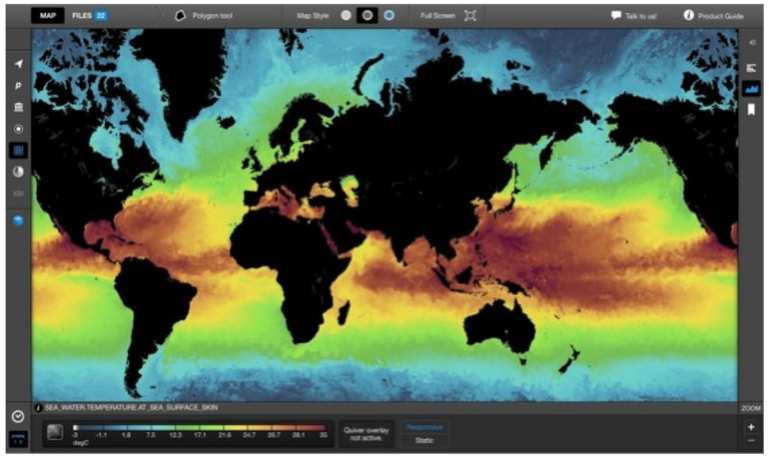


marinexplore.org — the biggest deployment of Planet OS

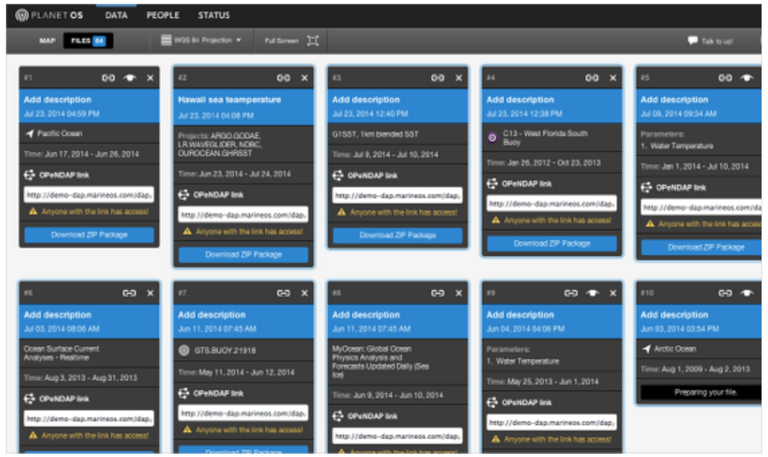
43,000+ data streams, 35 organizations, 8,000 users



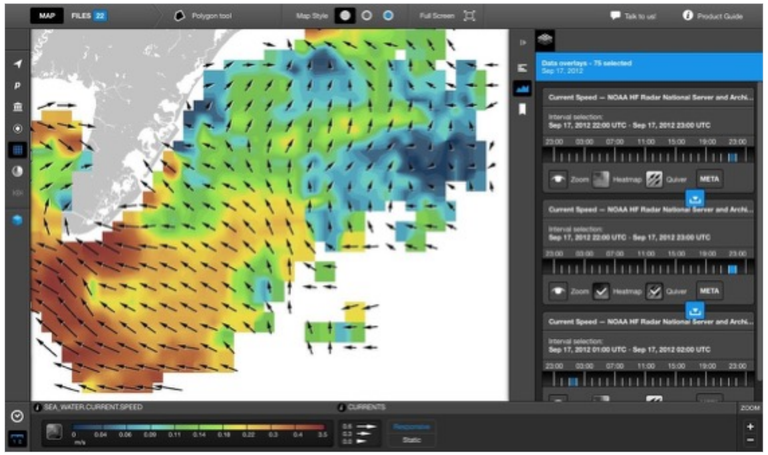
Data Discovery



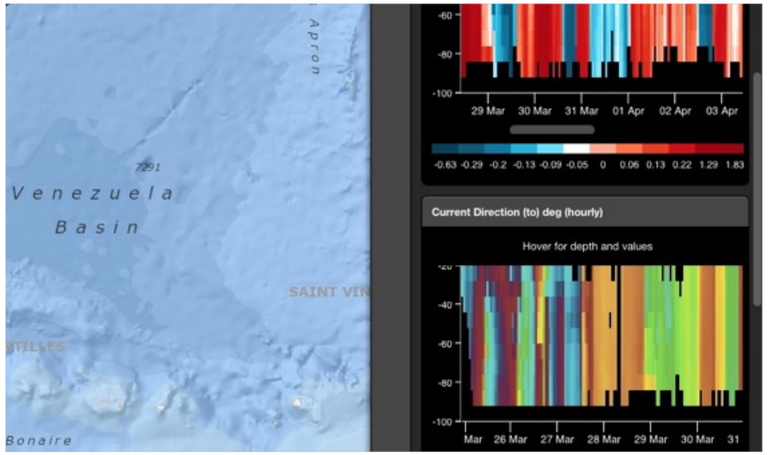
Raster data, heatmap overlays



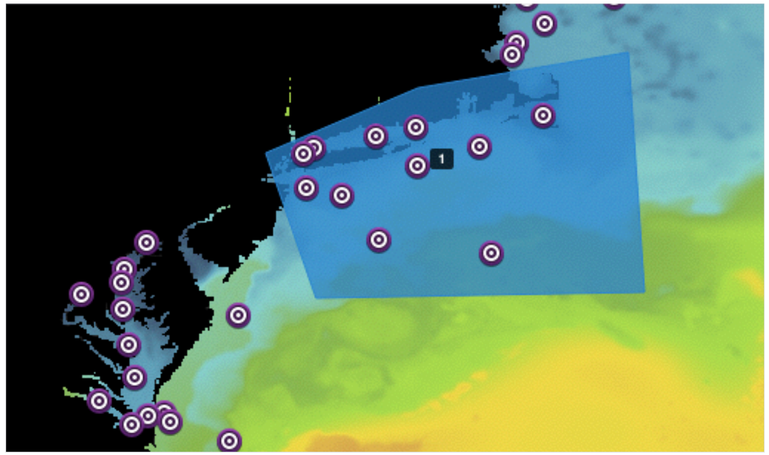
Access with third party applications



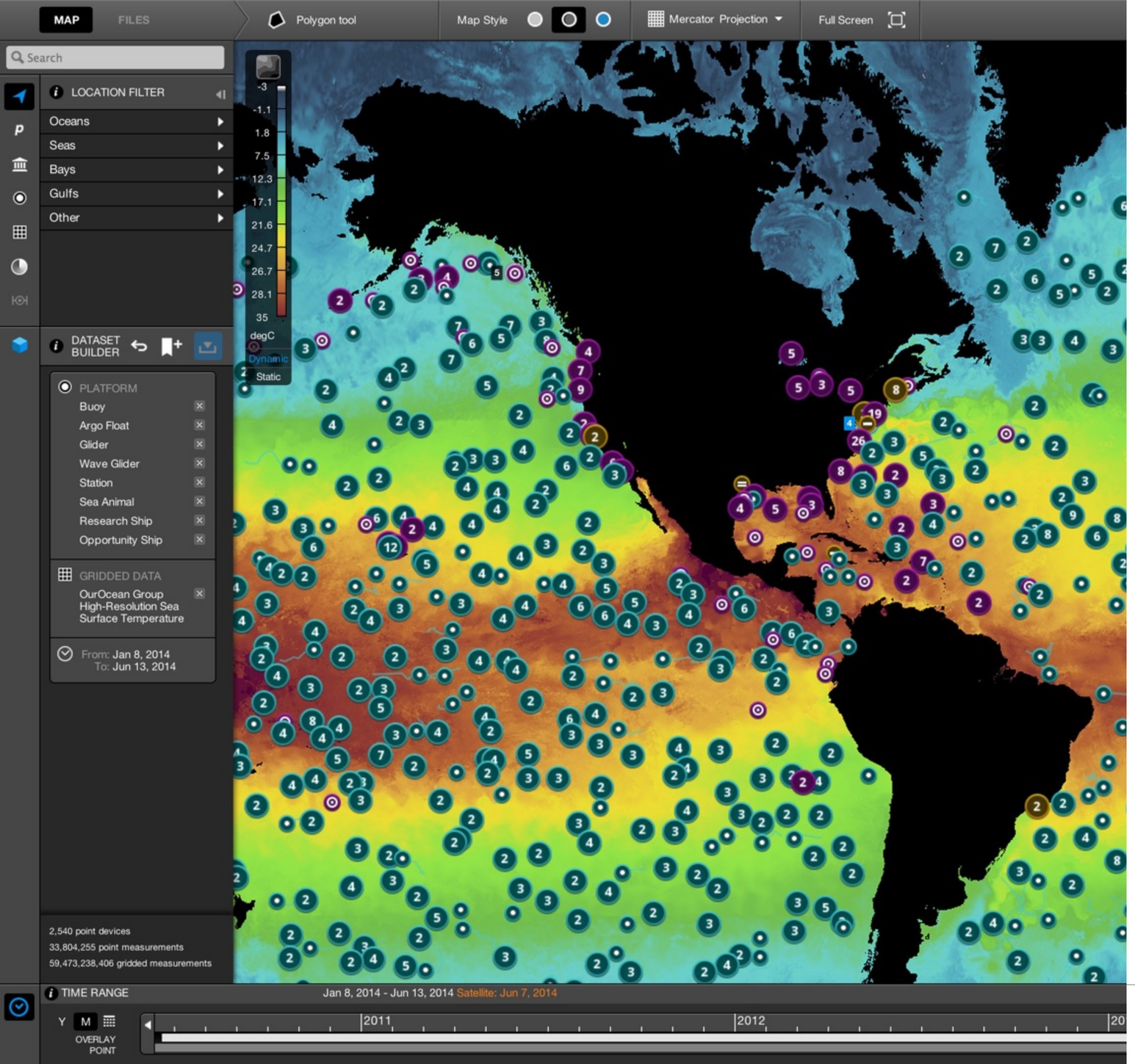
Raster data with quiver plot overlays



Rich graph visualizations



Build custom datasets



marinexplore.org

Marine data website

Visualising
Browsing
Filtering
Export

Built with:

Python/Cython
RabbitMQ, Celery
Postgres, PostGIS, Vertica
GIS Libraries (raster, geometry)
NumPy

How to make NOAA's large-scale weather and climate data easily discoverable and machine-readable?

Real-time weather & climate data is a global multi-billion dollar opportunity

NOAA's Challenge

- Tens of thousands of devices deployed in the ocean, on land, and space
- Critical for the government and industries
- Hundreds scattered web services (FTPs, flat files, THREDDS/OPeNDAP API)
- Data grows 10TB+ per day
- 26,595 NOAA datasets with ISO-19139 metadata
- 4,894 NOAA datasets with OpenDAP interface

So what's wrong with how it's done now?

An enthusiastic young researcher starts downloading data to an external HDD connected to his laptop — data keeps coming, external HDDs pile up…

the beard's getting longer and longer, and when the data is finally downloaded, we have a middle-aged, bored professor

Technical Challenges in the NOAA Big Data Project

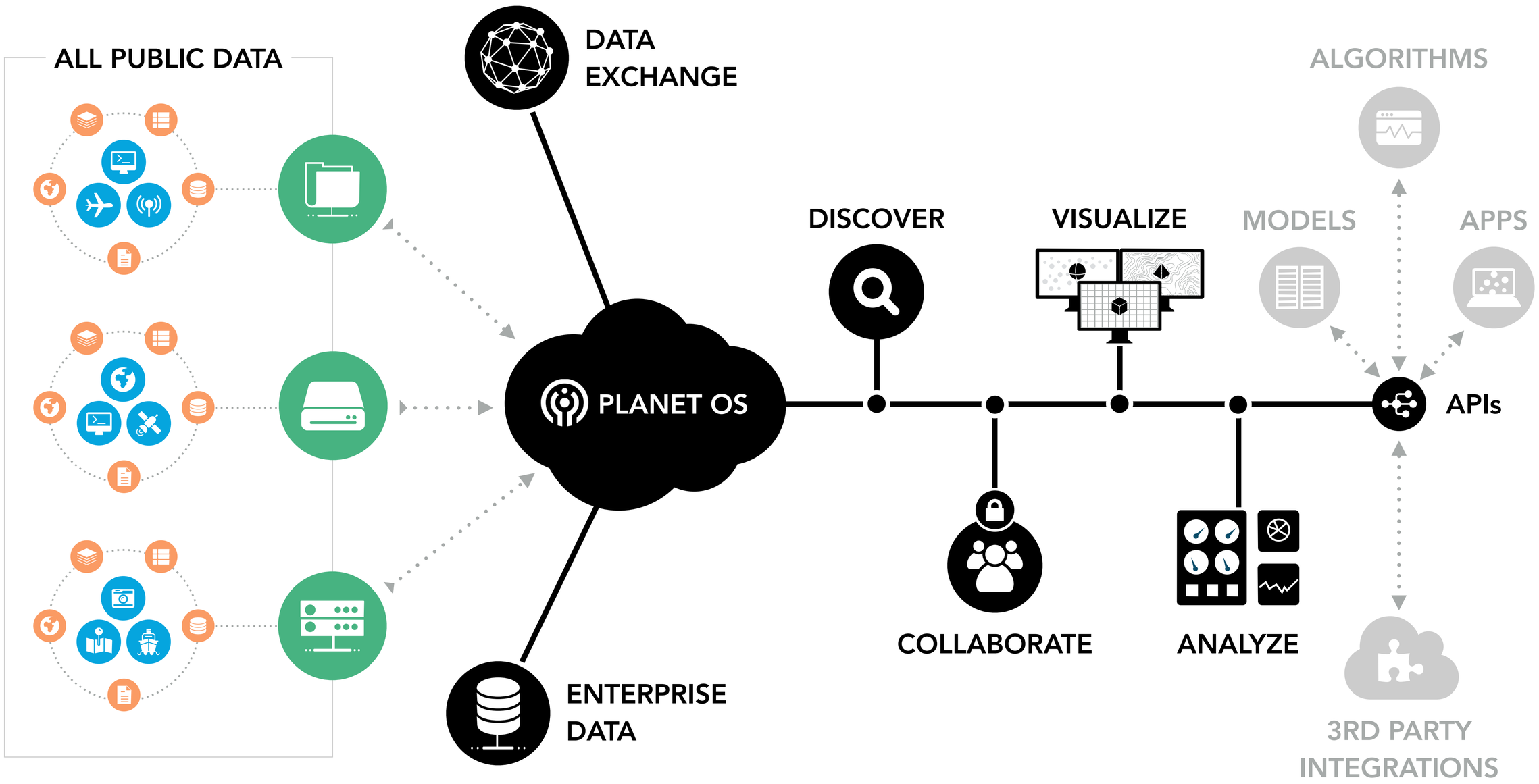
- Storing, processing, and indexing spatio-temporal time-series & array data
- Processing data at 10s (100s) of TB/day
- Transporting and processing archives at volumes 10s (100s) of PB
- Disseminating real-time data at latency of minutes
- Indexing 100K+ logical datasets and 100M+ technical datasets/files
- Providing uniform API/export for various data formats/protocols/projections

Potential solutions under consideration

- Indexing spatio-temporal and semantic metadata
- Indexing downsampled remote datasets acquired via OpenDAP and others
- Store chunked array data (MBs) in object store (e.g. Amazon S3)
- Provide on-demand computational infrastructure for analyzing data (e.g. Amazon)

What would make it even better?

- Incremental data compression
- BitTorrent-like data dissemination
- Sending pre-filtered data to consumers (e.g. by area)
- Computations scheduled next to the data storage
- Fast interconnect (10 GBit / Inifiniband) and GPUs
- Run analytical scripts (eg IPython Notebook, Matlab) to work with array data in the cloud


ALL PUBLIC DATA
DATA EXCHANGE
ENTERPRISE DATA
PLANET OS
DISCOVER
VISUALIZE
COLLABORATE
ANALYZE
ALGORITHMS
MODELS
APPS
APIs
3RD PARTY INTEGRATIONS

PLANET OS

We index your world