Flood Detection

Providing early warning of floods

eNomads

Team Members

- Andy Bowes @AndyJBowes
- Amit Varshney @Amitheunlimited
- Paulraj Madasamy @coolpaulraj
- Priyaa Thavasimani @PriyaaRamesh
- Siddhant Baviskar @siddhantfri3nds

Big Problem

Huge amount of weather data

Complex data structures

Propriety data formats

Locating the data sources is difficult

Lack of real-time data causes real world problems (e.g flooding, heavy snowfall etc)

Big Idea

Open up weather data to 3rd parties

Provide a simple interface

Don’t swamp users with too much data

Provide information at almost real-time

Provide sample applications which demonstrate the potential

Overview of Solution

Existing

Data provided from many sources.

Held in complex formats (e.g. HDF5)

Volume of available data is very large.

Target

Pool information into a single, trusted source.

Provide access via a simple API

Only key pertinent, information

Sources of Information

Percipitation (Completed)

NASA’s Percipitation Measurement Mission - http://pmm.nasa.gov/

Held in HDF5 format

Drought (Planned)

NASA’s Earth Observatory - http://earthobservatory.nasa.gov/

Social Media

Monitor Social Media Interactions to look for mentions of specific key words (e.g. ‘#flood’) and capture geolocation of these tweets to look for clusters.

Technical Solution

Provide a globally available REST Interface which returns manageable chunks of data in JSON format.

Create extractors to download & parse HDF5 Percipitation records.

Store Data in highly performant NoSql Database.

Provide a sample Web Application which demonstrates the capabilites of the API.

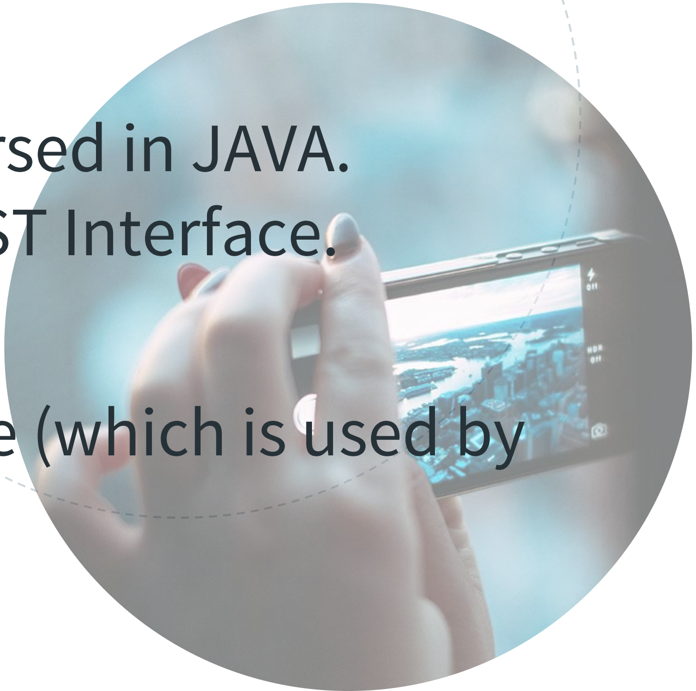
Technologies Employed

Web Application is created using R & Shiny

Highly performant REST API implemented using Python/Tornado.

Information is downloaded and parsed in JAVA. Uploaded to the database via a REST Interface.

Data stored in Basho/Riak database (which is used by “The Weather Channel”.



Example REST Request

Fetch Rainfall for a given location between 2 dates:

http://earthlive.westeurope.cloudapp.azure.com:8888/rainfall/53o222/11o111/20160411/20160415

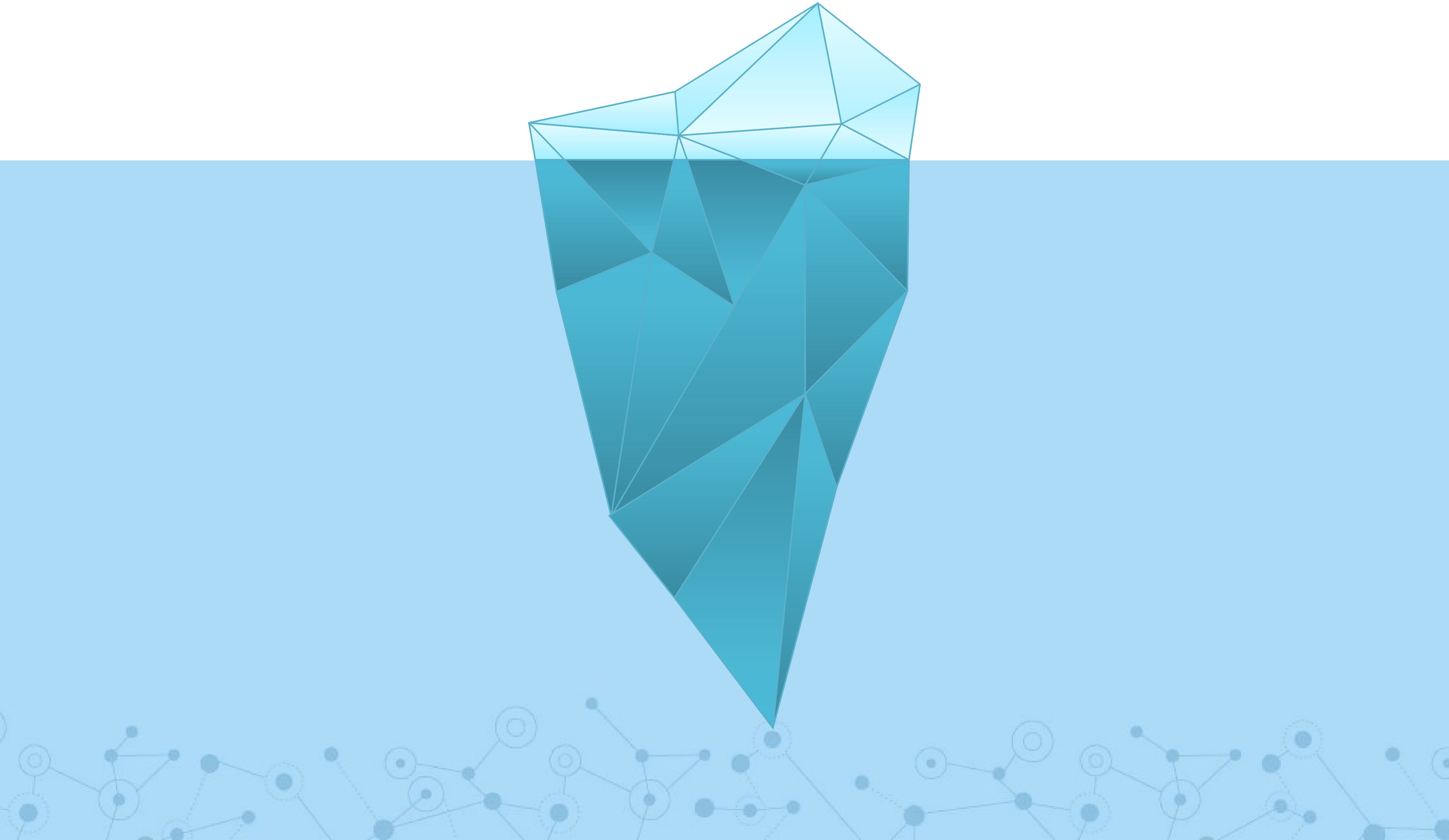
Returns:

{"rainfall": [[{"latitude": 55.00984, "timestamp": "201604131602", "precipitation": 2, "longitude": -1.559},

{"latitude": 55.00952, "timestamp": "201604131602", "precipitation": 33, "longitude": -1.564},

{"latitude": 55.00944, "timestamp": "201604131602", "precipitation": 38, "longitude": -1.5527}..

This is just the tip of the app of the iceberg ....



Thanks!

Any questions?