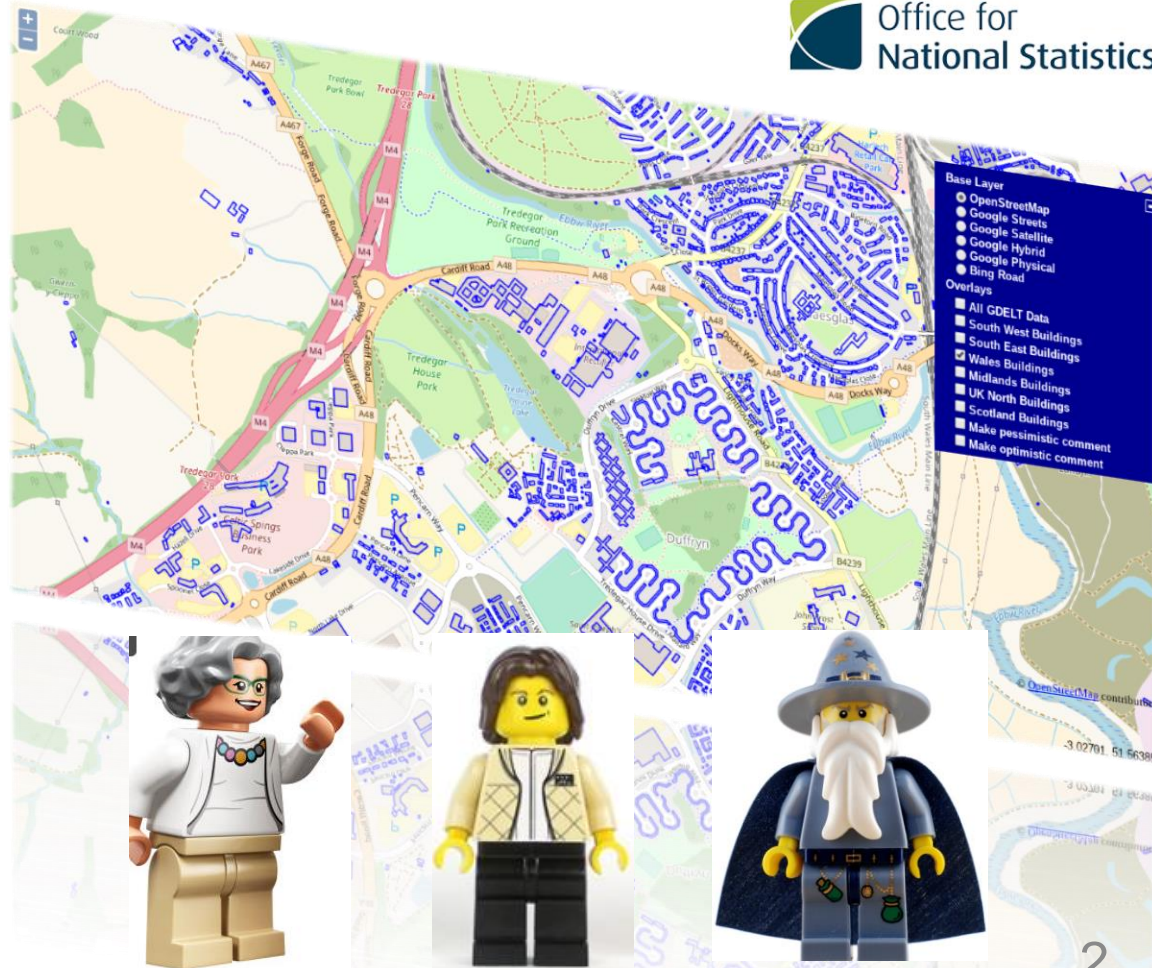


UN Global **Trusted** Platform Data, Applications and Services

Proof of Concepts

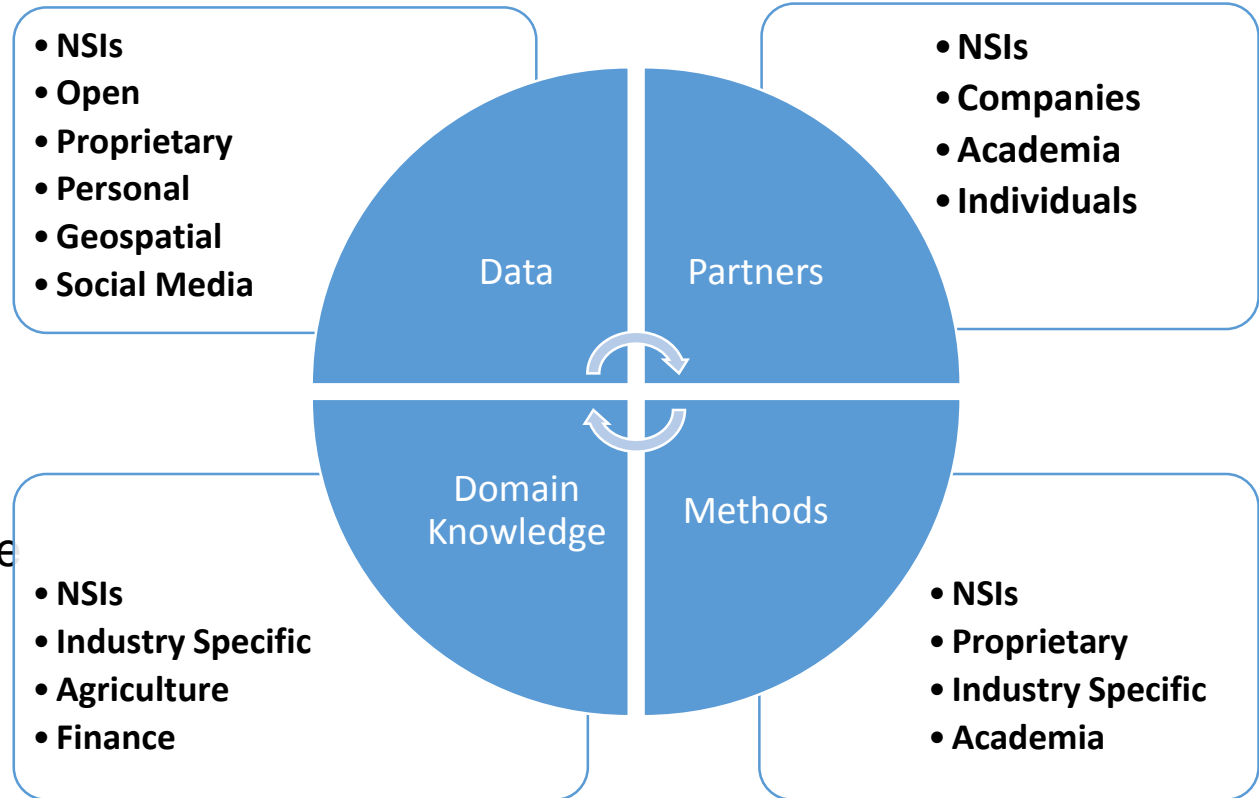
Approach

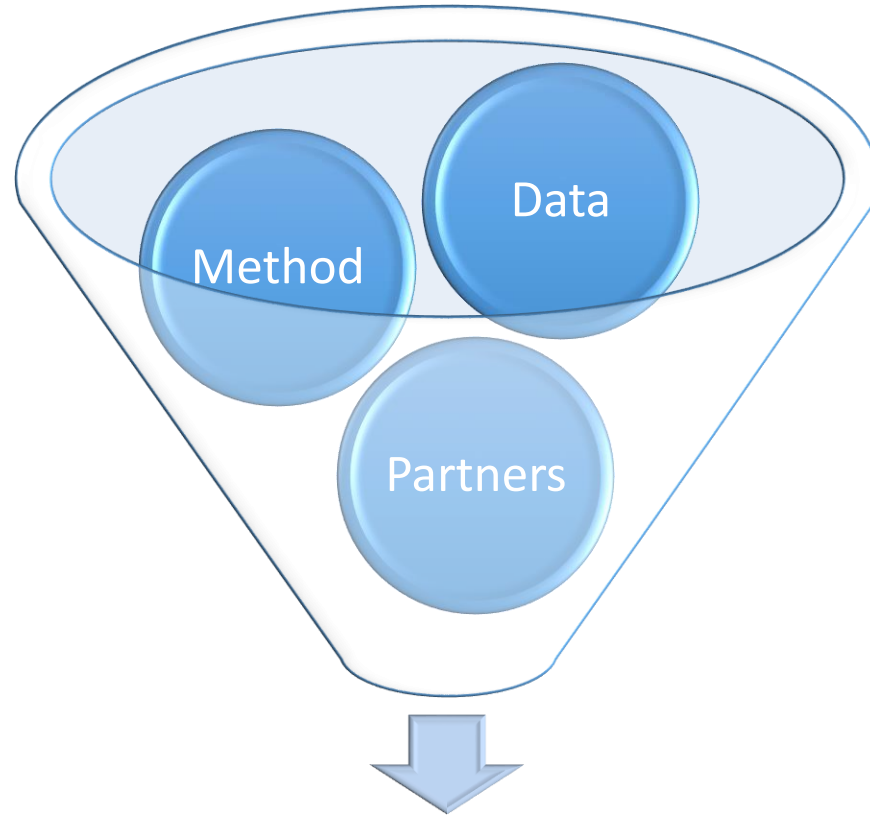
- Minimal Viable Team
 - 2 x Technical Resources
- Wardley Maps
- Principles
- Configure / Show / Iterate
- Cloud Computing
 - Consumption Based Pricing
 - Commodity Infrastructure
- Agile Delivery
 - Working with users
 - Prototyping
 - Iterative Development



Research

- 160+
 - Platforms
 - Products
 - Services
- What did we find
 - Data
 - Partners
 - Domain Knowledge
 - Methods
- Trust





Data Collaborative

Data Collaboratives / Use Cases

Satellite Imagery
Geospatial

Social Media

Scanner Data &
Online

Multi Source Data

Mobile Data

Training / Skills
Capacity Building

Data Collaborative Accelerator

Problem Marketplace / Collaboration Requests

Trusted
Data & Metadata

Trusted
Methods

Documentation
Code Library

Trusted
Partners

People Bios
Company Profiles

Trusted
Learning

Core Services

Sensitive
Data
Services

Marketplace
Services

Collaboration
Development
Environment
(IDE)

Workflow
Services

Publishing
Services

Identity
Management

Analytics
Services

Data Ingest
Services

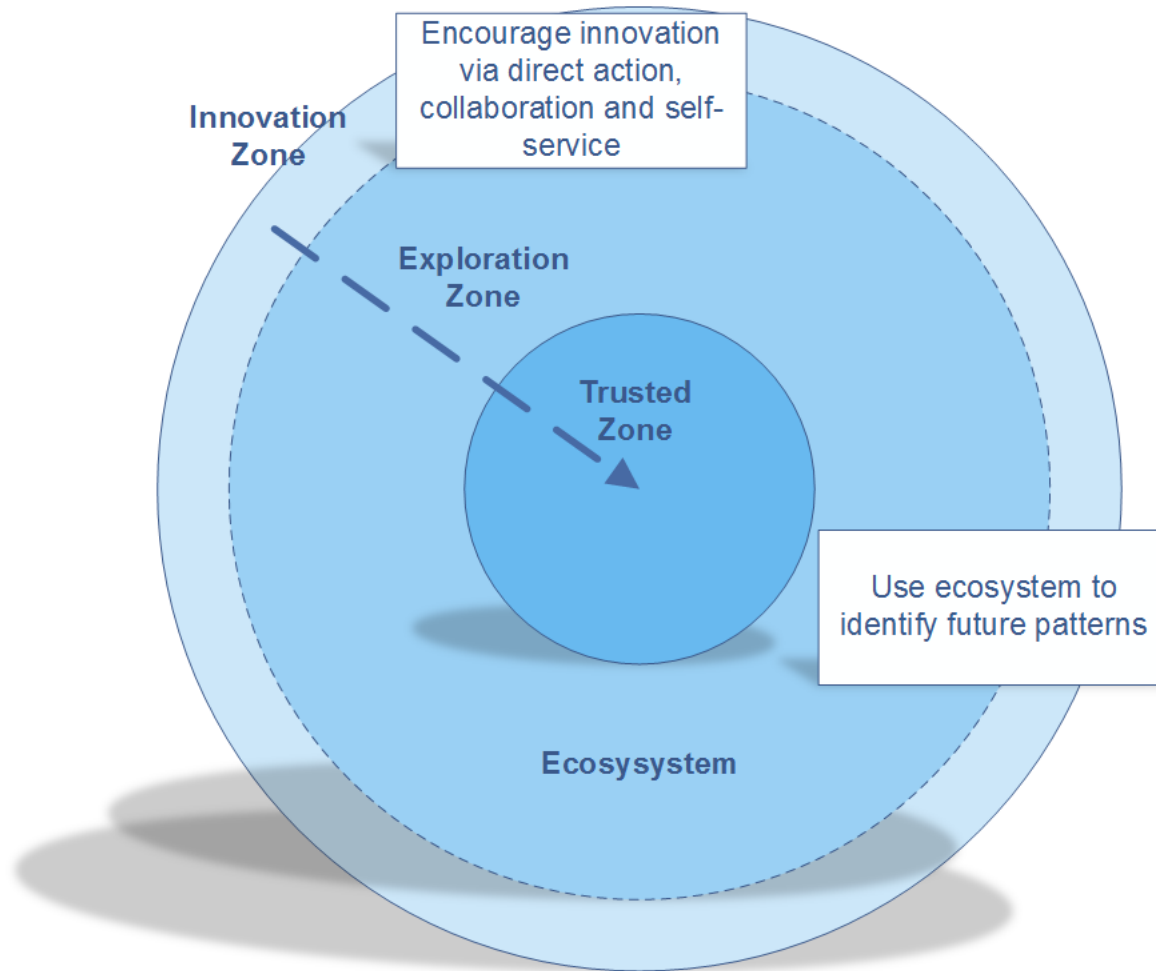
Cloud Infrastructure (IaaS)

Compute

Storage

Data Policy Framework

Common Technology Infrastructure

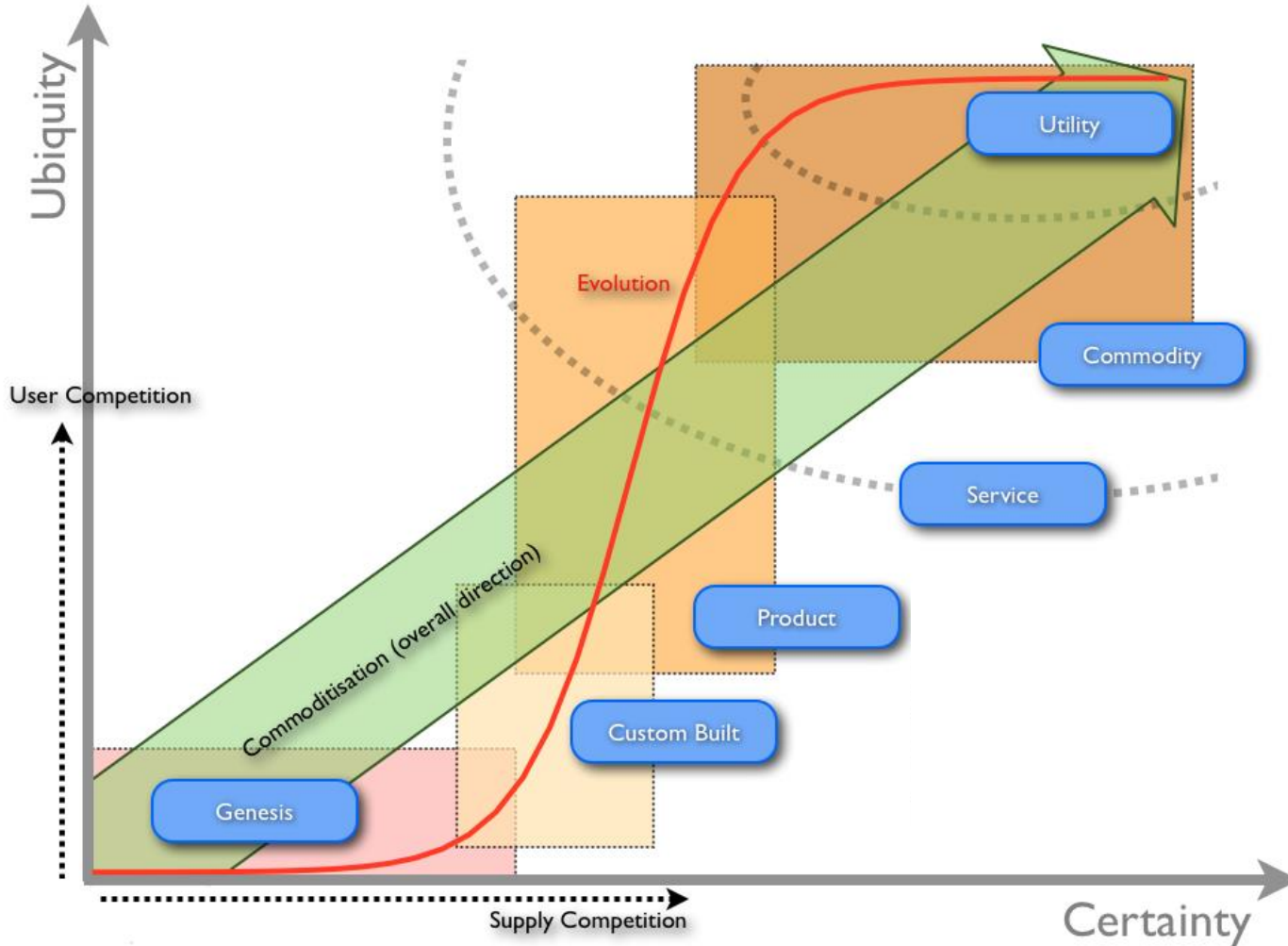


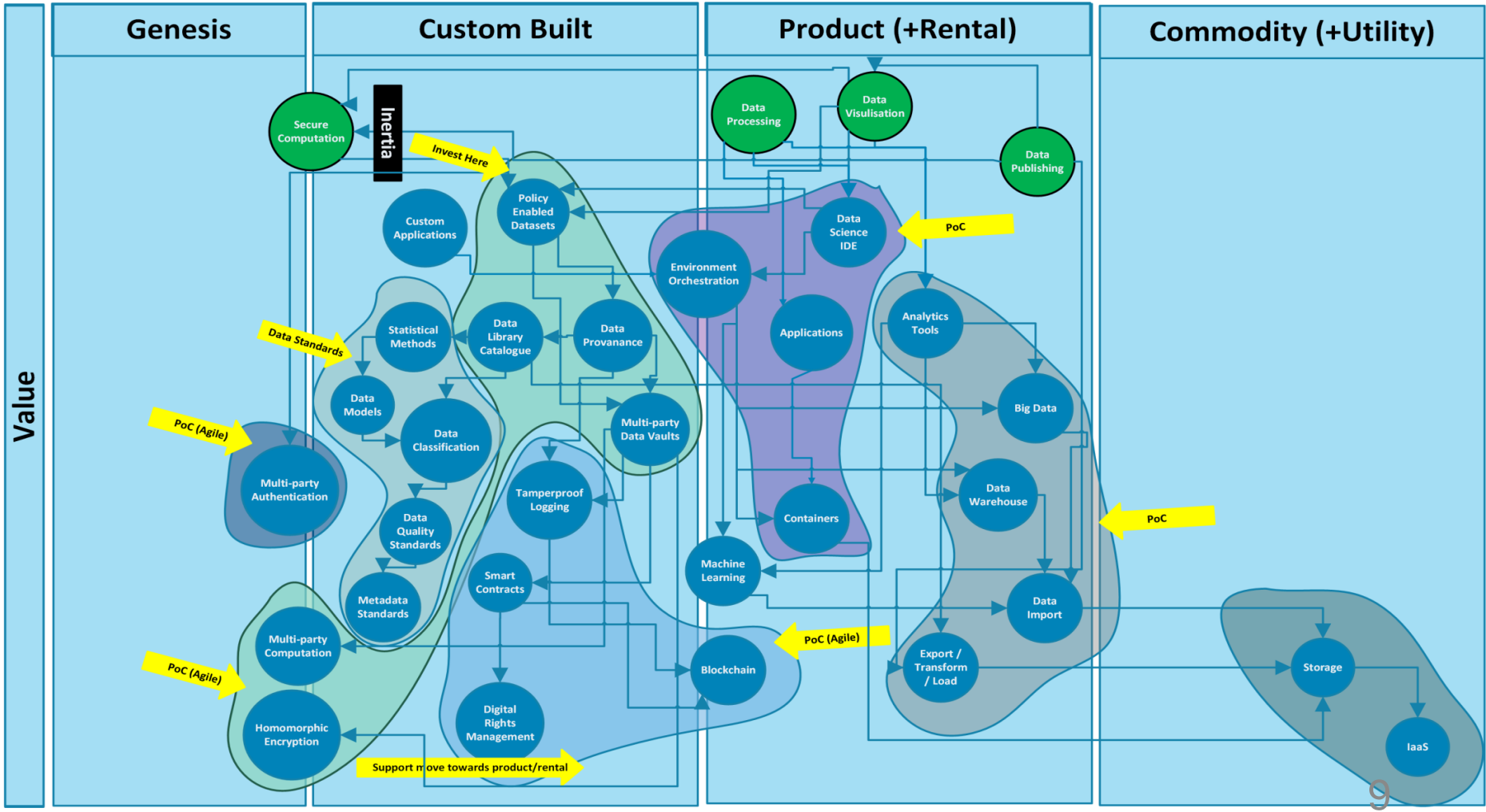
Proof of Concept Wardley Maps

<https://medium.com/wardleymaps/on-being-lost-2ef5f05eb1ec>

<https://www.youtube.com/watch?v=LZrINICJxUY>

The process of evolution repeats ...





Gaps / Next Steps

- Standards
 - Trusted Data
 - Trusted Methods
 - Trusted Partners
 - Trusted Learning
- Data Policy Framework
- Common Technology Infrastructure

Gaps / Next Steps

- Build Marketplace/Accelerator
 - Use to test business model/case
 - Collecting intelligence on data collaboratives
 - NSIs

Procurement

This page is intentionally left blank

Amazon AWS Costs

July \$30

August \$480

September \$683

October \$1,664

Total \$2,857

Proof of Concept **Demonstrations**

Data Collaboratives / Use Cases

Satellite Imagery
Geospatial

Social Media

Scanner Data &
Online

Multi Source Data

Mobile Data

Training / Skills
Capacity Building

Data Collaborative Accelerator

Problem Marketplace / Collaboration Requests

Trusted
Data & Metadata

Trusted
Methods

Documentation
Code Library

Trusted
Partners

People Bios
Company Profiles

Trusted
Learning

Core Services

Sensitive
Data
Services

Marketplace
Services

Collaboration
Development
Environment
(IDE)

Workflow
Services

Publishing
Services

Identity
Management

Analytics
Services

Data Ingest
Services

Cloud Infrastructure (IaaS)

Compute

Storage

Data Policy Framework

Common Technology Infrastructure

Data Collaboratives / Use Cases

Satellite Imagery
Geospatial

Social Media

Scanner Data &
Online

Multi Source Data

Mobile Data

Training / Skills
Capacity Building

Data Collaborative Accelerator

Problem Marketplace / Collaboration Requests

Trusted
Data & Metadata

Trusted
Methods

Documentation
Code Library

Trusted
Partners

People Bios
Company Profiles

Trusted
Learning

Core Services

JANA

Marketplace
Services

Anaconda
Enterprise

Workflow
Services

GeoServer
Shiny

Identity
Management

Rstudio
Jupyter
Notebooks
Athena

Apache
NiFi

Cloud Infrastructure (IaaS)

AWS S3 / EC2

AWS EC2

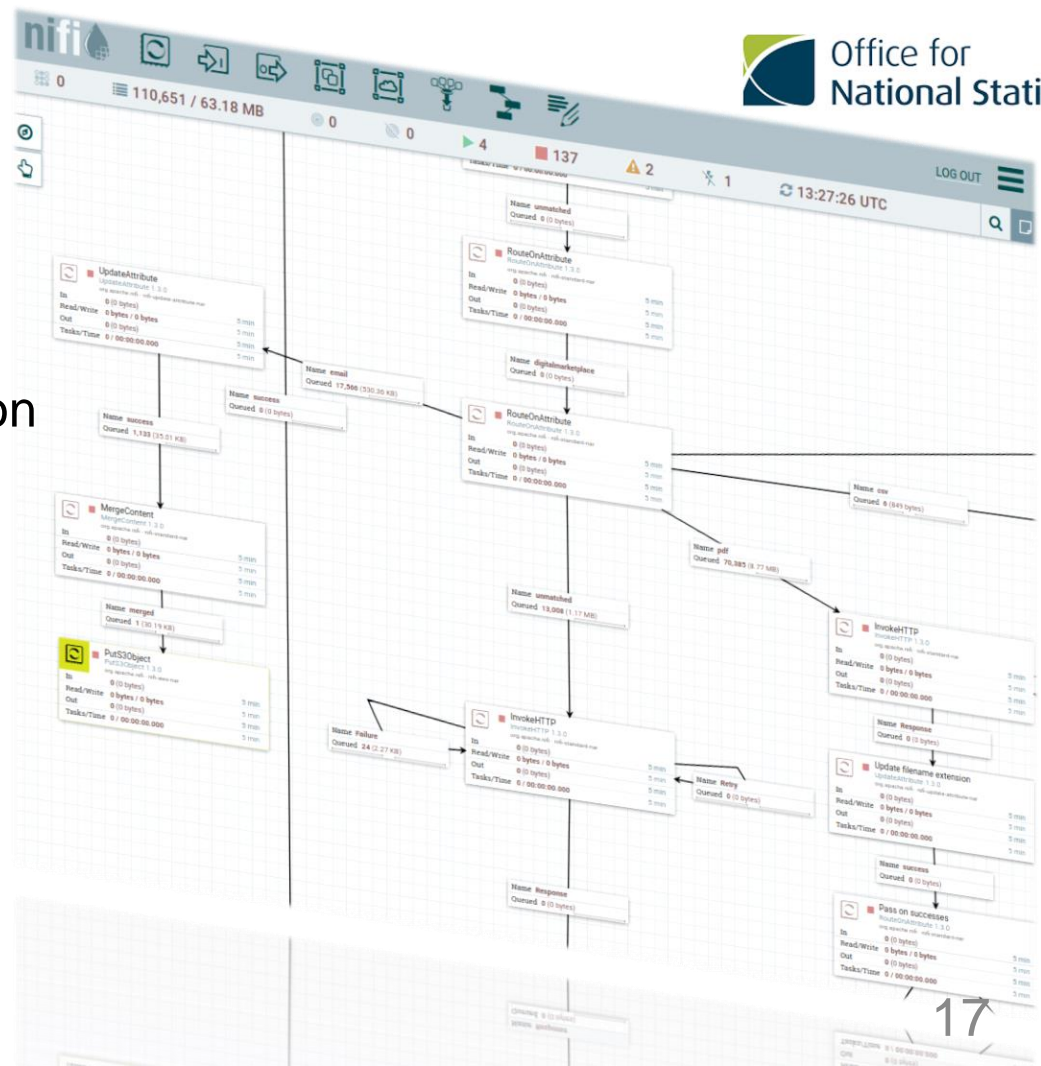
Data Policy Framework

Common Technology Infrastructure

Proof of Concept **Apache NiFi**

Apache NiFi

- CSV/Excel/File Processing
- Data Validation/Transformation
- Social Media – Twitter
- Satellite Data
- Integration with AWS Athena
- Integration with Solr/Banana
- Web Scrapping
- Public APIs
- [Apache NiFi](#)



Apache NiFi

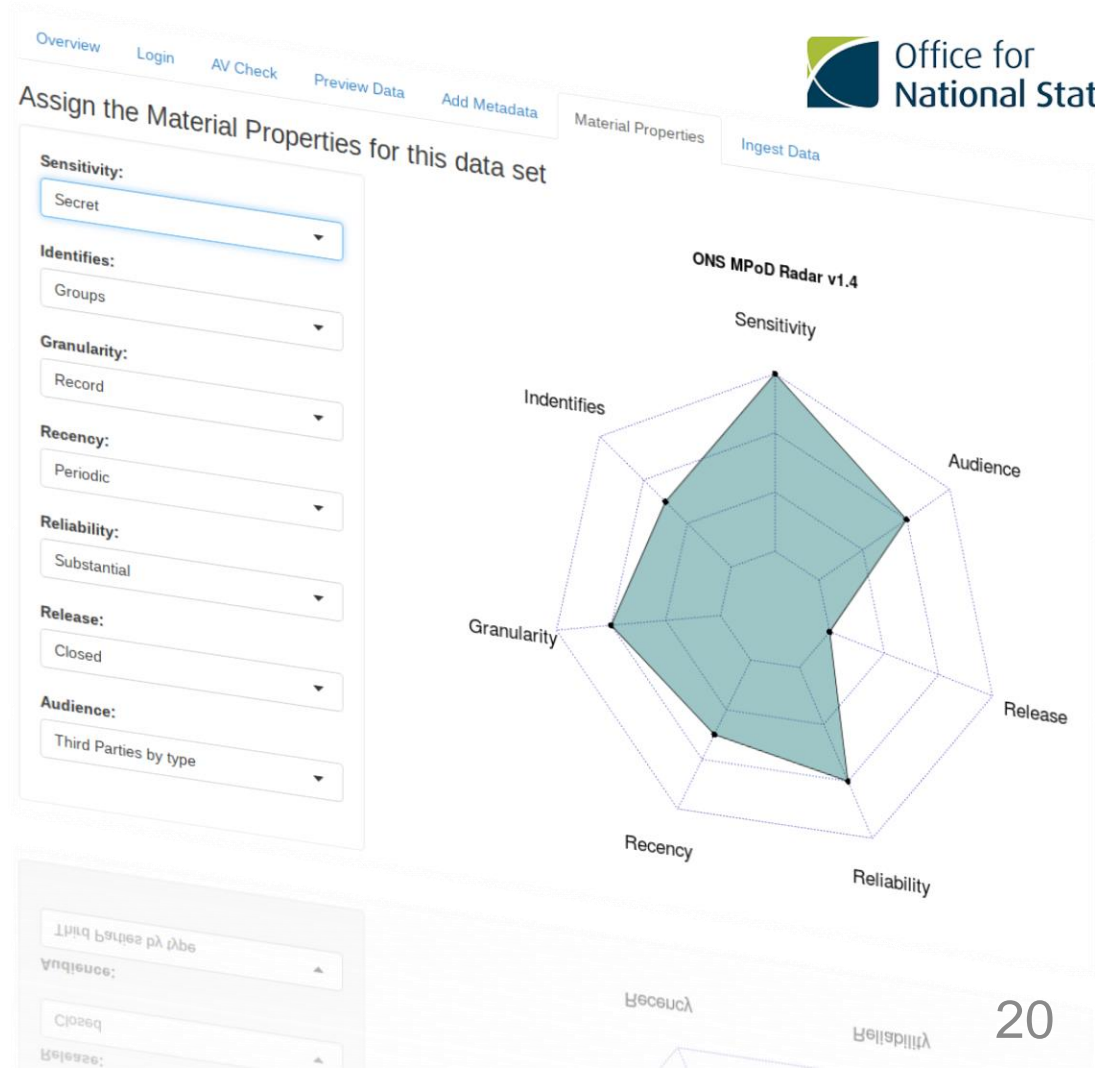
- Open Source
- Niagara Files (NyFy)
- Highly Scalable
- Cluster Support
- Cloud Agnostic



Proof of Concept **Material Properties of Data**

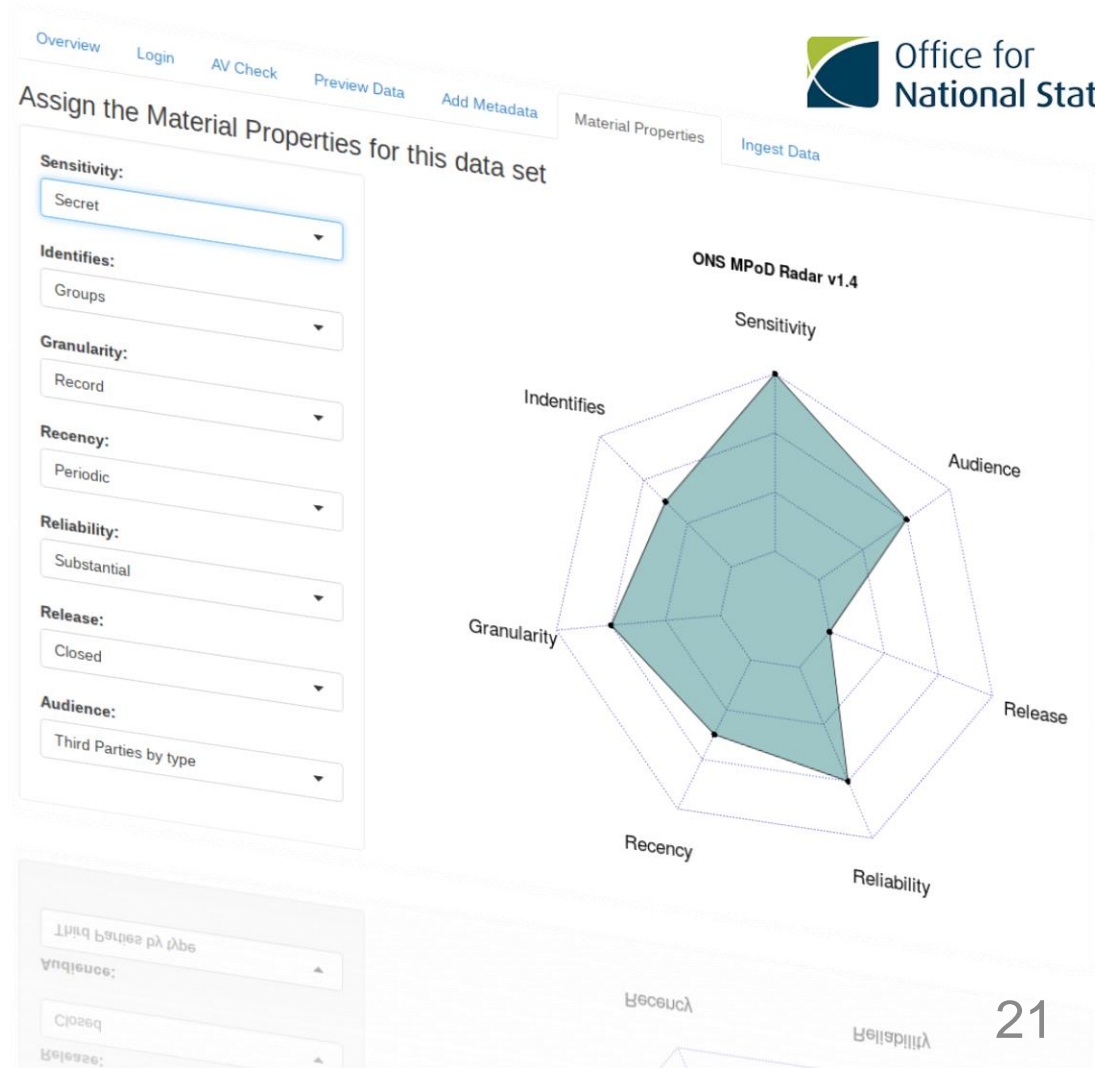
Material Properties of Data

- Data Ingest Workflow
- Antivirus Check
- Preview Data
- Classify Data
- Material Properties
- Ingest Data



Material Properties of Data

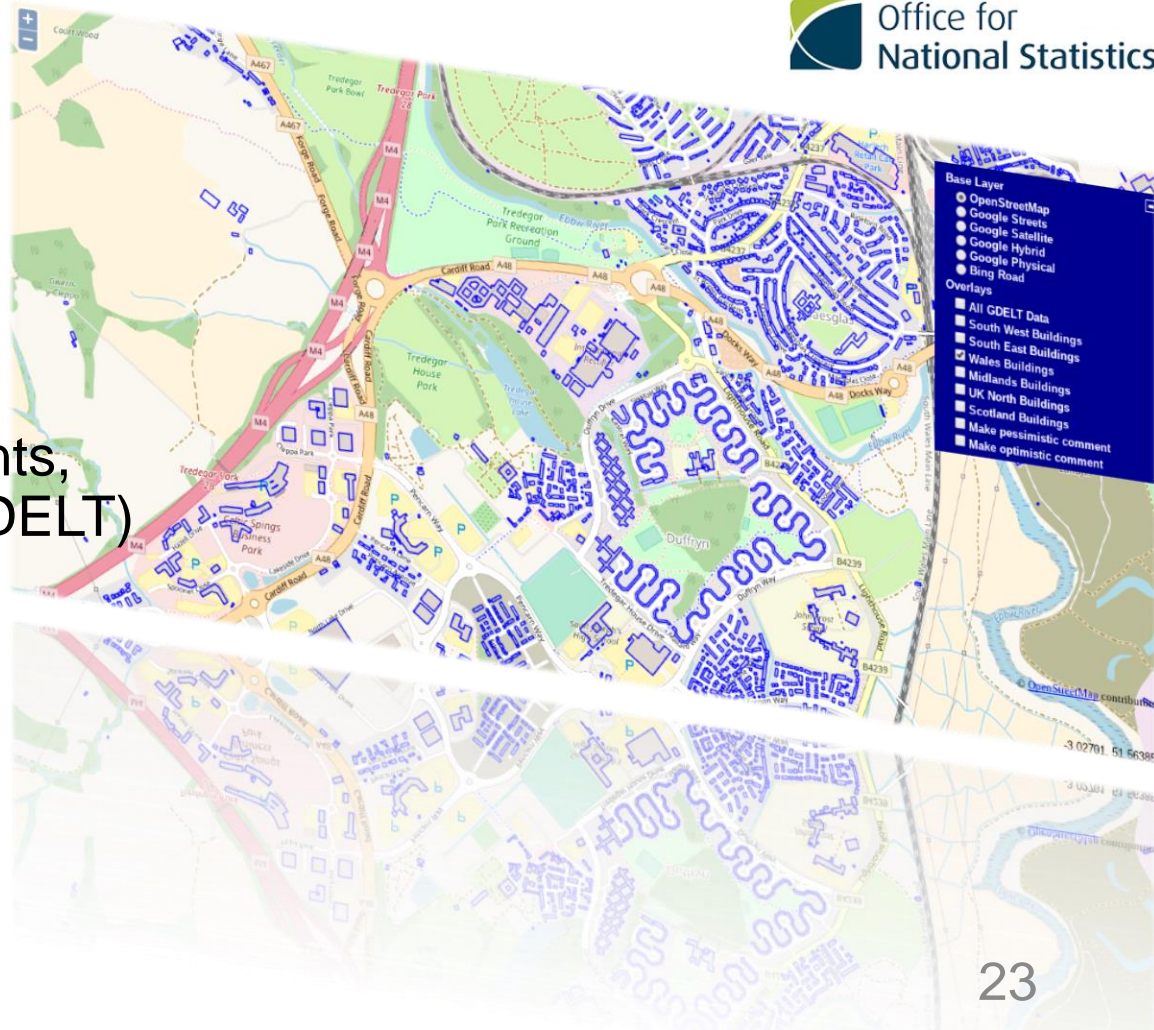
- Data Ingest Workflow
- Apache NiFi
- R
- RStudio
- Shiny



Proof of Concept **Satellite / Vector Data**

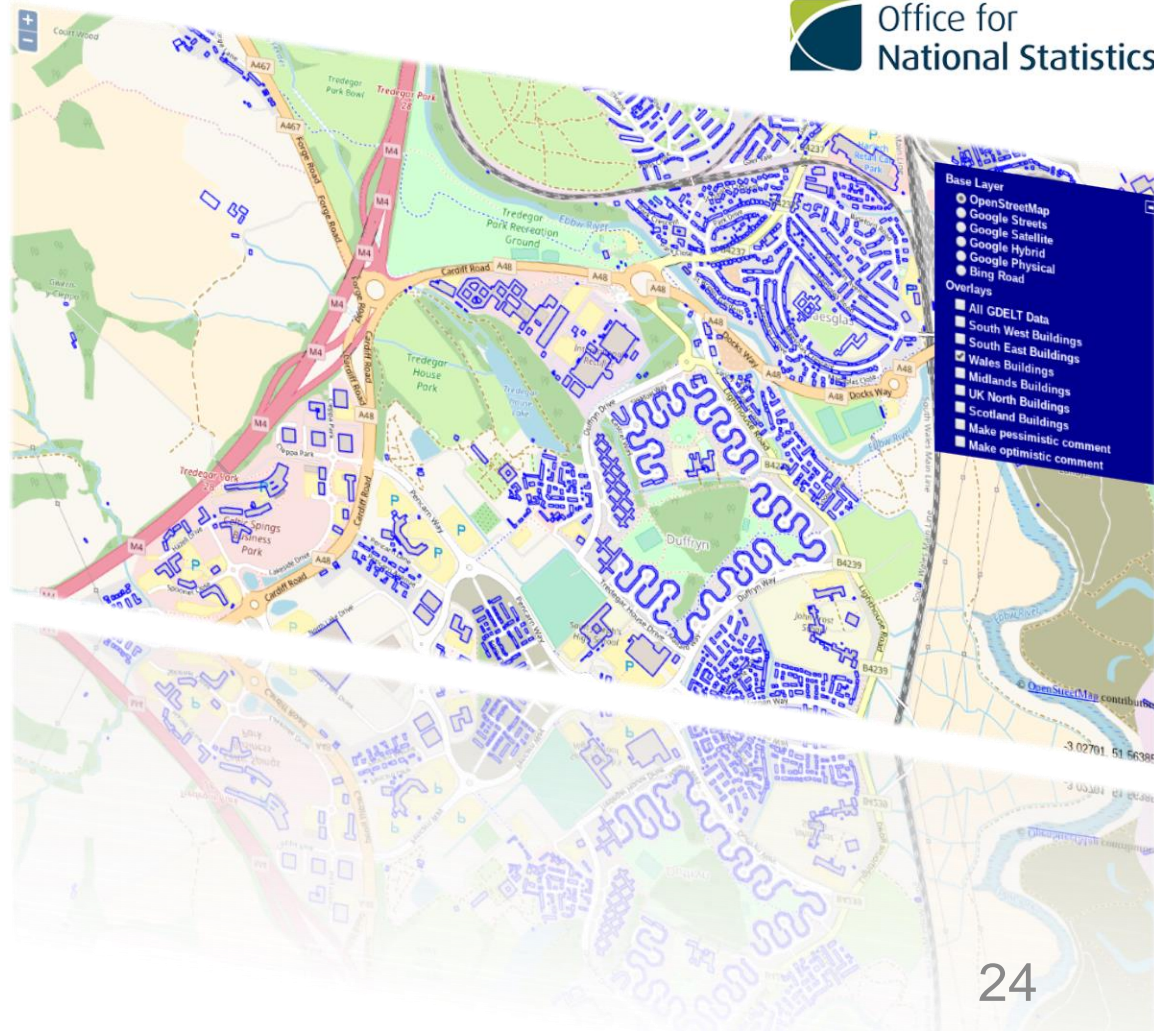
Satellite / Vector

- UK Buildings Shapefiles
 - All UK Buildings
- Global Database of Events, Language, and Tone (GDELT)
 - Imported from AWS S3
 - Imported 6M Events



Satellite / Vector

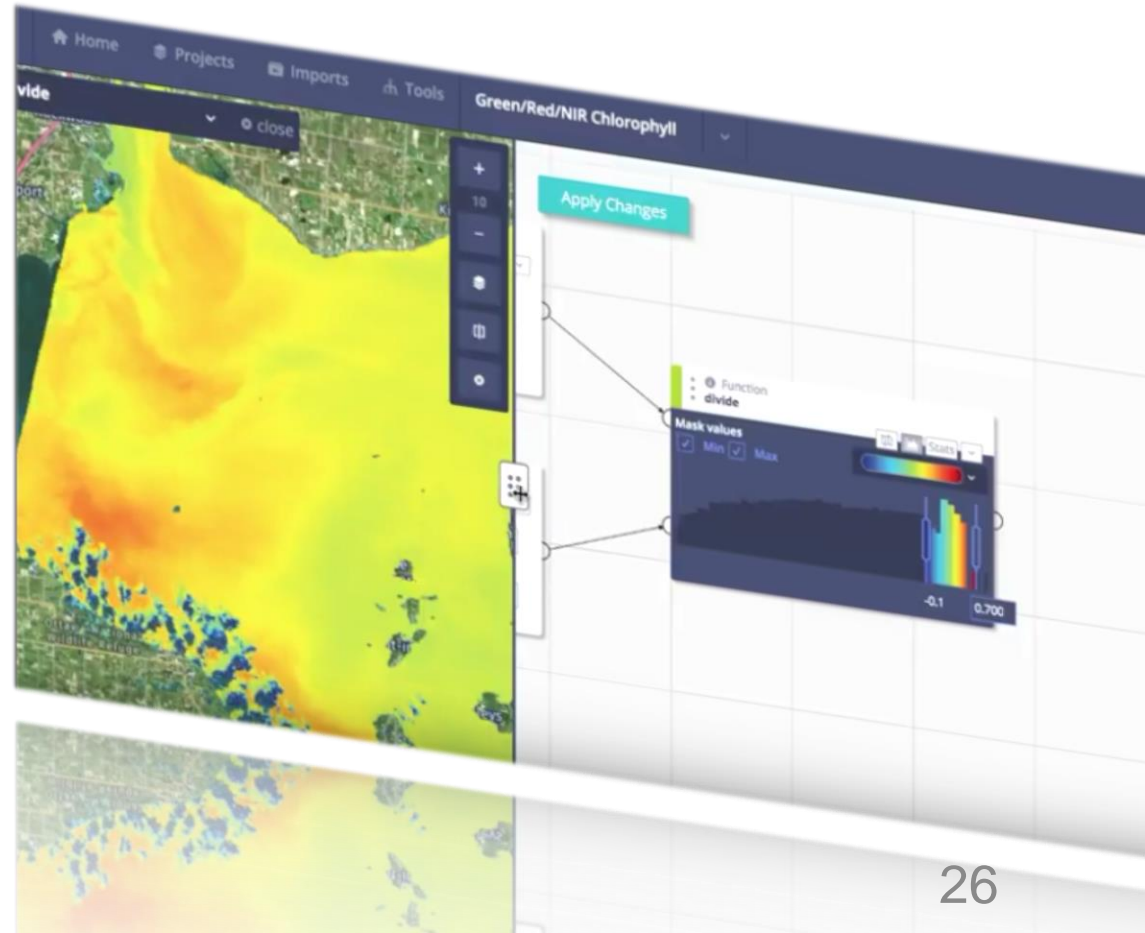
- Open Source
 - GeoMesa
 - GeoServer
 - GeoWebCache
 - Spark
 - Accumulo
 - Hadoop
- AWS EMR



Proof of Concept **Satellite / Raster Data**

Satellite / Raster

- Find, combine and analyse earth observation data
- Browse existing satellite datasets
- Stitch together imagery
- Build analyse automated pipelines
- Edit, iterate quickly (Real-time)



Satellite / Raster

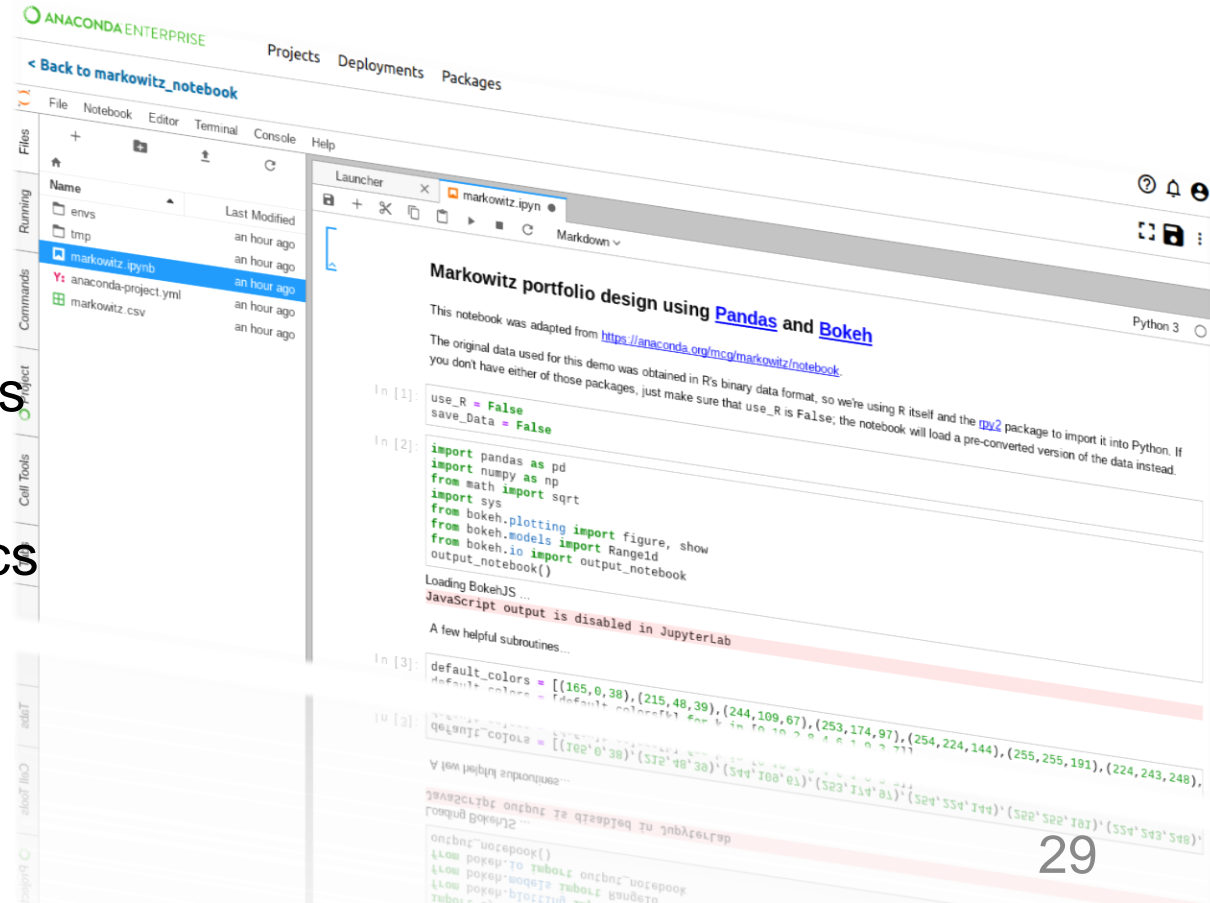
- Process large/small raster datasets with low latency using multiple clusters and multiple threads
 - GeoTrellis
 - GeoMesa
 - Spark
 - Accumulo
 - Hadoop
 - Scala



Proof of Concept **Integrated Development Environment (IDE)**

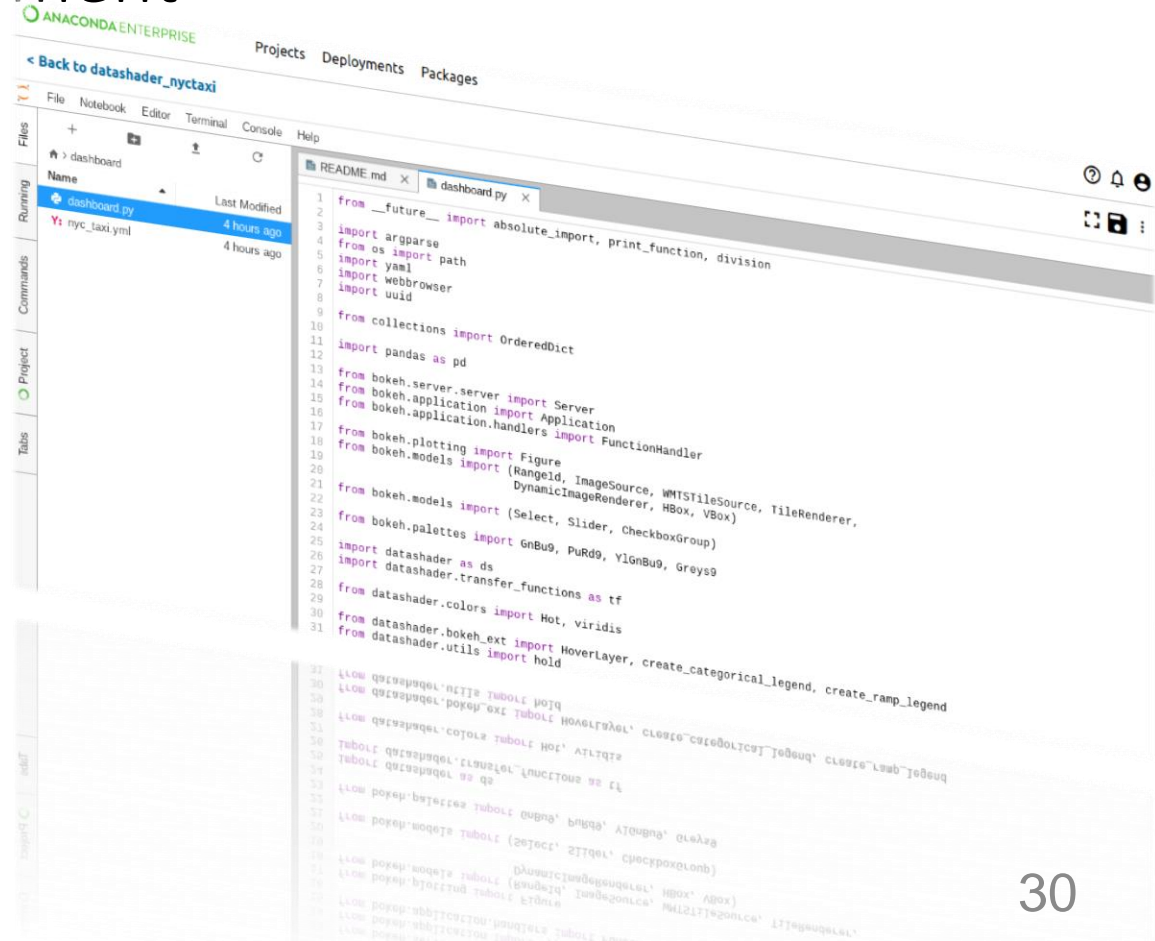
Integrated Development Environment

- Methods
 - Create / Share Code
 - Share Data
- Manage Dependencies
- Collaboration
- Windows / Linux / Macs
- Local / Cloud
- Assured Packages



Integrated Development Environment

- Anaconda 2/3
- JupyterLab
- Shiny
- Supports R, Python

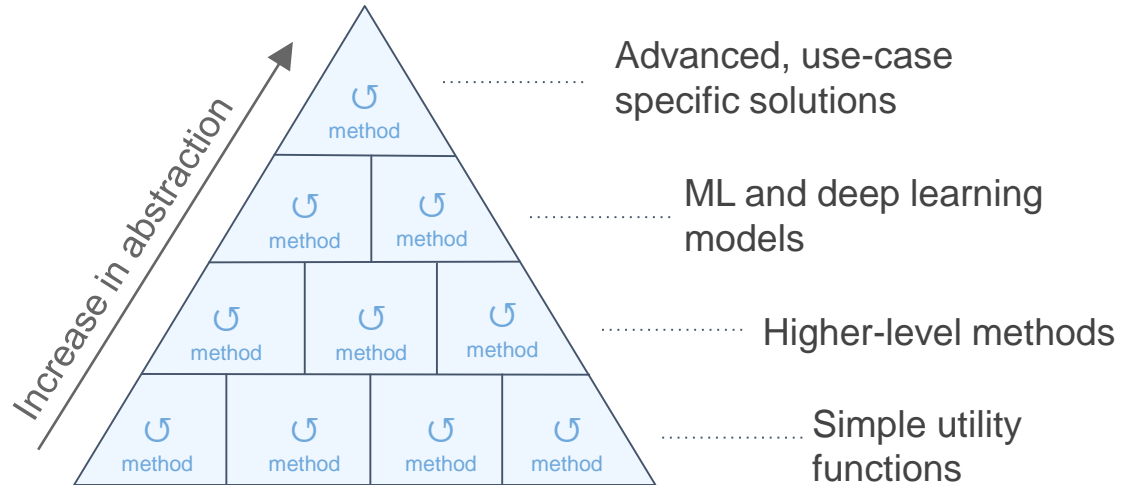


```
1 from __future__ import absolute_import, print_function, division
2
3
4 import argparse
5 from os import path
6 import yaml
7 import webbrowser
8 import uuid
9
10 from collections import OrderedDict
11
12 import pandas as pd
13
14 from bokeh.server.server import Server
15 from bokeh.application.server import Application
16 from bokeh.application.handlers import FunctionHandler
17 from bokeh.plotting import Figure
18 from bokeh.models import (Range1d, ImageSource, WMTStyleSource, TileRenderer,
19                           DynamicImageRenderer, HBox, VBox)
20
21 from bokeh.models import (Select, Slider, CheckBoxGroup)
22
23 from bokeh.palettes import GnBu9, PuRd9, YlGnBu9, Greys9
24
25 import datashader as ds
26 import datashader.transfer_functions as tf
27
28 from datashader.colors import Hot, viridis
29
30 from datashader.bokeh_ext import HoverLayer, create_categorical_legend, create_ramp_legend
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Proof of Concept **Methods Library**

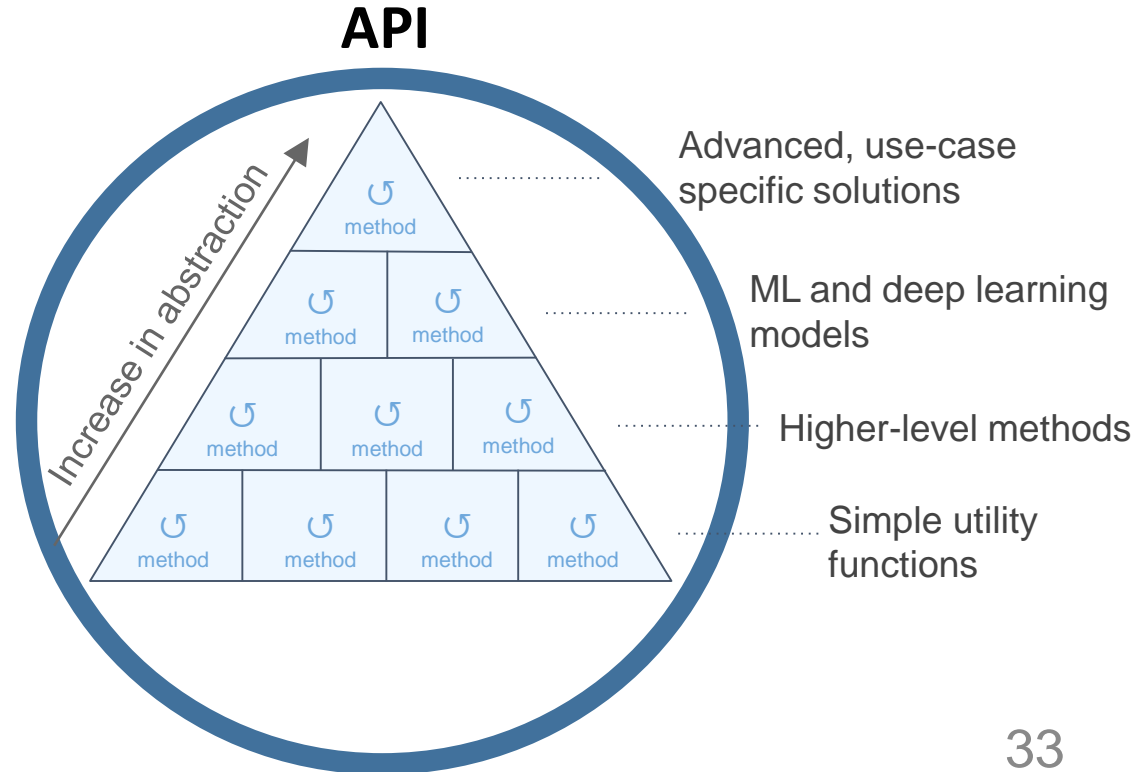
Methods Library

- Easy to use
- Easy to find
- Easy to compose
 - Building blocks
- Add to workflow



Methods as a Service

- Access via APIs
 - Methods as a Service
- Documentation
- ISTAT R Package
 - [R Evolved Generalized Software for Sampling Estimates and Errors in Surveys](#)



Example Services

Example Services/Products/Code

- Kylo
- GeoTrellis
- Australian Open Data Cube
- GeoNotebook
- Rstudio Server
- Cartoview
- Vinsight
- Multi-Party Authentication
- GeoNode
- OpenMined
- DataShield
- rSDMX
- FarmShot
- Plantix

Useful Links

- [GitHub Repository](#)

Additional Slides

Proof of Concept Calculations on Encrypted Data

Calculations on Encrypted Data

- DARPA Project
- Jana Database
 - Open Source
- Encrypted At Source
- Multi-Party Computation
- Differential Privacy
- Homomorphic Encryption
- High-Tech

```

query> list
relation
-----
community
dbversion
nation
person
person2diseaseriskfactor
person2diseasestate
policyauthority
policyauthority2community
policyauthority2nation

query> describe person
Relation: person
attribute | type
-----|-----
person_id | SQLUnsignedInt32
lastname  | SQLString
firstname | SQLString
birthdate | SQLDate
gender    | SQLString
residence | SQLString
citizenship | SQLUnsignedInt32

encryption | constraints
-----|-----
Share; DET |
Share      |
Share      |
Share; ORD with index | Pad to length 30
Share; DET with index | Pad to length 30
Share; DET with index | Pad to length 30
Share; DET with index | Pad to length 8

c1f1zse2p1b | z0g1u2i8ueq1uf35 | 2p1re:  |
1e2i9euce   | z0g1u2i8ueq1uf35 | 2p1re:  |
8eue6el    | z0g2f1u8          | 2p1re:  |
p1r1f1uq4e | z0g1d9e           | 2p1re:  |
41r2f1u8me | z0g2f1u8          | 2p1re:  |
1a2f1u8me  | z0g2f1u8          | 2p1re:  |
be1zou1iq   | z0g1u2i8ueq1uf35 | 2p1re:  |
g1f1r1p1e   |  | 2p1re:  |
k2j1f1ou: be1zou |  | 2p1re:  |
dne1l1> ne1 |  | 2p1re:  |
  
```

Jana Capabilities

- Functionality
 - Generous subset of SQL
 - RDBMS Atomicity, Consistency, Isolation, Durability (ACID) properties
- Privacy
 - Data in transit - Public key crypto + proxy re-encryption
 - Data at rest - Deterministic, random, searchable
 - Computation - in RDBMS using Deterministic Encryption (DET) & searchable, in SPDZ MPC
 - Results - Differential privacy applied while in Multi Party Computation (MPC)
- Performance
 - 10Ks of records moving to 100Ks, queries in seconds
- Deployment
 - Virtualized appliance with RESTful API

Types of Queries

- SPJ, UNION, INTERSECT, EXCEPT
- Integer, String, Boolean, Enum, Fixed-Point, Date
- Nested queries
- [Video Link](#)

```

SELECT person.person_id, lastname, firstname, diseasestate, gender, birthdate
FROM person
JOIN community ON community.community_id = person.residence
JOIN person2diseasestate ON person2diseasestate.person_id = person.person_id
JOIN policyauthority2community ON policyauthority2community.community_id = community.community_id
JOIN policyauthority ON policyauthority.authority_id = policyauthority2community.authority_id
WHERE person2diseasestate.transitiondate < '04-20-2017'
AND person2diseasestate.diseasestate IN ('I')
AND policyauthority.authority = 'CebuCityCommunityPA'
AND person.person_id NOT IN
(SELECT person.person_id
FROM person
JOIN community ON community.community_id = person.residence
JOIN person2diseasestate ON person2diseasestate.person_id = person.person_id
JOIN policyauthority2community ON policyauthority2community.community_id = community.community_id
JOIN policyauthority ON policyauthority.authority_id = policyauthority2community.authority_id
WHERE person2diseasestate.transitiondate < '04-20-2017'
AND person2diseasestate.diseasestate IN ('R', 'D')
AND policyauthority.authority = 'CebuCityCommunityPA');
  
```

```

AND policyauthority.authority = 'CebuCityCommunityPA');
FROM person
JOIN community ON community.community_id = person.residence
JOIN person2diseasestate ON person2diseasestate.person_id = person.person_id
JOIN policyauthority2community ON policyauthority2community.community_id = community.community_id
JOIN policyauthority ON policyauthority.authority_id = policyauthority2community.authority_id
WHERE person2diseasestate.transitiondate < '04-20-2017'
AND person2diseasestate.diseasestate IN ('R', 'D')
AND policyauthority.authority = 'CebuCityCommunityPA');
  
```