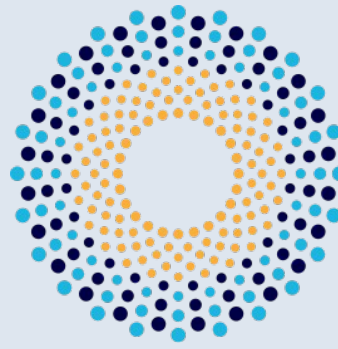




Science and  
Technology  
Facilities Council



**iris**

# IRIS Operational Intelligence:

Increasing data access to aid federation cohesiveness  
IRIS Digital Asset 2021/2022

Connor Pettitt

[connor.pettitt@stfc.ac.uk](mailto:connor.pettitt@stfc.ac.uk)

SCD RAL

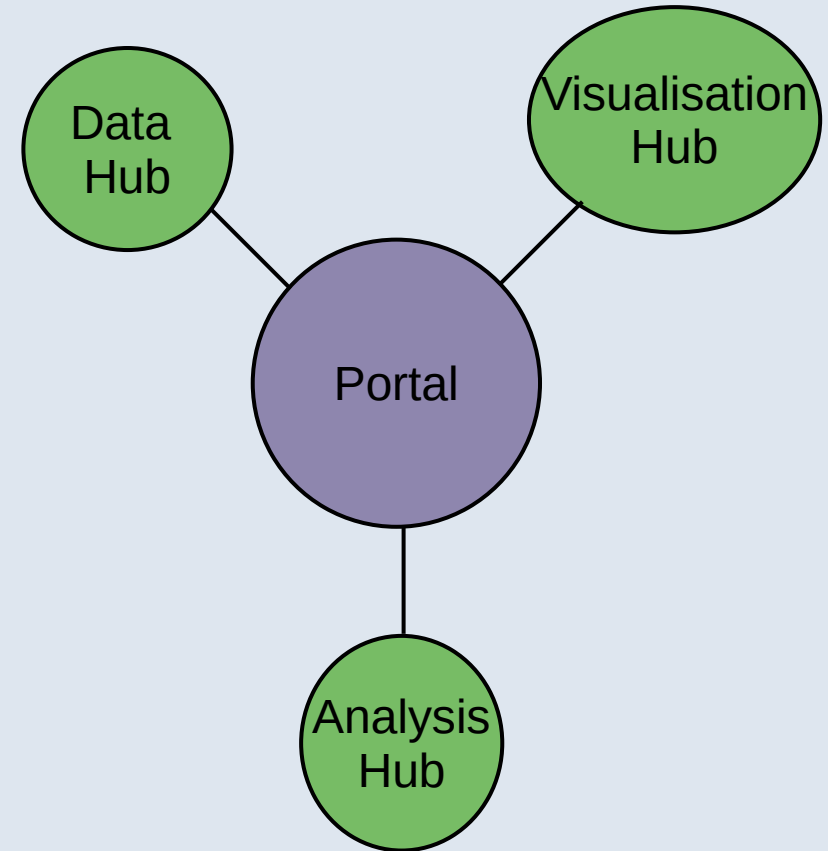
15th February 2022



Science and  
Technology  
Facilities Council

# Operational Intelligence

## New services



# Operational Intelligence

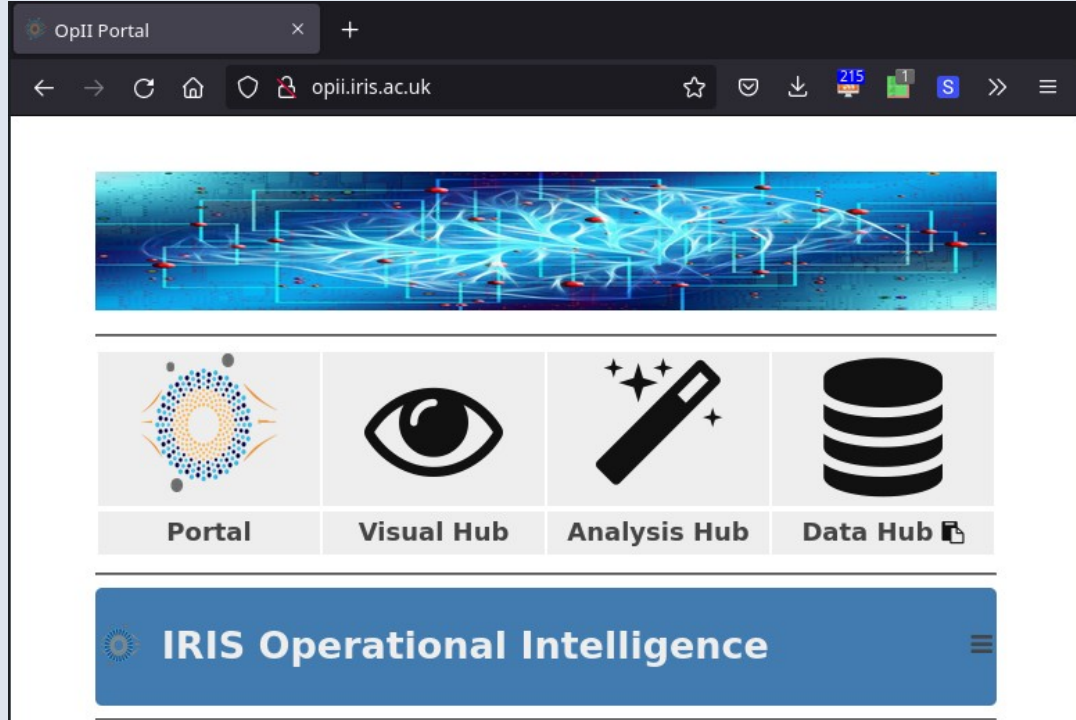
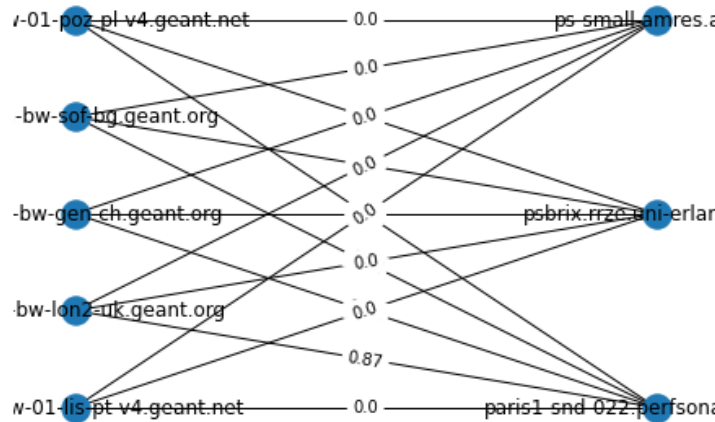
```
0 %curl -i api.opii.iris.ac.uk/help
Please follow this link to login:
https://iris-iam.stfc.ac.uk/login?s=ae8593f112

Login successful.
Session key: x0ivqb8gdf3kriz8l0bc7ytiknd826ug

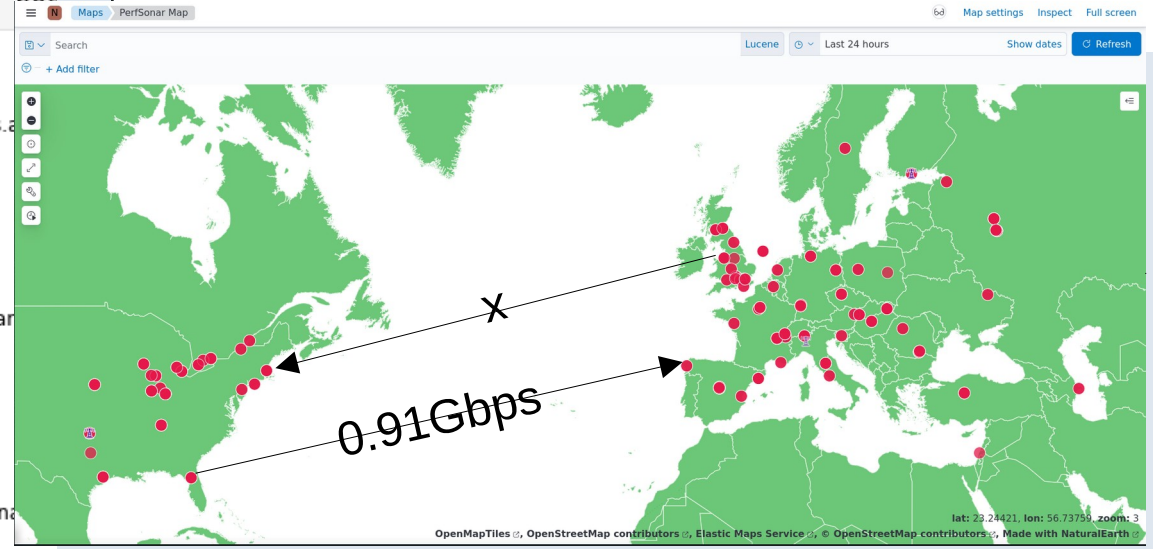
0 %curl -i api.opii.iris.ac.uk/help
-h 'auth:x0ivqb8gdf3kriz8l0bc7ytiknd826ug'
{
  "response": {
    "/cloud": "Cloud",
    "/cvmfs": "CVMFS",
    "/godcb": "GOCDB",
    "/help": "Return this information",
    "/perfsonar": "PerfSONAR"
  },
  "time": 1643211777
}
```

```
[31]: conn_graph = nx.from_dict_of_dicts(small)
[32]: labels = {e: conn_graph.edges[e]['weight'] for e in conn_graph.edges}
[33]: print('A small sample of all throughput tests')
pos = nx.bipartite_layout(conn_graph, destination_labels=labels)
_ = nx.draw(conn_graph, pos, with_labels=True)
_ = nx.draw_networkx_edge_labels(conn_graph, pos, edge_labels=labels)
```

A small sample of all throughput tests



Created 2022/02/09 by Connor Pettitt



# Data Hub with REST API

```
0 %curl -i api.opii.iris.ac.uk/help
```

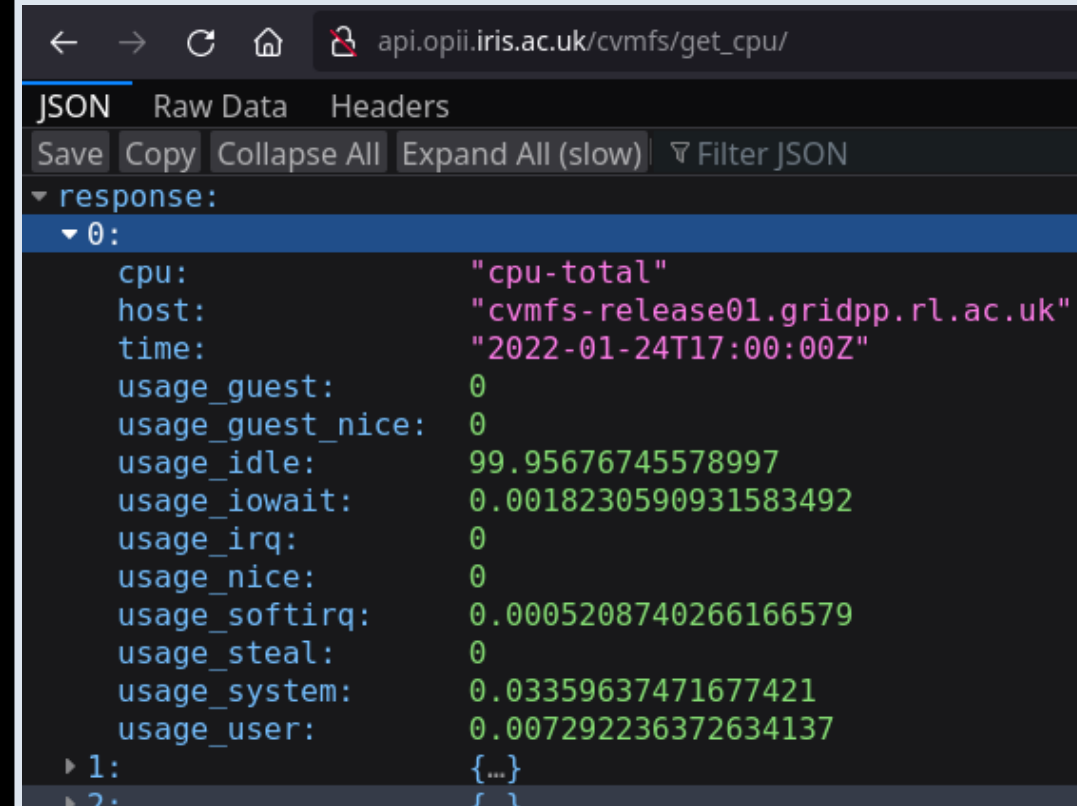
```
Please follow this link to login:  
https://iris-iam.stfc.ac.uk/login?s=ae8593f112
```

```
Login successful.  
Session key: x0ivqb8gdf3kriz8l0bc7ytiknd826ug
```

```
0 %curl -i api.opii.iris.ac.uk/help  
-h 'auth:x0ivqb8gdf3kriz8l0bc7ytiknd826ug'
```

```
{  
  "response": {  
    "/cloud": "Cloud",  
    "/cvmfs": "CVMFS",  
    "/gocdb": "GOCDB",  
    "/help": "Return this information.",  
    "/perfsonar": "PerfSONAR"  
  },  
  "time": 1643211777  
}
```

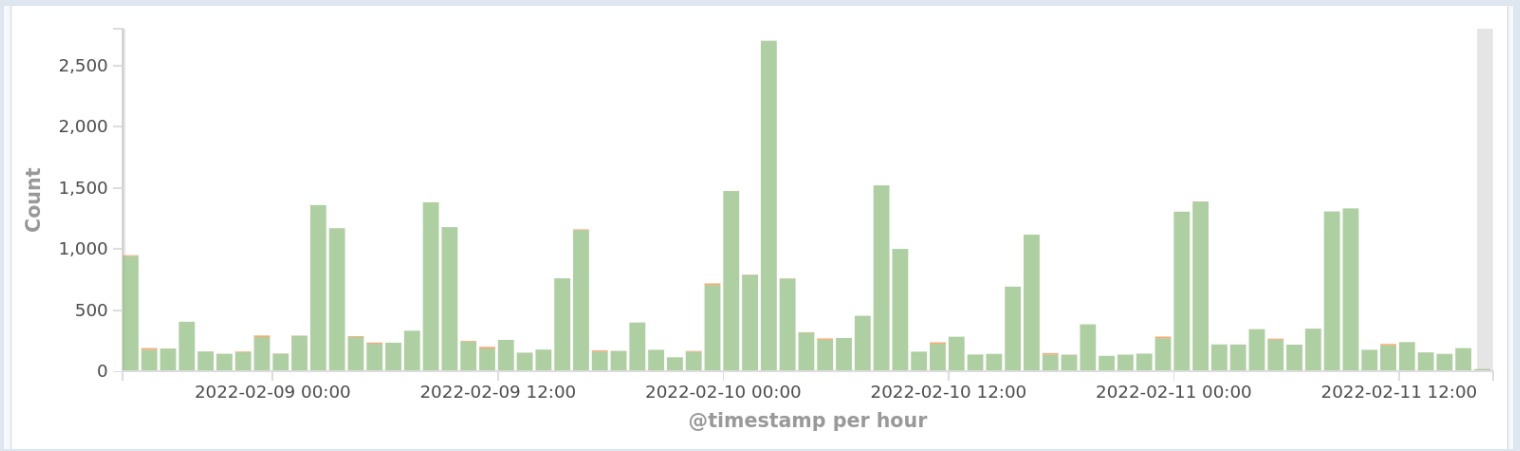
Create your own application using data API.  
Automate the feedback cycle.



```
api.opii.iris.ac.uk/cvmfs/get_cpu/  
JSON Raw Data Headers  
Save Copy Collapse All Expand All (slow) Filter JSON  
▼ response:  
  ▼ 0:  
    cpu: "cpu-total"  
    host: "cvmfs-release01.gridpp.rl.ac.uk"  
    time: "2022-01-24T17:00:00Z"  
    usage_guest: 0  
    usage_guest_nice: 0  
    usage_idle: 99.95676745578997  
    usage_iowait: 0.0018230590931583492  
    usage_irq: 0  
    usage_nice: 0  
    usage_softirq: 0.0005208740266166579  
    usage_steal: 0  
    usage_system: 0.03359637471677421  
    usage_user: 0.007292236372634137  
  ▶ 1: {...}  
  ▶ 2: {...}
```

# Visualisation Hub with Kibana

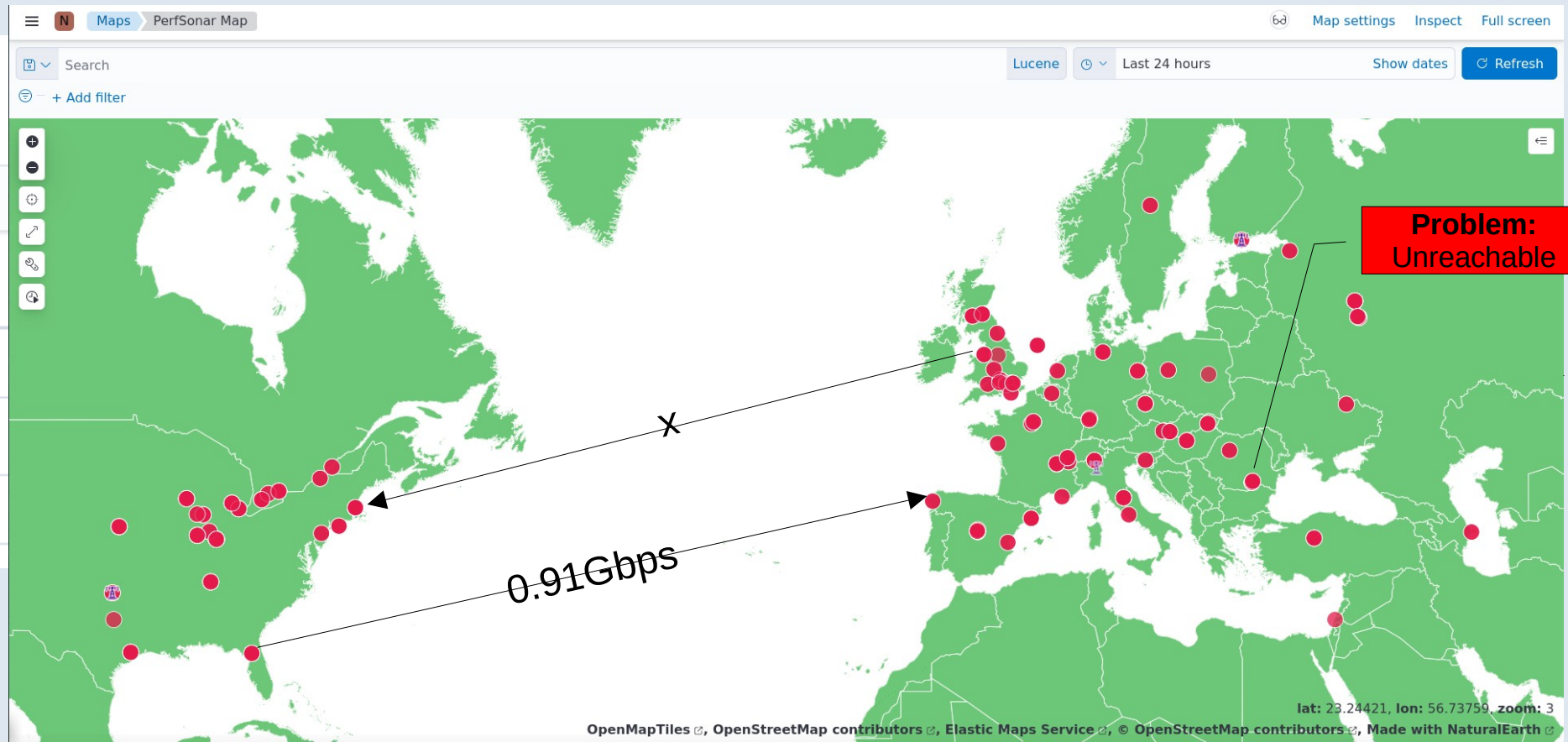
See friendly user focused visualisations.



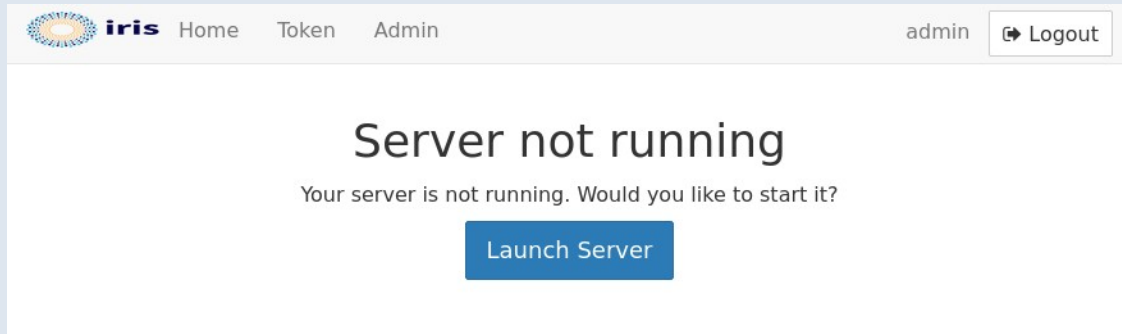
## Dashboards

Search...

- Title ↑
- CVMFS
- GOCDB
- Network



# Analysis Hub with JupyterHub

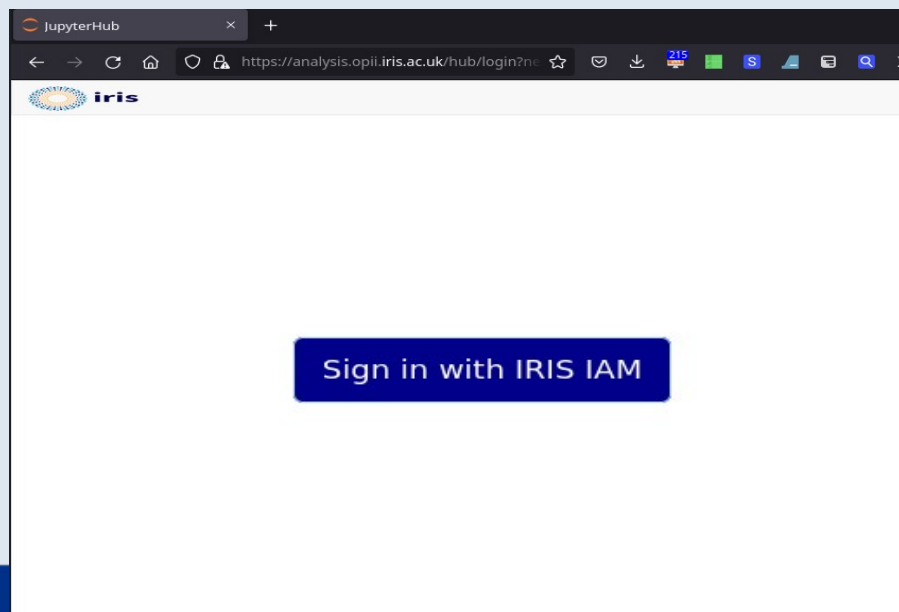


iris Home Token Admin admin Logout

## Server not running

Your server is not running. Would you like to start it?

Launch Server



JupyterHub

https://analysis.opii.iris.ac.uk/hub/login?n

iris

Sign in with IRIS IAM



```
[1]: from datetime import datetime
print('Last run:', datetime.now())

Last run: 2021-12-07 16:42:44.797834

[2]: import opiipy
from opiipy.services import cvmfs

[50]: # Init CVMFS data instance
cv = cvmfs.CVMFS()

[51]: # Run a few queries to the CVMFS InfluxDB database

[5]: # See measurement names
cv.query('show measurements')

[5]: [{ 'name': 'cpu'},
      { 'name': 'disk'},
      { 'name': 'diskio'},
      { 'name': 'kernel'},
      { 'name': 'mem'},
      { 'name': 'n_files'},
      { 'name': 'net'},
      { 'name': 'netstat'},
      { 'name': 'processes'},
      { 'name': 'size'},
      { 'name': 'swap'},
      { 'name': 'system'}]

[6]: # Get n_files data
data_n_files = cv.query('select * from n_files group by repository orde

[7]: data_n_files

[7]: [{ 'time': '2021-12-07T16:09:59Z',
      'host': 'cvmfs-uploader02.gridpp.rl.ac.uk',
      'value': 1317624,
      'repository': 'west-life.egi.eu'},
```

# Analysis Hub with JupyterHub

Create your own complex  
visualisations.

Example visualisation of a perfSONAR  
network graph with connection  
information in a notebook.



```
[6]: # Get n_files data
data_perfsonar = n.get_perfsonar()
```

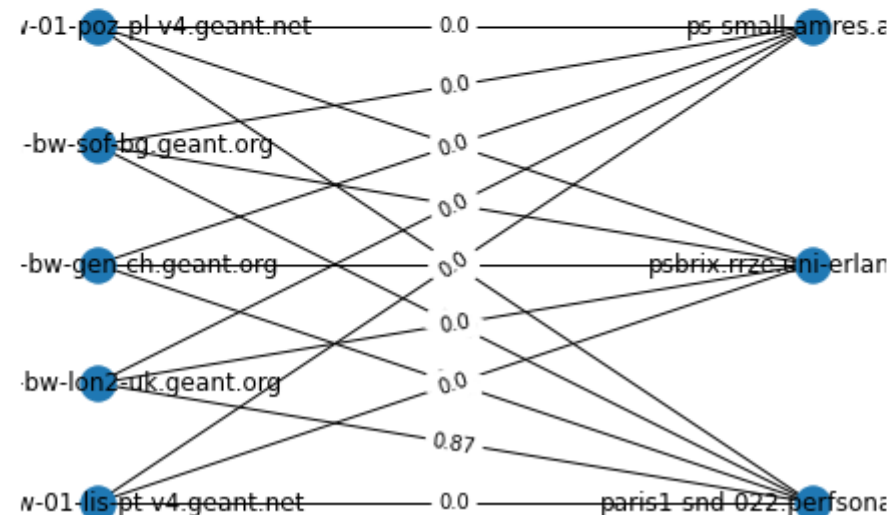
```
[Parallel(n_jobs=8)]: Using backend ThreadingBackend with 8 concurrent
workers.
[Parallel(n_jobs=8)]: Done 34 tasks      | elapsed:    3.5s
[Parallel(n_jobs=8)]: Done 184 tasks    | elapsed:   15.3s
[Parallel(n_jobs=8)]: Done 420 out of 420 | elapsed:   34.6s finished
```

```
[31]: conn_graph = nx.from_dict_of_dicts(small)
```

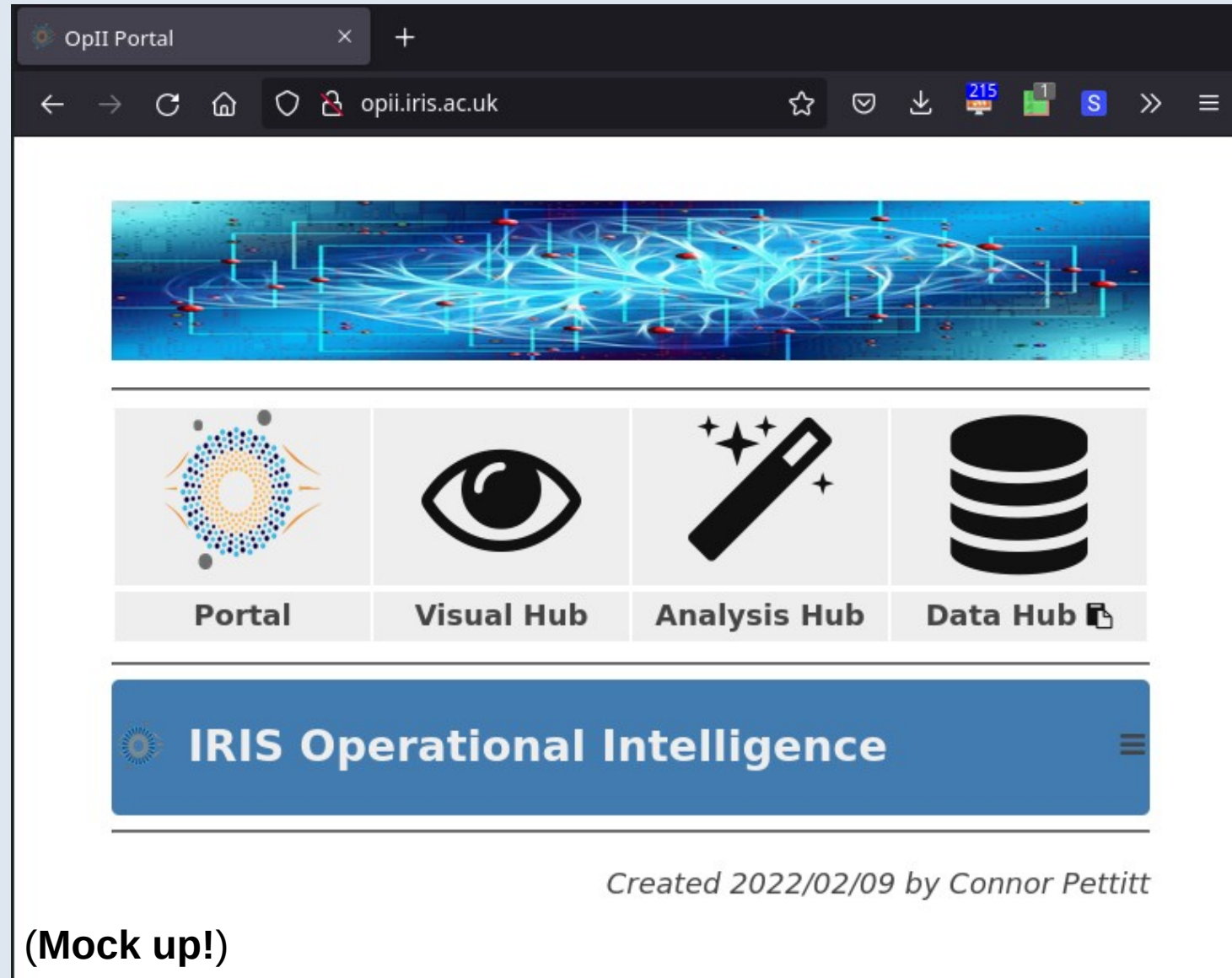
```
[32]: labels = {e: conn_graph.edges[e]['weight'] for e in conn_graph.edges}
```

```
[33]: print('A small sample of all throughput tests')
pos = nx.bipartite_layout(conn_graph, destinations)
_ = nx.draw(conn_graph, pos, with_labels=True)
_ = nx.draw_networkx_edge_labels(conn_graph, pos, edge_labels=labels)
```

A small sample of all throughput tests



# Portal





# Context

- 6 month **exploratory** project ending in toy implementation.
- Looking for **feedback**.
- Ideas for future **use cases**?
- **Perspectives:** Service vs Activity vs User

# Example use cases

# Example use cases

## Manager:

- See the current realtime effectiveness of a service by a new metric, e.g., “Estimated GHG emission vs utilisation.”

# Example use cases

## Manager:

- See the current realtime effectiveness of a service by a new metric, e.g., “Estimated GHG emission vs utilisation.”

## Activity:

- Automated feedback on real usage vs expected usage.

# Example use cases

## Manager:

- See the current realtime effectiveness of a service by a new metric, e.g., “Estimated GHG emission vs utilisation.”

## Activity:

- Automated feedback on real usage vs expected usage.

## User:

- Estimate the efficiency of a particular task on a given service, or “core utilisation and memory.”

# Example use cases

## Manager:

- See the current realtime effectiveness of a service by a new metric, e.g., “Estimated GHG emission vs utilisation.”

## Activity:

- Automated feedback on real usage vs expected usage.

## User:

- Estimate the efficiency of a particular task on a given service, or “core utilisation and memory.”

## Provider:

- Use statistical analysis and machine learning to categorise usage patterns.
- Near-term future projected usage.

# Example use cases

## Manager:

- See the current realtime effectiveness of a service by a new metric, e.g., “Estimated GHG emission vs utilisation.”

## Provider:

- Use statistical analysis and machine learning to categorise usage patterns.
- Near-term future projected usage.

## Activity:

- Automated feedback on real usage vs expected usage.

## All:

- Improve the feedback cycle between use and improvement.
- Draft and communicate the latest performance analyses.
- Quickly demonstrate service bugs and bottlenecks.

## User:

- Estimate the efficiency of a particular task on a given service, or “core utilisation and memory.”

# Motivation



# Motivation

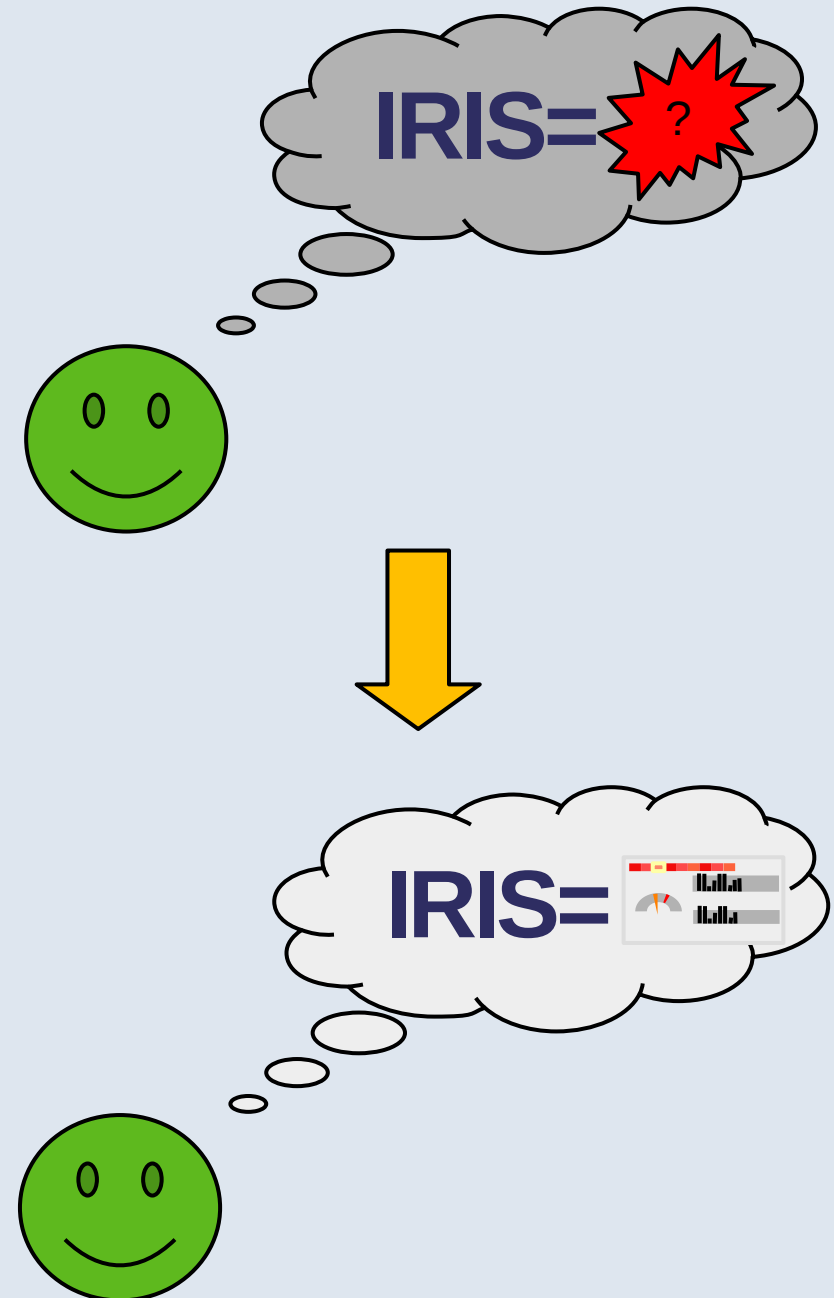
## Questions:

- How could IRIS appear more **coherent** to user/manager/activity?
- How could the current “**state**” of IRIS be made more visible?
- Who might benefit and how?

# Motivation

## Questions:

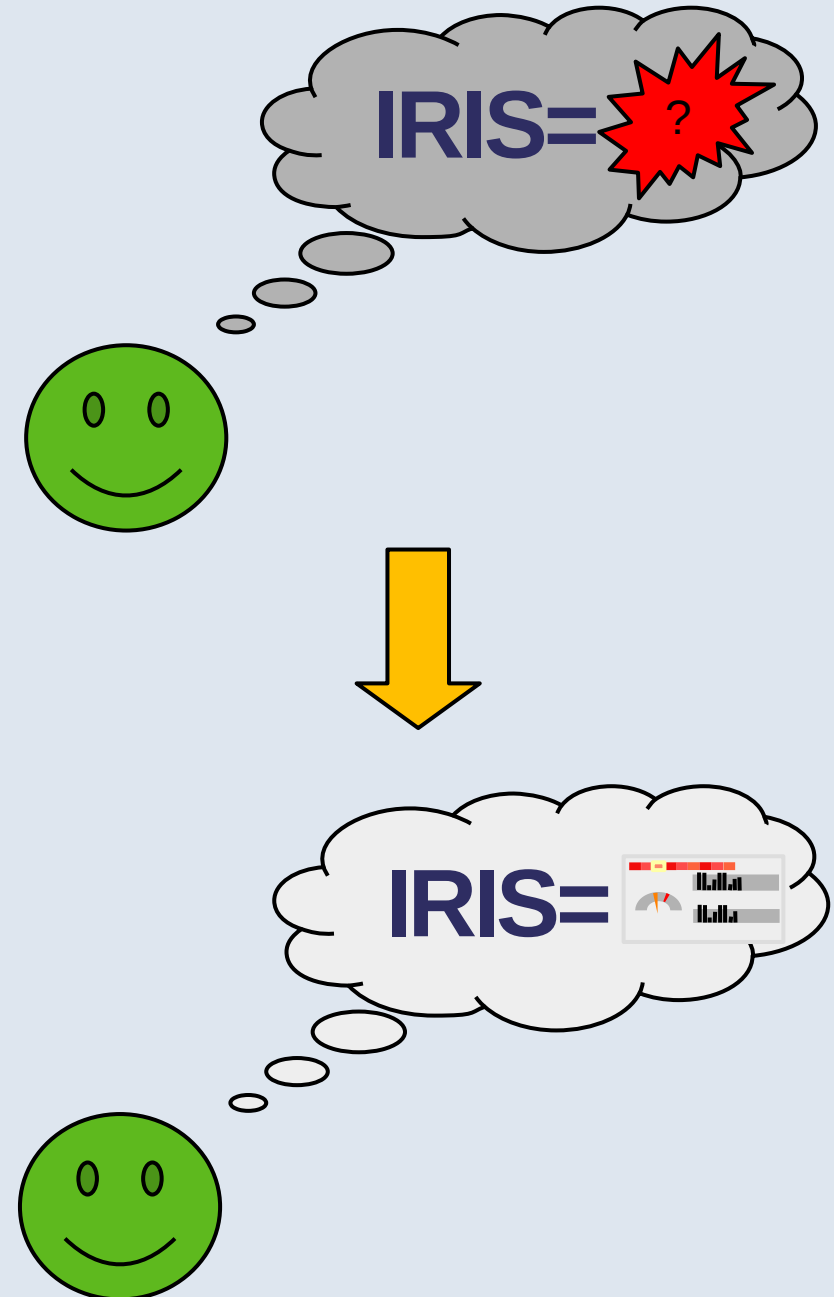
- How could IRIS appear more **coherent** to user/manager/activity?
- How could the current “**state**” of IRIS be made more visible?
- Who might benefit and how?



# Motivation

## Questions:

- How could IRIS appear more **coherent** to user/manager/activity?
  - How could the current “**state**” of IRIS be made more visible?
  - Who might benefit and how?
- 
- **Services** can **process** data to find **relationships**.
  - **Activities** can **automate** the usage-allocation feedback cycle.
  - **Project managers** can **see** the state and scale of operations.
  - **Sites** can display information as a point of **interest**.
  - **Users** can **demonstrate** during meetings/presentations.



# Challenges

# Challenges

## **Data, Analytics & Visualisation:**

- Relevance
- Testing & development
- Interpretability
- Sense of scale

## **Tools:**

- Interface
- Isolating IRIS metrics

## **Collaboration:**

- Specifying data
- Transport between sites
- Accessibility vs Security

# Data, Analytics & Visualisation

# Data, Analytics & Visualisations

## Data:

- Relevant machine metrics

## Examples:

- Service load (cpu, mem, disk, ...)
- Active jobs across domains
- Activities being served
- Transfer progress

# Data, Analytics & Visualisations

## Data:

- Relevant machine metrics

## Examples:

- Service load (cpu, mem, disk, ...)
- Active jobs across domains
- Activities being served
- Transfer progress

## Analytics:

- Quantify service “goodness”
- Allow comparisons

## Examples:

- Service “availability”
- Network “weather” (perfSONAR)
- Change in load
- Projected load vs capacity
- Energy used
- Efficiency
- Carbon emitted
- Basements flooded



# Data, Analytics & Visualisations

## Data:

- Relevant machine metrics

## Examples:

- Service load (cpu, mem, disk, ...)
- Active jobs across domains
- Activities being served
- Transfer progress

## Analytics:

- Quantify service “goodness”
- Allow comparisons

## Examples:

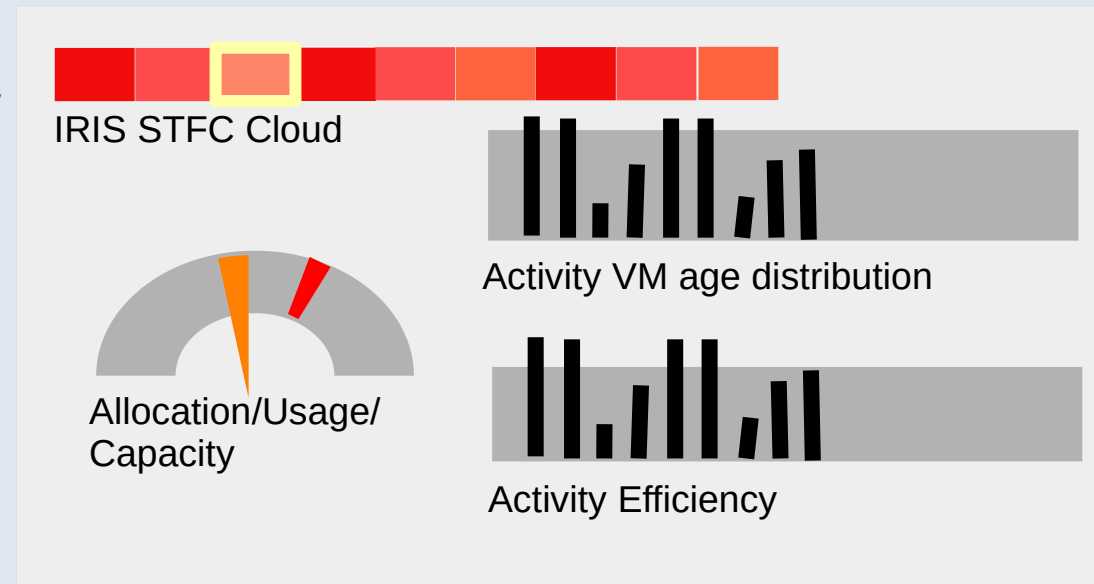
- Service “availability”
- Network “weather” (perfSONAR)
- Change in load
- Projected load vs capacity
- Energy used
- Efficiency
- Carbon emitted
- Basements flooded

## Visualisations:

- Easy to interpret
- “Sense” of the service

## Example:

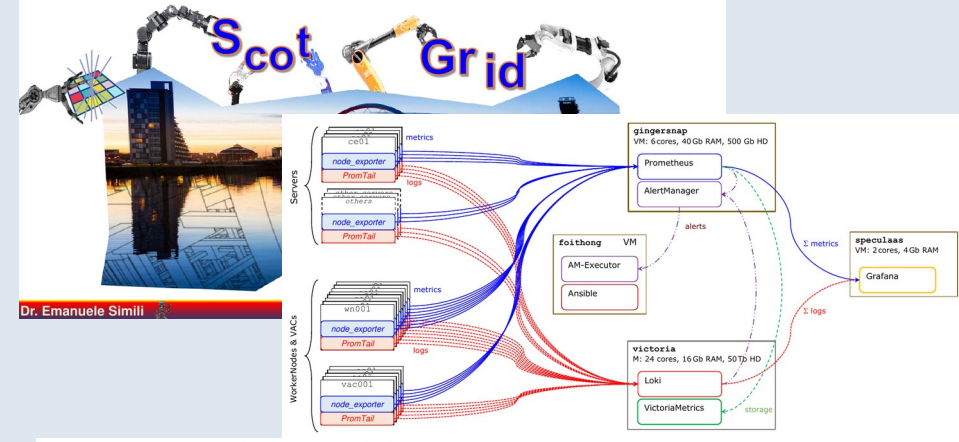
- A basic “front view” of the Cloud



# Tools

# Tools:

## Operational Intelligence @ Glasgow

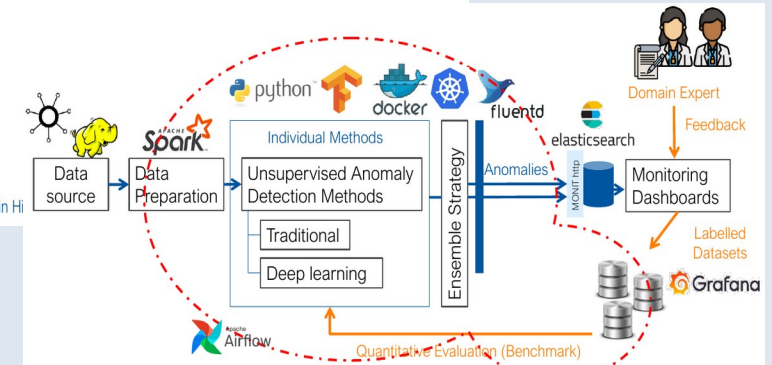


## Anomaly detection in the CERN cloud infrastructure

D. Giordano<sup>1</sup>, M. Paltenghi<sup>1</sup>, S. Metaj<sup>1</sup>, A. Dvorak<sup>2</sup>

<sup>1</sup>CERNIT, <sup>2</sup>Nuclear Physics Institute of the Czech Academy of Sciences

25th International Conference on Computing in Hi



# Tools:

## Glasgow:

- node\_exporter -> **Prometheus** -> Grafana
- PromTail -> **Loki** -> Grafana

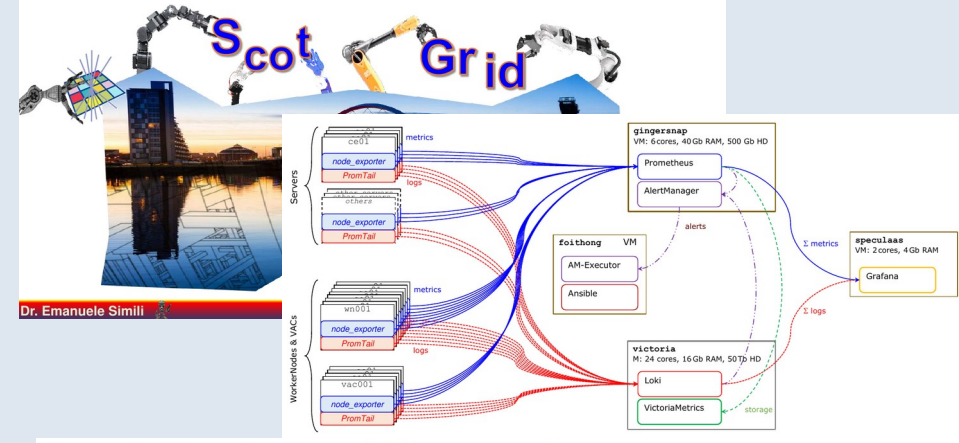
## CERN:

- Collectd -> Fluentd -> **ElasticSearch** -> Grafana

## SCD RAL:

- Telegraf -> **InfluxDB** -> Grafana
- Filebeat -> Logstash -> **ElasticSearch** -> Kibana
- ? -> **Ganglia** -> Icinga
- **APEL**
- **GOCDB**

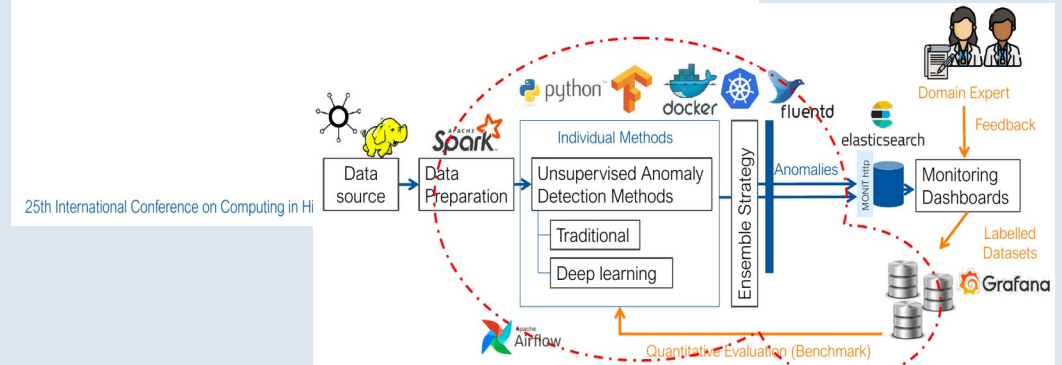
## Operational Intelligence @ Glasgow



## Anomaly detection in the CERN cloud infrastructure

D. Giordano<sup>1</sup>, M. Paltenghi<sup>1</sup>, S. Metaj<sup>1</sup>, A. Dvorak<sup>2</sup>

<sup>1</sup>CERNIT, <sup>2</sup>Nuclear Physics Institute of the Czech Academy of Sciences



# Tools:

## Glasgow:

- node\_exporter -> **Prometheus** -> Grafana
- PromTail -> **Loki** -> Grafana

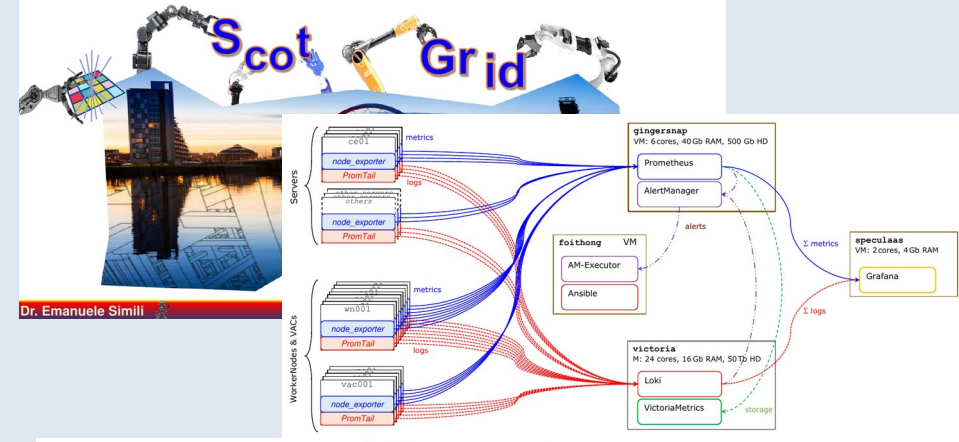
## CERN:

- Collectd -> Fluentd -> **ElasticSearch** -> Grafana

## SCD RAL:

- Telegraf -> **InfluxDB** -> Grafana
- Filebeat -> Logstash -> **ElasticSearch** -> Kibana
- ? -> **Ganglia** -> Icinga
- **APEL**
- **GOCDB**

## Operational Intelligence @ Glasgow

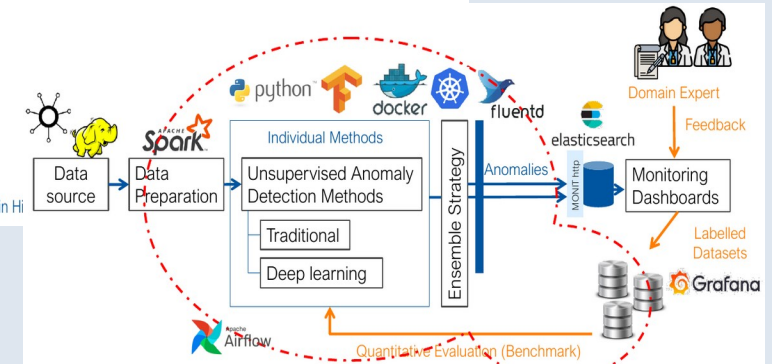


## Anomaly detection in the CERN cloud infrastructure

D. Giordano<sup>1</sup>, M. Paltenghi<sup>1</sup>, S. Metaj<sup>1</sup>, A. Dvorak<sup>2</sup>

<sup>1</sup>CERNIT, <sup>2</sup>Nuclear Physics Institute of the Czech Academy of Sciences

25th International Conference on Computing in Hi



## Databases:

- InfluxDB
- APEL
- GOCDB
- Elasticsearch
- Prometheus
- Loki
- Ganglia

# Collaboration



# Collaboration

Data transport:

Participation:



# Collaboration

## Data transport:

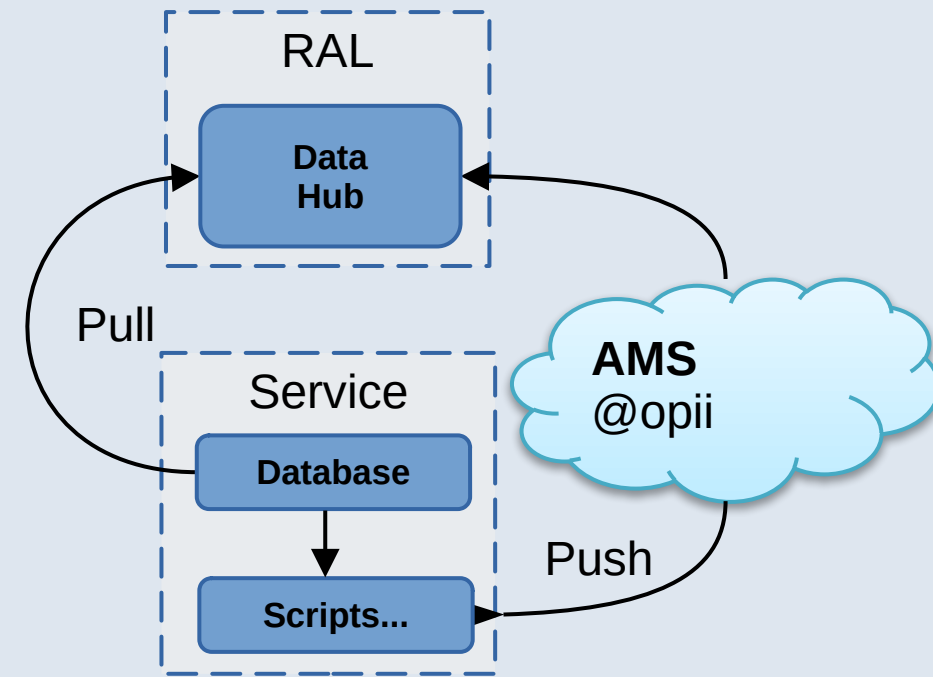
HTTPS

- **Data Hub** can **pull** from databases using **limited read accounts**.

AMS

- **Python-opii** (later) can **push** to the **AMS @opii** queue.

## Participation:





# Collaboration

## Data transport:

### HTTPS

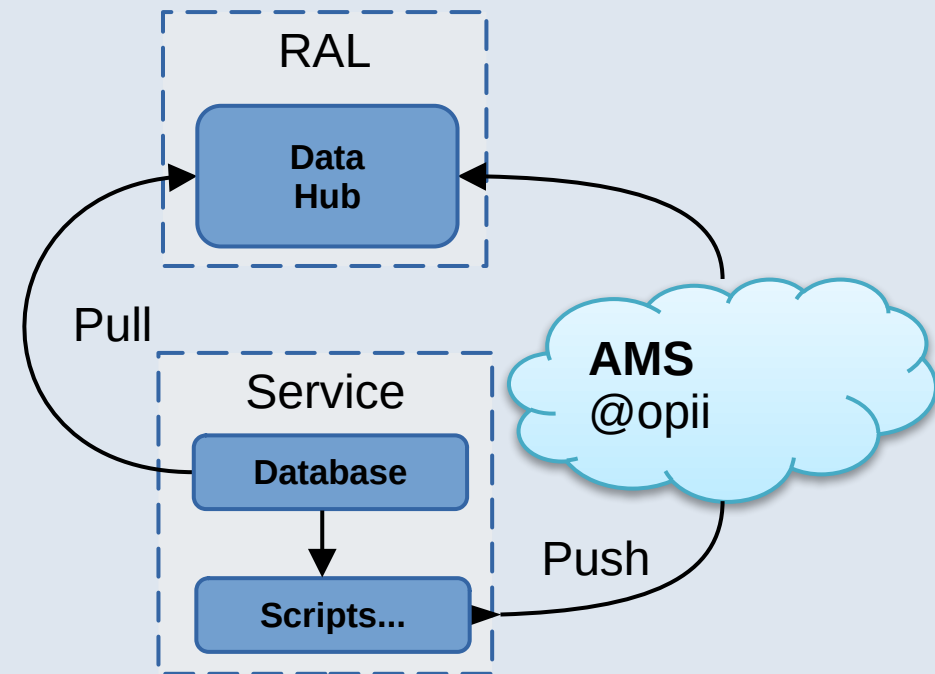
- **Data Hub** can **pull** from databases using **limited read accounts**.

### AMS

- **Python-opii** (later) can **push** to the **AMS @opii** queue.

## Participation:

- Services collect own metrics.
- Flexible publishing (e.g. push or pull.)
- Standard data/methods document format.
- Focus on security and access control mechanisms.



# Collaboration

## Data transport:

HTTPS

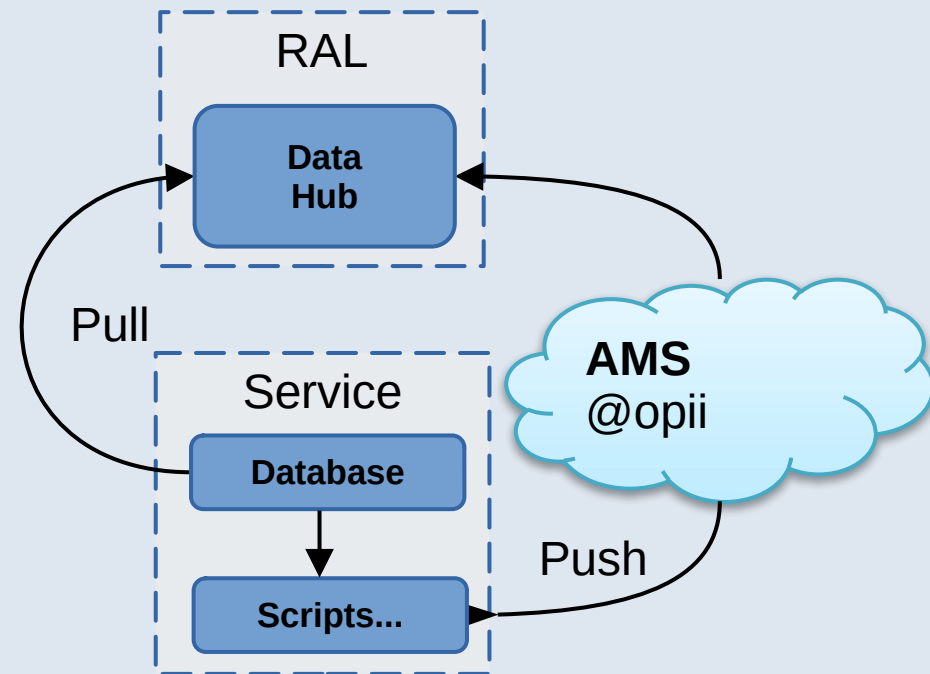
- **Data Hub** can **pull** from databases using **limited read accounts**.

AMS

- **Python-opii** (later) can **push** to the **AMS @opii** queue.

## Participation:

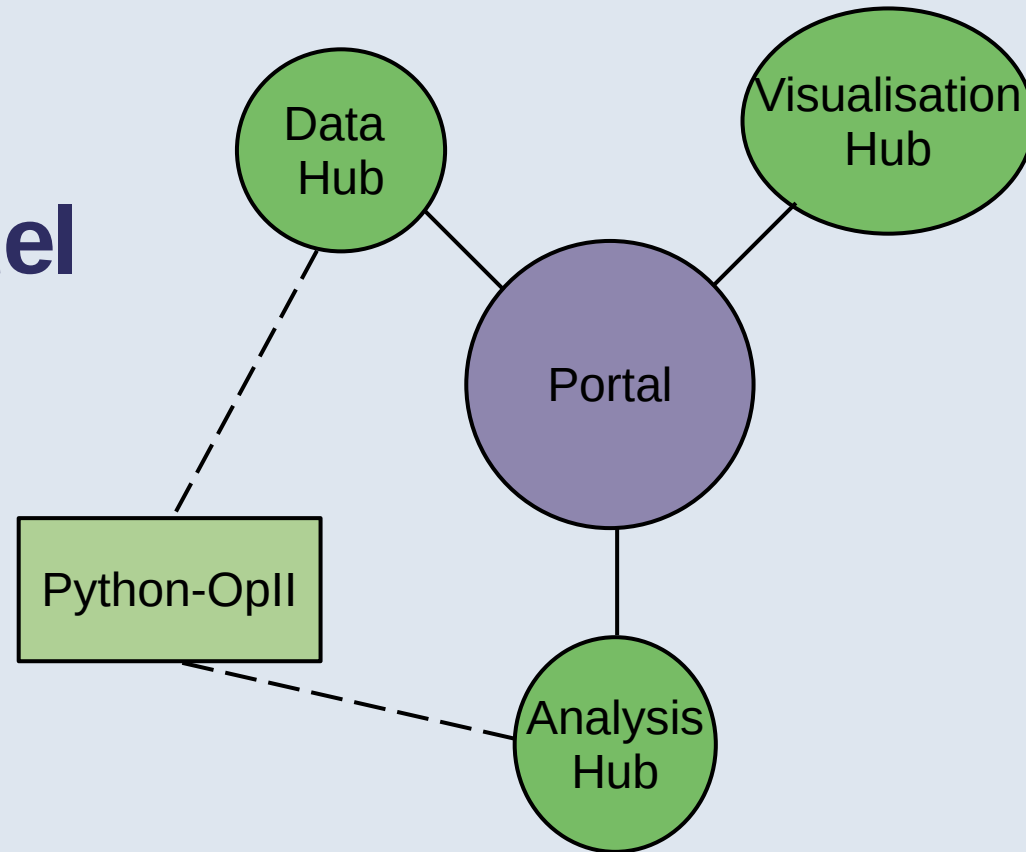
- Services collect own metrics.
- Flexible publishing (e.g. push or pull.)
- Standard data/methods document format.
- Focus on security and access control mechanisms.



	A	B
1	Service	SuperService
2		
3	Data	cpu
4	Description	Telegraf [cpu] output
5	Delivery	Pull
6	Database	InfluxDB
7	Endpoint	influxdb.ss.ac.uk
8	Credentials	
9	Query	SELECT * FROM cpu where time > Now() - 1w;
10		
11	Data	transfers
12	Description	ARK CE transfer jobs
13	Delivery	Push
14	AMS Queue	@opii
15	Format	json
16	ID	ss-transfers

# The current model

- Python package
- Architecture
- Interactions
- Implementation



# OpII Python Package

(Operational Intelligence for IRIS)



# OpII Python Package

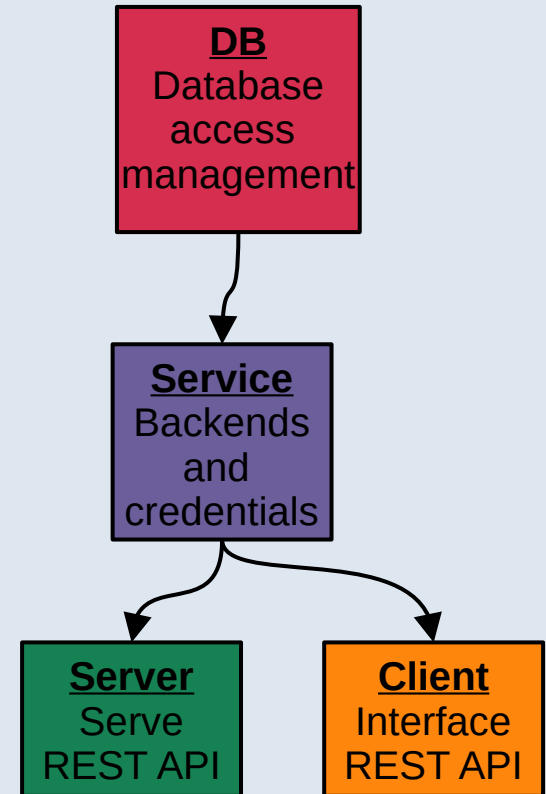
## User functions:

- Pull from service databases / Data Hub REST API
- Run analyses
- Send to the Data Hub

# OpII Python Package

## User functions:

- Pull from service databases / Data Hub REST API
- Run analyses
- Send to the Data Hub



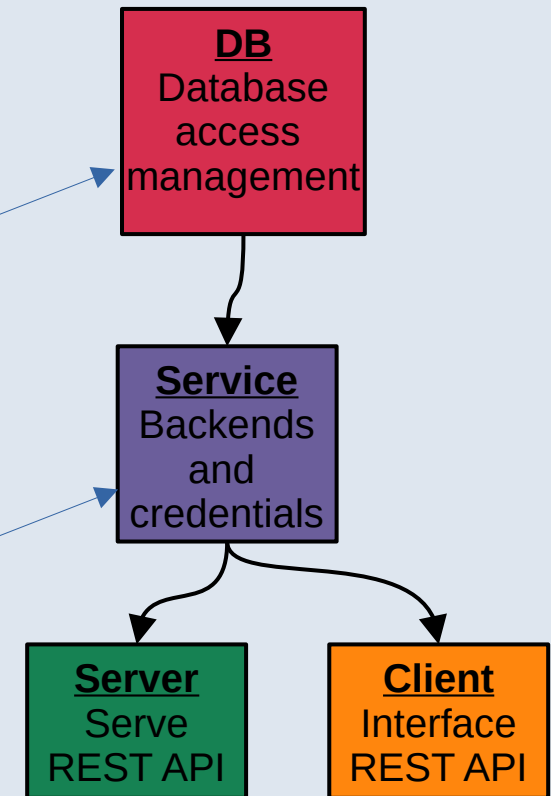
# OpII Python Package

## User functions:

- Pull from service databases / Data Hub REST API
- Run analyses
- Send to the Data Hub

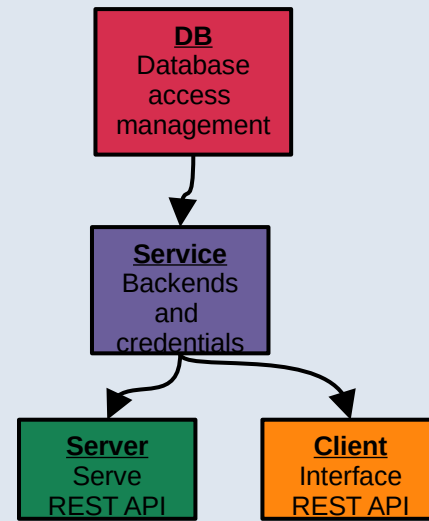
e.g. InfluxDB, GOCDB, APEL,  
Prometheus, Loki, ...

e.g. Access metrics/usage in Cloud,  
CVMFS, SCD Compute, ...



# OpII Python Package

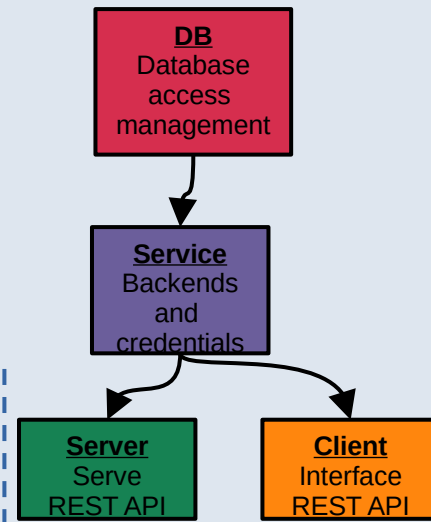
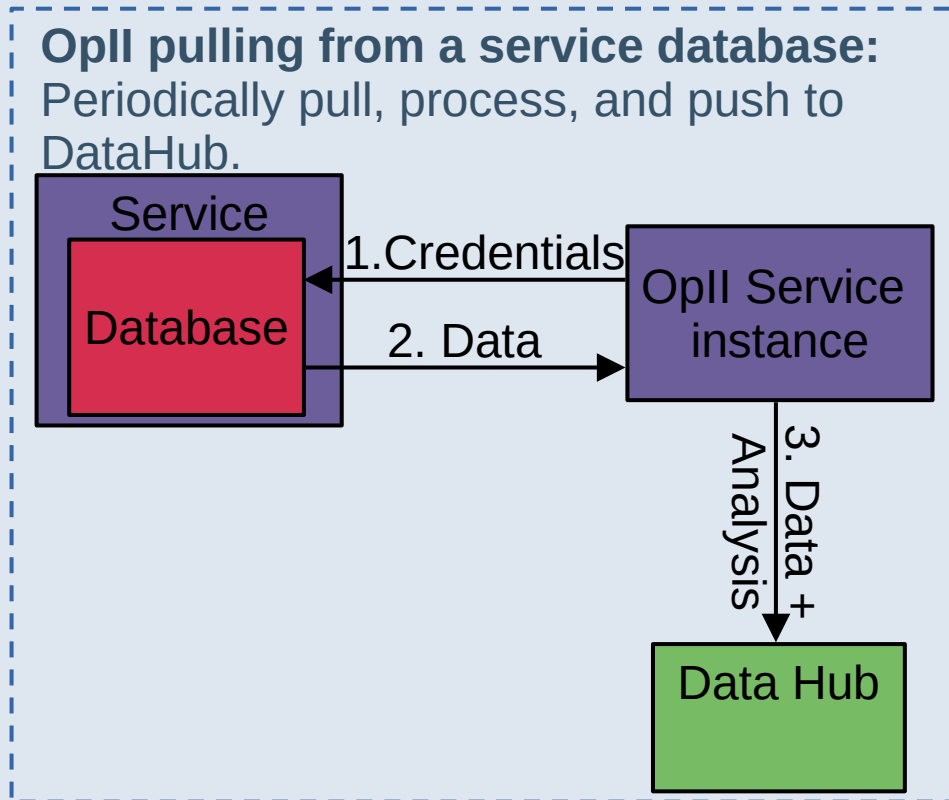
## Example workflows





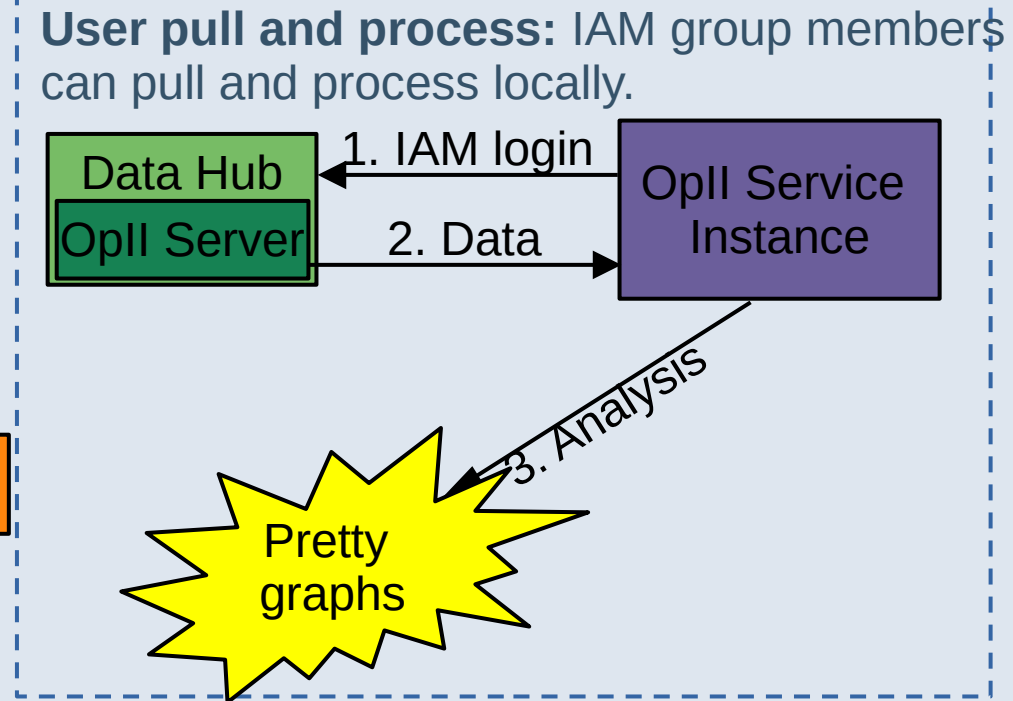
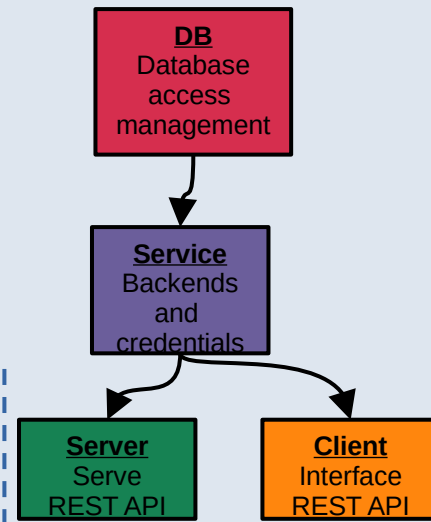
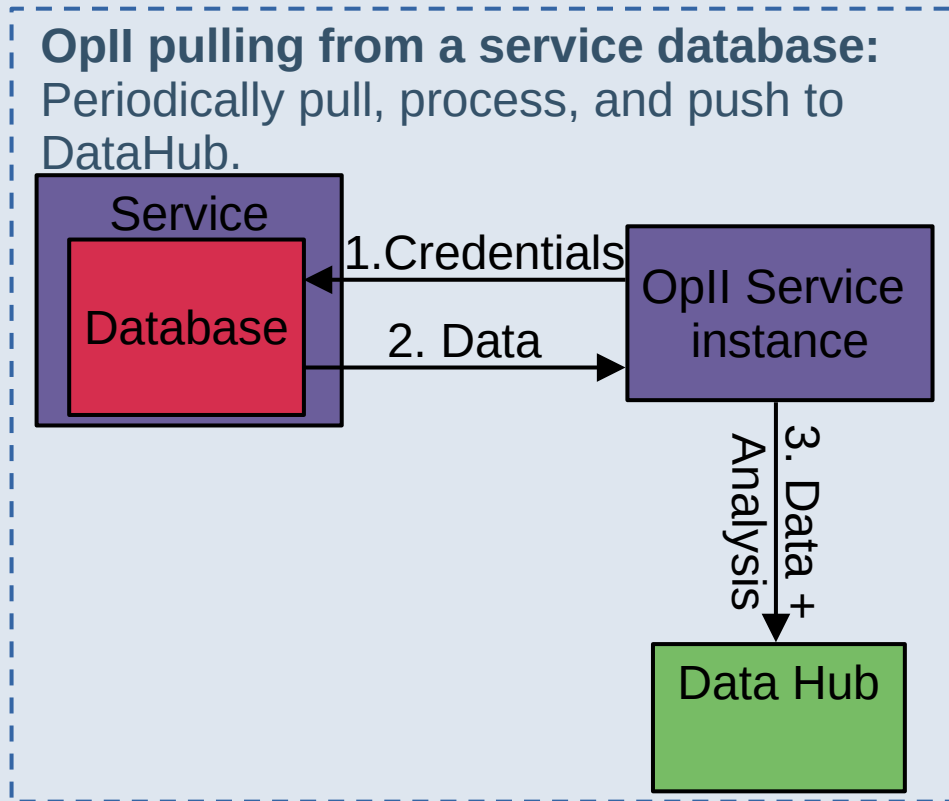
# OpII Python Package

## Example workflows



# OpII Python Package

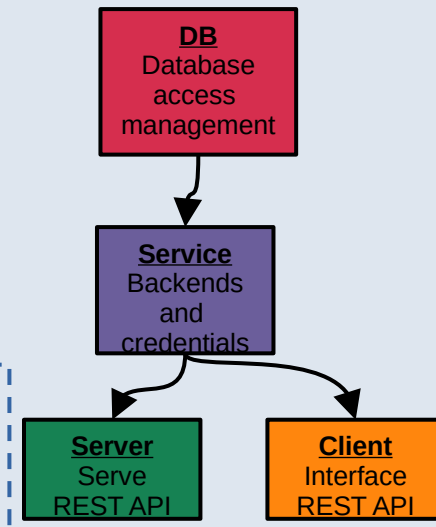
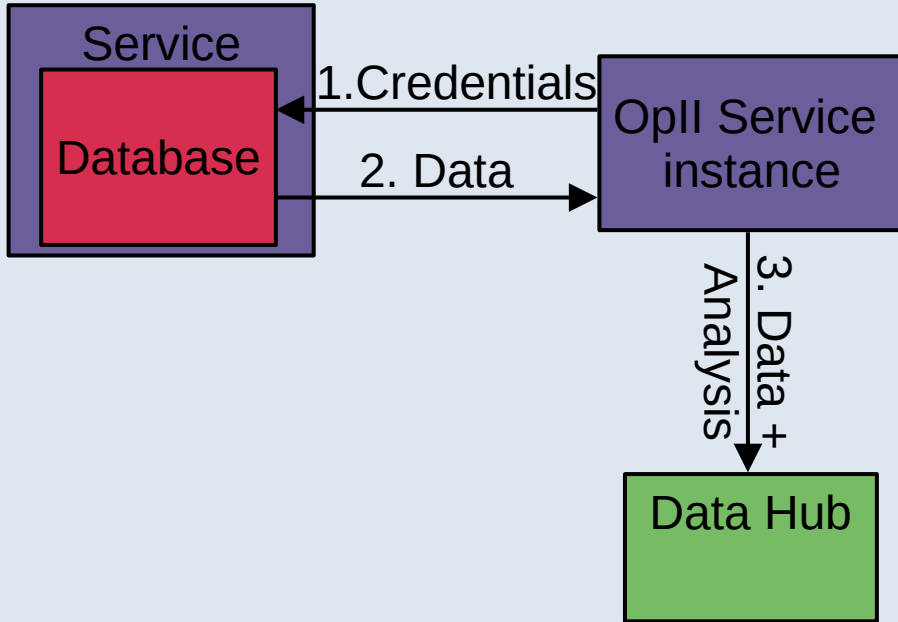
## Example workflows



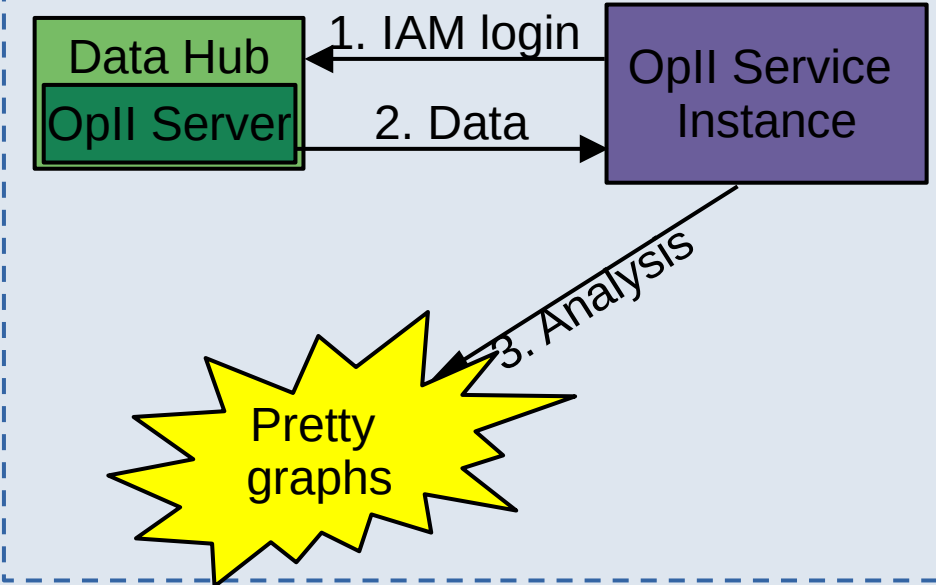
# OpII Python Package

## Example workflows

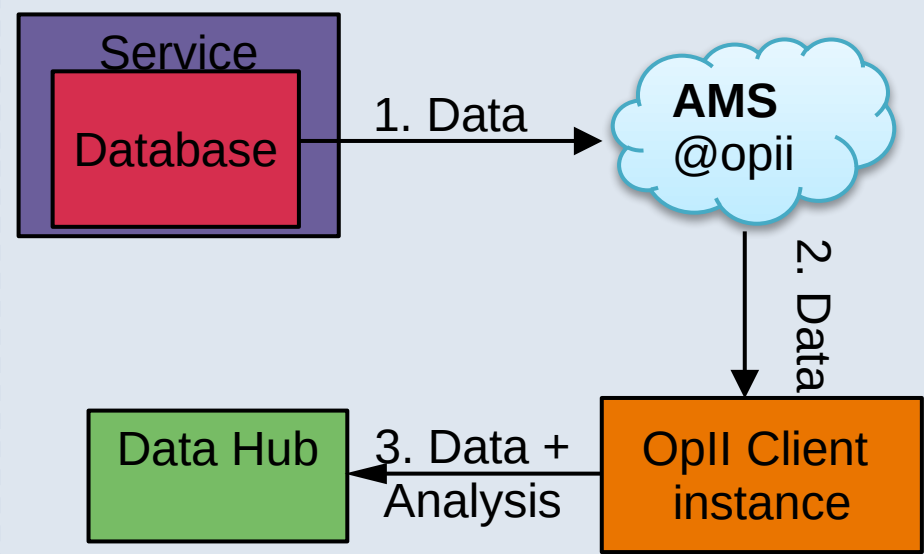
**OpII pulling from a service database:**  
Periodically pull, process, and push to DataHub.



**User pull and process:** IAM group members can pull and process locally.



**Service pushing to AMS:** Services define when data is pushed out.

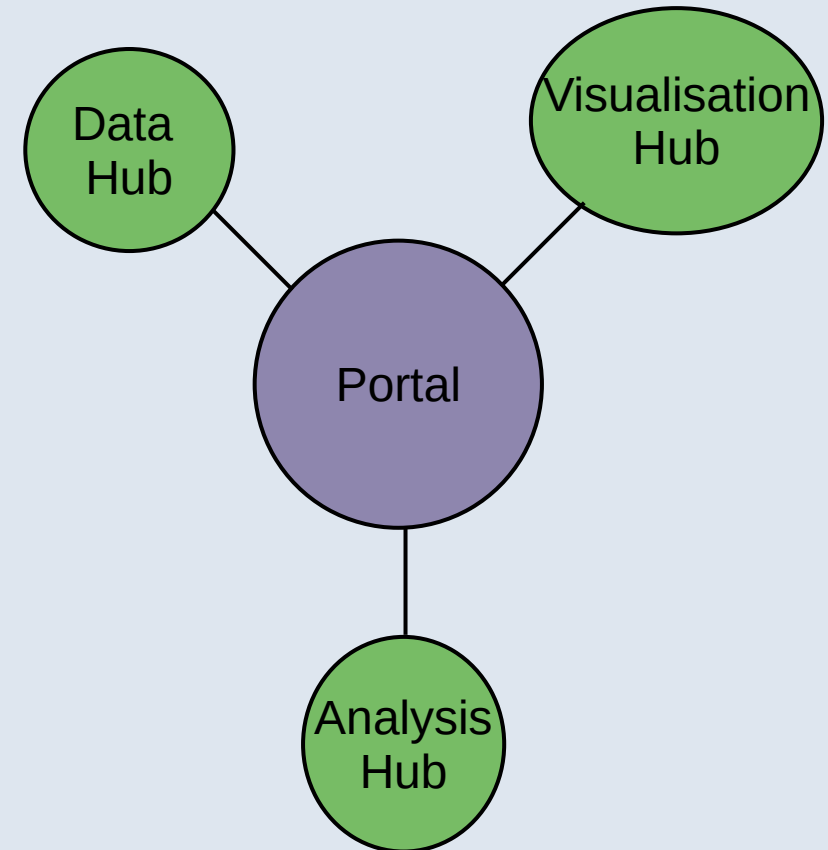


# Infrastructure

# Infrastructure

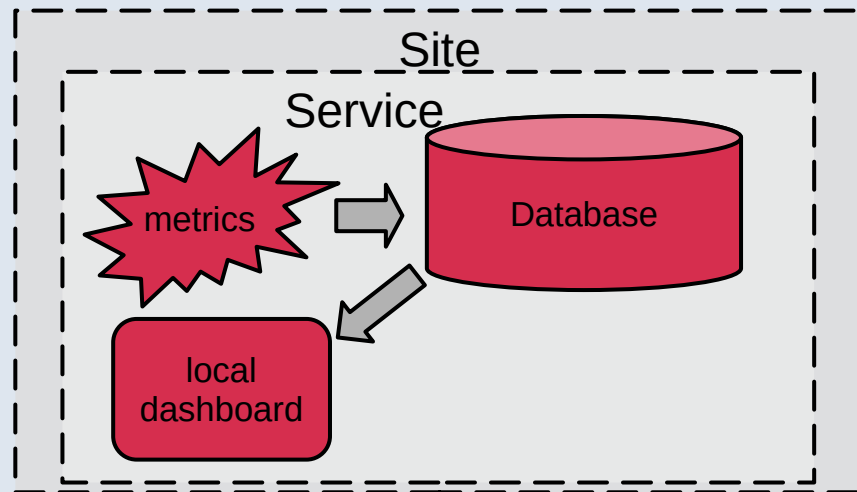
**Servers (“Hubs”)** specialising in serving, processing, or visualising IRIS information.

- **Data Hub** periodically pulls from service databases, runs a REST API, and processes/stores data.
- **Analysis Hub** provides a **user space** for creating analysis workflows and dashboard **mockups**.
- **Visualisation Hub** provides a shared space to create stable **views** on this information.
- A central **portal** which improves findability and provides updates.



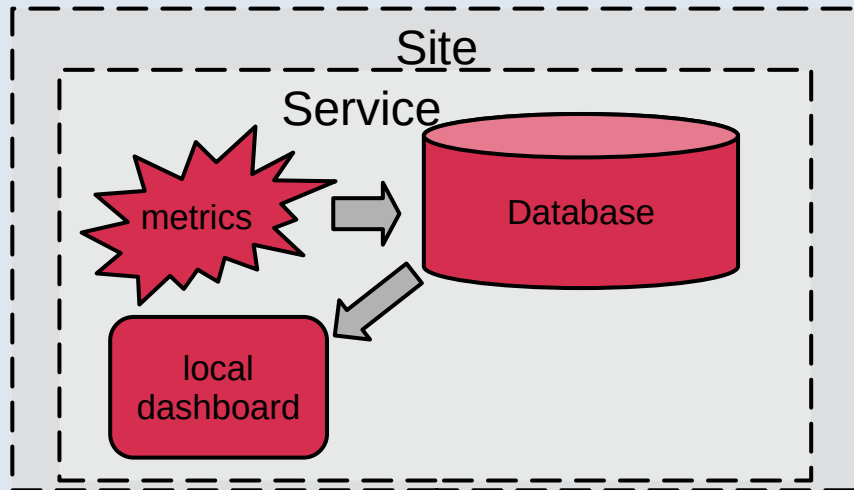
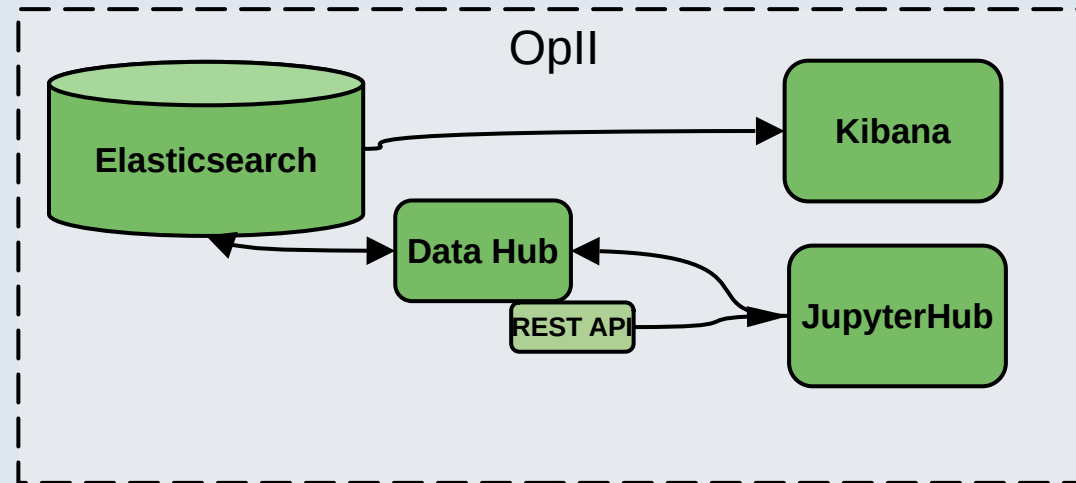
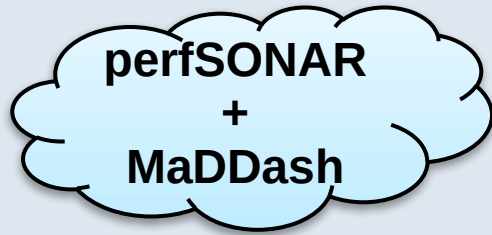
# Infrastructure:

## Architecture model



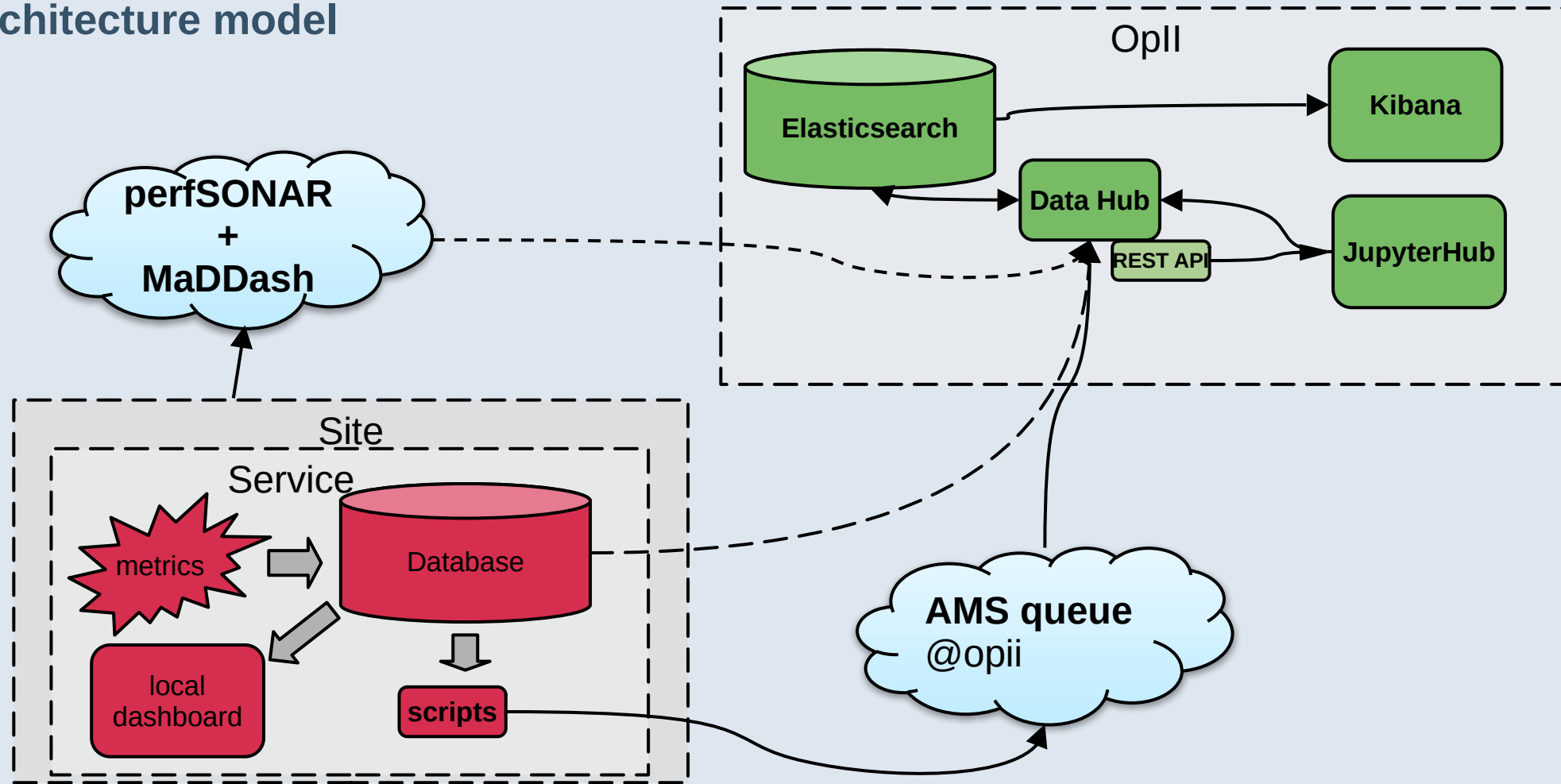
# Infrastructure:

## Architecture model



# Infrastructure:

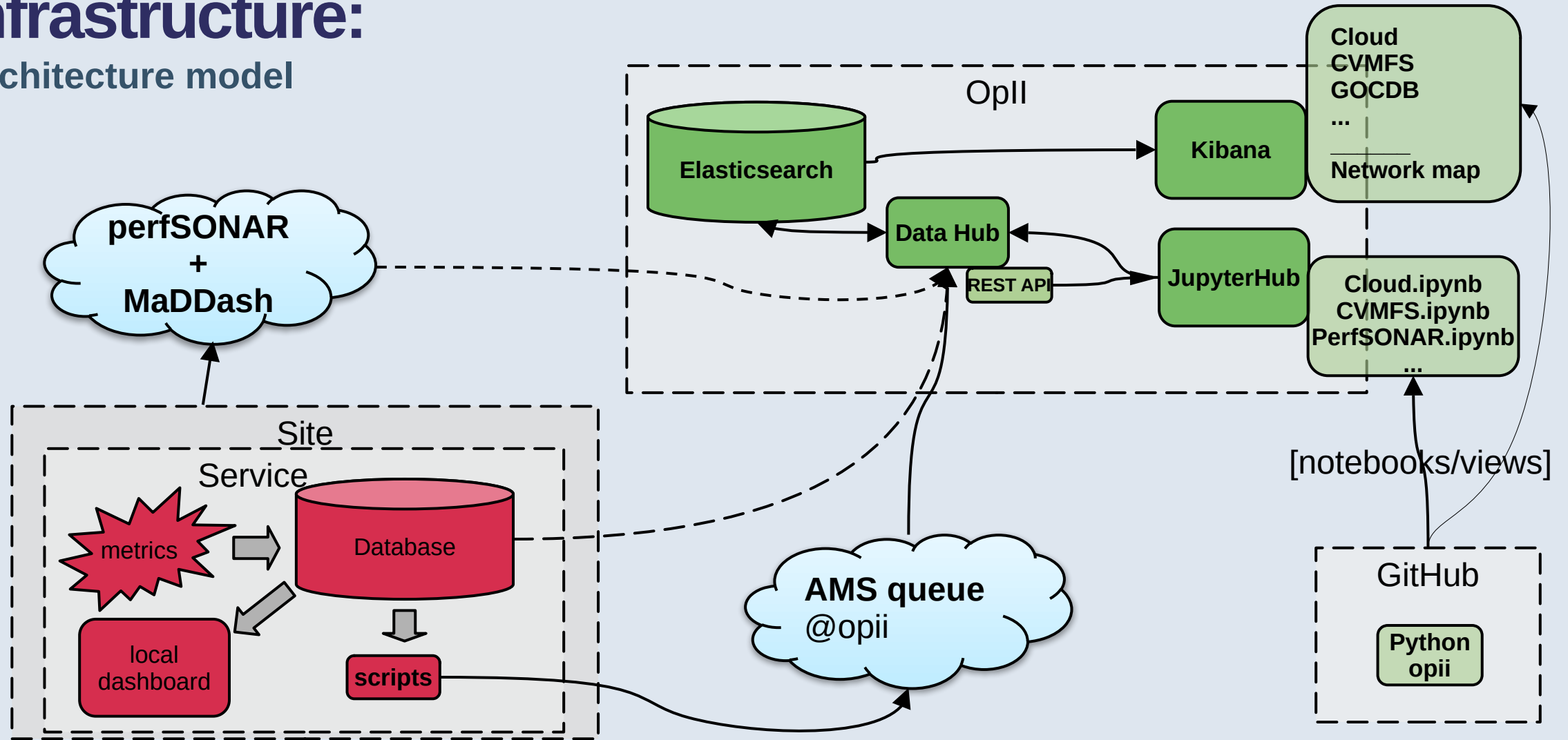
## Architecture model





# Infrastructure:

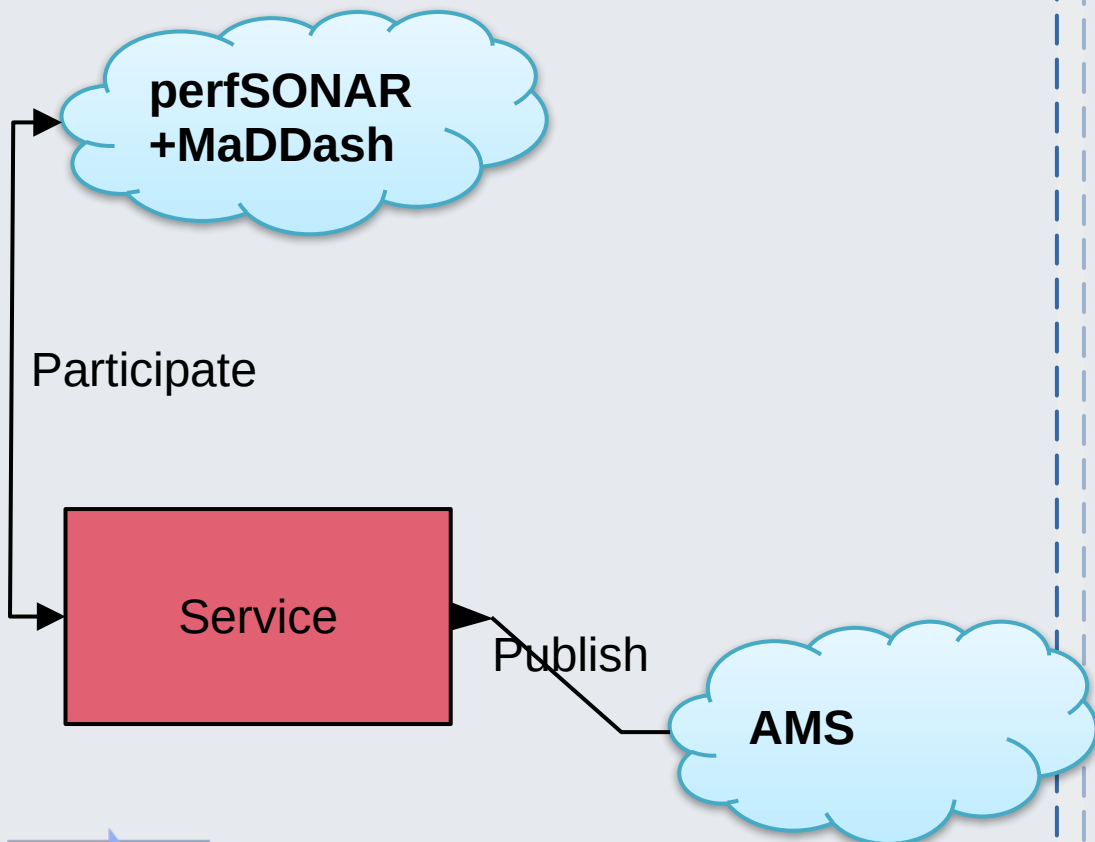
## Architecture model



# Infrastructure:

## Interaction model

Backend

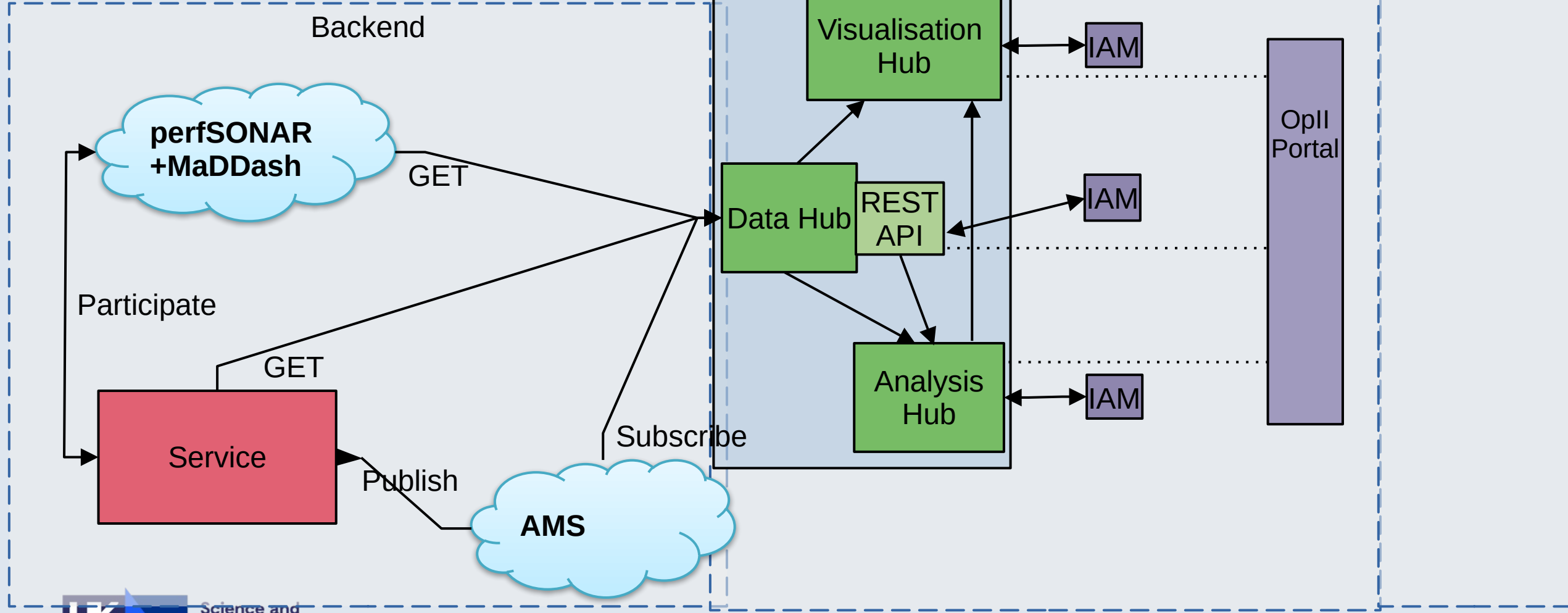


Interface

User

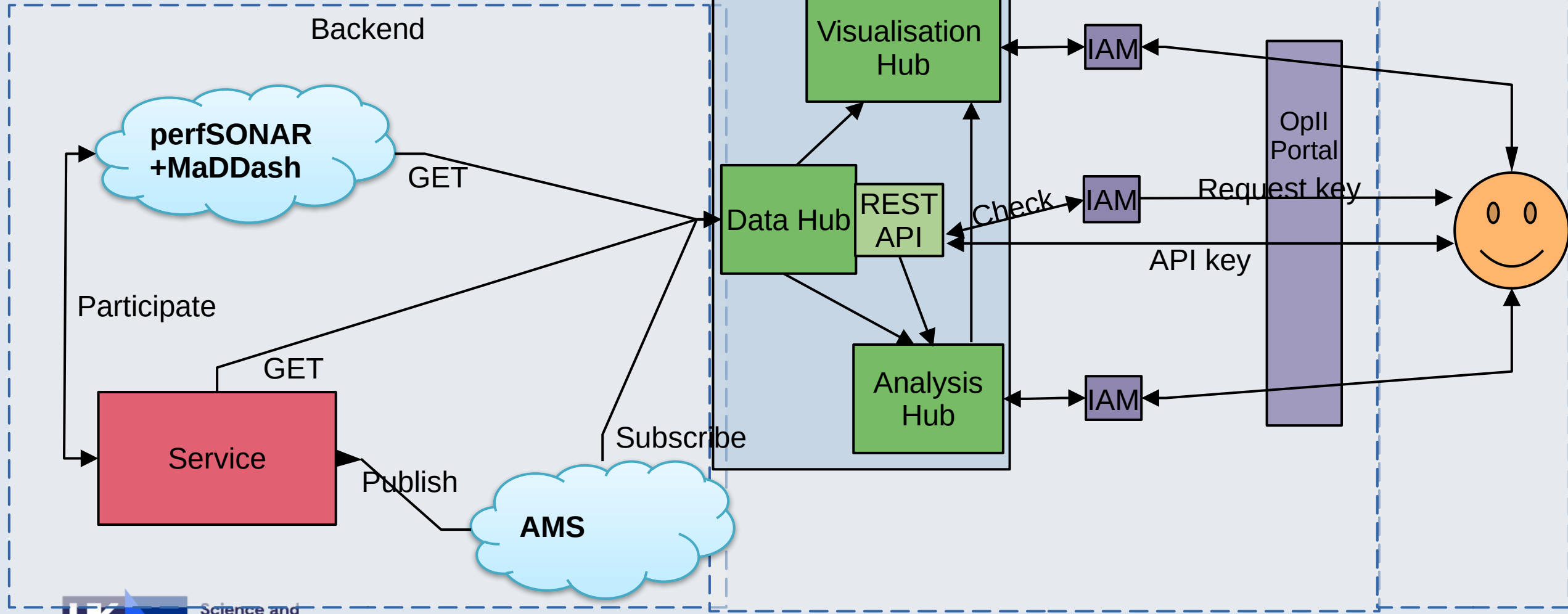
# Infrastructure:

## Interaction model



# Infrastructure:

## Interaction model

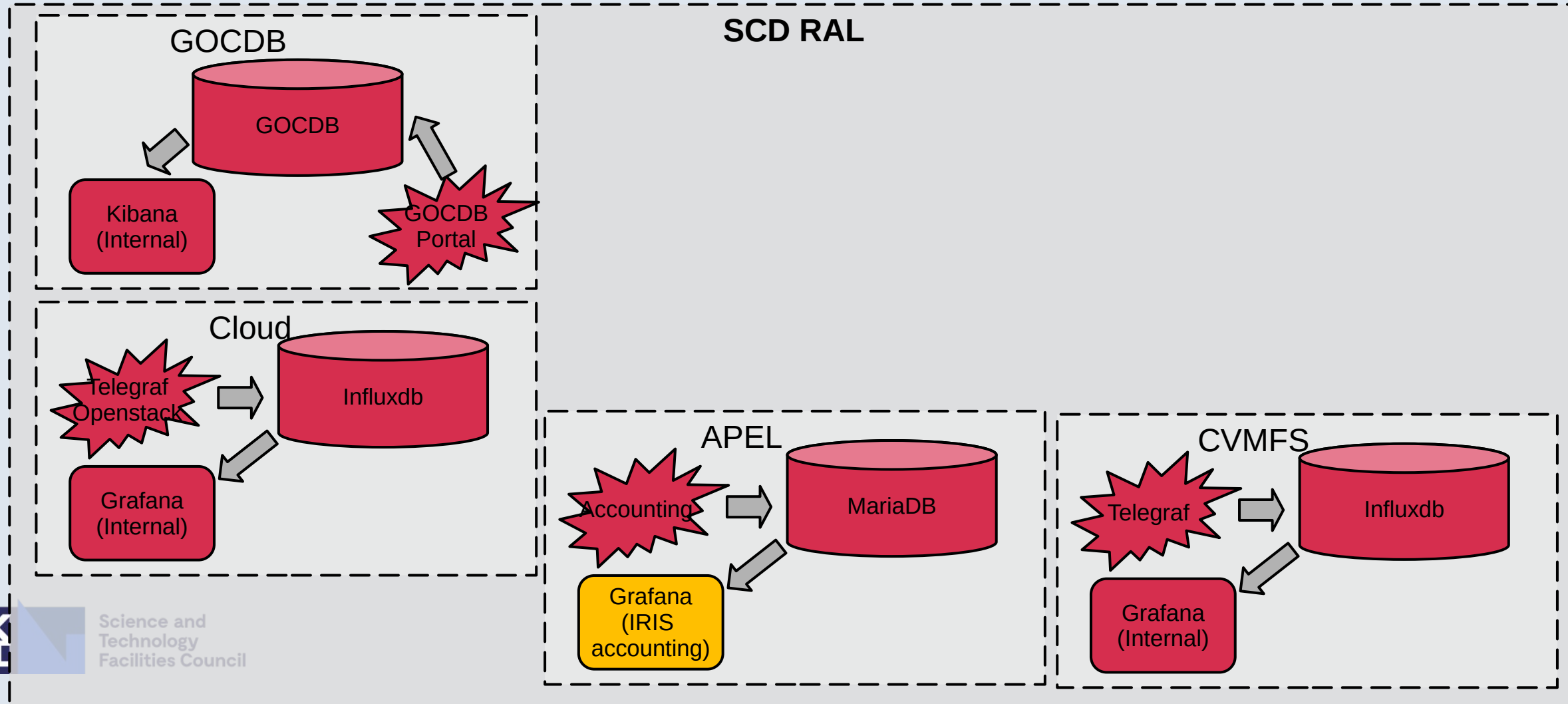


Science and  
Technology  
Facilities Council

# Infrastructure:

## SCD RAL

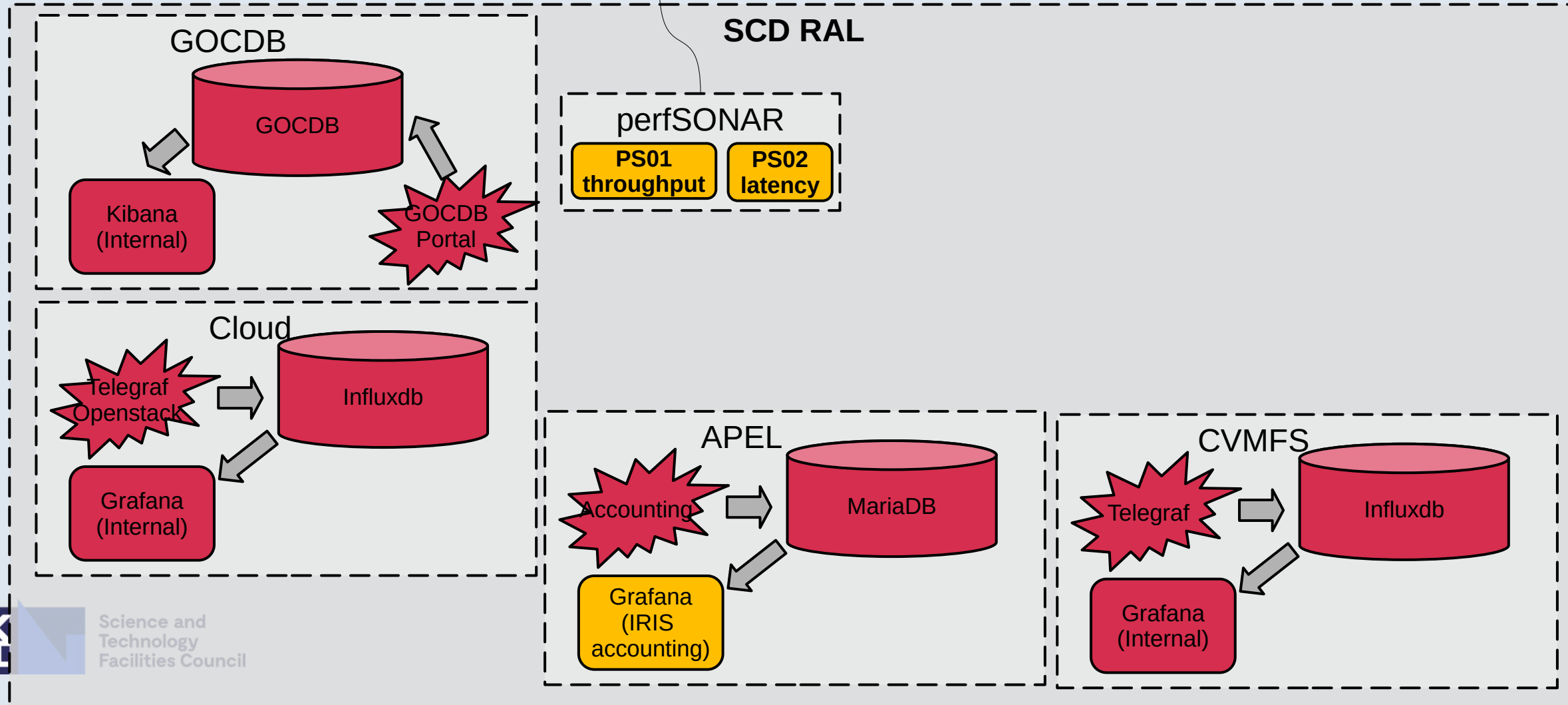
(Internal toy implementation  
as of 2022/02/15)



# Infrastructure:

Implementation at SCD RAL

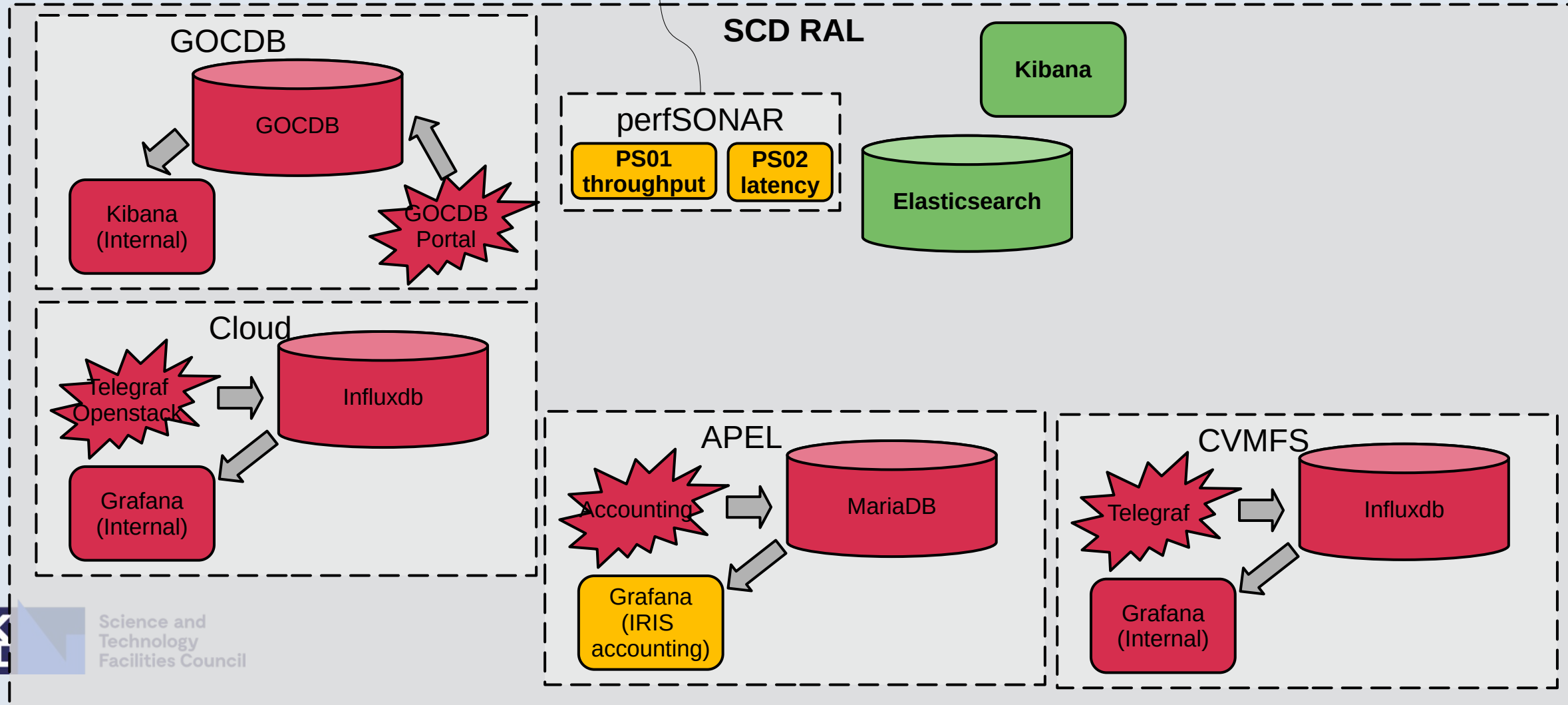
(Internal toy implementation on 2022/02/15)



# Infrastructure:

## SCD RAL

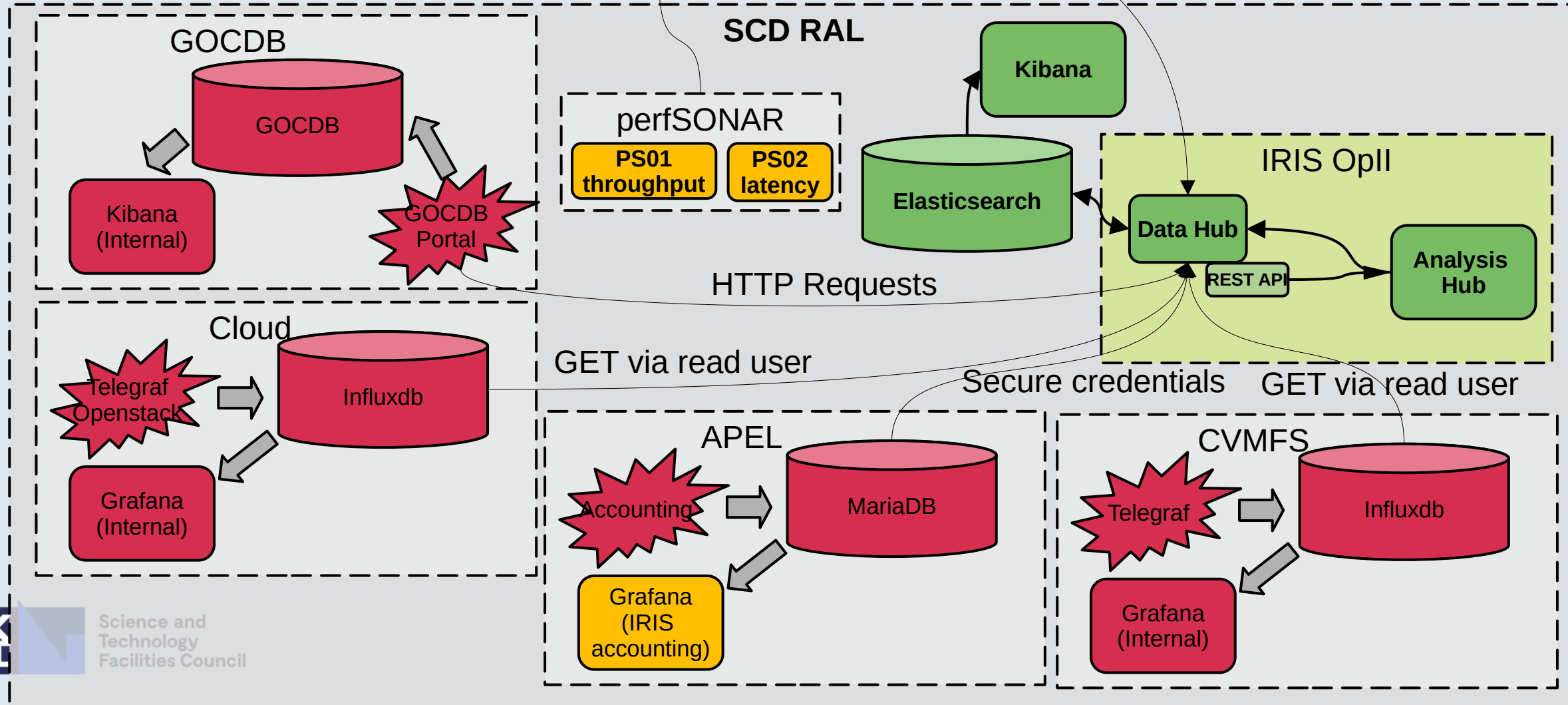
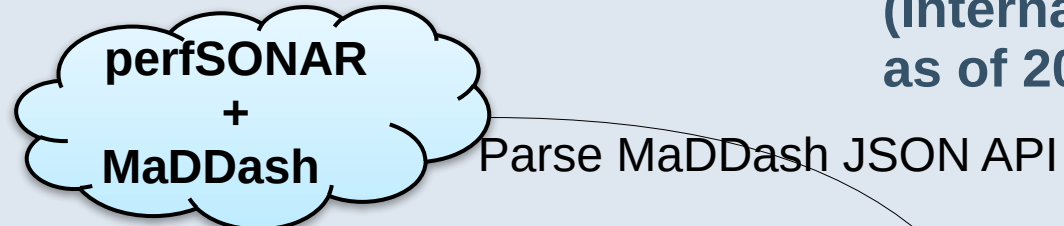
(Internal toy implementation as of 2022/02/15)



# Infrastructure:

## SCD RAL

(Internal toy implementation as of 2022/02/15)





# IRIS Services

**APEL**

**Cloud STFC**

**GOCDB**

Compute at RAL

Certificate Authority

Compute at DiRAC (Leicester)

Compute at GridPP

**CVMFS**

Disk/Tape/CephFS Storage

DIRAC (*Imperial*)

FAST-HEP (Bristol)

High Memory Nodes (Manchester)

**IRIS-IAM**

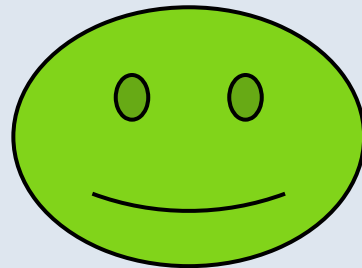
Scientific Openstack (Stackhpc)

RUCIO/FTS

VCycle (CERN)



# Thank You



# perfSONAR



# perfSONAR

Sites can participate by creating perfSONAR “nodes” and providing network test data (from their location) to WLCG/OSG MaDDash via the Esmond API.

## WLCG/OSG New Production perfSONAR Dashboard

☰ Dashboards ☰ Reports ⚙ Settings ↗ WLCG/OSG Networking Resources

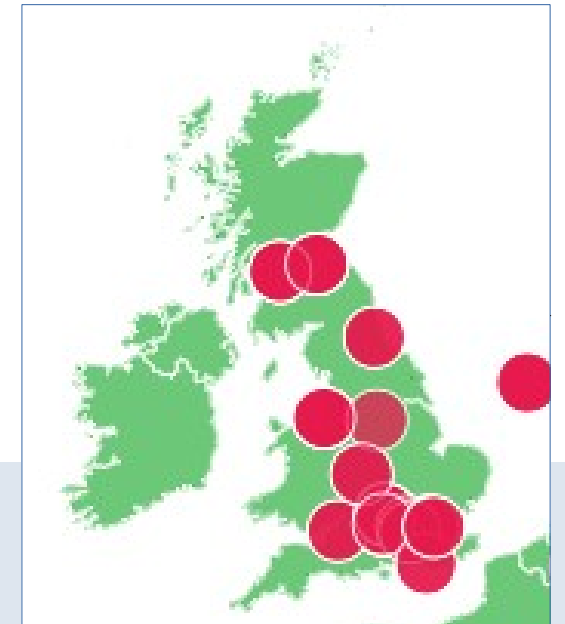
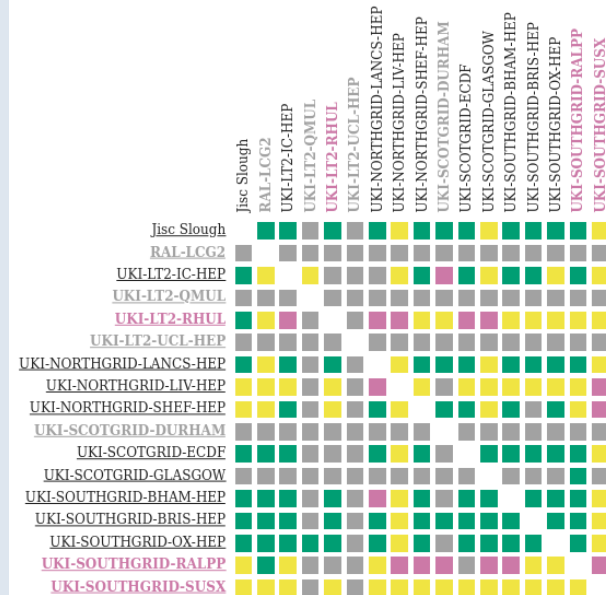
Last page refresh time: January 25, 2022 12:48:25 PM Greenwich Mean Time

### UK Mesh Config Dashboard

UK Mesh Config - UK IPv4 Bandwidth - Throughput

■ Throughput >= 2Gbps ■ Throughput < 2Gbps ■ Throughput <= .5Gbps ■ Unable to find test data ■ Check has not run yet

⚠ Found a total of 7 problems involving 7 hosts in the grid



# Tools: Interfacing

## Databases with query formats:

- Prometheus ----- 

```
$ curl 'http://localhost:9090/api/v1/query?query=up&time=2015-07-01T20:10:51.781Z'
```
- Loki ----- 

```
$ curl -G -s "http://localhost:3100/loki/api/v1/query" --data-urlencode 'query=sum(rate({job="varlogs"}[10m])) by (l
```
- ElasticSearch ----- 

```
0 %curl -i -u el_reader:pa55w0d elasticsearch.ss.ac.uk:5001/log_index/_search/
```
- InfluxDB ----- 

```
6 %curl -i --get 'http://influxdb.ss.ac.uk:8086/query?u=db_reader2&p=pa$$word' --data-urlencode "db=cloud" --data-urlencode "q=SHOW MEASUREMENTS;"
```
  
- APEL ----- 

```
0 %mysql -u supporter -p'as3@k29f/49' -h apel.tld
```
- GOCDB ----- 

```
0 %curl https://gocdb.iris.ac.uk/gocdbpi/public/?method=get_site&sitename=RAL-LCG2
```
- MaDDash ----- 

```
%curl --insecure https://psmad.opensciencegrid.org/maddash/grids "name":"ALICE - ALICE T2 QWAMP Latency - Loss" "uri":"/maddash/c
```

(with perfSONAR, shown later)

# Distribution

## APEL scale:

- APEL scale
  - 131 sites serving 100 VOs with 1168 DNS
  - 4M records/day [Kibana]
  - [?] Messages/day
  - ~ 250byte / record ~ 1GB/day via AMS

## Metrics scale:

- Metric size: 100\*FLOAT32 ~ 3kB
- Machines: 8k/site ~ 1M machines
- Collection precision: “realtime” ~ 1sec - 1min
- Update rate: “realtime” ~ 1min - 60mins

Min\max update rate:

24\1440 messages / day

→ If we want to match the scale of APEL (~1GB/day)

all machines: ~ 0.3 datapoints / machine / day

(or 2400 datapoints / site / day)

aggregate by grid,cloud,storage:

~0 - 3k datapoints / site / day

If we want “realtime”:

all: 12M - 700M dp / site / day, (36GB - 2.1TB/site)

aggregate: 1.5k - 88k dp / site / day, (5MB - 265MB/site)

(AMS is used for realtime metrics already)