Storage Developments at Edinburgh

Peter Clarke, **Rob Currie**, James Perry, Wenlong Yuan

Storage Development

User Facing Developments:

- DUNE Rucio monitoring
- Centralized distributed XCache monitoring dashboard
- LSST Rucio monitoring (WIP)

Work behind the scenes:

- Better protocol support (S3 in Rucio)
- Tool/service debugging/fixes (XRootD)
- StashCache service (another XRootD service)
- Monitoring framework(s) building/design

Monitoring for Rucio

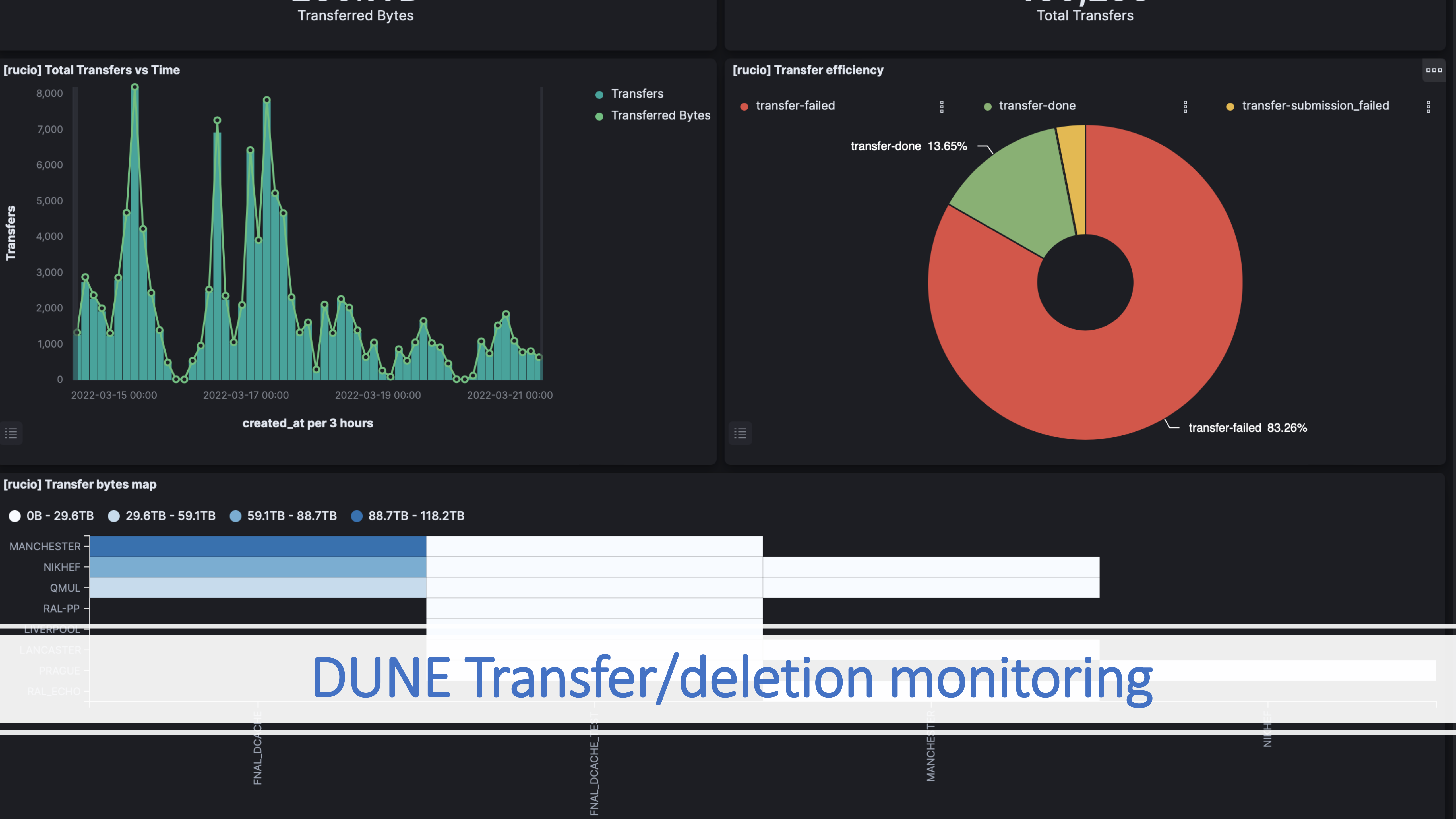
Rucio as a Service

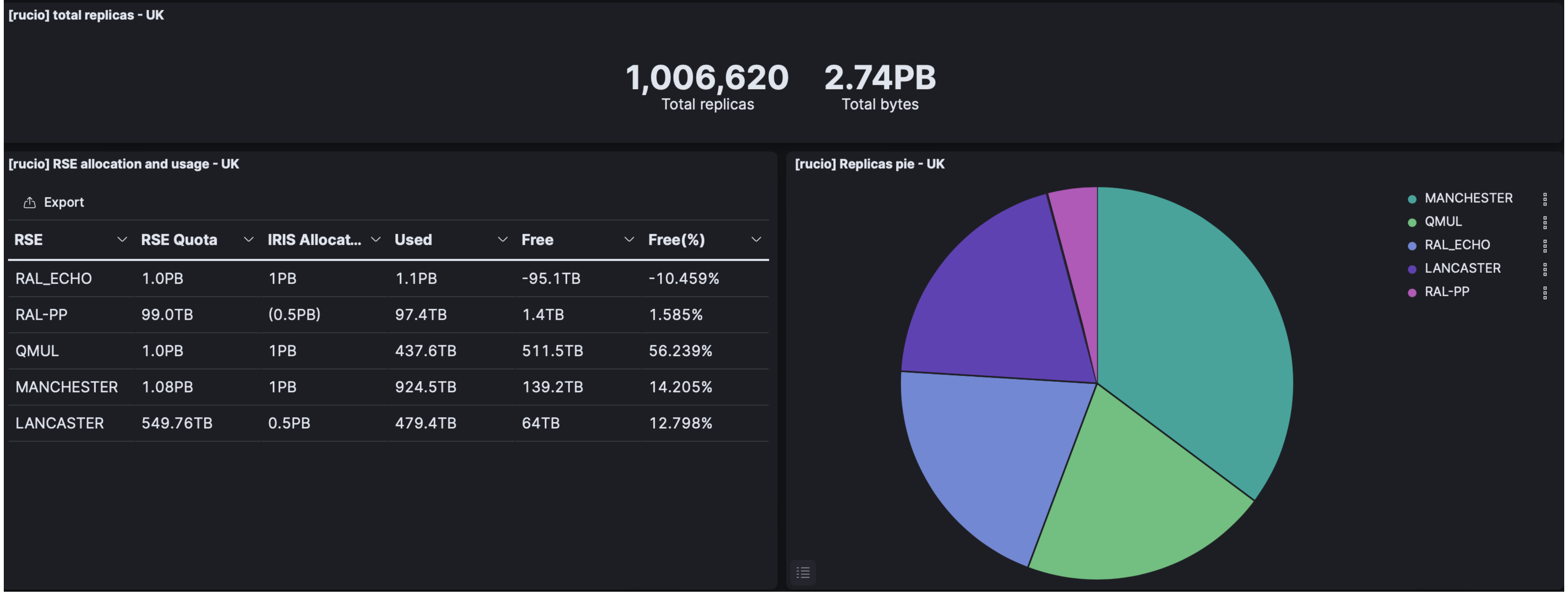
- Storage system health
- Summary of SEs, data location, accounting etc.
- Trace data transferring activities
- Data access pattern analysis

VO support work in Edinburgh

- Deployed Rucio monitoring for DUNE, running as a remote DUNE Rucio monitoring site
- Now Deploying a Rucio monitoring system for LSST

DUNE Transfer/deletion monitoring





[rucio] total replicas - UK

1,006,620 Total replicas

2.74PB Total bytes

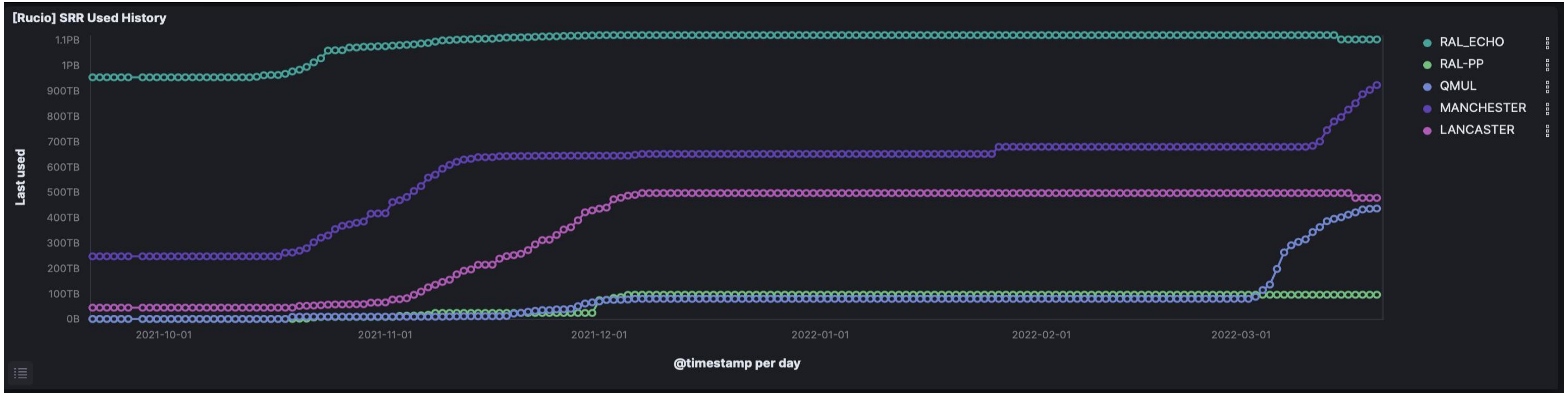
[rucio] RSE allocation and usage - UK

Export

RSE	RSE Quota	IRIS Allocat...	Used	Free	Free(%)
RAL_ECHO	1.0PB	1PB	1.1PB	-95.1TB	-10.459%
RAL-PP	99.0TB	(0.5PB)	97.4TB	1.4TB	1.585%
QMUL	1.0PB	1PB	437.6TB	511.5TB	56.239%
MANCHESTER	1.08PB	1PB	924.5TB	139.2TB	14.205%
LANCASTER	549.76TB	0.5PB	479.4TB	64TB	12.798%

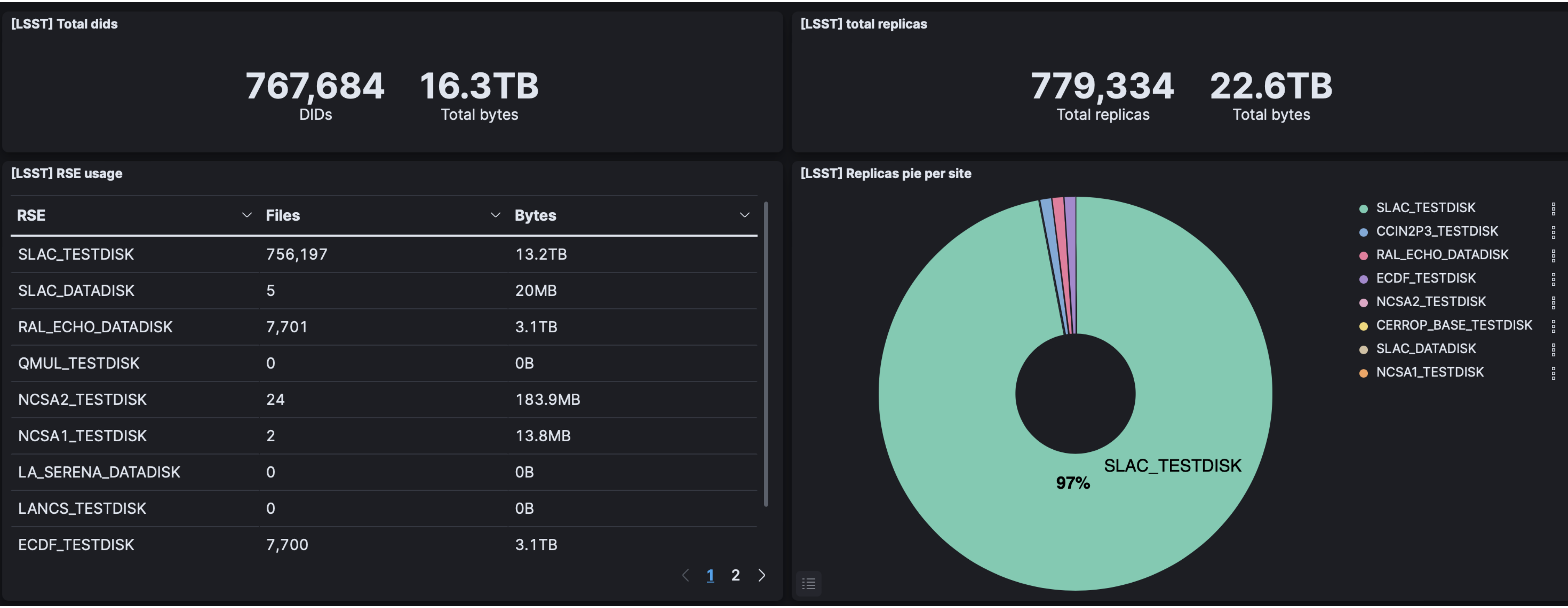
[rucio] Replicas pie - UK

DUNE Transfer/deletion monitoring



[Rucio] SRR Used History

Early Stage LSST Rucio monitoring



[LSST] Total dids

767,684 DIDs

16.3TB Total bytes

[LSST] total replicas

779,334 Total replicas

22.6TB Total bytes

[LSST] RSE usage

RSE	Files	Bytes
SLAC_TESTDISK	756,197	13.2TB
SLAC_DATADISK	5	20MB
RAL_ECHO_DATADISK	7,701	3.1TB
QMUL_TESTDISK	0	0B
NCSA2_TESTDISK	24	183.9MB
NCSA1_TESTDISK	2	13.8MB
LA_SERENA_DATADISK	0	0B
LANCS_TESTDISK	0	0B
ECDF_TESTDISK	7,700	3.1TB

1 2

[LSST] Replicas pie per site

Monitoring Rucio activity

Internal metrics

- Graphite metrics sent by Rucio core and various daemons

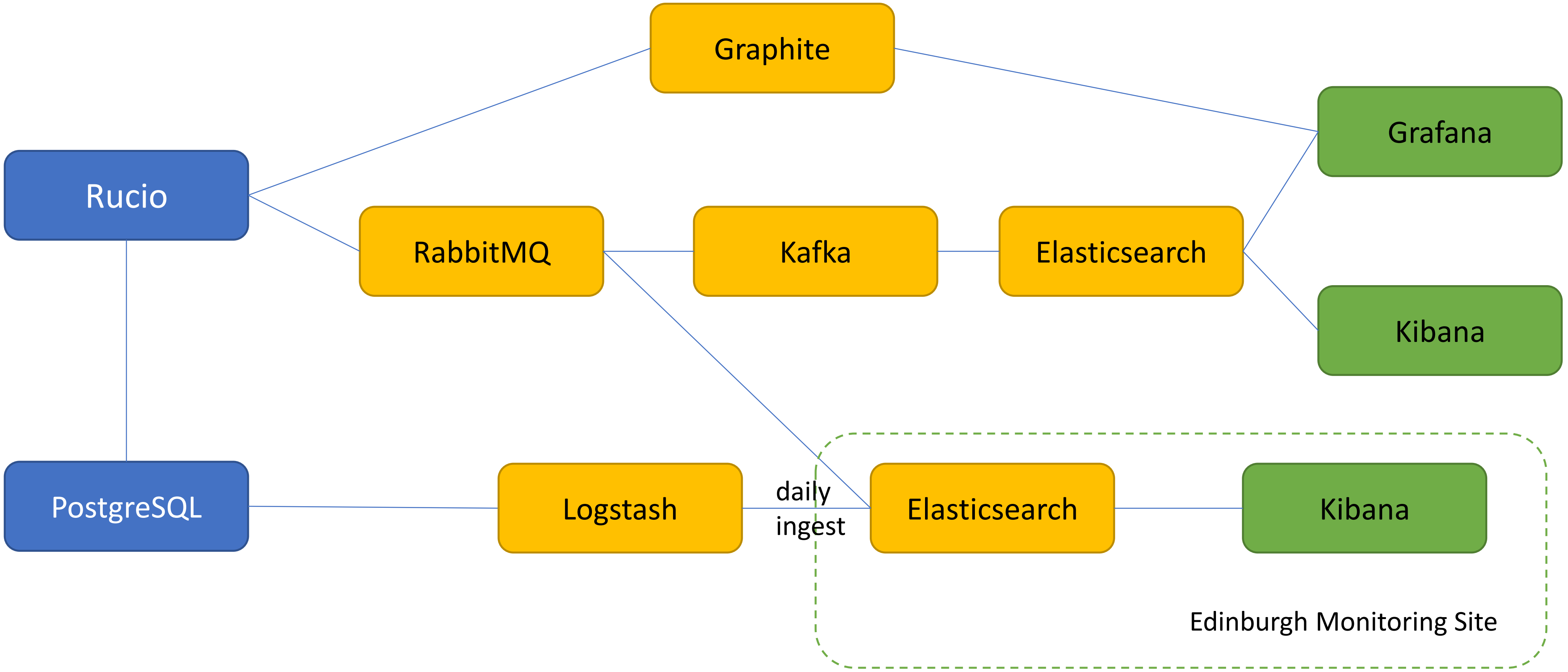
Transfer/deletion monitoring

- Transfer status: submitted, queued, waiting, done or failed messages are sent to a message queue via **Hermes**
- Messages then dumped into ElasticSearch to be visualised using Kibana/Grafana
- **Hermes2** can send messages to ElasticSearch directly

File/dataset/accounting trace

- Trace data are recorded in the Rucio internal database
  - DIDs (data identifier), Replicas (data location), Accounting (RSEs, user accounts) …
- DB tables are dumped to Edinbrugh ElasticSearch cluster periodically to be visualised
  - Daily dumps from FNAL for DUNE, from SLAC for LSST

DUNE Rucio Monitoring infrastructure



Recent core-Rucio developments

New communities have been happy with their adoption of Rucio for distributed file-management.

- One of Rucio's advantages is its ability to plug into an external infrastructure
- To avoid fragmentation and reduce VO-specific code within Rucio "*Policy Packages*" have been developed
- Supporting this has required cross-VO collaboration/investment as well as documentation to support the community
- DUNE was one of the first customers of this

Recent core-Rucio developments (2/2)

- Policy Packages for DUNE has allowed them to customize their Rucio deployment
- One of the key things is that this package allows DUNE to integrate Rucio with their *Metacat* service to have custom LFN2PFN mappings

- We have also worked to support “s3” as a first-class protocol within Rucio
- In addition to this, working with DUNE and other communities there is an ongoing effort to reduce the requirements of the rucio-clients which benefits multiple-VOs

XRootD Behind the Scenes

This protocol/service is widely tested/used/relied-on across HEP which allows us to manage data at scale using X509 based authentication/security.

- Service has recently undergone a major behind the scenes re-write with long-term support in mind
- Evolving landscape is putting new requirements on this service (as well as others!), e.g. token support, macaroons, etc…
- Some corner-cases are starting to creep in regarding XRootD and advanced configurations/setups.
  IMO this emphasises that more testing and more eyes/development is needed

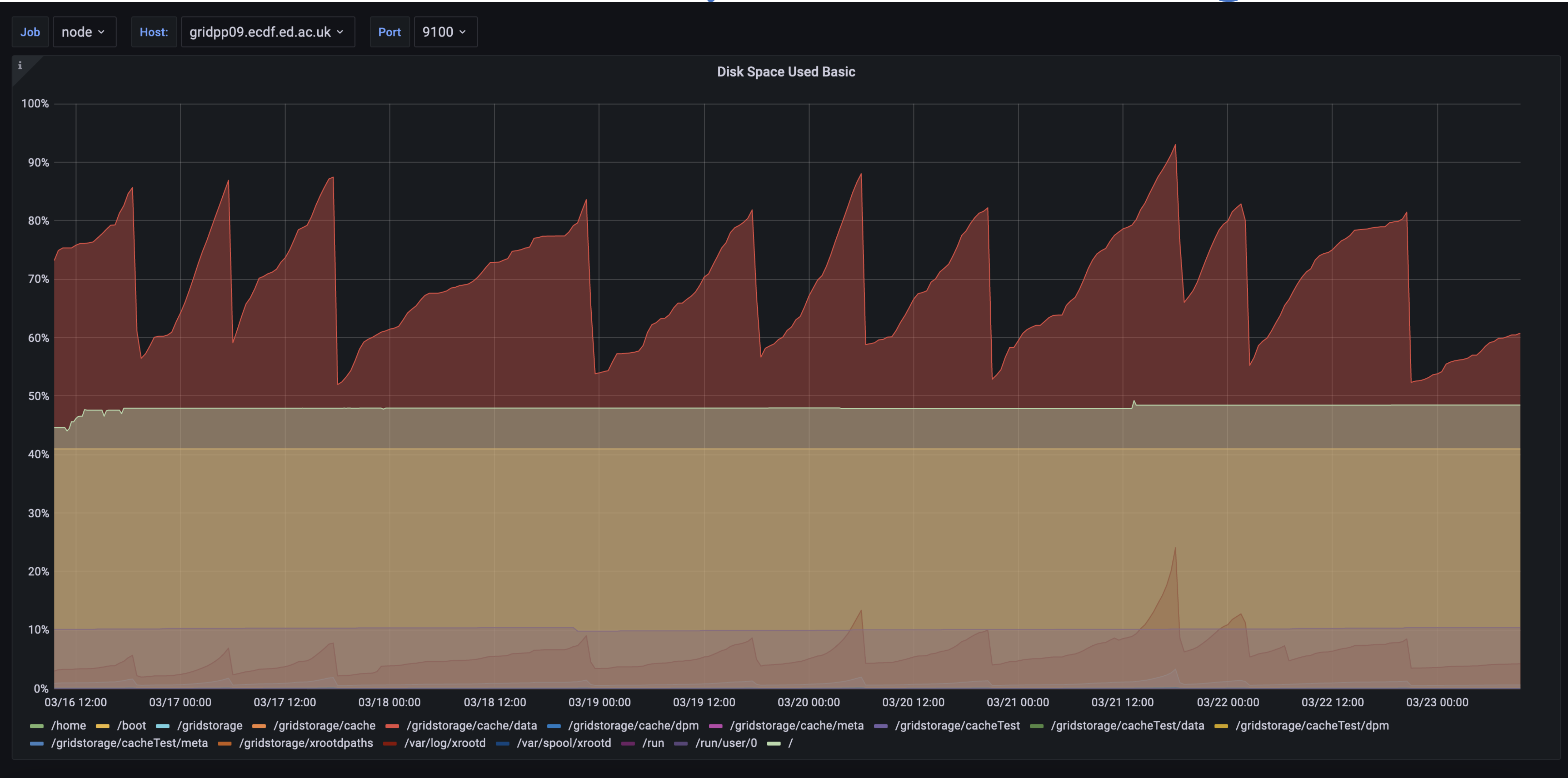
XRootD as a Service

- XRootD as a service has some long-term stability issues
- Common to restart it as a service ~every 24hr
  (Ideally this shouldn't be needed)
- Debugging crashes at Edinburgh we've identified a lot of problems as being related to the CentOS7-host (specifically OpenSSL-1.0.2)
- We're working to understand the full impact of this, but will likely advise an OS upgrade for XCache services once we've finished looking into this in more detail…

- Main advantage of this has been developing a familiarity with the XRootD framework codebase and build system

- Plan is to optimise the behaviour of our XCache by combining ML/AI heuristics with XRootD to improve file caching/purging decisions

XCache Filesystem Monitoring



Job: node

Host: gridpp09.ecdf.ed.ac.uk

Port: 9100

StashCache Service

- StashCache is used by some VOs such as DUNE as an alternative to CVMFS when transferring large files in a similar way to WN (http over XRootD)
- To support this, we have deployed a testing instance at Edinburgh
- Installing this from scratch required working with the OSG such that Edinburgh and the cache are registered in the appropriate systems
- Setting this up is a relatively simple process as the service is based on XRootD+plugins from an OSG repo
- Monitoring this will require us to fall back on our experiencing monitoring other services at the site

Production Monitoring at Edinburgh

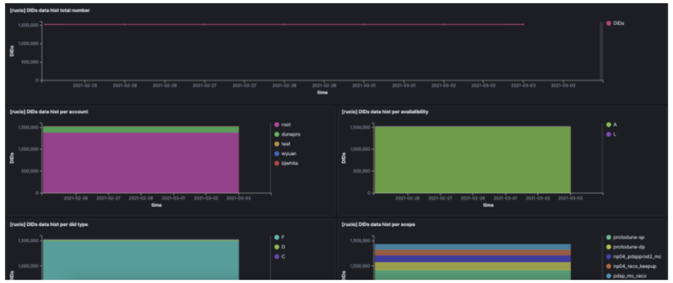
- Currently Supporting DUNE and LSST VOs as well as XCache-UK monitoring using single ELK stack
- https://monitoring.edi.scotgrid.ac.uk/
- Notionally “small” hardware requirements, so running on retired storage node for now
- Ingesting data both directly and via a RabbitMQ messaging system

Edinburgh-GridPP Monitoring

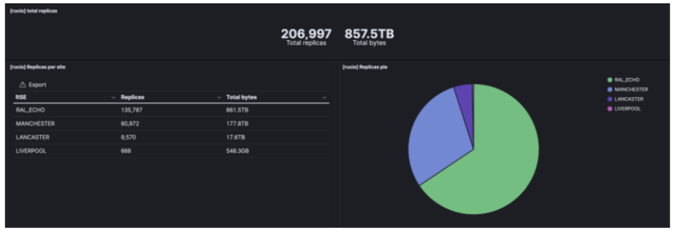
---

new for 2021!

DUNE 7Day RUCIO Monitoring


[rucio] DIDs data hist total number
DIDs
[rucio] DIDs data hist per account
root
dunepro
test
wyuan
bjwhite
[rucio] DIDs data hist per availability
A
L
[rucio] DIDs data hist per did type
F
D
C
[rucio] DIDs data hist per scope
protodune-sp
protodune-dp
np04_pdspprod2_mc
np04_reco_keepup
pdsp_mc_reco
time
2021-02-26
2021-02-27
2021-02-28
2021-03-01
2021-03-02
2021-03-03

DUNE UK Monitoring


[rucio] total replicas

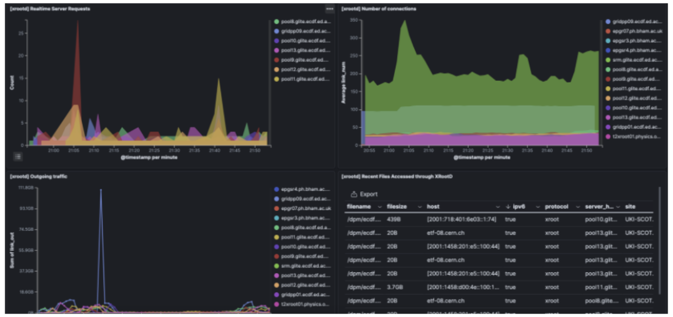
206,997 Total replicas

857.5TB Total bytes

[rucio] Replicas per site

RSE	Replicas	Total bytes
RAL_ECHO	135,787	661.5TB
MANCHESTER	60,972	177.8TB
LANCASTER	9,570	17.6TB
LIVERPOOL	668	548.3GB

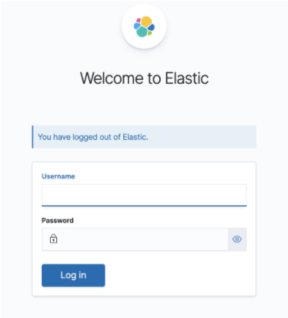
GridPP XRootD Monitoring



[xrootd] Recent Files Accessed through XRootD

filename	filesize	host	ipv6	protocol	server_h...	site
/dpm/ecdf....	439B	[2001:718:401:6e03::1:74]	true	xroot	pool10.glit...	UKI-SCOT.
/dpm/ecdf....	20B	etf-08.cern.ch	true	xroot	pool13.glit...	UKI-SCOT.
/dpm/ecdf....	20B	[2001:1458:201:e5::100:44]	true	xroot	pool13.glit...	UKI-SCOT.
/dpm/ecdf....	20B	etf-08.cern.ch	true	xroot	pool13.glit...	UKI-SCOT.
/dpm/ecdf....	20B	[2001:1458:201:e5::100:44]	true	xroot	pool13.glit...	UKI-SCOT.
/dpm/ecdf....	3.7GB	[2001:1458:d00:4e::100:1...	true	xroot	pool11.glit...	UKI-SCOT.
/dpm/ecdf....	20B	etf-08.cern.ch	true	xroot	pool8.glite....	UKI-SCOT.

Dev Kibana Instance


Welcome to Elastic

You have logged out of Elastic.

Username

Password

Production Monitoring at Edinburgh (2/2)

- Have discovered more tasks could be simplified by improving our site monitoring
- Plan to use same infrastructure to support our local HEP group by ingesting clean-room monitoring data feeds into our ELK stack for remote/centralised monitoring of air-quality https://gitlab.cern.ch/guescini/canary/-/wikis/home
- Our production ELK stack was our first attempt at building a monitoring stack.

  Can we now do better?

Building a new Monitoring Stack

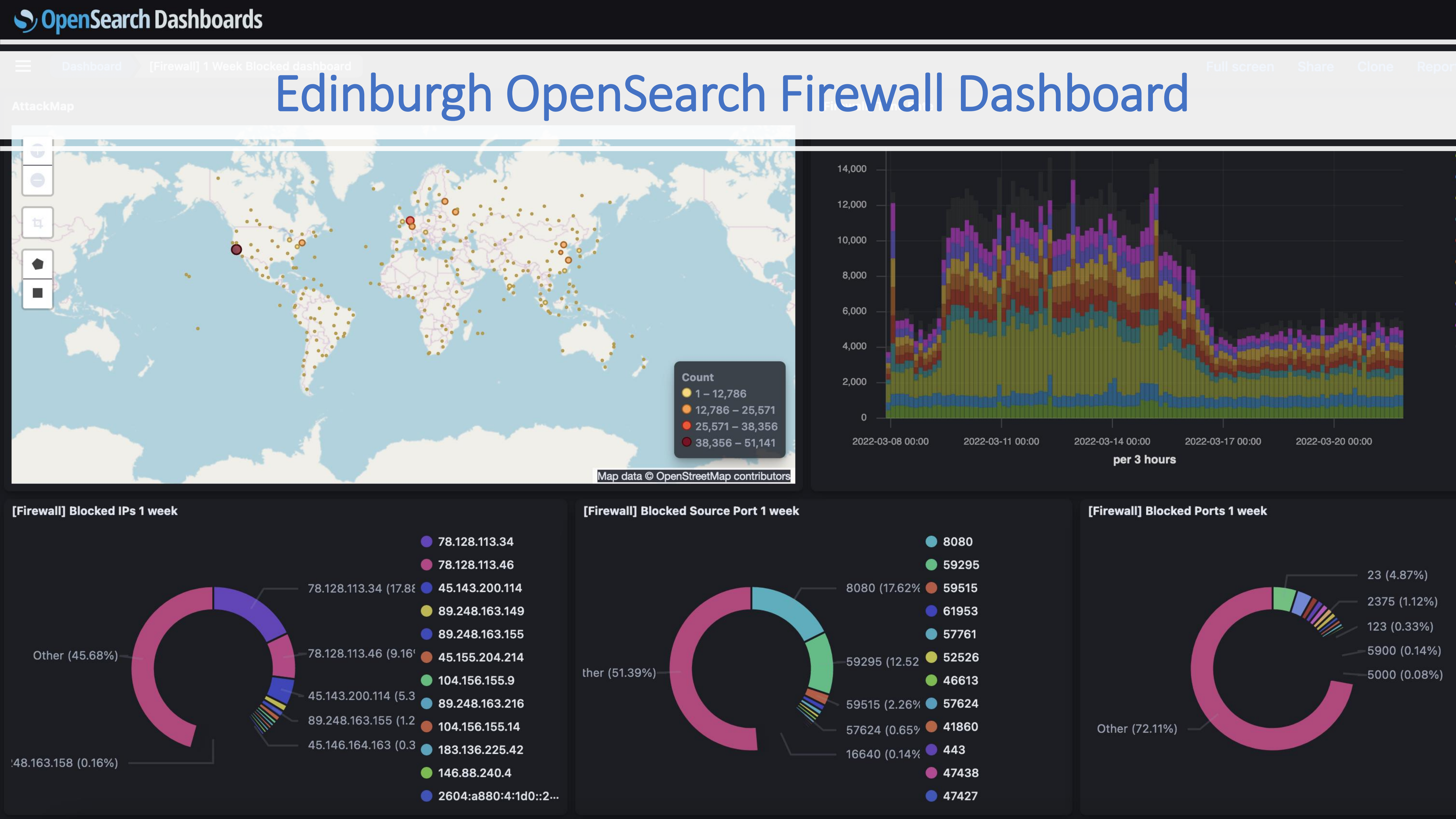
- Since we deployed our ELK cluster, the OpenSearch fork has gained popularity.
- We have recently tested a new OpenSearch based cluster for comparison to ELK.
- Behind the scenes there are battles going on between OpenSearch-(Amazon.com) and ELK-(elastic.io).

  Who can win over most of the community/ industry-customers?
- My quick summary is:

  IMO OpenSearch offers more to us as a community (HEP/GridPP). I'm aware there are some larger deployments being planned reflecting this.

  This has less emphasis on paid-for features, and we're interested in potentially developing our own tooling atop these tools already used in industry.

Edinburgh OpenSearch Firewall Dashboard



Conclusions

- We are supporting DUNE and LSST with monitoring of their RUCIO services and extracting high-level data from their systems

- Built a system for remote monitoring of XRootD instances, will be watching how this compares to the new WLCG XRootD monitoring system, we may also find having a GridPP instance useful for different reasons

- Have developed a familiarity with different monitoring technologies and how to integrate them successfully (and lots of what not to do… see backups for more)

- Are working closely with different VOs to support tooling required for many different storage workflows and different uses of Rucio

BACKUPS

Production Monitoring at Edinburgh

- Original ELK stack was setup circa 2016 to meet a minimally defined set of requirements
- Containerised deployment has helped in upgrading/maintaining
- Have learned a lot more since then about ELK systems as well as best practice when deploying similar technologies
- ElasticSearch is like a large database in many ways
- Good Kibana use requires a good understanding of the whole ELK model
- Ingesting data is difficult to get right, there is logstash, but this has proven difficult to use/maintain (based on our testing)

Why does Monitoring Infrastructure Design Matter?

1. **Well defined things I know about.**

   CPU/Memory usage?
   How many logins have there been?
   What is the IP of the incoming connection?

   For situations like this you have: schema-on-write

2. **Things that aren’t known in advance**.

   How did X happen?
   What happened during a (security) incident?
   What went wrong in an unexpected way when …?

   For these situations you can use: schema-on-read

Monitoring Infrastructure

Fair to say that “monitoring” and “big data” are on a collision course. (Some would say they have already collided)

If care isn’t taken, can quickly end up with a very fragmented ecosystem, however still no 1 tool meets all requirements.

“Newest” players in system monitoring are:

1. PLG (Prometheus Loki Grafana)
2. ELK (ElasticSearch LogStash Kibana)
3. OFD (OpenSearch FluentD Dashboards)

Which Infrastructure Should I use?

	PLG	ELK	OFD
Pros	• Easy to Setup • Simple user-interface • Lots of shared projects from community (drag&drop solutions) • Simple non privileged exporter	• Tested with industry experience • Advanced tooling available • Allows examining data post-collection ***schema-on-read*** • http(s) based protocol for all access	• Active open development across multiple projects • Features such as anomaly detection built-in (for free!) • Strong backing from industry projects • Builds atop experience from ELK • Allows examining data post-collection ***schema-on-read***
Cons	• Ecosystem built around ***schema-on-write*** • Scalability more difficult	• Licensing is difficult/annoying • Advanced features are not-free in cost of freedoms • Complex/Difficult permissions model(s) • Complex UI/management • Increasingly cloud-orientated model	• Ecosystem is evolving rapidly • Complex/Difficult permissions model(s) • Compatibility issues regarding ELK • Exporting/ingesting data is potentially difficult

So, what monitoring should I use?

Not a straight-forward question to answer. Ultimately, whatever works best for you.

- For well defined metrics, PLG is such a pleasant experience to setup/use I still recommend it
- For ingesting logs and searching them after-the-fact I would seriously push you to OFD
- FluentD is potentially a much better tool than logstash IMO and offers much more flexibility in setting up data ingestion