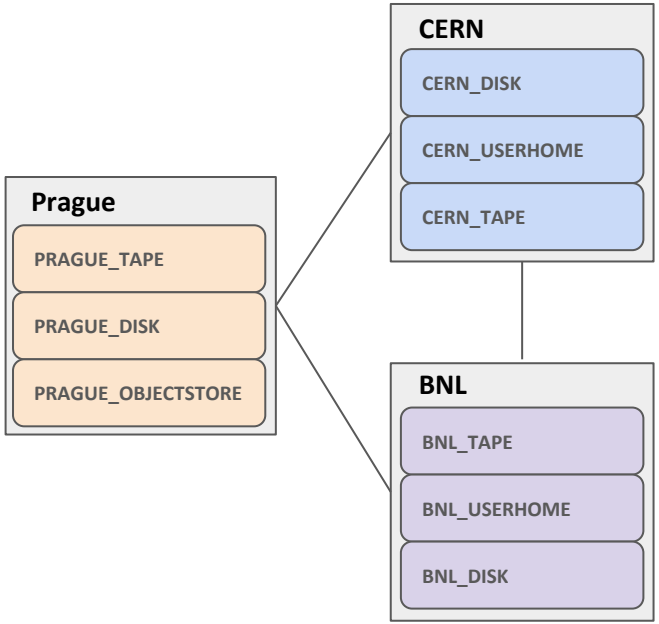
Chapter 2: Storage

The *Rucio Storage Element* (RSE)

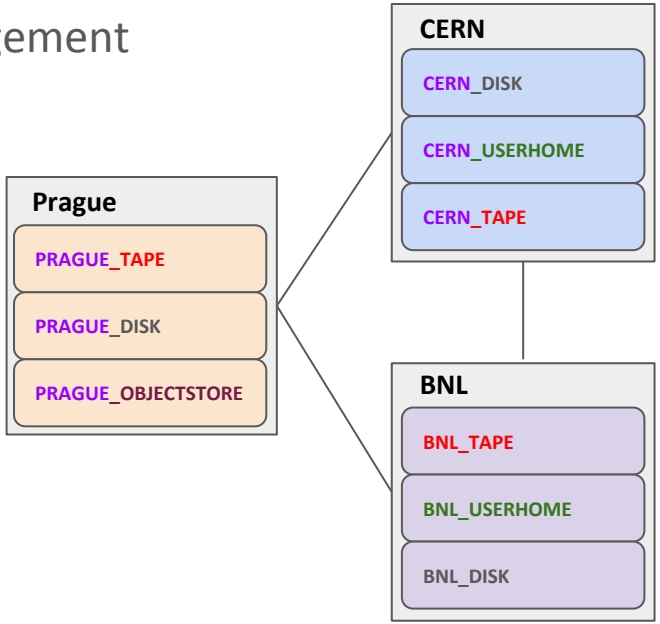
- What it is not (to get it out of the way)
  - A piece of software that you need to run at a data centre
- An RSE is an abstraction of a storage endpoint
  - A unique identifier of any potential storage area
  - A logical entity of space
  - The smallest addressable unit of storage
  - An entry in the Rucio database
  - A collection of arbitrary metadata
  - The target for quotas, limits, and accounting



RSE grouping and metadata

- The set of all RSEs form the topology for data management
- Fixed set of necessary metadata
  - `name, type, availability, …`
- To give users flexibility each RSE can host an arbitrary amount of custom metadata
- Can be set and redefined at runtime
- Can be used to build replication rules (cf. Ch 3)

- Examples
  - `is_tape=True`
  - `user_software_available=True`
  - `min_free_space=10000`
  - `region=Europe`



Protocols

- Each RSE can support multiple data access protocols
  - Data centre administrators should have their choice of preferred protocols
- Currently supported protocols
  - XRootD, HTTP/WebDAV, GridFTP/GSIFTP, S3, POSIX, SRM
- Priorities can be set for *read*, *write*, *delete*, and *third-party-copy* operations
  - Additionally with WAN and LAN support, if necessary, to support customised storage access

- Example
  - BNL_DATADISK **WAN**: *read*:gsiftp,srm *write*:- *delete*:gsiftp *third-party-copy*:webdav
    LAN: *read*:root,webdav,gsiftp *write*:root *delete*:-
  - CERN_TAPE **WAN**: *read*:- *write*:- *delete*:srm *third-party-copy*:srm

Replicas

- File DIDs eventually lead to replicas
  - A replica is the physical representation of the file, i.e., bytes on storage
  - There can be files with zero replicas
- For existing files on storage
  - Can be registered as-is directly into the Rucio catalogue
  - File DIDs will retain their full path information as given by the client/user
- When uploading new data there are two possibilities
  - Leave the decision of the path on storage to Rucio (= automatically managed storage namespace)
  - Continue to provide full paths on the storage to the file
- Automatically managed storage namespace is function-based and customisable

Replica resolution

- Locating and accessing the data is dependent on the client location
  - Client locally at data centre with attached storage (LAN) or access remotely via the network (WAN)

- Example resolving DID *user.jdoe:my-analysis-data-123.tar.gz* (scope: user.jdoe, name: my-analysis-data-123.tar.gz) on three different RSEs

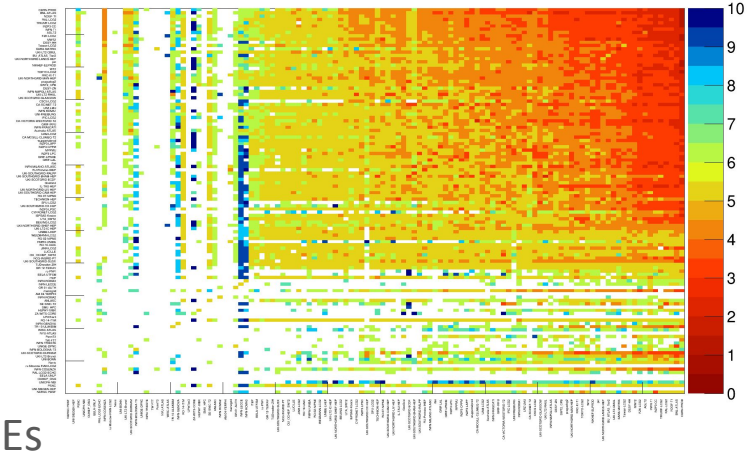
RSE	Protocol	Host	Port	Prefix	Path
A	https	mystorage.ch	443	/user/area/	
B	root	server.edu	1094	/storage/	
C	gsiftp	securedisk.de	8446		/user/defined/datafile.gz

Automatic
A → `https://mystorage.ch:443/user/area/user.jdoe/34/65/my-analysis-data-123.tar.gz`
B → `root://server.edu:1094/storage/user.jdoe/34/65/my-analysis-data-123.tar.gz`

Client-provided
C → `gsiftp://securedisk.de:8446/user/defined/datafile.tar.gz`

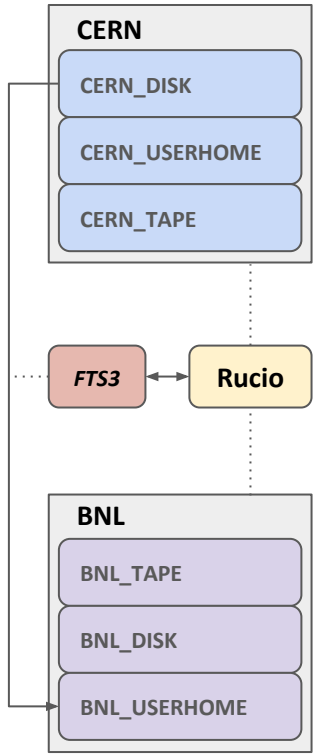
Distance

- Connected RSEs have an arbitrary non-zero integer distance value
- Distance influences the sorting of files when considering sources for transfers
- Distinct concept from the more dynamic network metrics
- Zero distance blacklists a link between two RSEs
- We also support geographical distance (GeoIP) and transfer-queue lengths
- Plumbing
  - Monthly reevaluation of the collected average throughput of file transfers between RSEs
  - Logarithmic application of values (0 .. site local, 10GB/s .. 1, 1KB/s .. 11)



Third party copy

- Orchestrated large-scale data movement
  - Reliable direct file transfers between two RSEs
  - Delegated to a file transfer service
- Rucio provides a generic transfertool API for third party copy
  - Independent of any underlying transfer service
  - Asynchronous interface to any potential third-party tool
- Currently available implementation of transfertool API is FTS3
- Other transfertools such as GlobusOnline can be integrated
- Interaction with transfertool is highly optimised
  - Bulk operations, message passing, retrials, multi-source sorting, …



Object Stores

- Rucio supports S3-style object stores directly
  - Such as Ceph/Rados, OpenStack Swift, Amazon S3, Google Cloud Platform
  - Can use signed URLs and access keys authentication
- Third party copy supported with HTTP/WebDAV
- Useful for large volumes of small objects
  - E.g,. in ATLAS as backend for logfiles and opportunistic compute outputs

Caches

- RSEs can be tagged with the special metadata `volatile=True`
- Volatile RSEs can have data movement and deletion not orchestrated by Rucio
- External cache-controller must register/unregister replicas to Rucio
- Message passing is supported to synchronise cache updates
- Two caches are directly supported
  - ARC-Cache — cf. Friday morning presentation by D.Cameron
  - Xcache — dedicated to the XrootD protocol

Questions?

If you have a question but don't get the chance to ask it directly during the session, you can do it here: https://goo.gl/BdSGoC