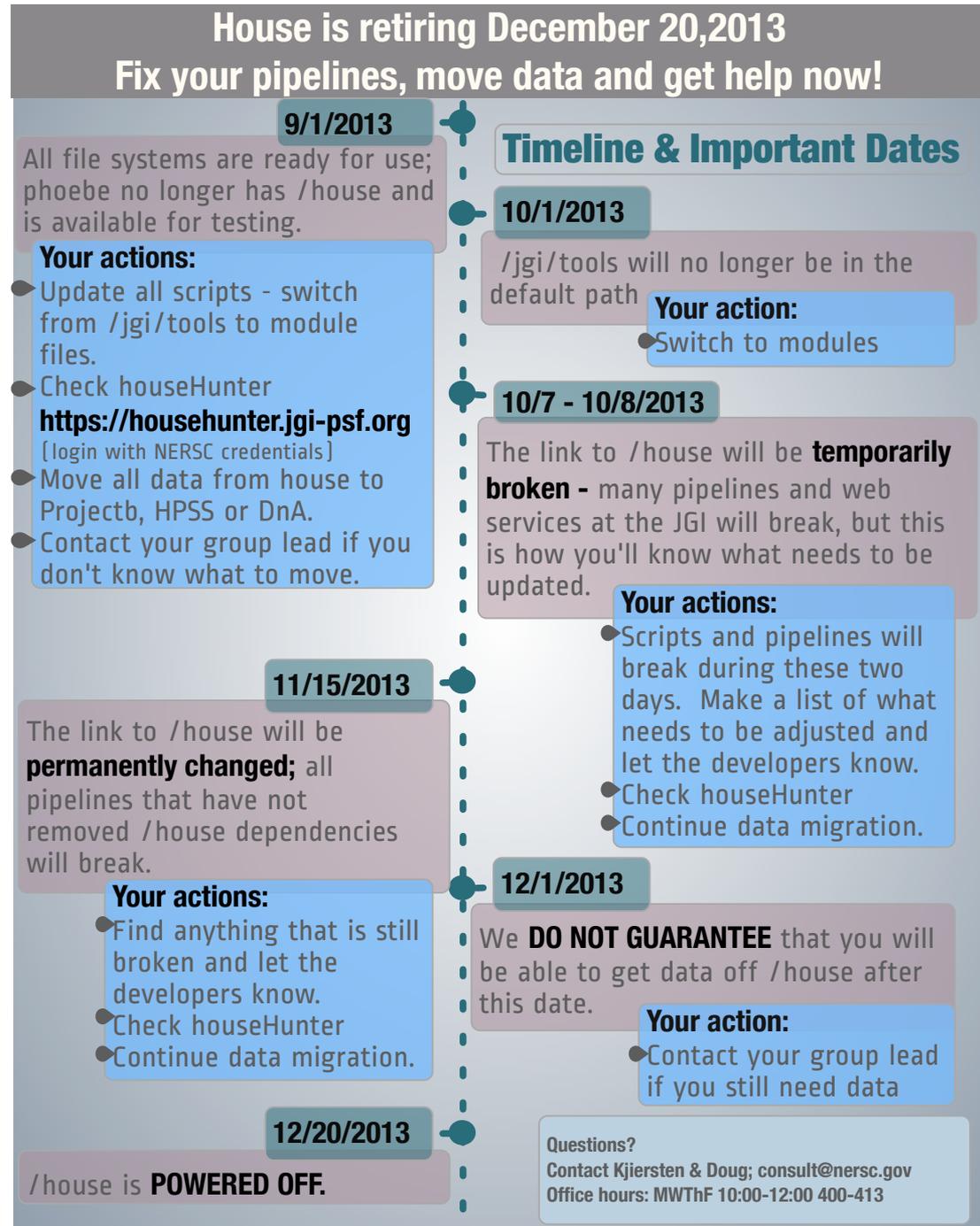


House Retirement Timeline

If we agree on the timeline, this is going to be posted in the restrooms as "house training"



Data Management at the JGI – Lessons learned

Many file systems needed to support JGI's diverse workload

- Webfs for web servers
- Projectb for active projects (tuned for heavier I/O)
- DnA for shared data and automated archiving service
- SeqFS file system for sequence data

File system needs to be tuned for optimal block size and large amount of metadata (the JGI has lots of small data files)

Large data files need to be accessible across multiple disk controllers (this seems to be a big problem with the Thumpers – huge number of partitions and controllers, but most sit idle)

- This is something GPFS does well

10 years worth of data on a variety of disk arrays, NetApps and /house

- Data “archived in place” on spinning disk, symlinks and organizational structure make it challenging to find data again

Mirroring local disk leads to a significant loss of space, meaning less data can be transferred to local disk for processing.

Data Management at the JGI - Lessons learned

Many file systems needed to support JGI's diverse workload

- Webfs for web servers
- Projectb for active projects (tuned for heavier I/O)
- DnA for shared data and automated archiving service
- SeqFS file system for sequence data

File system needs to be tuned for optimal block size and large amount of metadata (the JGI has lots of small data files)

Large data files need to be accessible across multiple disk controllers (this seems to be a big problem with the Thumpers - huge number of partitions and controllers, but most sit idle)

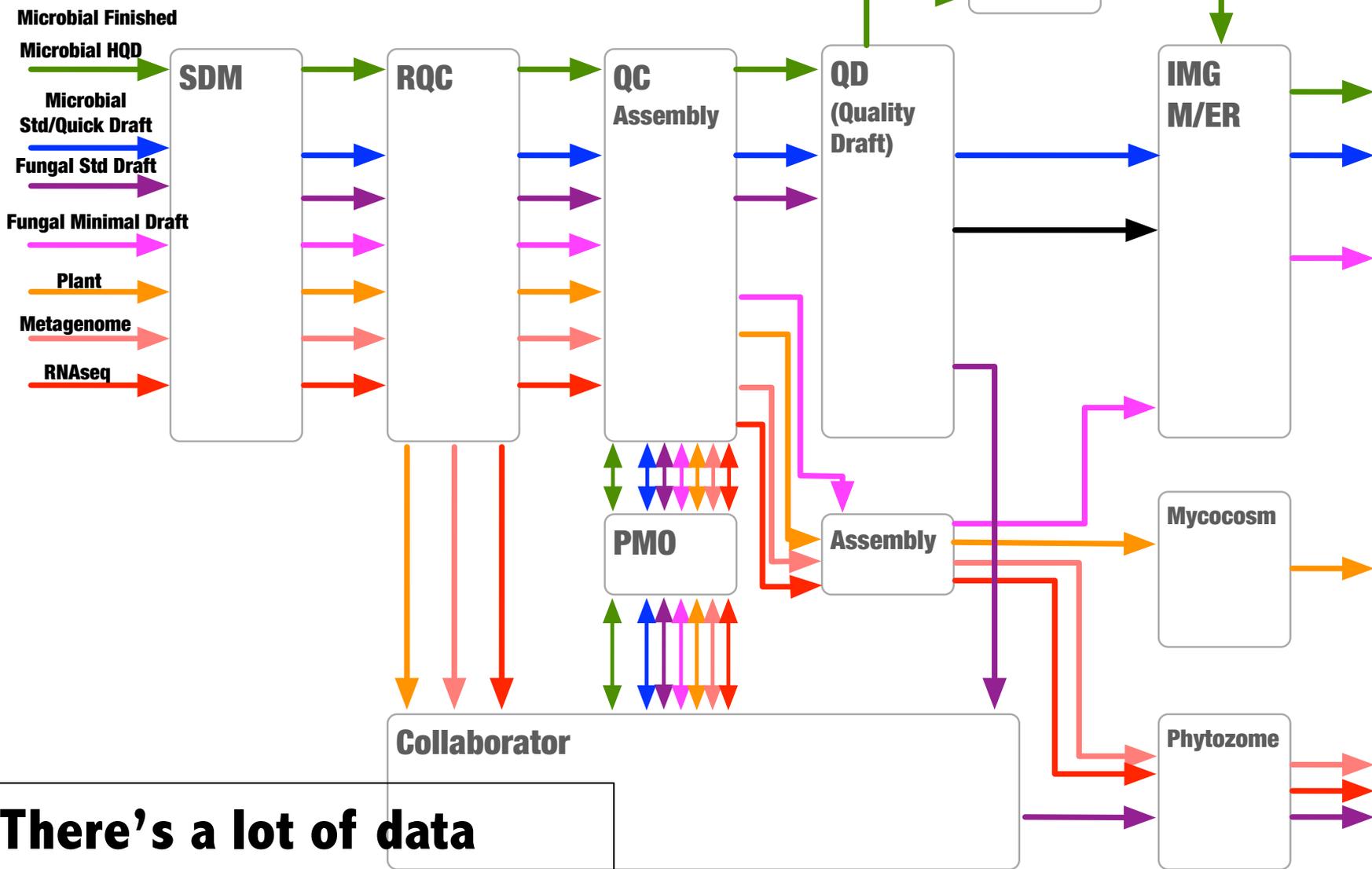
- This is something GPFS does well

10 years worth of data on a variety of disk arrays, NetApps and /house

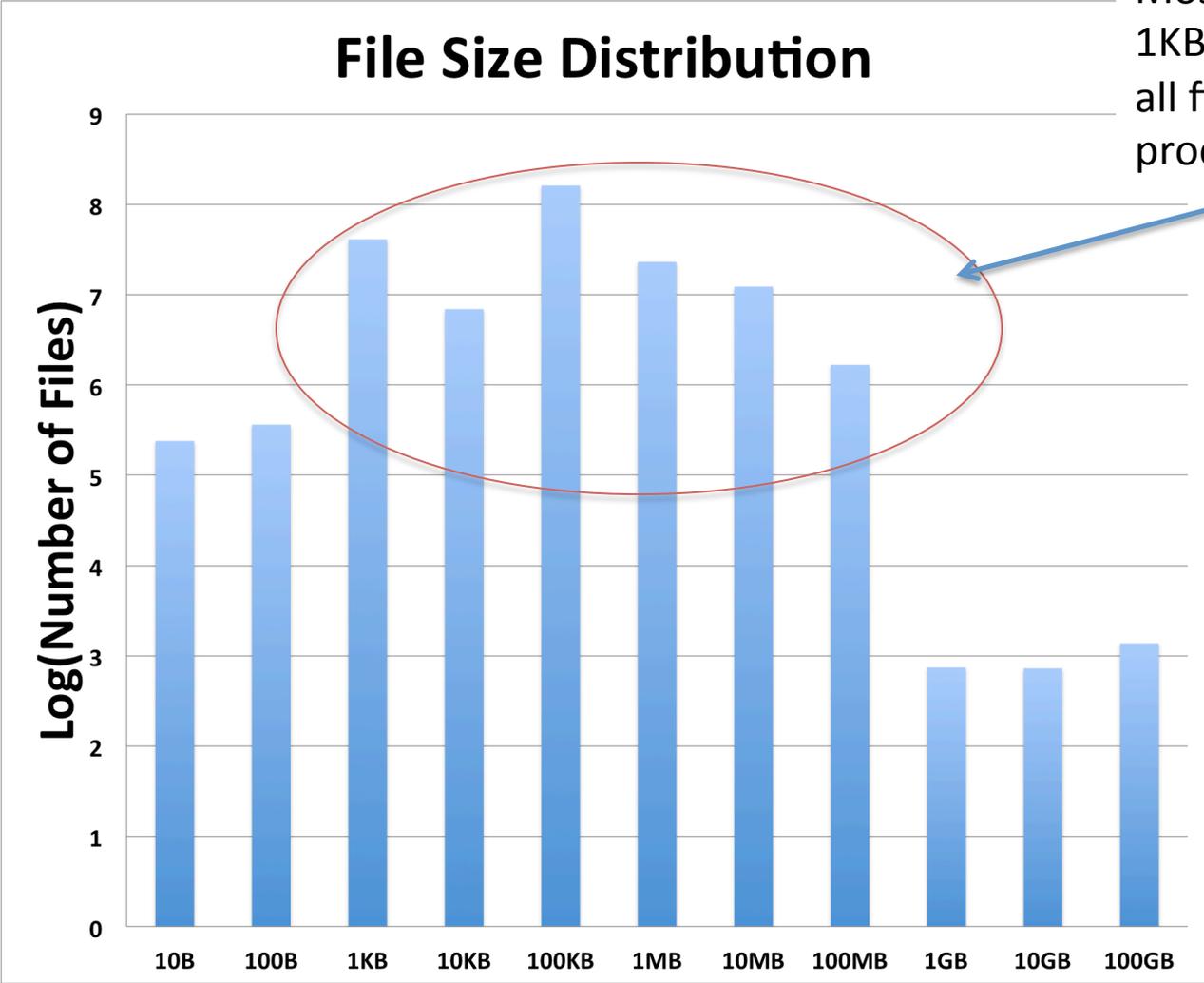
- Data "archived in place" on spinning disk, symlinks and organizational structure make it challenging to find data again

Mirroring local disk leads to a significant loss of space, meaning less data can be transferred to local disk for processing.

Path of Sample Processing



Sequencer Data File Size Distribution

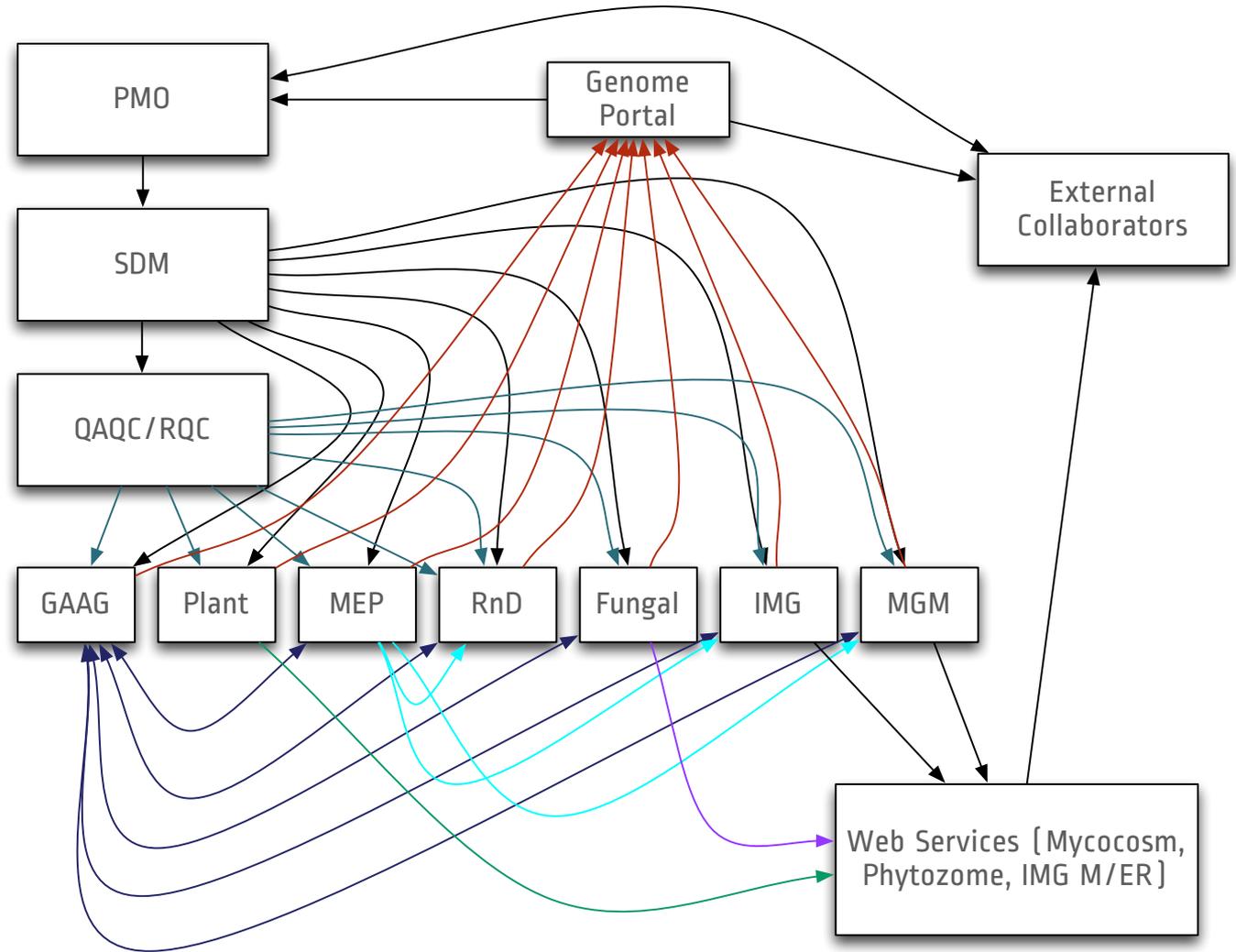


Most files are in the 1KB-10MB size and all files need to be processed by QAQC

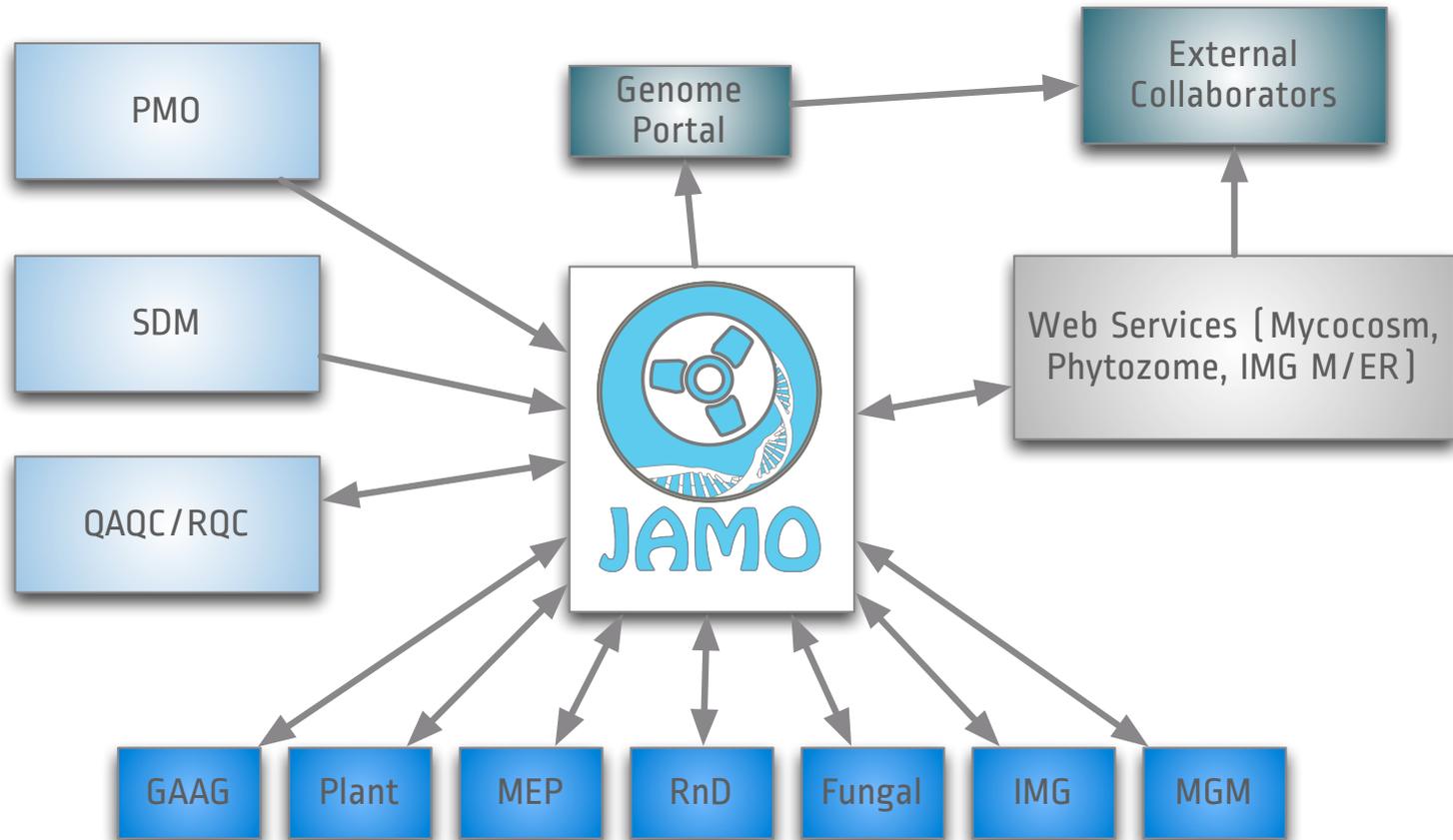
How is the data shared?

Each group manages data access in a different way:

- Google documents
- databases
- scripts to query specific locations on disk
- Symlinks



Proposal: Improve data sharing at the JGI through a central data management repository



Groups will find data by querying a centralized service either through a Python API or a RESTful interface.

Benefit of this system: Concrete Example

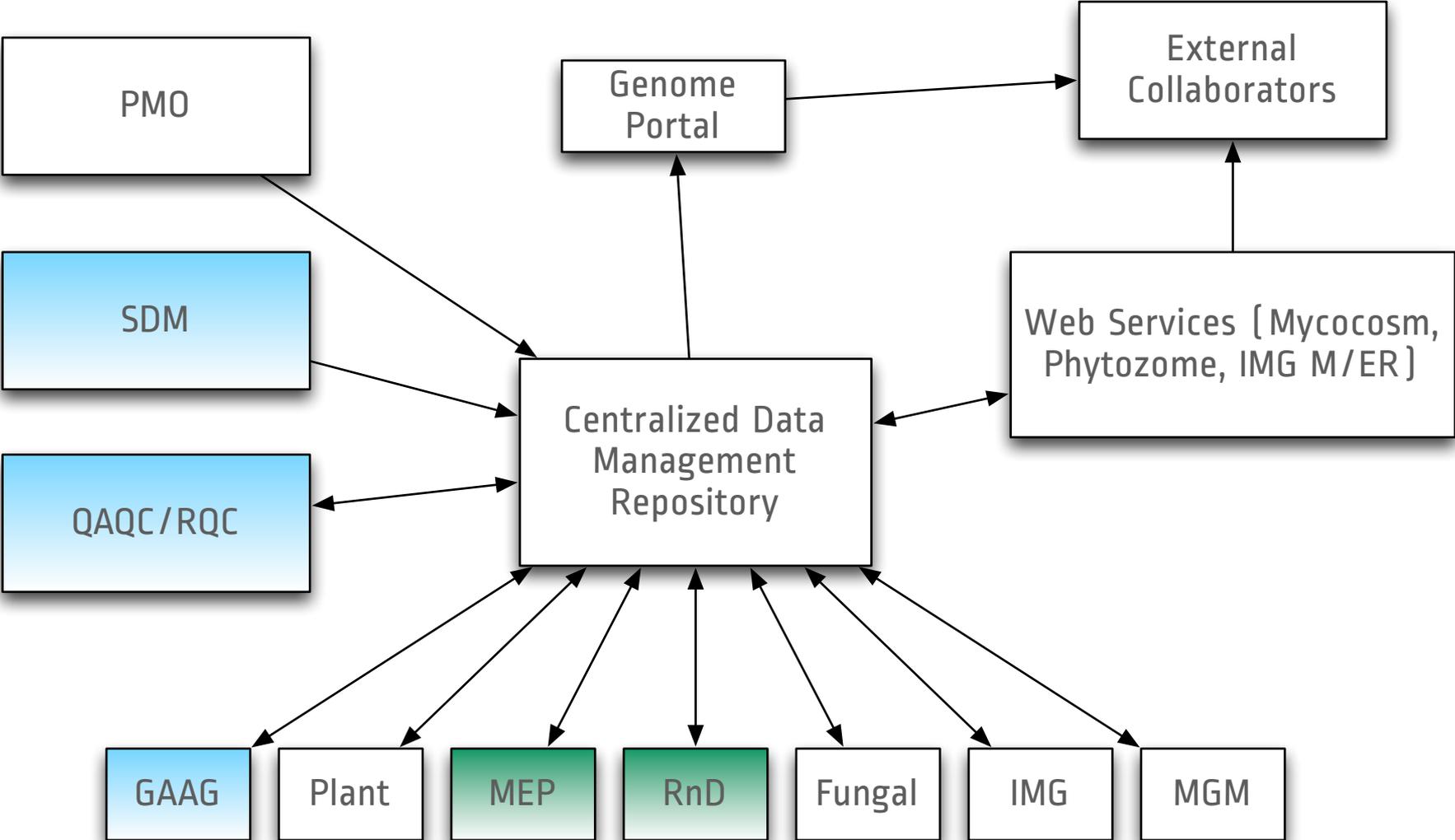
Collaborators come to the JGI years after their analysis has been completed and need to find their data.

Status today: This data can not easily be found

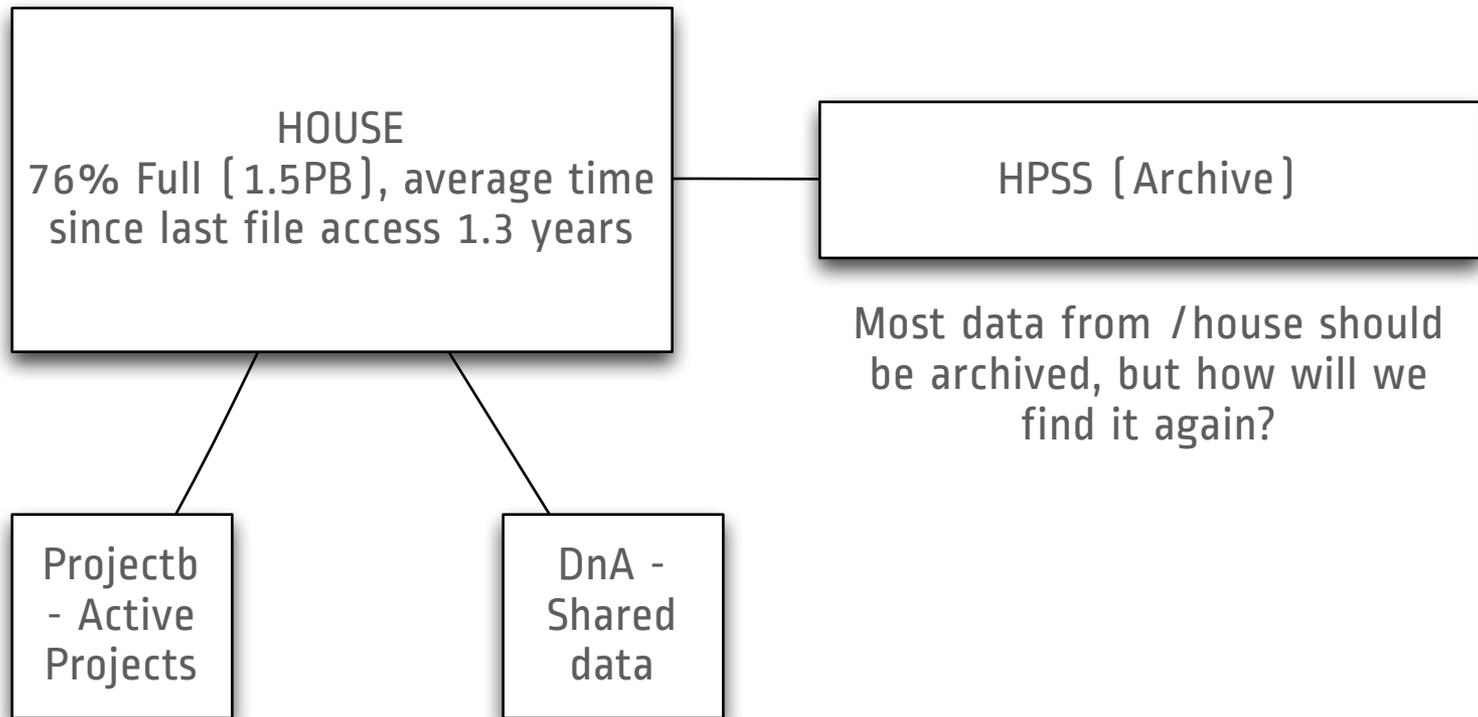
Process: Collaborators must find a person they worked with who remembers where the data was on disk - UNSUSTAINABLE

Future: Data can be found through a UI or with a command-line query making it straight-forward to find the data and retrieve it from tape (if the file still exists)

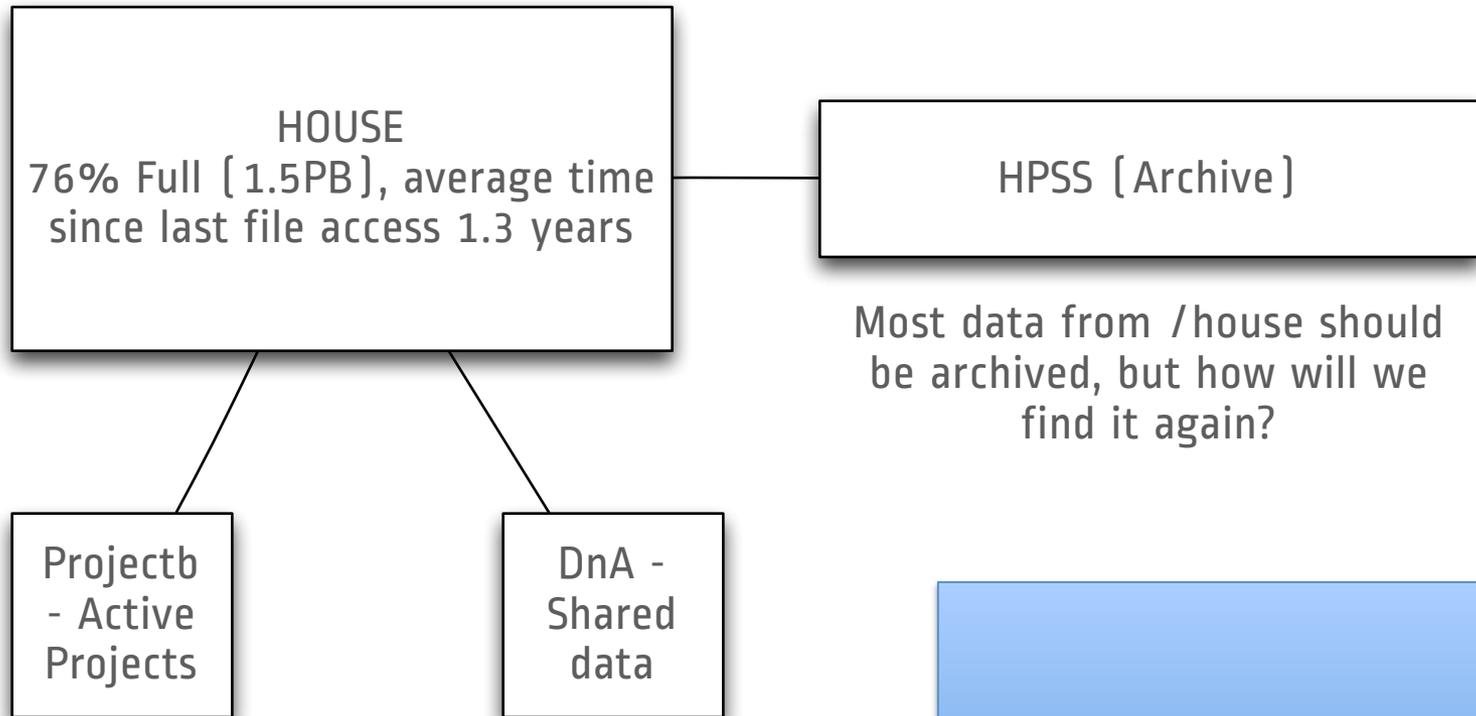
Progress: Steps that have been taken, some groups already have data in this system



Proposal: JGI policy that all groups use this system to back up legacy data from /house



Proposal: JGI policy that all groups use this system to back up legacy data from /house



Most data from /house should be archived, but how will we find it again?

Use the metadata system!

File System Mount Points

Read/Write

Read Only

Unavailable

File System	Description	Compute Nodes	Data Transfer Nodes	GPwebs	GPints	GP logins
Projectb	Data generated by compute jobs. Optimized for heavy I/O.	RW mount point: /global/projectb	RW mount point: /global/projectb	RW mount point: /global/projectb	RW mount point: /global/projectb	RW mount point: /global/projectb
WebFS	File system to support services running on the GPwebs	unavailable	unavailable	RW mount point: /webfs	unavailable	unavailable
DNA	File system for shared data, completed results and data that is ready to be archived. Optimized for read performance. Mounted by NFS/GPFS on the compute nodes.	Read Only mount point: \$DNAFS=/global/dna \$DNA_NFS=/global/nfsdna/shared/	RW mount point: \$DNAFS=/global/dna	Read Only mount point: \$DNAFS=/global/dna \$DNA_NFS=/global/nfsdna/shared/	Read Only mount point: \$DNAFS=/global/dna \$DNA_NFS=/global/nfsdna/shared/	Read Only mount point: \$DNAFS=/global/dna \$DNA_NFS=/global/nfsdna/shared/

Data Transfer Examples

Getting data from Genepool login nodes to /webfs - **use rsync!**

```
rsync -v -e ssh <files> gpweb<num>.nersc.gov:/webfs/<dir>
```

```
kmfagnan@genepool12 ~ $  
rsync -v -e ssh test_transfer gpweb08.nersc.gov:/webfs/scratch/portal  
  
test_transfer  
  
sent 281218 bytes  received 42 bytes  187506.67 bytes/sec  
total size is 281092  speedup is 1.00  
kmfagnan@genepool12 ~ $
```

Data Transfer Examples

Getting data from Genepool cluster to /webfs - **use rsync!**

```
rsync -v -e ssh <files> gpweb<num>.nersc.gov:/webfs/<dir>
```

```
kmfagnan@genepool12 ~ $ cat test_trans.sh
```

```
#!/bin/bash -l
```

```
#$ -N data_transfer
```

```
#$ -cwd
```

```
rsync -v -e ssh files gpweb08.nersc.gov:/webfs/<dir>
```

```
kmfagnan@genepool12 ~ $ qsub test_trans.sh
```

Data Transfer Examples

Getting data from Genepool cluster to /global/dna – use the **mv command!**

```
kmfagnan@genepool12 ~ $ cat projb_to_dna.sh
#!/bin/bash -l
#$ -N projb2dna
#$ -l xfer.c

mv files $DNAFS/projectdirs/<dir>

kmfagnan@genepool12 ~ $ qsub projb_to_dna.sh
```

Data Transfer Examples

Getting data from Genepool cluster to /global/dna – use the **mv command!**

```
kmfagnan@genepool12 ~ $ cat projb_to_dna.sh
#!/bin/bash -l
#$ -N projb2dna
#$ -q xfer.q

mv files $DNAFS/projectdirs/<dir>

kmfagnan@genepool12 ~ $ qsub projb_to_dna.sh
```

xfer queue performance

We examined the performance of the nodes in Genepool Phase 2 that have been set aside for data transfers:

1GB Transfers

7GB Transfers

300GB Transfers

