

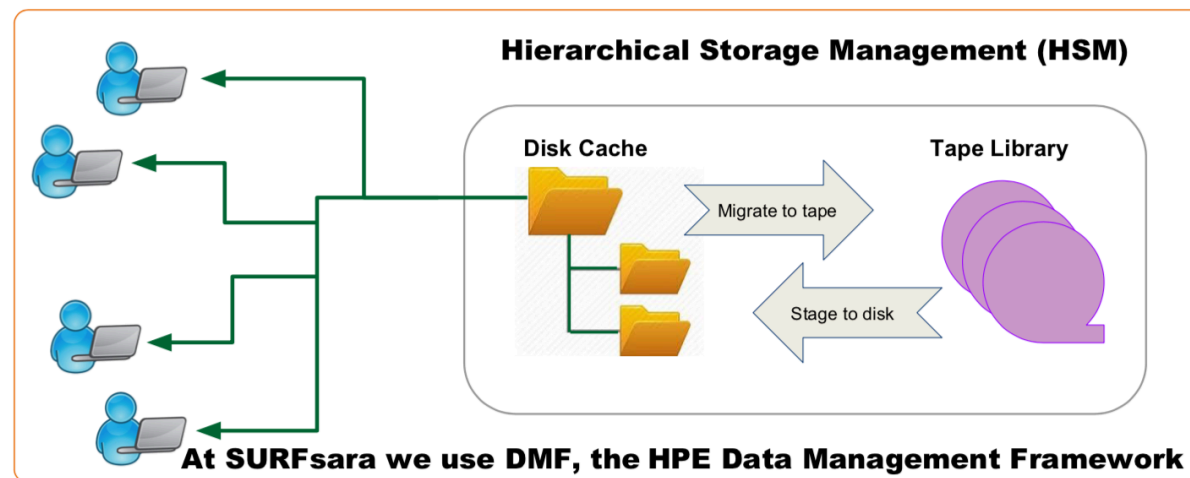
# iARCH

Asynchronous file handling with iRODS tape resources

<https://github.com/sara-nl/surfsara-dmf-irods-client>

# SURFsara and iRODS services

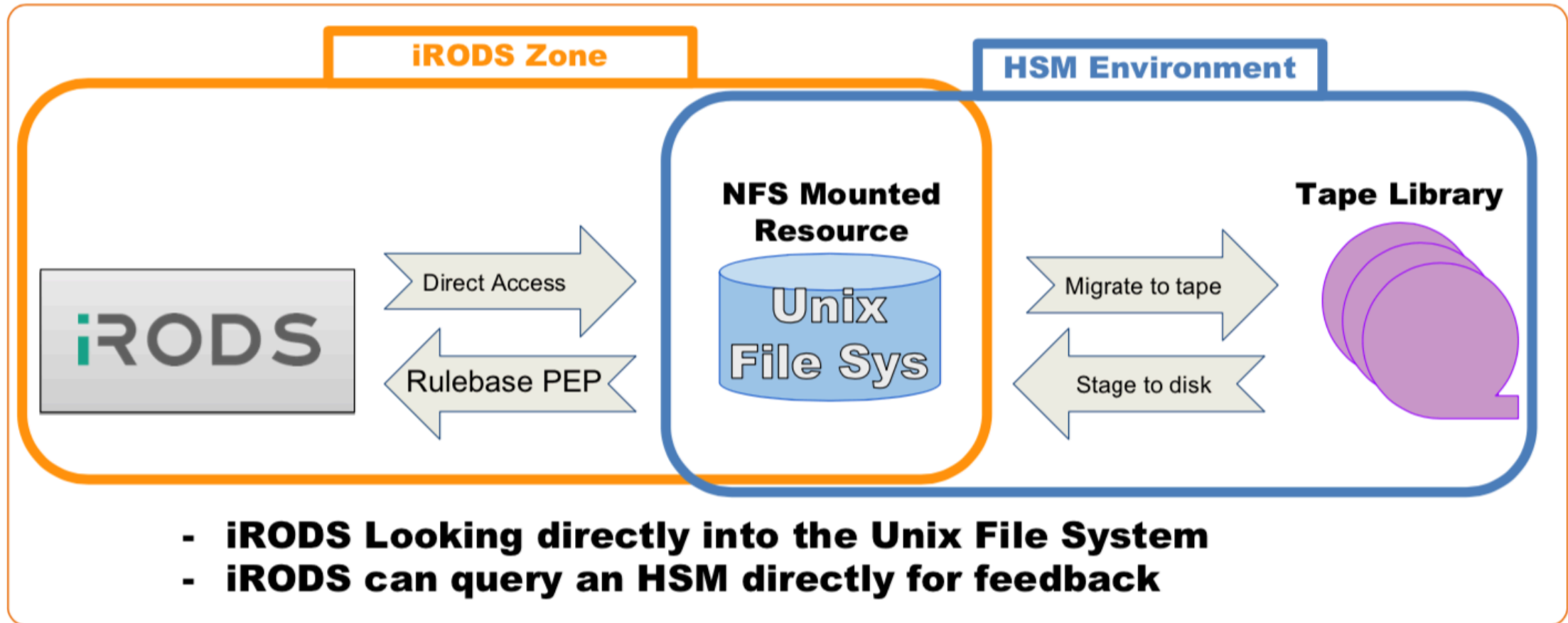
- SURFsara is the Dutch High Performance Computing centre supporting Dutch researchers via services, training and consultancy
- SURFsara has a storage scale out service for iRODS instances hosted at universities
- Resources from SURFsara (disk storage, object store, data archive) can be leveraged for scalable data infrastructures on university premise
- SURFsara Data Archive uses tape technologies where data is stored online on a disk cache, or offline on tape managed by DMF
- How did we connect iRODS to the Data Archive? Topic of our talk last year..



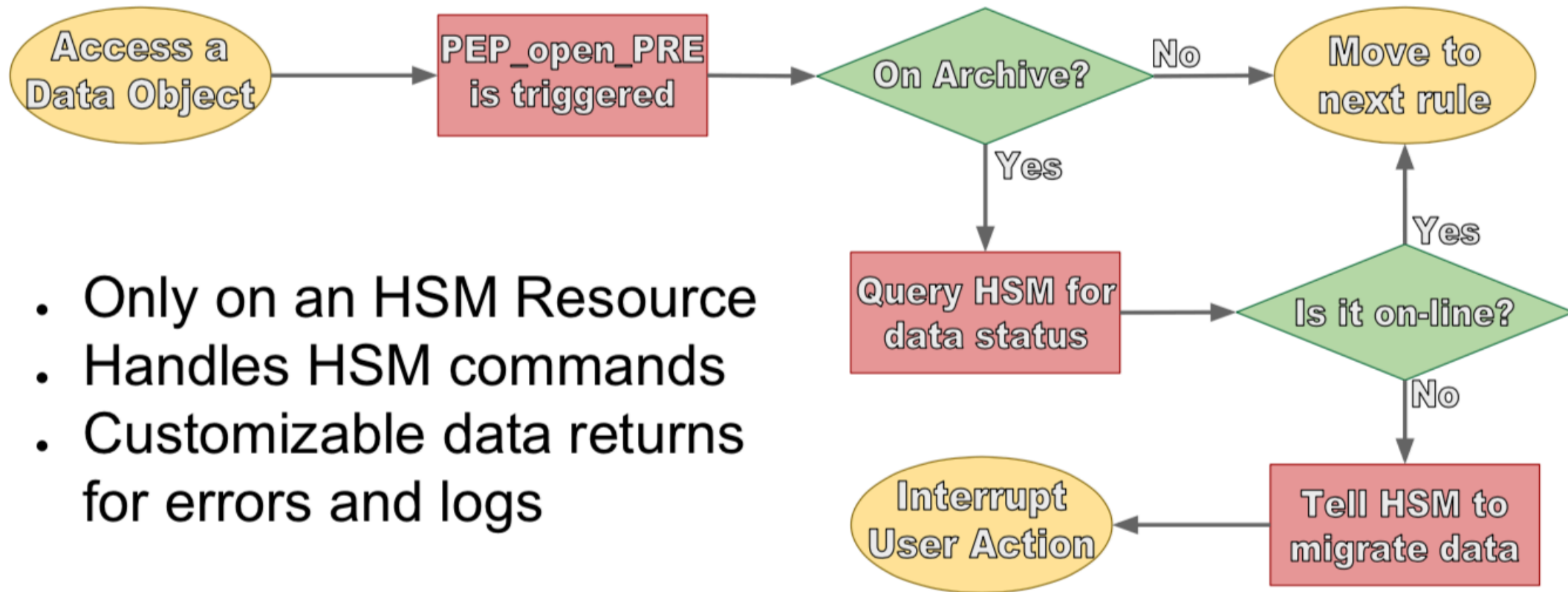
# iRODS scale out to tape storage

- iRODS does not know about 'offline' data, although the compound resource could be used..
- After testing, we concluded that the compound resource was inefficient for our use case
- Instead we use a NFS server as unix filesystem connected to the tape disk cache and we created a set of rules to make the Data Archive transparent
- Hand over of data is more transparent, inode still visible to iRODS also if bit stream is on tape.

# Overview of components for scale out to Data Archive



# Overview of rules that manages offline vs online data



- Only on an HSM Resource
- Handles HSM commands
- Customizable data returns for errors and logs

# Ongoing issue with handling asynchronous data

- User feedback not so friendly:

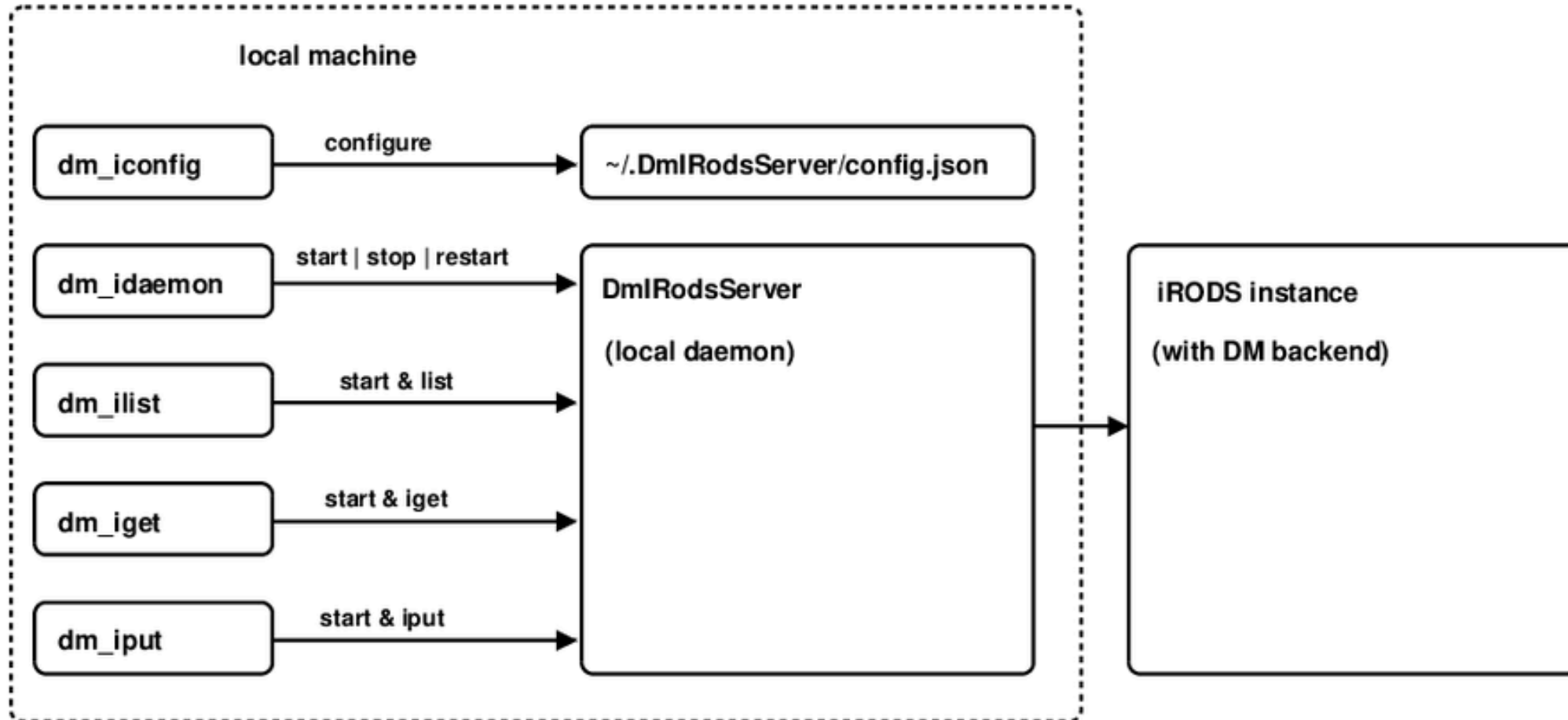
```
matthews$ iget test  
ERROR: getUtil: get error for ./test status = -1101000 RULE_FAILED_ERR  
Level 0: DEBUG: /surf/home/matthews/test is still on tape, but queued to be staged. Current data staged: 42%.
```

- Especially when researchers will be directly accessing data objects on the data archive resource, we need a more convenient way of communicating
- This issue will become more urgent thanks to better price point of SURF tape storage.
- A dedicated nimble data handling tool for researchers would be beneficial, e.g. for use on HPC systems in notebooks or data processing pipelines.

# iARCH: utility to handle offline/online data on iRODS

- commandline application which makes it easier for the user to download, upload and retrieve information about the state of data
- uses the iRODS python client
- The application is split into a set of CLI tools and a daemon-like application that handles requests and file transfers in the background.
- The daemon is automatically spawned as a non-root process upon the first request and stopped when idle for a specific time.

# iARCH: overview of commands





# iARCH: overview of commands

**dm\_iconfig:** initializes the connection similar to iinit.  
Stores the necessary information in the home folder

```
~/DmIRodsServer/config.json  
~/DmIRodsServer/completion.sh  
~/DmIRodsServer/.irodsA
```

**dm\_ilst:** lists all objects that were part of current/past processes started by the daemon, also objects whose status has not been changed by the daemon

```
> dm_ilst
```

DMF	TIME	STATUS	MOD	FILE	LOCAL_FILE
DUL	2018-10-10 15:52:38	DONE	100% GET	/surf/home/rods/1M_0003.dat	1M_0003.dat
DUL	2018-10-10 16:31:52	DONE	100% PUT	/surf/home/rods/1M_0001.dat	1M_0001.dat
DUL				/surf/home/rods/1G_0001.dat	
DUL				/surf/home/rods/	

# iARCH: overview of commands

**dm\_iput:** uploads data object to iRODS instance, not directly to the archive resource. Also no control over staging of data

```
> dm_iput test50.mb
STATUS          FILE
scheduled       ../test50mb <> /{zone}/home/{user}/test50mb
```

```
> dm_ilst
DMF TIME          STATUS          MOD FILE          LOCAL_FILE
DUL 2018-10-10 16:31:52 PUTTING    30% PUT /surf/home/rods/test50.mb test50.mb
```

**dm\_iget:** downloads the object. If object is offline, it will first stage it back to the disk cache of the data archive

```
> dm_iget /surf/home/rods/test50.mb
STATUS          FILE
scheduled       /surf/home/rods/test50.mb
```

# iARCH: future plans

- Increase functionality (at the moment only up/download), e.g. querying for data
- Abstract the concept of offline and online data objects further, i.e. handle data irrespective of which resource a data object resides (disk, zero-watt disks, tape etc)
- Implement in our HPC systems (Lisa, Cartesius)
- <https://github.com/sara-nl/surfsara-dmf-irods-client>

# iARCH: credits

- Stefan Wolfsheimer (developer)
- Sharif Islam (tester)
- Matthew Saum (tester)



Arthur Newton



E-mail: [info@surfsara.nl](mailto:info@surfsara.nl)



[www.surf.nl](http://www.surf.nl)



# Driving innovation together

**Driving innovation together**



# iARCH: overview of commands

**dm\_iinfo:** retrieve information on certain objects

```
> dm_iinfo /surf/home/rods/1M_0003.dat
-----
Transfer
-----
retries           : 3
status            : DONE
errmsg            :
time_created      : 2018-10-10 15:52:38
transferred       : 1048576
mode              : GET
-----
Local File
-----
local_file        : ~/1M_0003.dat
checksum          : 4r0H0reFDi9BgY9dHBg0dS92kgV0ChXzj0U+dEBfXe0=
...
-----
Remote Object
-----
remote_file       : /surf/home/rods/1M_0003.dat
remote_size       : 1048576
remote_create_time : 2018-10-02 13:48:07
...
-----
DMF Data
-----
DMF_state         : DUL
DMF_emask         : 160000
...
```

# iARCH: overview of commands

**dm\_idaemon:** control the daemon manually (mostly for debugging)

`dm_idaemon status`

`dm_idaemon start`

`dm_idaemon restart`

`dm_idaemon stop`

`dm_idaemon`