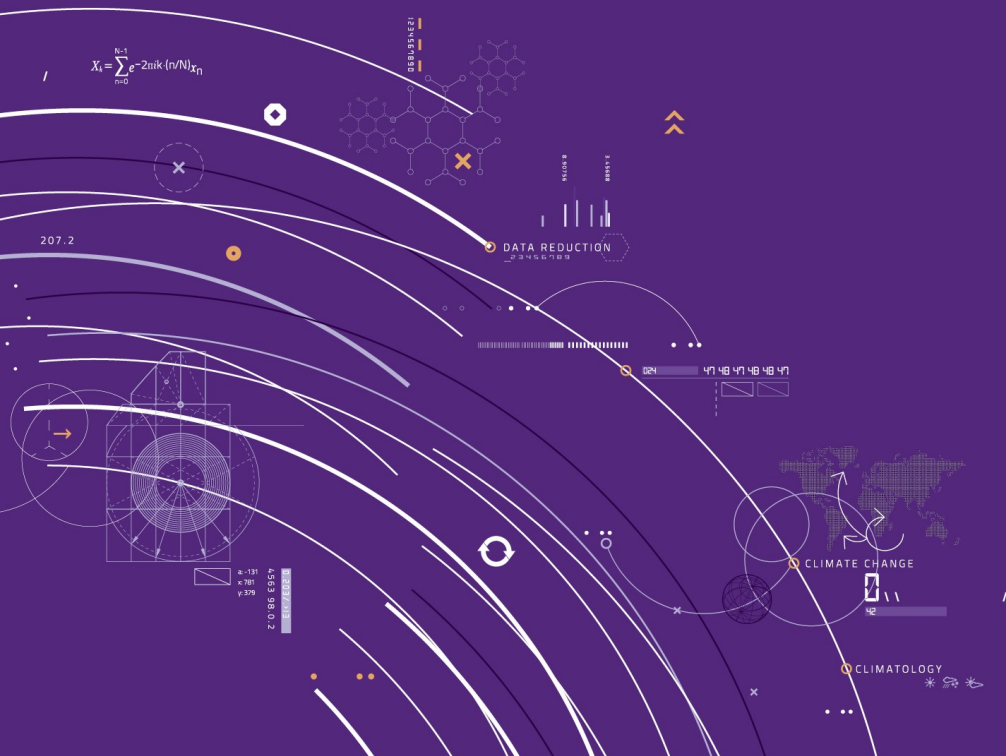


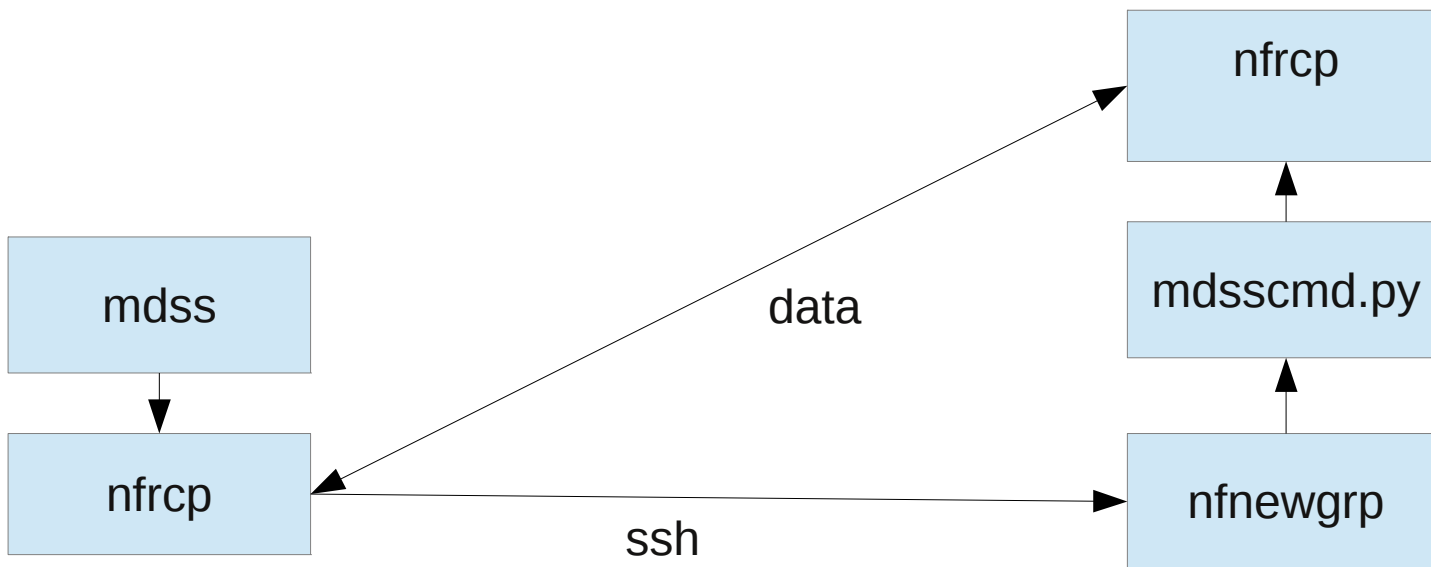
# mdss - NCI's mass-data storage system abstraction layer.



- mdss is a shell-level storage abstraction layer, not necessarily tied to DMF
- Rationale for not mounting storage directly - locality, reliability, control.
- ftp-style command set - get, put, ls, mkdir, rm etc.
- Lots of little things behind the scenes differentiate it from ftp, scp etc.

```
-bash-3.2$ mdss mkdir dh-test
-bash-3.2$ mdss put foo.dat dh-test
-bash-3.2$ mdss ls -l dh-test
total 4
-rw-r--r-- 1 djh900 z00 29 Sep 24 16:47 foo.dat
-bash-3.2$ mdss verify dh-test/foo.dat
dh-test/foo.dat: OK
-bash-3.2$ mdss get dh-test/foo.dat bar.dat
```

- Control flow - file copying



- Control flow - ls, mkdir etc.



- Centralised configuration and control (via mysql db). Non-interactive use polls until a backend host is available.
- Uses ssh authentication, including "mdss setup" to push keys
- Uses nfrpc, in-house fork of scp. Data sent/received by multi-threaded reader/writer across separate unencrypted socket. checksums over ssh. Very similar to bbcp, but pre-dates bbcp (?) and has xattr-based checksum mechanism.
- Runs several nfrcps simultaneously
- Checksums stored in xattrs. Verifies both network and remote filesystem integrity. There's an "mdss verify" command. Original checksum propagates along subsequent copies (providing xattrs enabled), so corruption doesn't get sanitised away. Uses ns timestamp and size info to validate checksum state.
- setuid wrapper "nfnewgrp" to change group for different projects
- Control flow in detail - mdss, mdss.py, [nfrpc], ssh, mdsh, mdsscmd, nfnewgrp, mdsscmd.py, [nfrpc]
- Optional restricted shell "mdsh" for users

- Issues dmget just after starting "mdss get" so first opened files get recalled before the rest. Checks for excessive recursion first.
- Optional dmpout -w after "mdss put".
- Lots of scope for optimisation here – e.g. get files in VSN order.