

Update on Implementation and Usage of HTCondor at DESY:

In this talk we provide new details of the DESY configurations for HTCondor. We focus on features needed for user registry integration, node maintenance operations and fair share / quota handling. We are working on Docker, Jupyter and GPU integration into our smooth and transparent operating model setup.

DESY/IT-Systems:

Thomas Finnern
Christoph Beyer
Martin Flemming
Yves Kemp



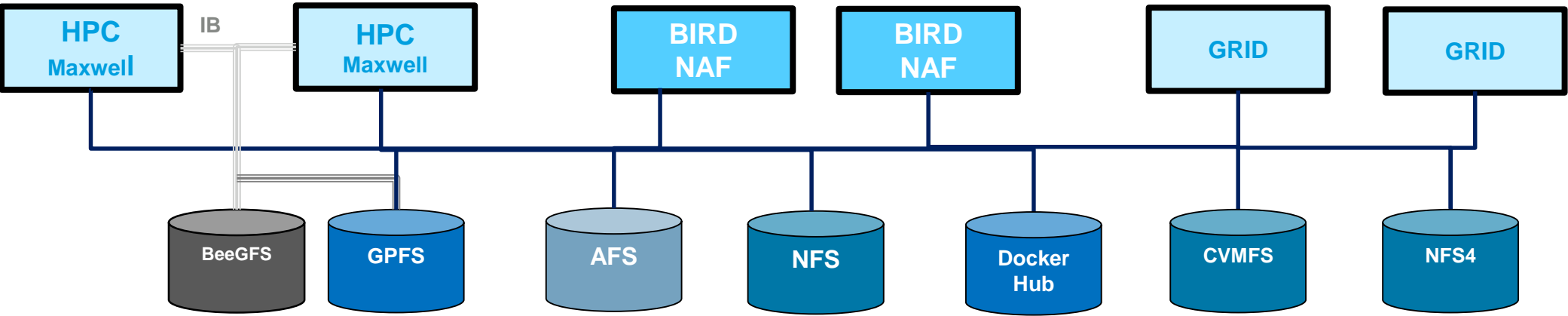
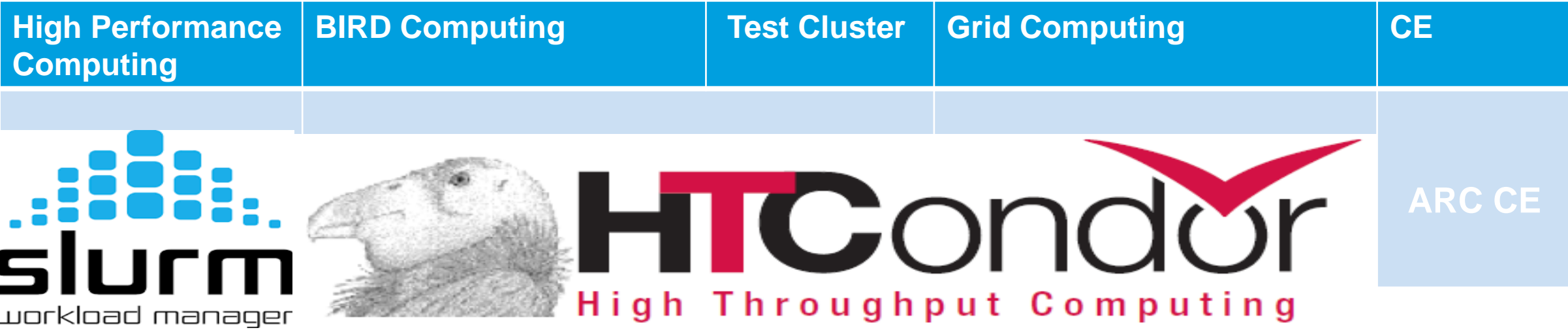
EC-JRC Ispra



Overview DESY Batch Infrastructure !



Basic Blocks



Outline of Talk

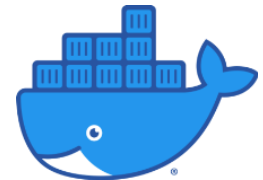
HTCondor DESY Environment



- Main Focus on BIRD Facility
 - BIRD/NAF Overview
- The Base: Job Classes
 - Implementing Dynamic Fair Share
 - Node Automation and Control
 - New Resource Optimization
- User Registry Integration
 - Creating User.Map and Share Groups
 - Adding Blacklists and Maintenance
 - Kerberos



- New Features
 - GPU Support
 - Jupyter Notebook Integration
 - Docker Service
 - **Function as a Service**
- Potential Pitfalls
- Outlook and Conclusions

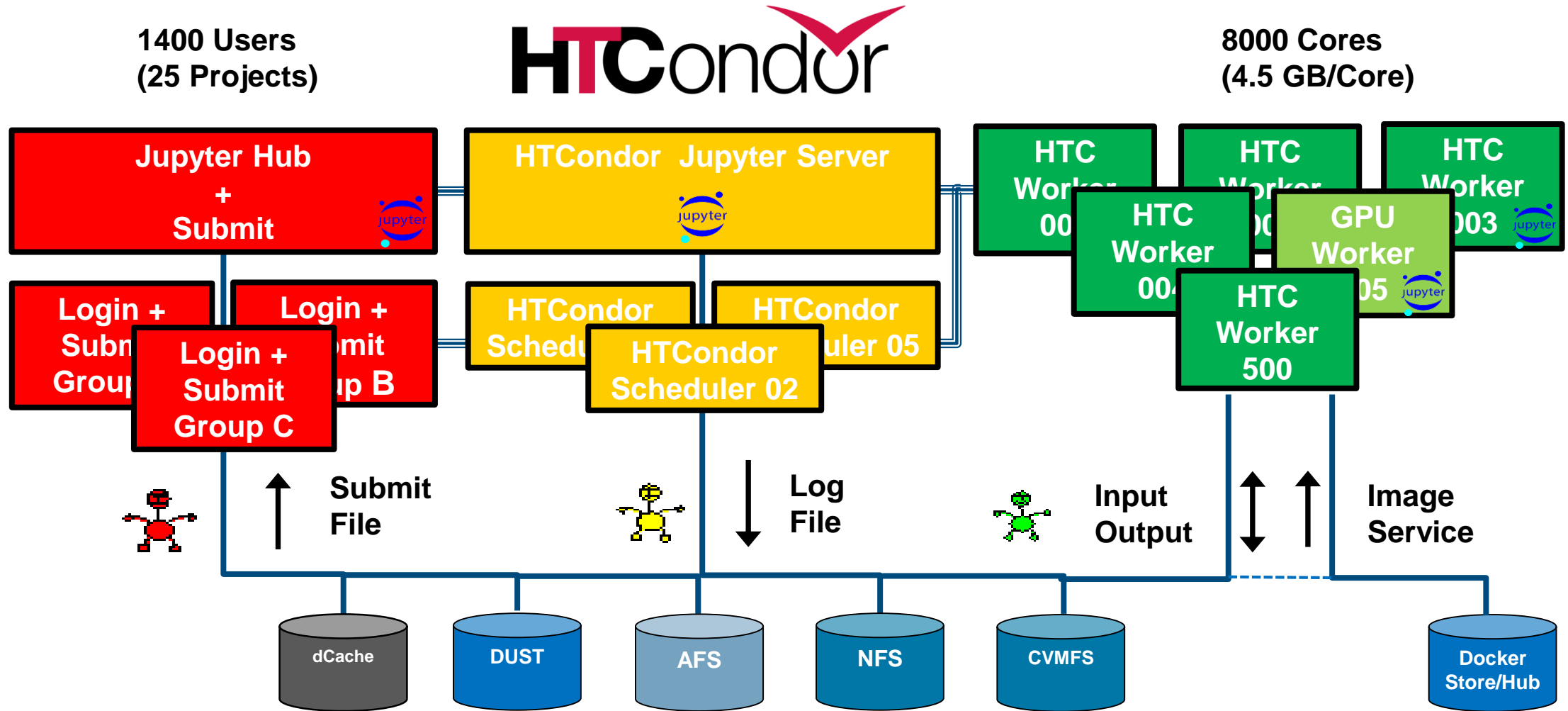


BIRD, NAF, HTC and HPC:
Batch Infrastructure Resource at DESY
National Analysis Facility
High Throughput Computing
High Performance Computing

BIRD/NAF Simple Block View



Block Numbers

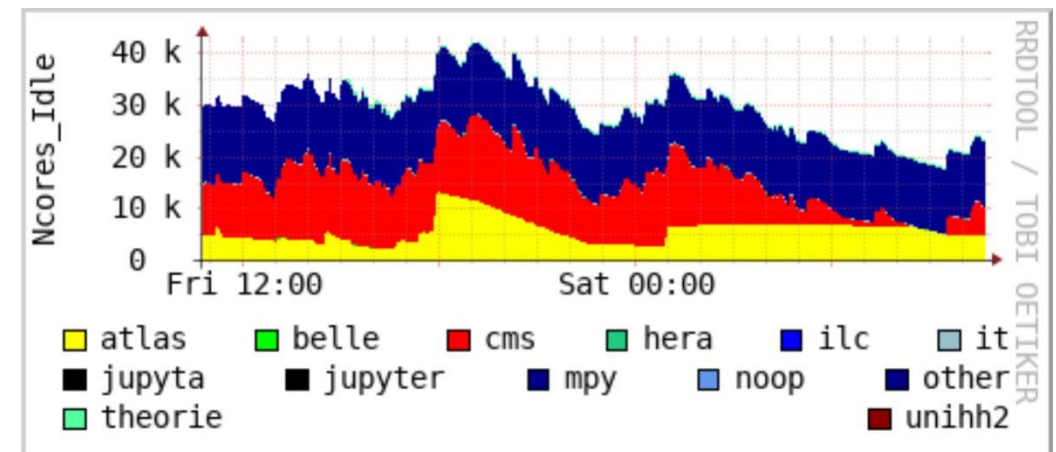
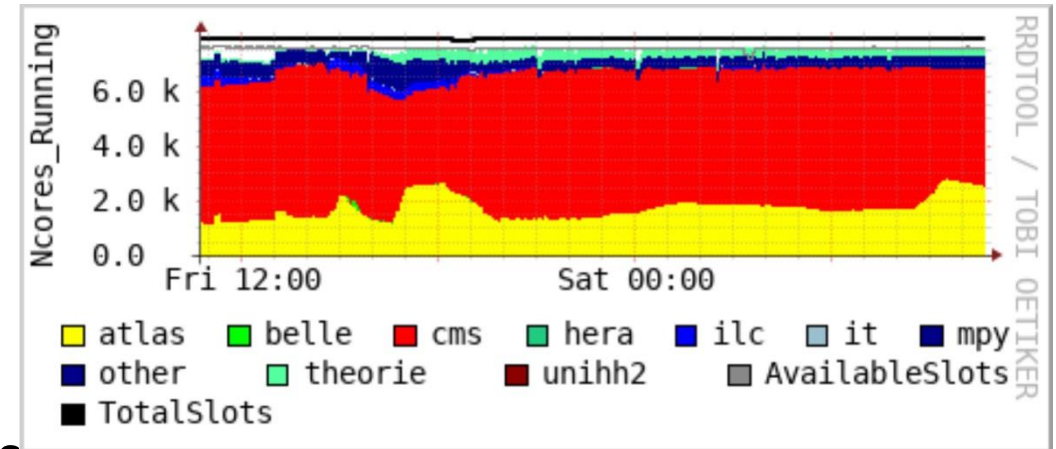


Organizing the Work Load Environment



„The start time of a job should be less or in the same order as the requested runtime“

- Policies set on Scheduler
 - Maximum, requested and default Job Runtimes
 - Resulting Drain Times
- Values
 - Default Runtime 3 hours
 - Runtimes requestable from 1 Week to a few Minutes
 - Vacate Time is 5 minutes as Part of Job Runtime
- Implementation
 - HTC Feature: Periodic Remove
 - Runtime Calculation within HTC Interface
 - Node Runtime essential in Process Management
 - Quota/Fairshare Overcommittment



The Base: Job Classes



Defines metrics for Quota/Fairshare and Node Management

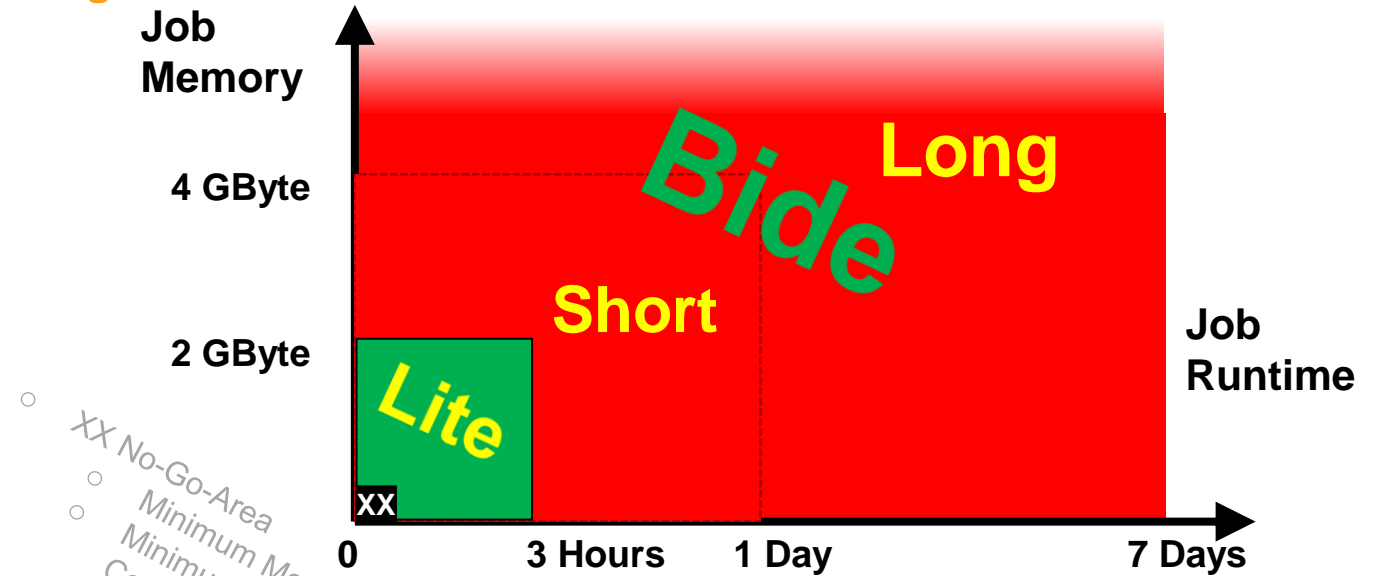
- Job Classes
 - Lite, Short and Long defined
 - Lite and Bide for Quotas and Shares
- Job Types
 - Single, Array, Multicore, Multiarray
 - For Informational Purpose
- Shared Quota
 - 300 % Oversubscription for Lite Jobs
 - 33 % Oversubscription for all Jobs (~ Entropy)

- System ClassAds

- SysProject
- DefProject
- Accounting Group
- Quota/Fairshare

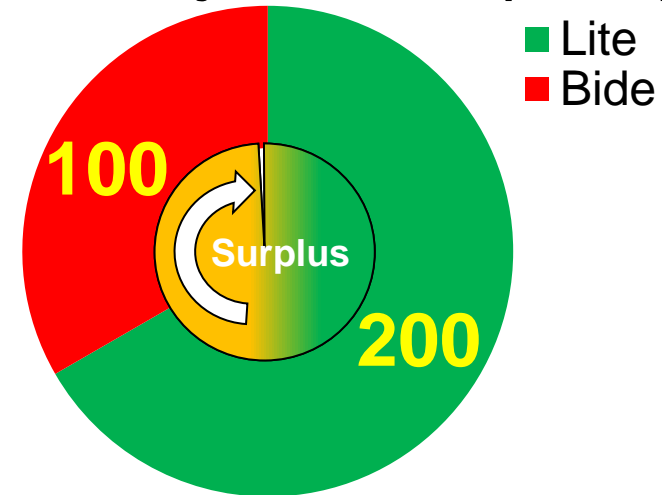
- User ClassAds

- RequestRuntime
- RequestMemory
- RequestDisk
- MyProject



○ XX No-Go-Area
 ○ Minimum Memory Setting
 ○ Minimum Runtime Out of Control

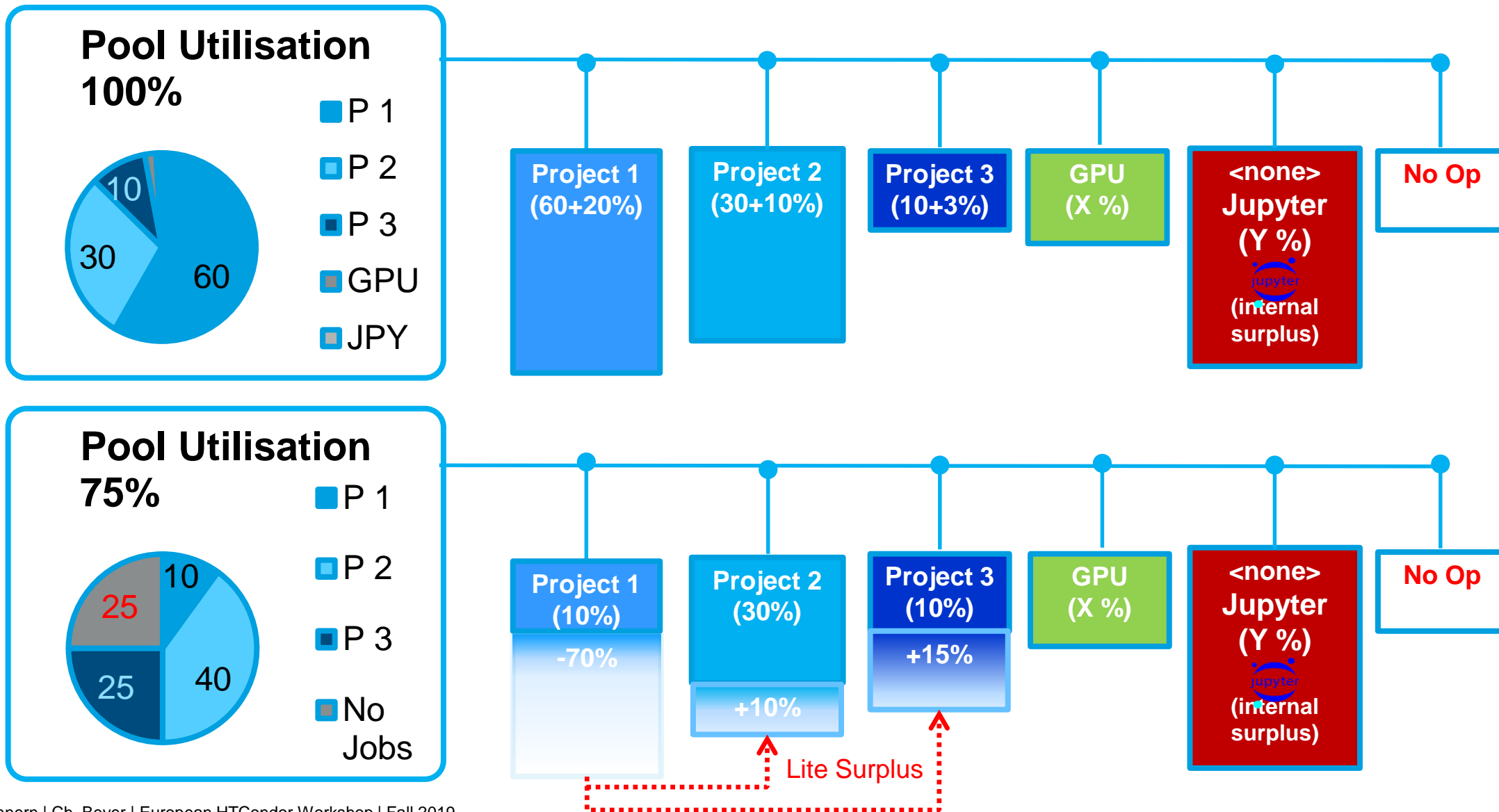
Project Quota (300%)



Implementing Dynamic Fair Share (133 %)



Two different utilisations



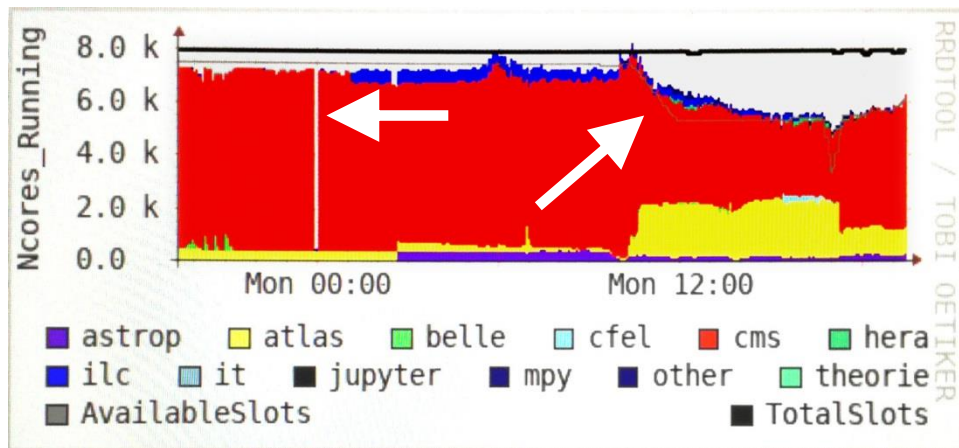
Node Automation and Control

Node Management simple and without job interference



- Automated Operation of Nodes
 - For Problems (e.g. node failures)
 - For Service (e.g. cluster kernel update)
 - Manually/CLI or by scripting
- Disable, drain, reboot and reset Nodes
 - No preemption or job killing
 - No specific operator knowledge needed

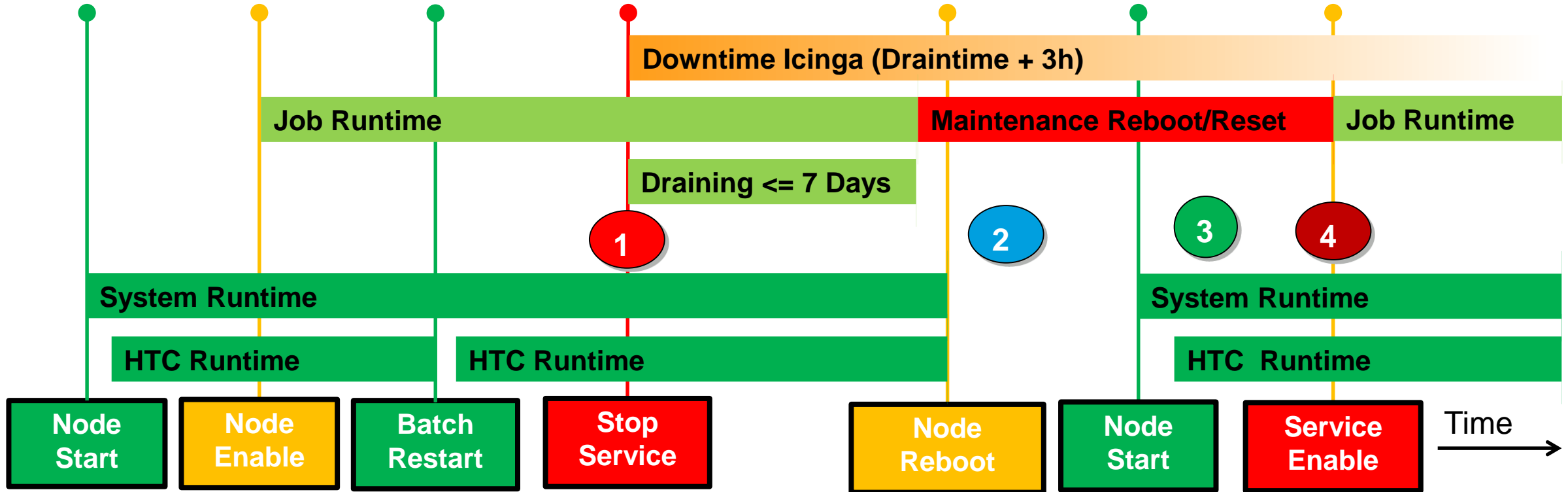
- Authentication
 - Based on **A**uthenticated **R**emote **C**ommand (**arc**)
 - User based (Operators, Admins) Batchnode.sh
 - Server based (Scripts, Cluster Reboot, Kernel Upgrades, ...)
 - Node based (local Monitoring, ...)
- Transparent
 - All states in one view
 - Hourly status update
 - Sets/resets exact icinga downtimes
 - Works for all pools
 - GRID, BIRD, TEST, ...



Node and Job Timing



Version A: Automatic Node Draining (Disable/)/Drain/Reboot/Reset



1	2	3	4
Batchnode.sh	Node.cron (Sys reboot)	Node.cron (node status)	Node.cron (node enable)
StartJobs = False Condor_drain -graceful	Cron.hourly	Cron.hourly	StartJobs = True Condor_drain -cancel

Optimised Node Automation and Control



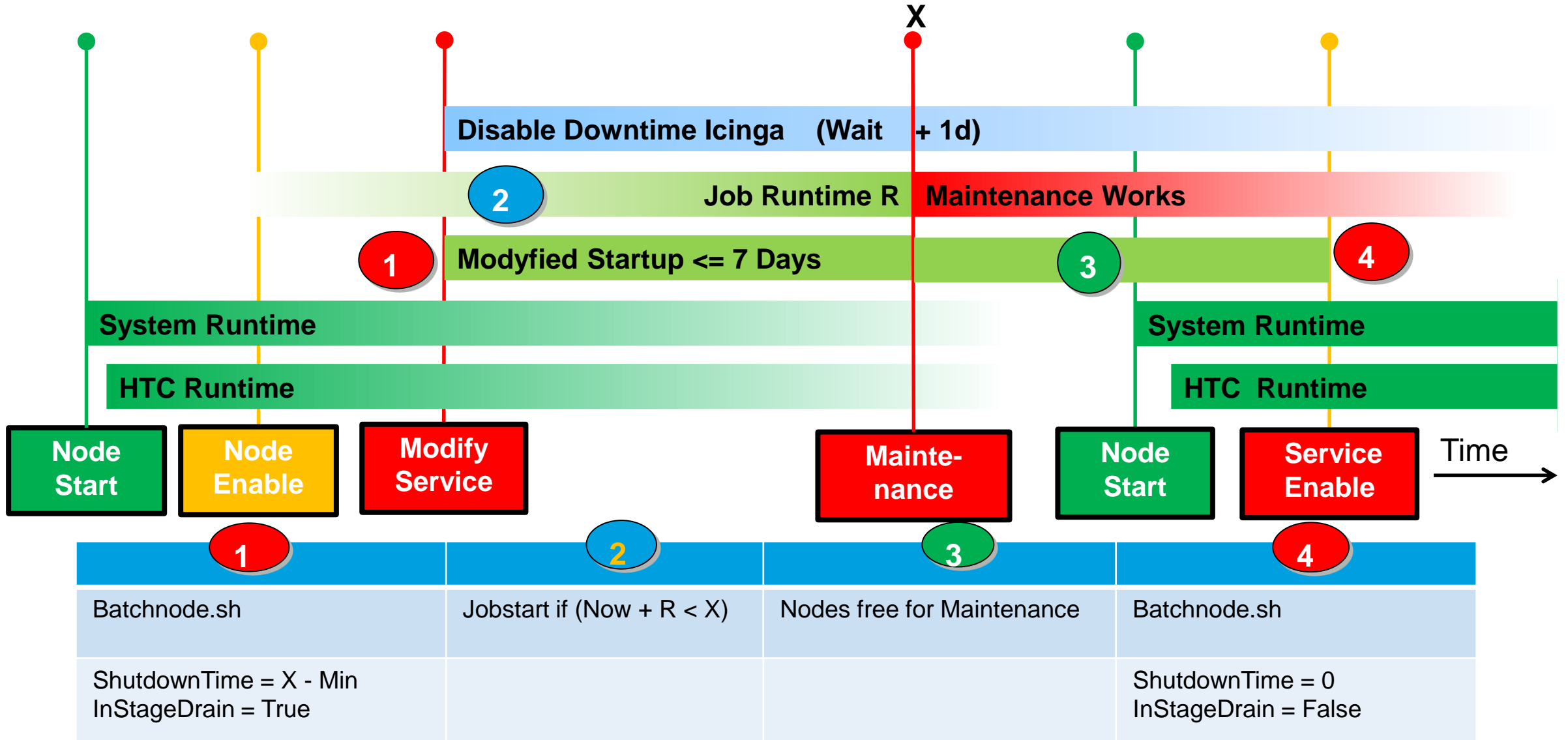
Version B: Draining without draining for planned maintenance

- Draining
 - Side Effect: Empty Resources
 - Not for Project specific Maintenance
- Solution
 - Maintenance and Project controlled Jobstart
 - Simple extension to Version A
- Draining and minimizing badput
- Healthcheck and Draining Project sensitive
- Support-Group still may run test jobs

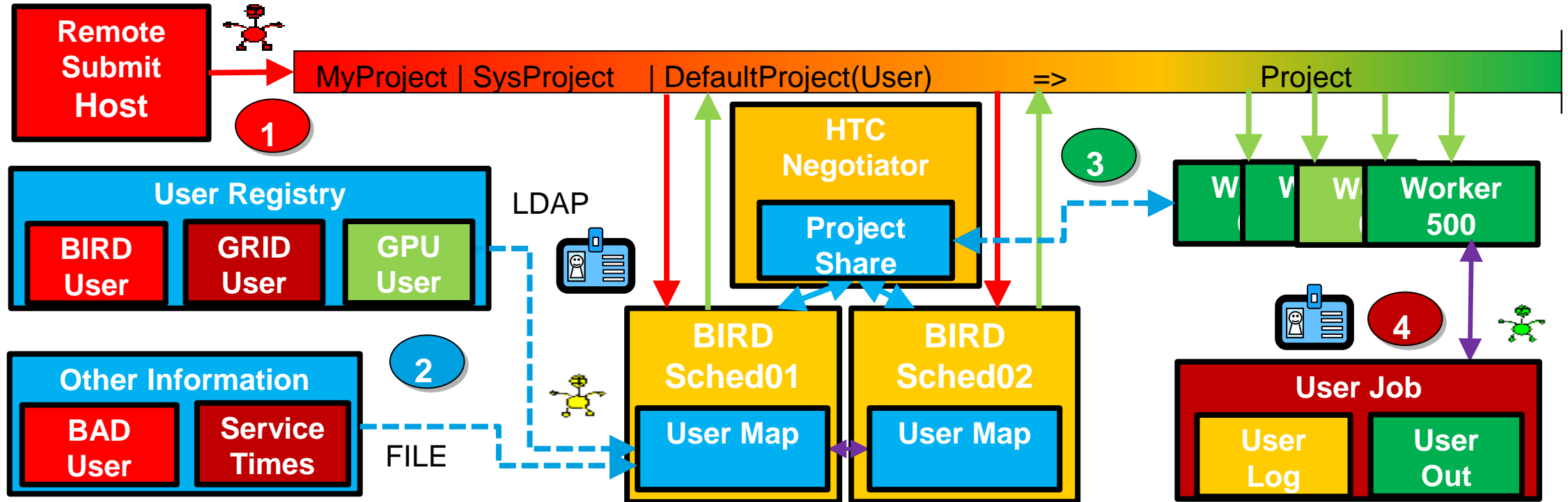
```
GroupsDisabled = "none"
InStageDrain = False
ShutdownTime = 0
IgnoreMaintenance = False
STARTD_ATTRS = InStageDrain, ShutdownTime, StartJobs, GroupsDisabled, BIRD_RESOURCE
STARTD.SETTABLE_ATTRS_ADMINISTRATOR = StartJobs, InStageDrain, ShutdownTime, GROUPS_DISABLED
DRAIN = ((InStageDrain == True && (time() + MaxJobRetirementTime < ShutdownTime)) || InStageDrain == False)
GROUP_NOT_DOWN = (StringListMember(DESyAcctGroup, GROUPS_DOWN) == False && StringListMember(DESyAcctGroup,
GroupsDisabled) == False)
IGNORE_MAINTENANCE = ((IgnoreMaintenance == True) && (Project == "support"))
START = ((StartJobs == True) && $(GROUP_NOT_DOWN) && $(DRAIN)) || $(IGNORE_MAINTENANCE)
```

CC and Job Timing

Version B: Preparing for a node for CC Maintenance



User Registry + User Blacklisting + Node Control



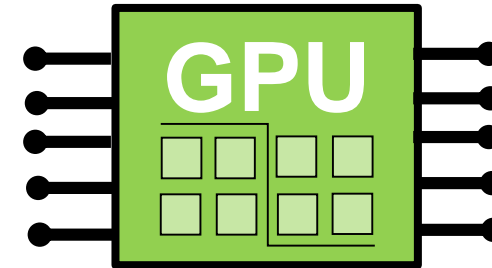
1	2	3	4
	Generate_UserMap.sh		Job_Wrapper.sh
Condor_submit	cron, ldap, Transforms.htc	Quota.htc	

GPU Support

Starting with a full node slot for a GPU



- Uses Enterprise Linux \geq Version 7
- Full Node Scheduling
 - 1 GPU-Slot per GPU-Node
 - In Combination with Jupyter Slot(s)
- User needs BIRD-GPU-Resource in Registry



```
JOB_TRANSFORM_T06AccountingGroup @=end
```

```
[  
eval_set_DESYAcctGroup = ifThenElse(Project =?= undefined, "BIRD_noop", \  
    ifThenElse(IsJupyterJob =?= True, "BIRD_jupyter", \  
    ifThenElse(RequestGPU != undefined && userMap("Projects.gpu",Owner) =?= undefined, "BIRD_noop", \  
    ifThenElse(userMap("Projects.noops",Owner) != undefined, "BIRD_noop", \  
    )  
]
```

```
JOB_TRANSFORM_T07AccountingStatusHold @=end
```

```
[  
eval_set_HoldReason = ifThenElse(RequestGPU != undefined && userMap("Projects.gpu",Owner) =?= undefined,  
"Unauthorized GPU request", \  
]
```

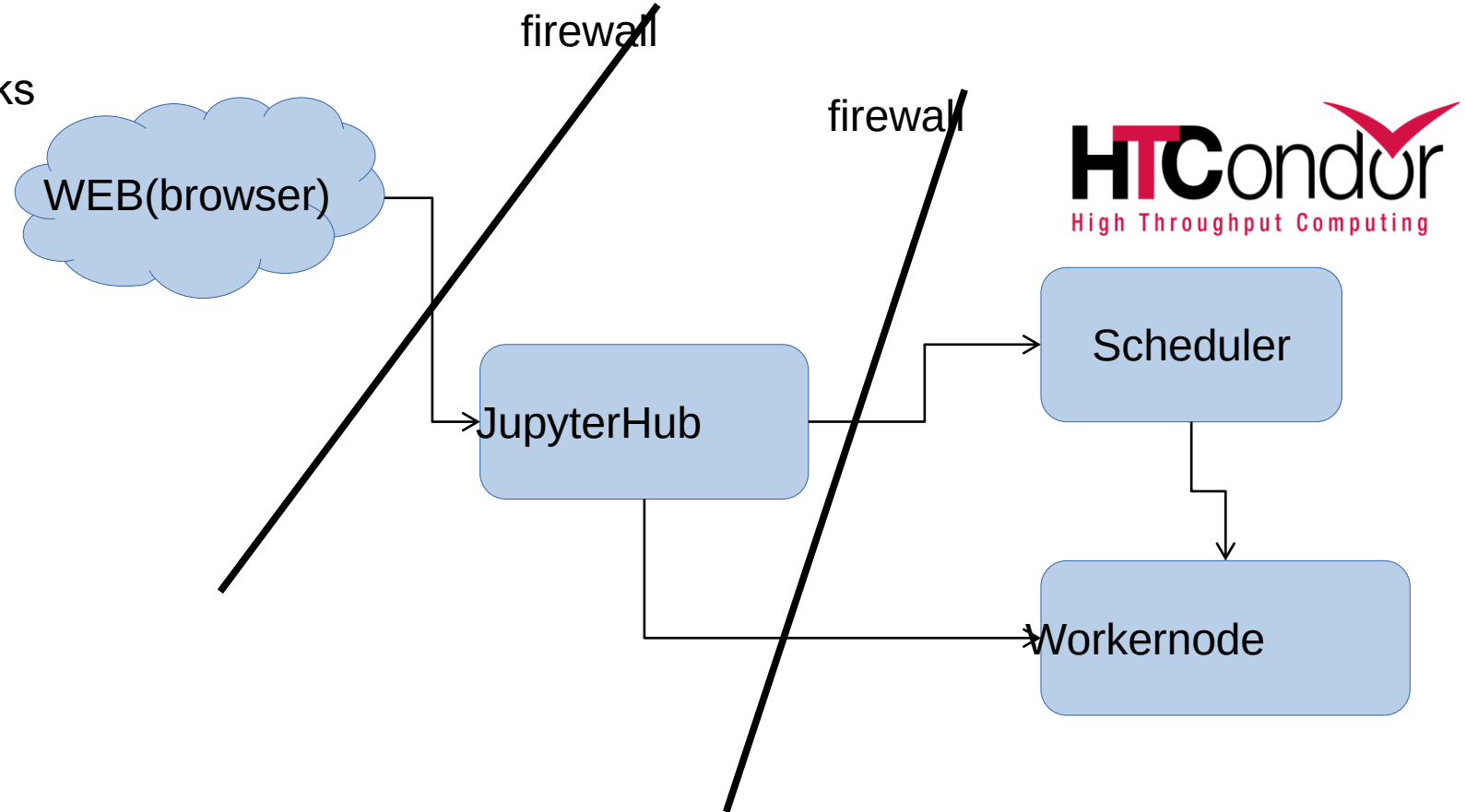


- Webinterface Jupyter Notebook

- Does the Proxying to Notebooks
- <https://naf-jhub.desy.de/>
- Runs the DESY DMZ
- Runs the Python Kernel

- Functionality Notebook Server

- Login of Users
- Database of logged in user
- Access to Data, Mounts etc.
- 2bg Ram soft limit
- 20h runtime “start your notebook once a day”



Integration of Jupyter Notebooks



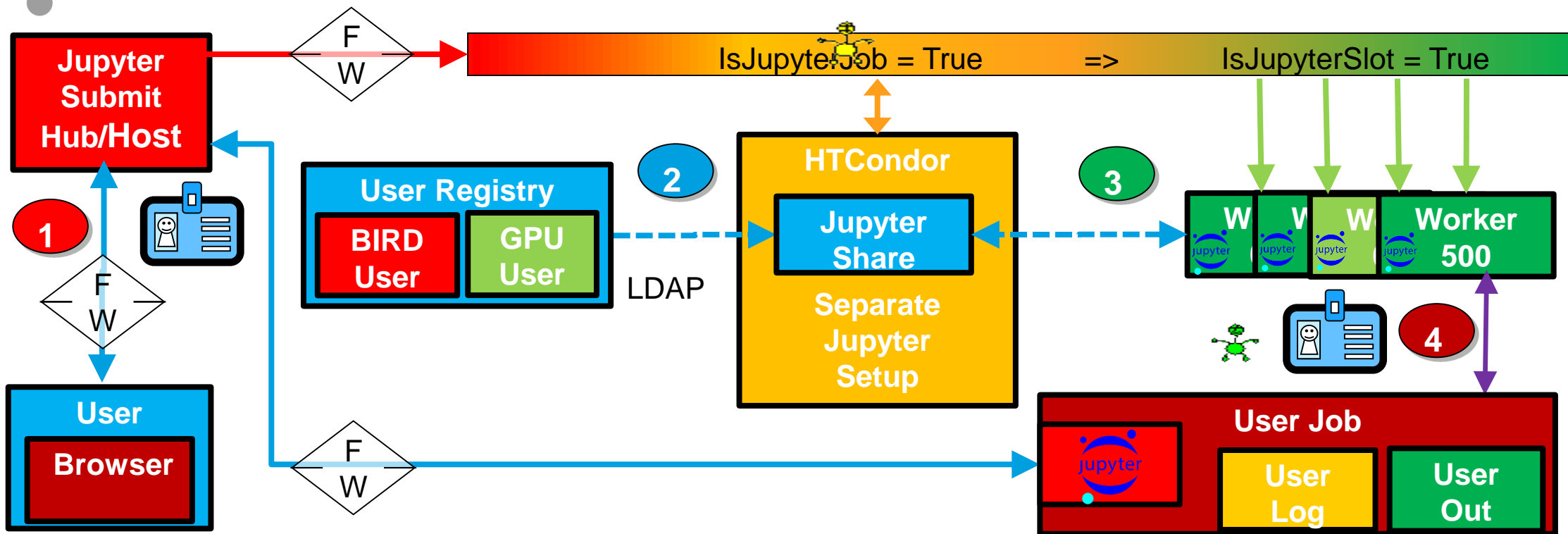
- BIRD/NAF
 - Jupyter-Hub for User Access
 - User needs BIRD-Resource in Registry
 - User don't needs BIRD-GPU-Resource
 - Jupyter Software and Configuration
 - HTC-Interface Configuration
 - Project, Runtime, ...
 - External Access
 - Open Port Ranges to HTCondor Schedds and Workers

```
JOB_TRANSFORM_T06AccountingGroup @=end
[
eval_set_DESYAcctGroup = ifThenElse(Project =?= undefined, "BIRD_noop", \
    ifThenElse(IsJupyterJob =?= True, "BIRD_jupyter", \
    ifThenElse(RequestGPU !== undefined && userMap("Projects.gpu",Owner) =?= undefined, "BIRD_noop", \
    ifThenElse(userMap("Projects.noops",Owner) !== undefined, "BIRD_noop",
```

- HTCondor-Backend Configuration
 - Needs Enterprise Linux \geq Version 7
 - Slots either shared on GPU nodes or oversubscribed on worker nodes
 - Scheduler Transforms for Automatic Setup
 - Special Jupyter Slots
 - Workernode Software Add On
 - Fast Startup for Interactive Usage

```
JOB_TRANSFORM_T18Jupyter @=end
[
Requirements = (IsJupyterJob =?= True && JobUniverse != 7);
copy_Requirements = "Base4Requirements";
set_Requirements = Base4Requirements && (IsJupyterSlot =?= True);
eval_set_JobHistory = False;
]
@=end
```

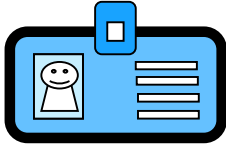

jupyter Communication Blocks



1	2	3	4
	Prepare Jupyter Settings	Fast Quota Bypass	Job_Wrapper.sh
Condor_submit	ldap, Transforms.htc	Quota.htc	

Docker Service

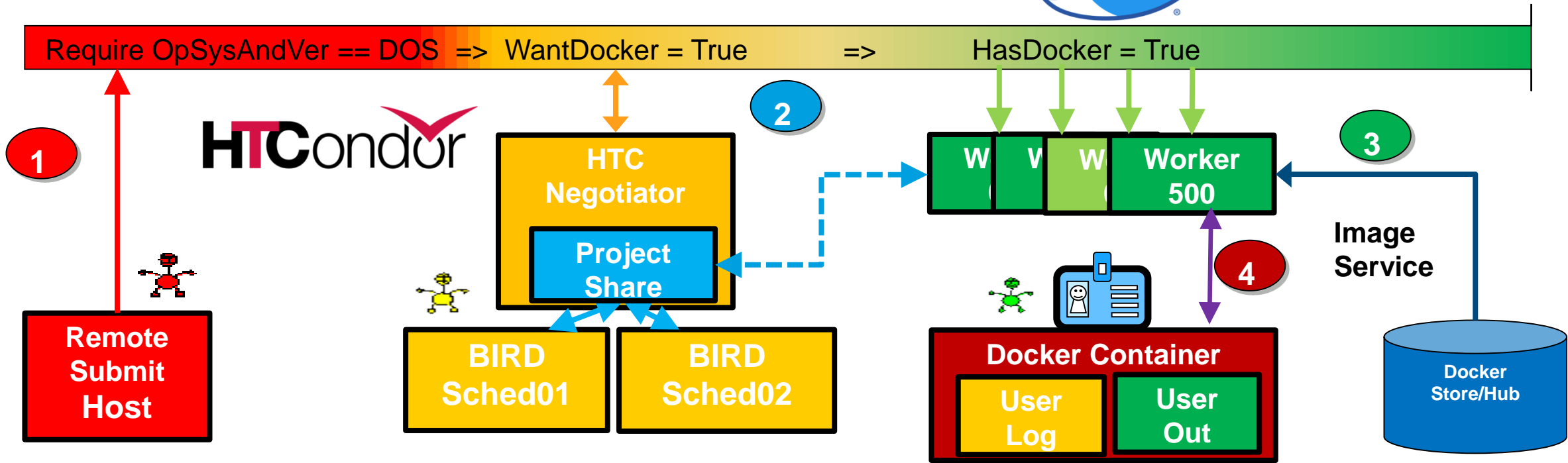
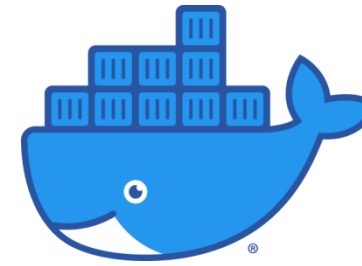


- BIRD/NAF
 - „Proof of Concept“ for planned feature done
 - Docker without Docker Universe
 - External mounts
 - Project settings
 - KRB/AFS
 - Additional Key Ring 
 - Image Support
 - System Images vs. User Images
 - Docker Hub with secure Images
 - Image Loading and Network Bandwidth
 - Storage Optimisation (Image Layer)

```
JOB_TRANSFORM_T09MapSL7 @=end
[
Requirements =
regexp("OpSysAndVer\\s*==\\s*"SL6on7"",
unparse(Requirements));
set_WantDocker = True;
#set_Universe = Docker;
set_DockerImage = "sl6krbafs:latest";
set_OpSysAndVer = "SL6on7";
]
@end
```



Docker Communication Blocks



1	2	3	4
	Prepare Docker Settings for DOS	Prepare DOS-Image	Job_Wrapper.sh
Condor_submit	Transforms.htc	Docker	Mount, configure

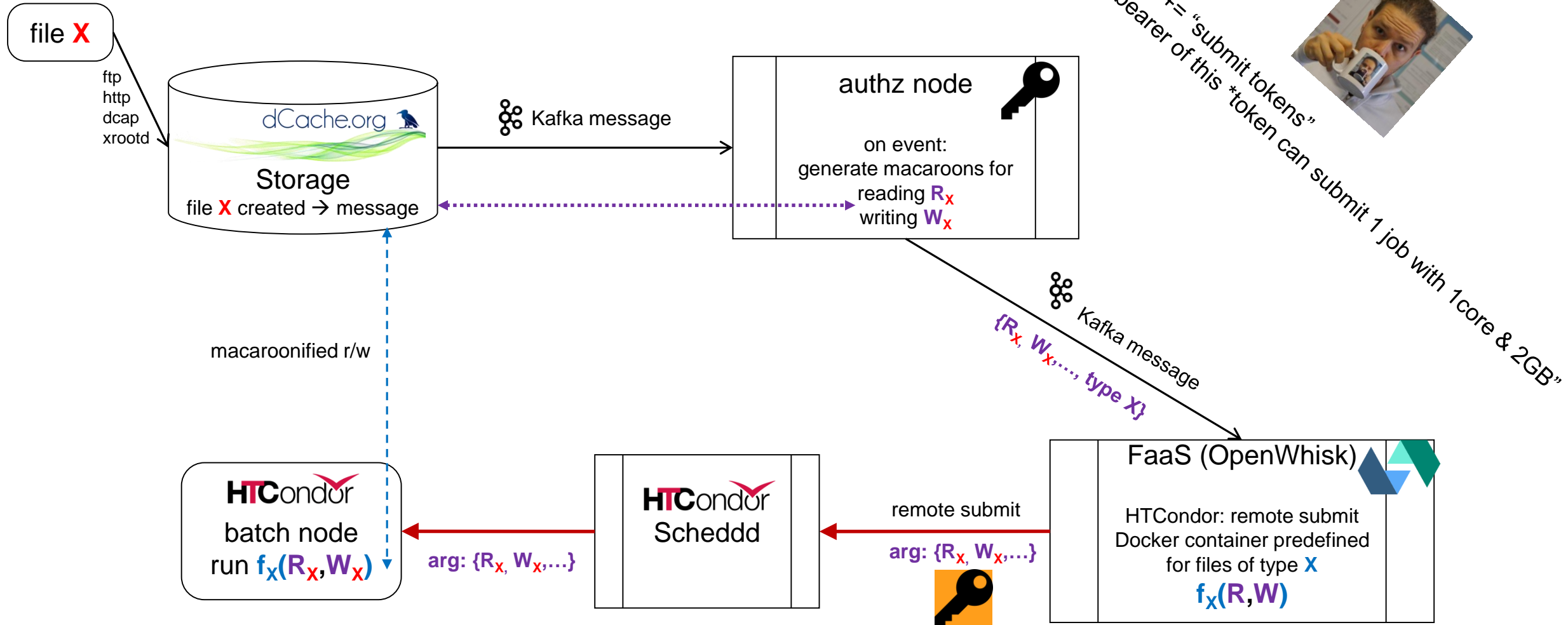
Function-as-a-Service

Idea: “anonymous” job workflows with authorization tokens


- “*personal job*”: submit as user ‘alice’ a job running under ‘alice’ to read file X in ‘alice’ resources
- “*anonymous job*”: job is run under “nobody”, reads/writes authorized through a transient token
- automated workflows: data events initiate jobs:
 - a new file **X** is written to the storage
 - files of type **X** are always to be processed with the same application **f**
 - let new files **X** trigger *themselves* their own processing with *function* **f(X)**
- use tokens for authorization
 - Storage Access with “*Macaroons*” as tokens
 - “the bearer of this token is allowed to read file **X** within the next 2 hours”
 - not need to carry any user authentication through the whole workflow

Event based FaaS with Condor as Backend

(draft)



Wishlist += "submit tokens"
 "the bearer of this *token can submit 1 job with 1core & 2GB"



submission still needs some form of authz...
 (... and have to do accounting right)

Potential Pitfalls

We are still learning ...

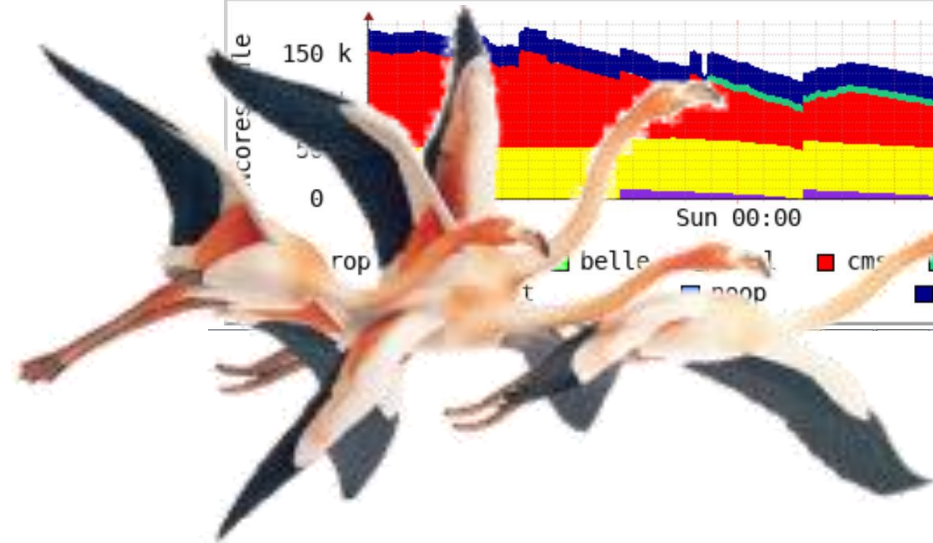
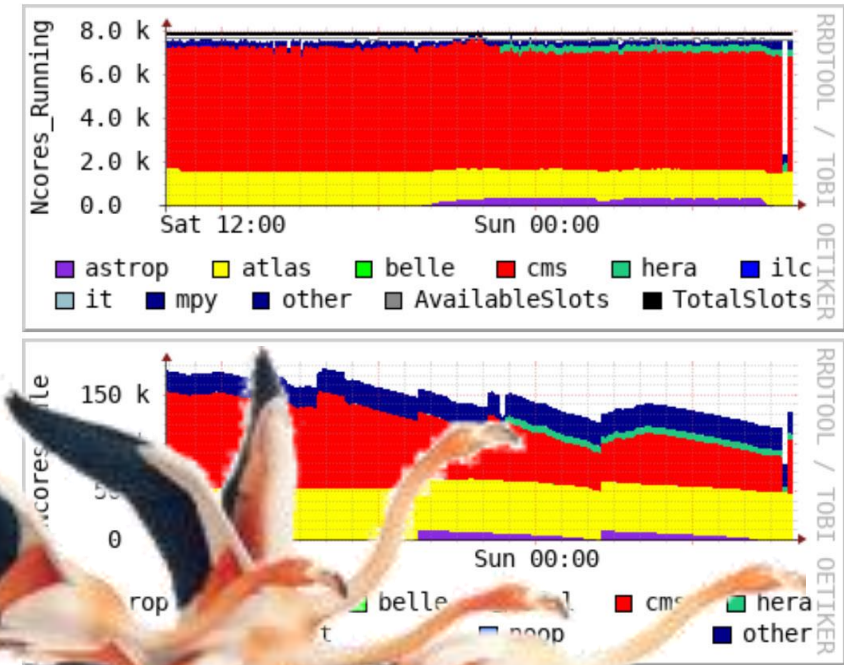


- Single users can disturb schedulers and file system access
 - Would like to have stable NFS access
 - Too many erroneous jobs make schedulers unresponsive
- No disable status on reinstalled machines
 - “StartJobs = False” survives reboot, but no reinstall
- Slow migration from SL6 (73%- Slots) to EL7 (27%+ Slots)
- Small job requests sometimes idle forever
- Environment Setup
 - Reentrant jobwrapper to realize LD_LIBRARY_PATH after setting new project
- Scope of ClassAds
 - New Version: Special Treatment RequestRuntime
 - MaxJobRetirementTime: From Worker or from Job ?
- Jupyter
 - Startup time of the notebook is work in progress
 - Message handling between condor and jupyter needs to be improved

Outlook and Conclusions

- BIRD/NAF
 - „Proof of Concept“ for planned Feature done
 - Waiting for HTCondor new Auth Features
 - Full Kerberos and AFS support
- Next Steps may be ...
 - Final Jupyter and GPU Configurations
 - No SL6 and NFS
 - Docker as SL6 Replacements
 - GRID uses Singularity for legacy jobs
 - Docker for different operating system flavours
 - **Function-As-A-Service**

Cores over time, different states:



Thank you for listening



Questions !

Answers ?



Appendix

A bit more details

Function as a Service

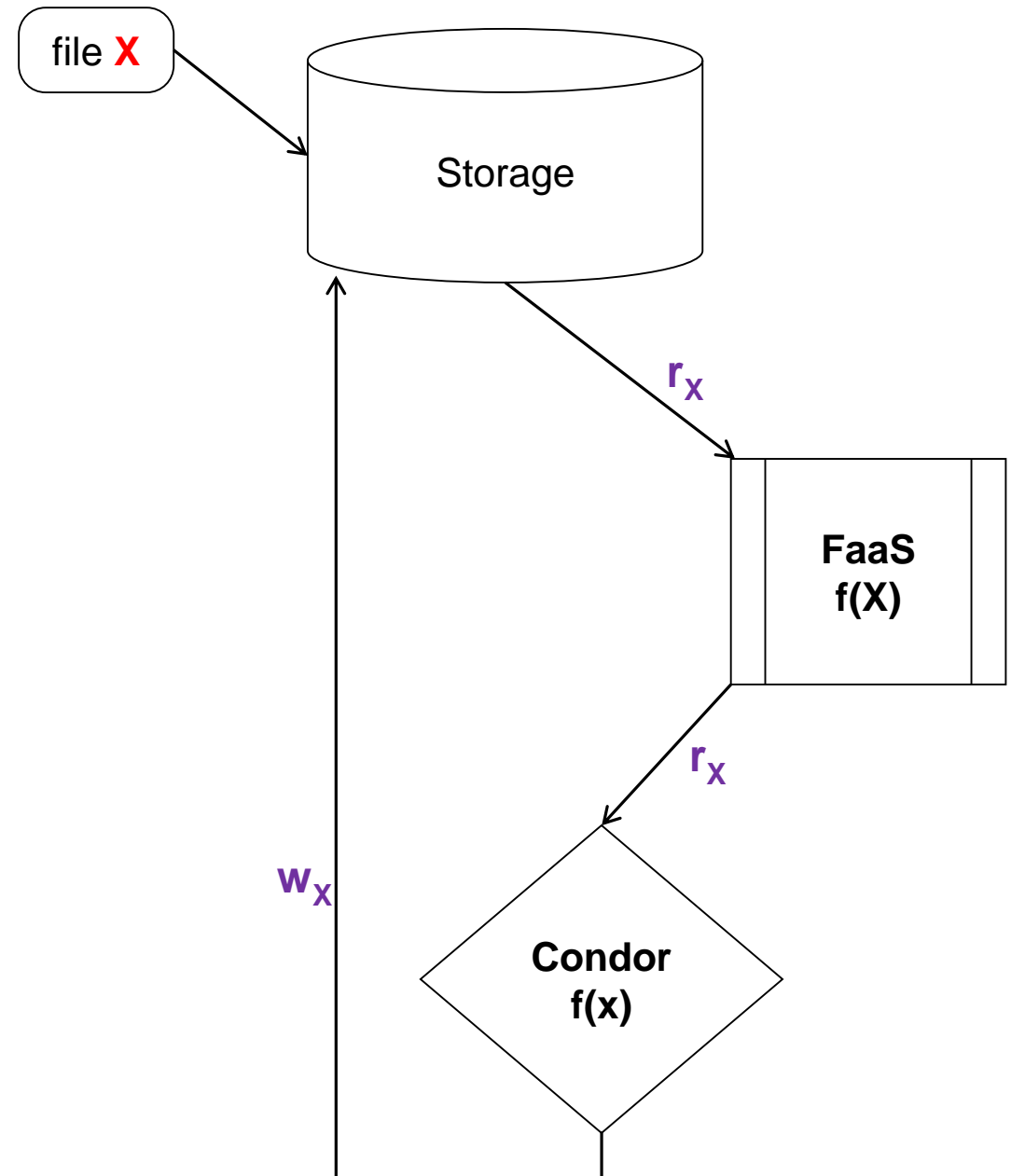
(with HTCondor as workhorse)

Idea

- automatic workflows triggered by *external* events

Example

- files/events need to be processed always the same way
- new file **X** comes into the storage
 - containing events be calibrated
 - create tokens for read r_x and write w_x
 - submit calibration job $f(X)$ to Condor
 - ...
 - success



FaaS with Condor

(quasi)-anonymous jobs with macaroons

- user FOO submits job 314159
 - job 314159 belongs to user FOO
 - can access resources/files belonging to FOO
- *anonymous* job → disentangle somewhat from the user (except for accounting of course)
 - job 161803 submitted by a generic user without *specific* resources
 - access to input/output files only by dedicated authorization tokens
 - can **only read** file X
 - can **only write** to Z.d

Macaroons

Access tokens generated by the dCache storage



macaroon: lengthy, cryptic “*enhanced cookie*”

- e.g.: MDAxY2xvY2F0aW9uI...2lnbmF0dXJlIDbpQS1h1zYdMh08

for the URI

- <https://dcache-se-doma.desy.de:2880/desy/Hamburg/MacaroonIO/read.d/test.root>

the bearer of this token is allowed to

- read and get metadata over WebDAV/HTTP

the resource

as long as

- the request is from subnets 131.169.180.0/24 or 188.184.9.0/
- and only within the next 2 hours

Macaroons

Access tokens generated by the dCache storage



similarly:

generate output directory and macaroon token, that allows

- to write

to

- <https://dcache-se-doma.desy.de:2880/desy/Hamburg/MacaroonIO/write.d>

with the limitations

- validity for the next 4 hours
- only from within the subnet 131.169.180.0/24

btw: macaroons can always be further restricted (but not extended)

→ the bearer of the token above could derive another token

- which could be further limited to subnet 131.168.180.214/32 only

Macaroons

Beefed-Up Cookies / Limiting access tokens to the necessary

e.g., requesting with curl a macaroon allowing reading a file for 1¹/₂ days and using a grid proxy for authz:

```
curl -L --key $X509_USER_PROXY --cert $X509_USER_PROXY --cacert $X509_USER_PROXY --capath $X509_CAPATH \
-X POST -H 'Content-Type: application/macaroon-request' \
-d '{"caveats": ["activity:DOWNLOAD,LIST"], "path":"/desy/Hamburg/MacaroonIO/read/foo.root", "validity": "P1DT12H"}' \
https://dcache-se-doma.desy.de:2880
```

return:

```
"macaroon": "MDAzNW...PE1Cg",
  "uri": {
    "targetWithMacaroon": "https://dcache-se-
doma.desy.de:2880/desy/Hamburg/MacaroonIO/read/foo.root?authz=MDAzNWx...",
    "baseWithMacaroon": "https://dcache-se-doma.desy.de:2880/?authz=MDAzNWxvY...",
    "target": "https://dcache-se-doma.desy.de:2880/desy/Hamburg/MacaroonIO/read",
    "base": "https://dcache-se-doma.desy.de:2880/"
  }
}
```


example: dCache storage event message

```
{ "date": "2019-07-31T15:31:10.388+02:00",
  "msgType": "transfer",
  "transferTime": 41,
  "cellName": "dcache-desy15-03",
  "session": "pool:dcache-desy15-03@dcache-desy15-03Domain:1564579870388-221080",
  "subject": [
    "UidPrincipal[34839]",
    "GidPrincipal[5296,primary]",
    "GidPrincipal[5148]",
    "GidPrincipal[5296]",
    "GidPrincipal[5847]",
    "GidPrincipal[1094578758]",
    "Origin[2001:638:700:10a8::1:58],",
  ],
  "initiator": "door:nfs4-wgs@dcache-door-desy02_nfs4Domain:AAWO+iNkqWg:1564579870321001",
  "transferPath": "/",
  "meanReadBandwidth": 394645595.2664036,
  "version": "1.0",
  "storageInfo": "belle:local@osm",
  "readIdle": "PT0.009028762S",
  "transferSize": 76516,
  "protocolInfo": {
    "protocol": "NFS4",
    "port": 437,
    "host": "2001:638:700:10a8:0:0:1:58",
    "versionMajor": 4,
    "versionMinor": 1 },
  "cellType": "pool",
  "readActive": "PT0.032327636S",
  "fileSize": 76516,
  "queuingTime": 0,
  "cellDomain": "dcache-desy15-03Domain",
  "isP2p": false, "pnfsid": "0000F2FDE34028D340CEA98ABA87ED0E4336",
  "billingPath": "/",
  "isWrite": "read",
  "status": {
    "msg": "",
    "code": 0
  }
}
```