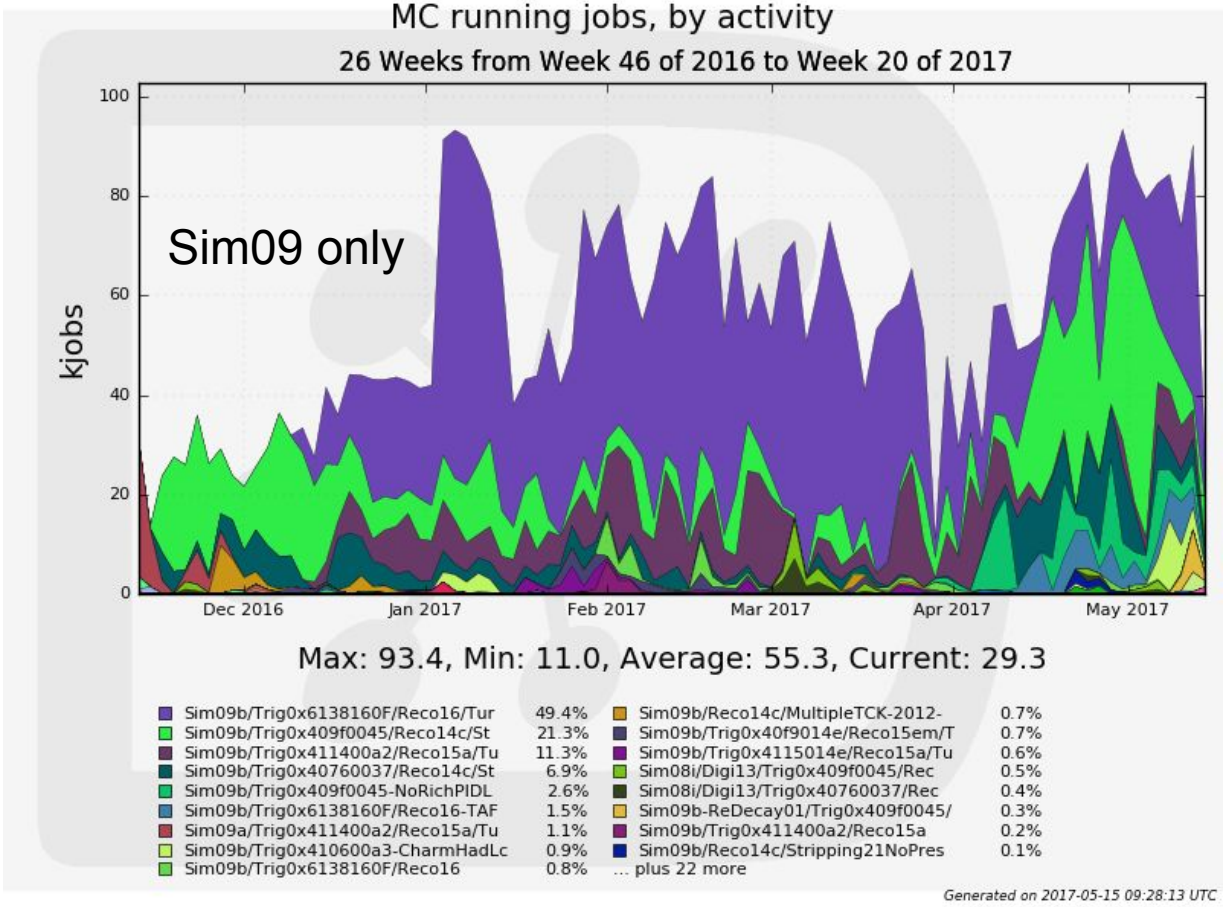
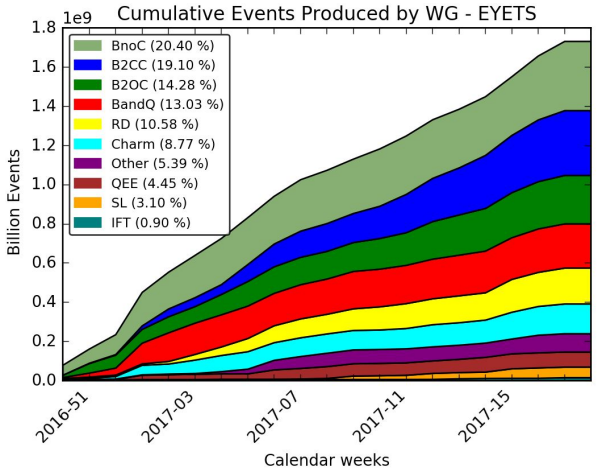
LHCbDIRAC

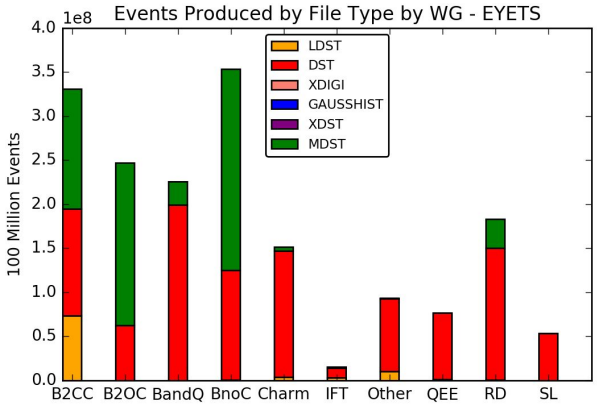
Activities, recent and upcoming developments

Distributed computing resources: usage

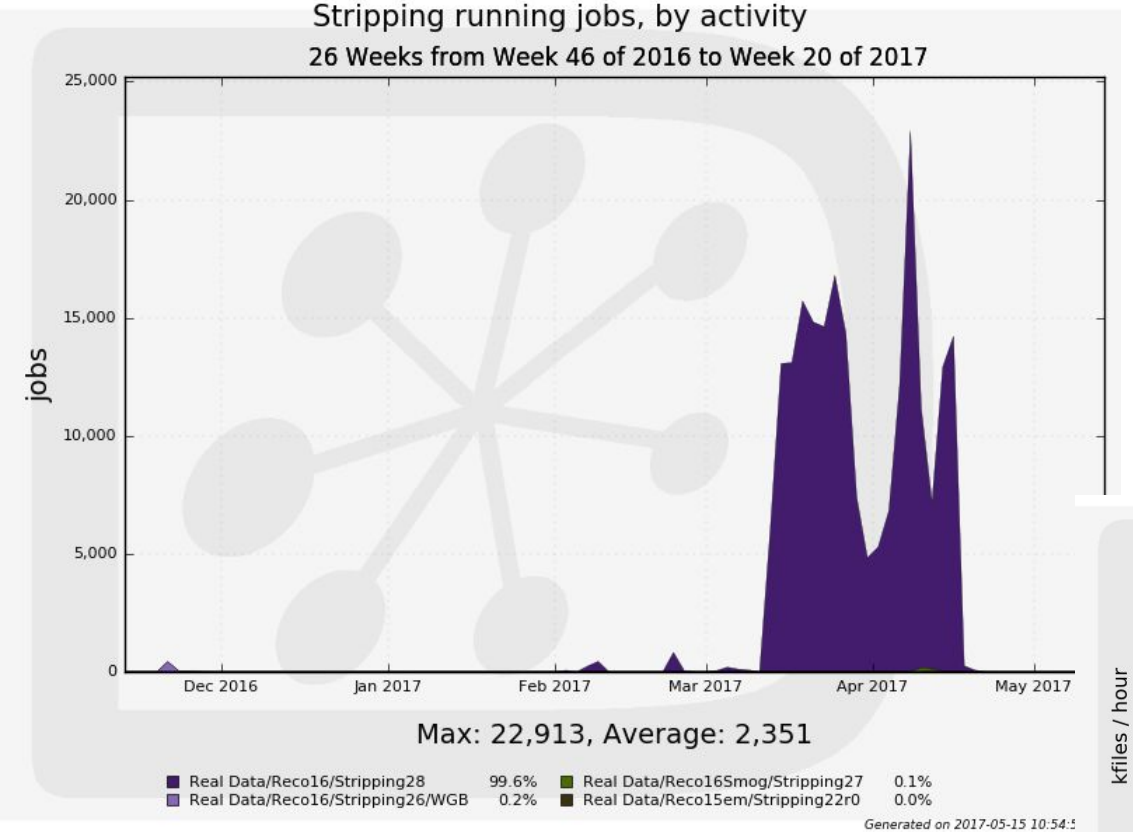
EYETS activities: MonteCarlo Simulations



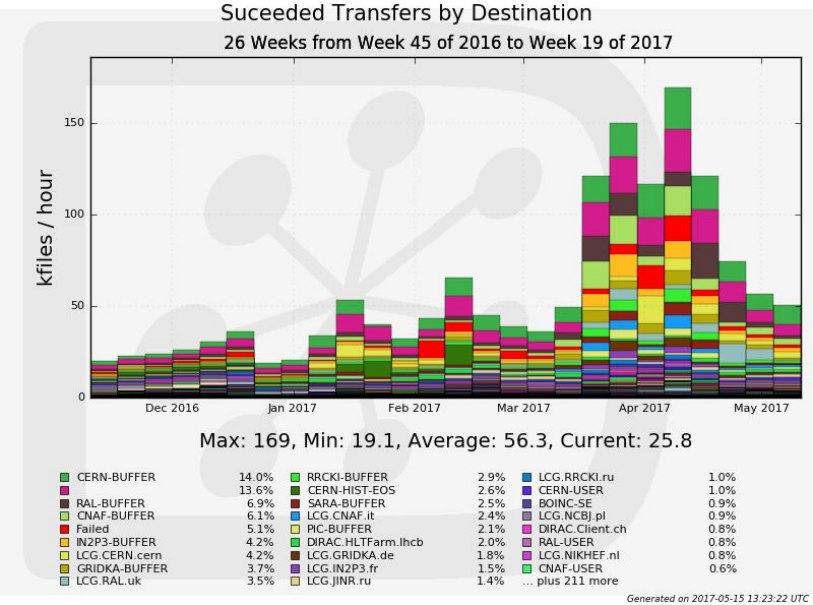




EYETS activities: DataStripping

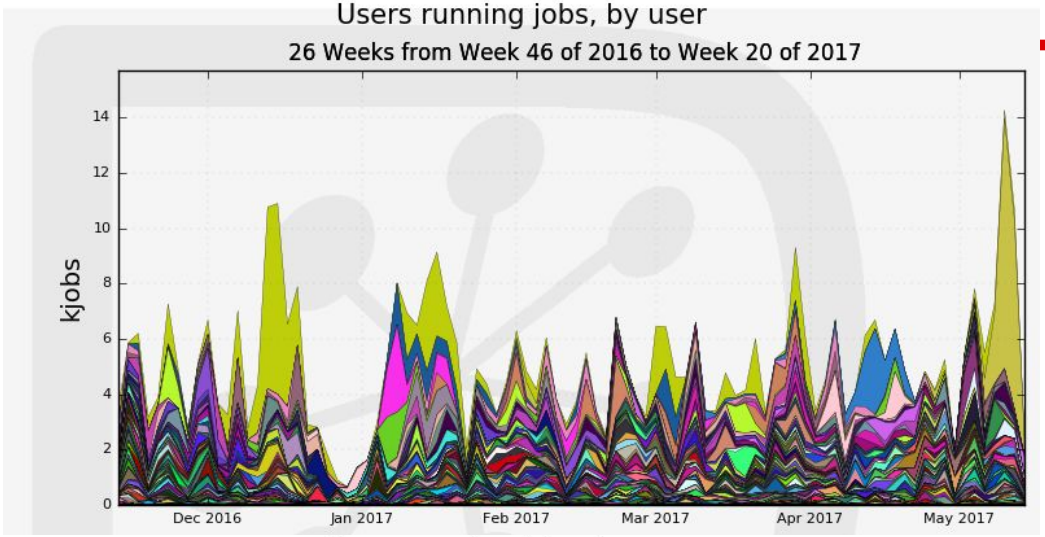


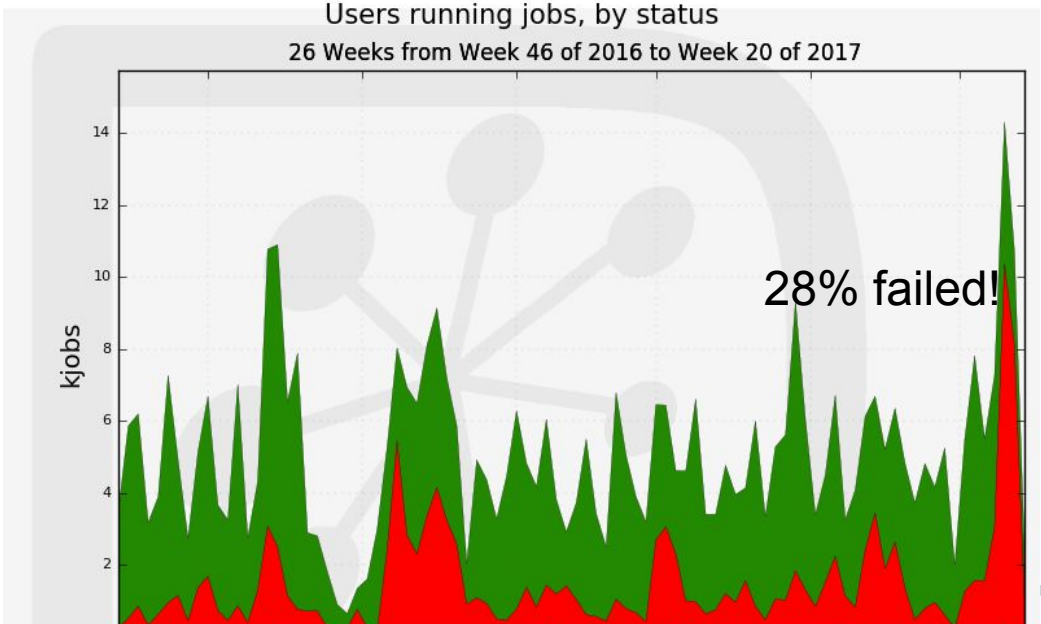
- Dominated by Stripping28
- 4 intense weeks

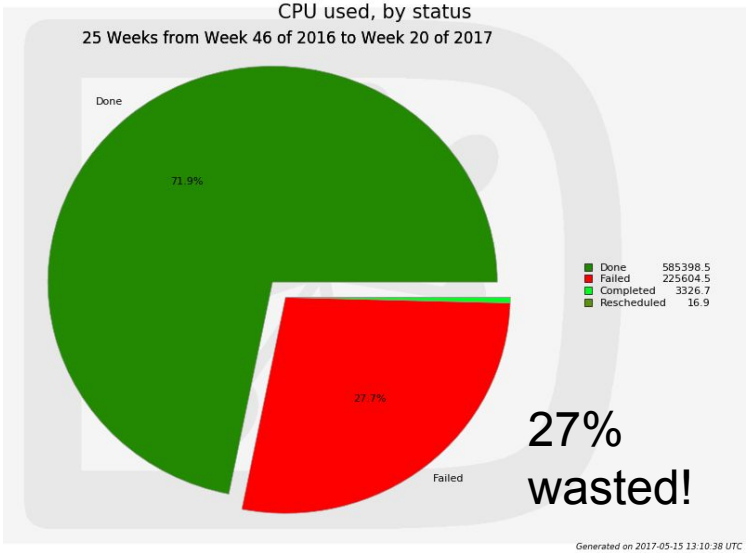


- Incremental Stripping of 2015 to be started soon

EYETS activities: Users' jobs

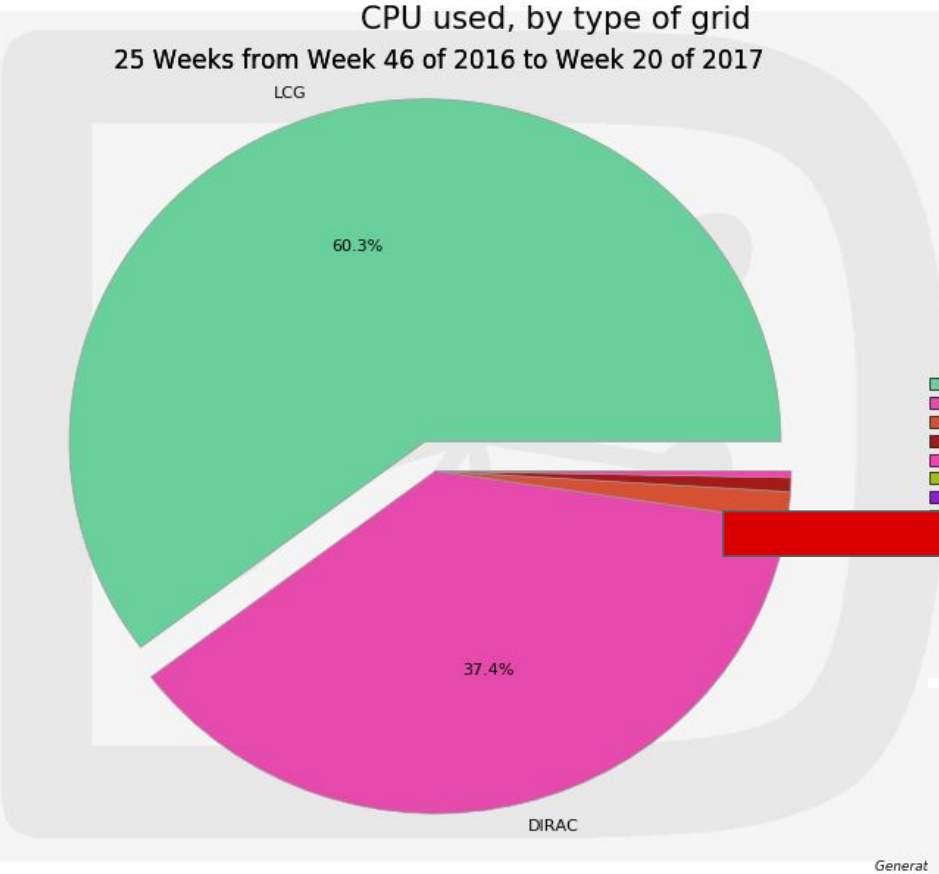


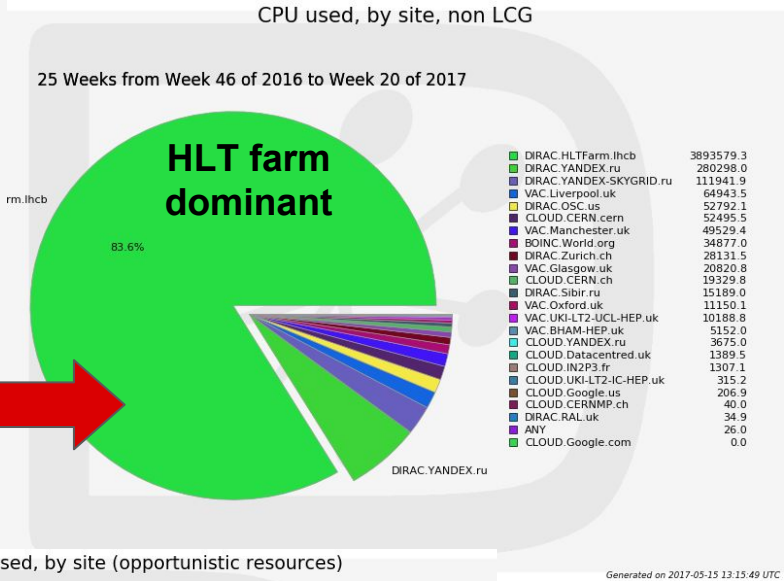


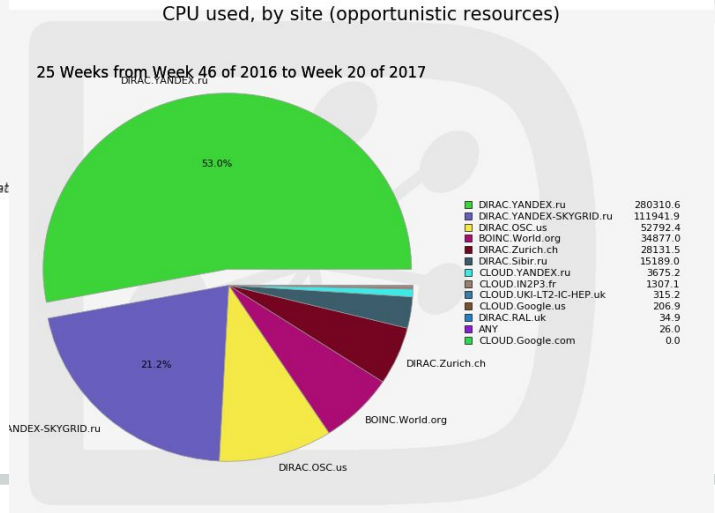


27% wasted!

EYETS activities: Offline resources







- Plan to continue to use HLT farm even during LHC ‘17 ramp up

LHCb DIRAC management

LHCb DIRAC Pillars

- We are operating a service: need to keep a running system working, with continuity
  - Partial down-times may anyway happen
  - Up to now we managed to schedule the most important ones
- There won't be any revolution
  - The system is evolving gradually
  - We are introducing new/better/faster stuff
    - Users should not notice about (most of) them
- Usability and scalability are the end goals

Reminders

- DIRAC is a fully open source project
  - Extended in LHCb
- Used by 40+ communities
- 220K+130K lines of python 2
    - Plus some .sh, .js

- Maintenance is a daily work

Towards the upgrade: Done/ToDo (1)

- Certification process and team - DONE
  - needs proper maintenance and team ready to act
- Python backward/forward compatibility - STARTED
  - Pilot: compatibility for python 2.6, 2.7(.5, .12)
    - Not yet the server (no real need)
  - Python 3 later
- Systematic testing for performances - TODO

Towards the upgrade: Done/ToDo (2)

- Introducing real-time monitoring backend - DONE
  - ElasticSearch
  - Using it - partially DONE
- Replace custom libraries - IN PROGRESS
  - Also for performance reasons
  - Student (ISIMA) working on DIRAC RPC
- Packaging (e.g. via docker) - ~DONE
  - Not published on docker store
- High availability and load balancing - IN PROGRESS
  - Mesos setup already used in certification setup
- Logging - IN PROGRESS
  - Student project (ISIMA)
  - Centralized
  - Structured
  - Pars/Grep/Grok-able

Towards the upgrade: Done/ToDo (3)

- Introducing MQ systems in DIRAC - DONE
  - Using them in LHCb - TODO
    - We have some very basic usage already
- Bulk jobs submission - IN PROGRESS
  - A first version is ready, not considered safe by LHCb
  - Not anymore highest priority

Towards the upgrade: Done/ToDo (4)

- Revision of SQL schemas - TO RESTART
- Using EOS for LHCb logs - IN PROGRESS
  - Just restarted (no progress - low priority)
- Integration of block storages - TODO

Timeline (restricted) (by TDR roadmap)

	2016 Q1	2016 Q2	2016 Q3	2016 Q4	2017 Q1	2017 Q2	2017 Q3	2017 Q4
Non-SQL integrations		X		X				
MQs integrations			X	X				
New logging						X		
Python 3 plans							X	

New (restricted) timeline

	2017 Q2	2017 Q3	2017 Q4	2018 Q1	2018 Q2	2018 Q3	2018 Q4
New logging	X →	→					
Python 3 plans		X →	→				
Consumer/listeners	→	→	→	→	→	→	

Few “Done” items

- Abandoned completely SetupProject
  - Now always lb-run
    - --use-grid
    - --platform best (from v8r8 - next week)
      - when no platform is specified
- Test of development version of LbScripts
- BKK changed partitioning and hosting
  - better performances
- Steps manager changes (v8r8 - next week)
  - move of output files visibility flag out of application steps

Few “ToDo” items

- Matching on Grid WNs using LbPlatformUtils
- MC changes (various merges)
- Interruptable MC
  - Currently tested on the HLT farm
    - LHCbDIRAC OK with that
  - Extending it outside the farm: needs MJF

Versioning

- LHCbDIRAC release cycle: weekly patches
  - Time-based release process
  - Release prepared on Friday, deployed on Monday
  - In operation for more than a year now
    - good feedback

- DIRAC: not the same.

Questions

?

BACKUP Slides

The certification process

- We automate what we can automate (Jenkins, Travis)
  - Static code analysis
  - Unit tests
  - Integration tests
  - Regression tests
- Then, there are well documented tasks
  - For system tests
    - A pretty long list of them, and keeps increasing
- Continuous feedback is a key

Message Queues

- We are introducing MQ (stomp)
  - Consumers as DIRAC components? → RFC
- And listeners
  - To topics
- We can replace/complement agents or services with consumers/listeners
  - and also (especially?) executors
  - Agents, executors, consumers as a single component?

Replace custom libraries

- pyGSI
  - In python 2.7.9+ ssl module(s) have been backported from python 3
- Multithreading and multiprocessing
  - Futures?
- gLogger -> logging
- DEncode -> json

- DISET protocol -> ?
  - Tokens?
  - REST?