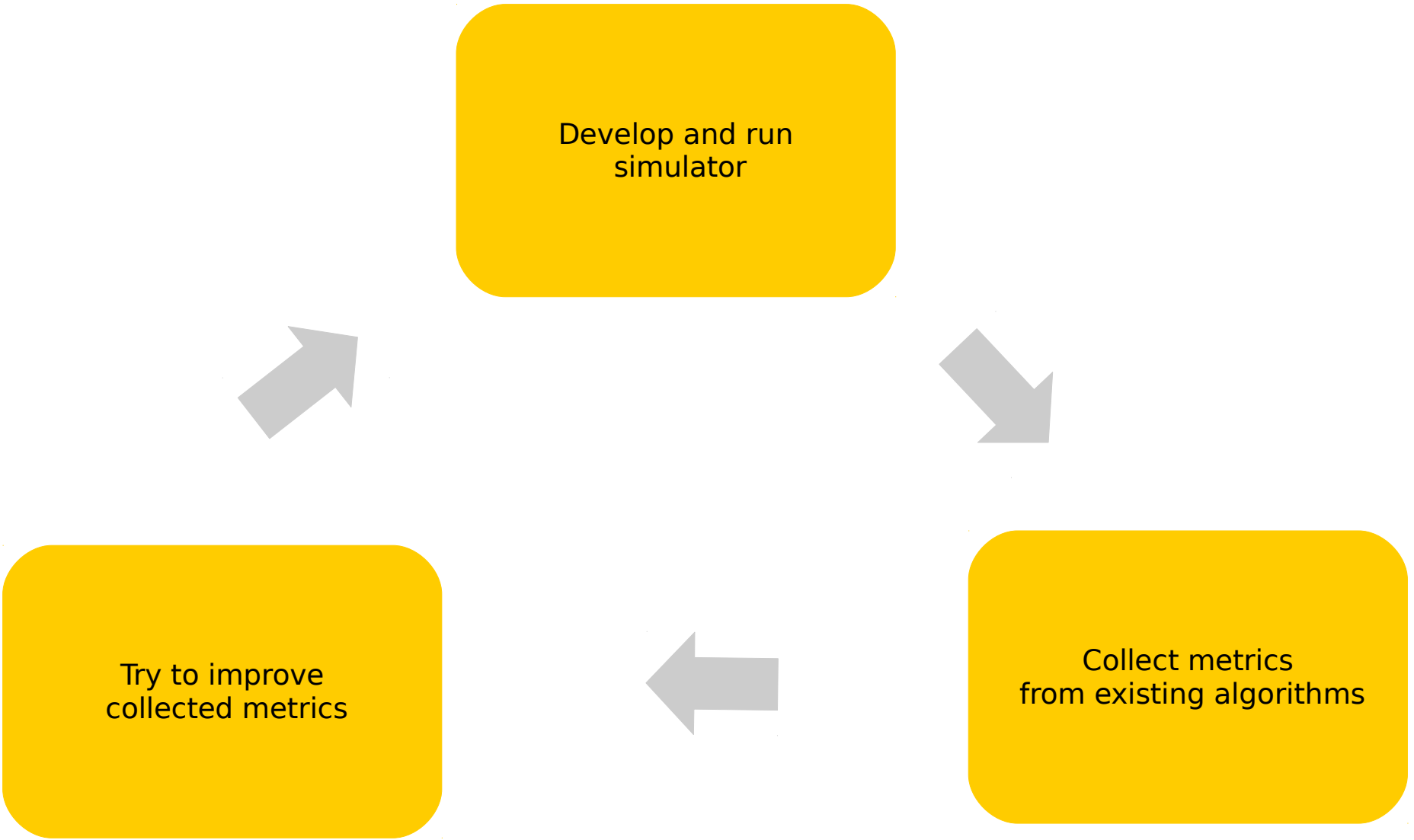
LHCb Grid Simulation

Goals and problems

- Try to develop smarter algorithms for
  - Data management
  - Predict anomalies
  - Job scheduling
- Need a simulator to test such things
- Use grid simulation to optimize algorithms/compare with existing ones
- Have a feedback from you

Strategy


Develop and run simulator
Collect metrics from existing algorithms
Try to improve collected metrics

Simulation process

Current state of the project

- More real input jobs
- Storage
- Real links
- Multi-core (1job per core)
- Tracing metrics

Input job parameters

- Name
- Type *(User, MC, Reconstruction, Stripping , …)*
- Time of job submission to queue
- Amount of flops needed to execute job
- Name of input dataset file
  - Size of dataset
  - Number of available replicas
  - Types (disk or tape) of available replicas
  - Locations of dataset
- Name of output dataset
  - Size of dataset
  - Number of output replicas
  - Locations and storage type (disk or tape) of output replicas
- Maybe something else?

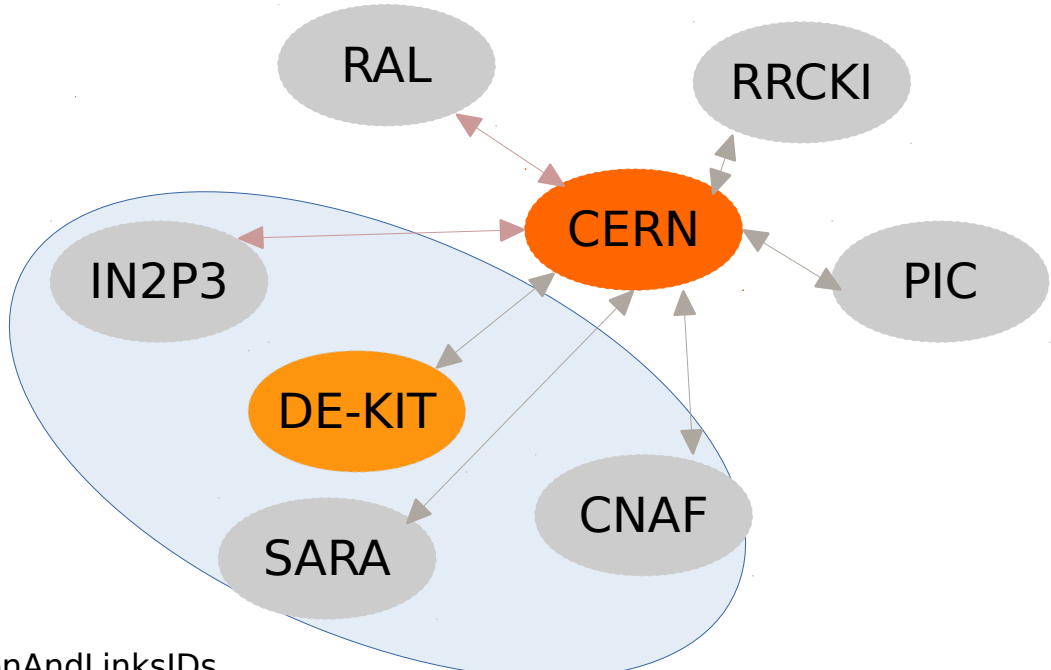
Network

Tier0-Tier1 links

Link Id	T1	Use	Bandwidth
CERN-CNAF-LHCOPN-001	CNAF	Primary	10G
CERN-GRIDKA-LHCOPN-001	GRIDKA	Primary	10G
CERN-IN2P3-LHCOPN-001	IN2P3	Primary	10G
CERN-PIC-LHCOPN-001	PIC	Primary	10G
CERN-RAL-LHCOPN-001	RAL	Primary	10G
CERN-SARA-LHCOPN-001	SARA	Primary	10G
CERN-RRCK1-LHCOPN-001	RRCKI	Primary	2G
CERN-RAL-LHCOPN-002	RAL	Backup	10G
CERN-PIC-LHCOPN-002	PIC	Backup	1G

Tier1-Tier1 links

Link Id	Bandwidth
CNAF-GRIDKA-LHCOPN-001	10G
GRIDKA-IN2P3-LHCOPN-001	10G
GRIDKA-SARA-LHCOPN-001	10G



Data taken from
https://twiki.cern.ch/twiki/bin/view/LHCOPN/NamingConventionAndLinksIDs

Tier

- Number of cores
- Flops per core
- Anomalies
  - Schedule of anomalies
- Links
- Storage
  - Disk
  - Tape

Storage

Each tier has
- 1 attached disk
- 1 attached tape
- 7 mounted disk (remote access to another Tier1s’ storages)
- 7 mounted tape (remote access)

Each storage is characterized by write/read/connection rate

Each storage has a file content catalog which contains name and size of datasets

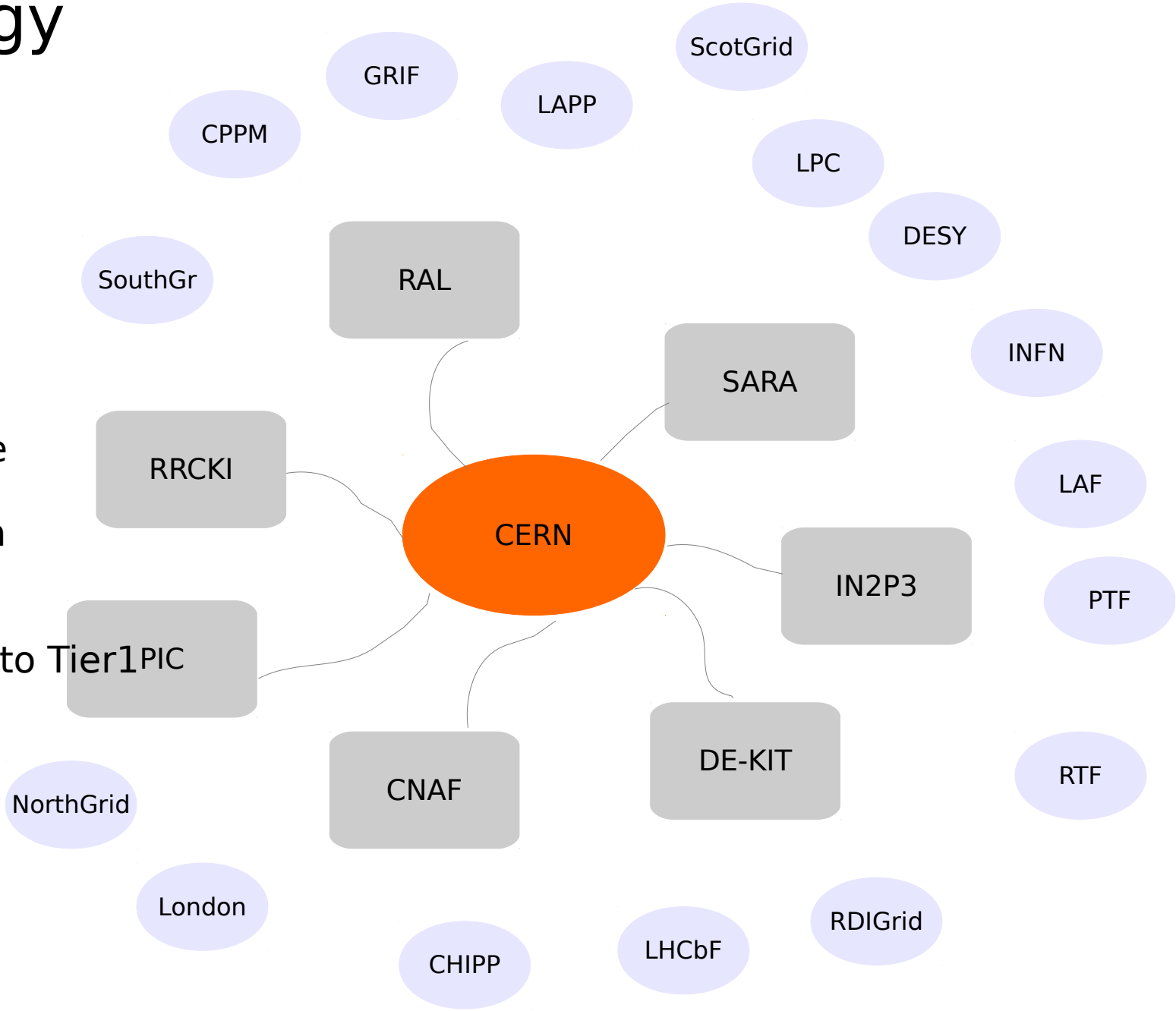
Data popularity

- Data management: how increase or decrease in the number of replicas affects job's wall time
- Optimizing disk space by deleting less popular files. Different strategies can be tested
  - LRU
  - LFU
- Namenode contains all relevant info about file popularity:
  - Filename
  - File size
  - Array of clock times when file was requested
- Every *N* days file-deleter seeks for less popular files

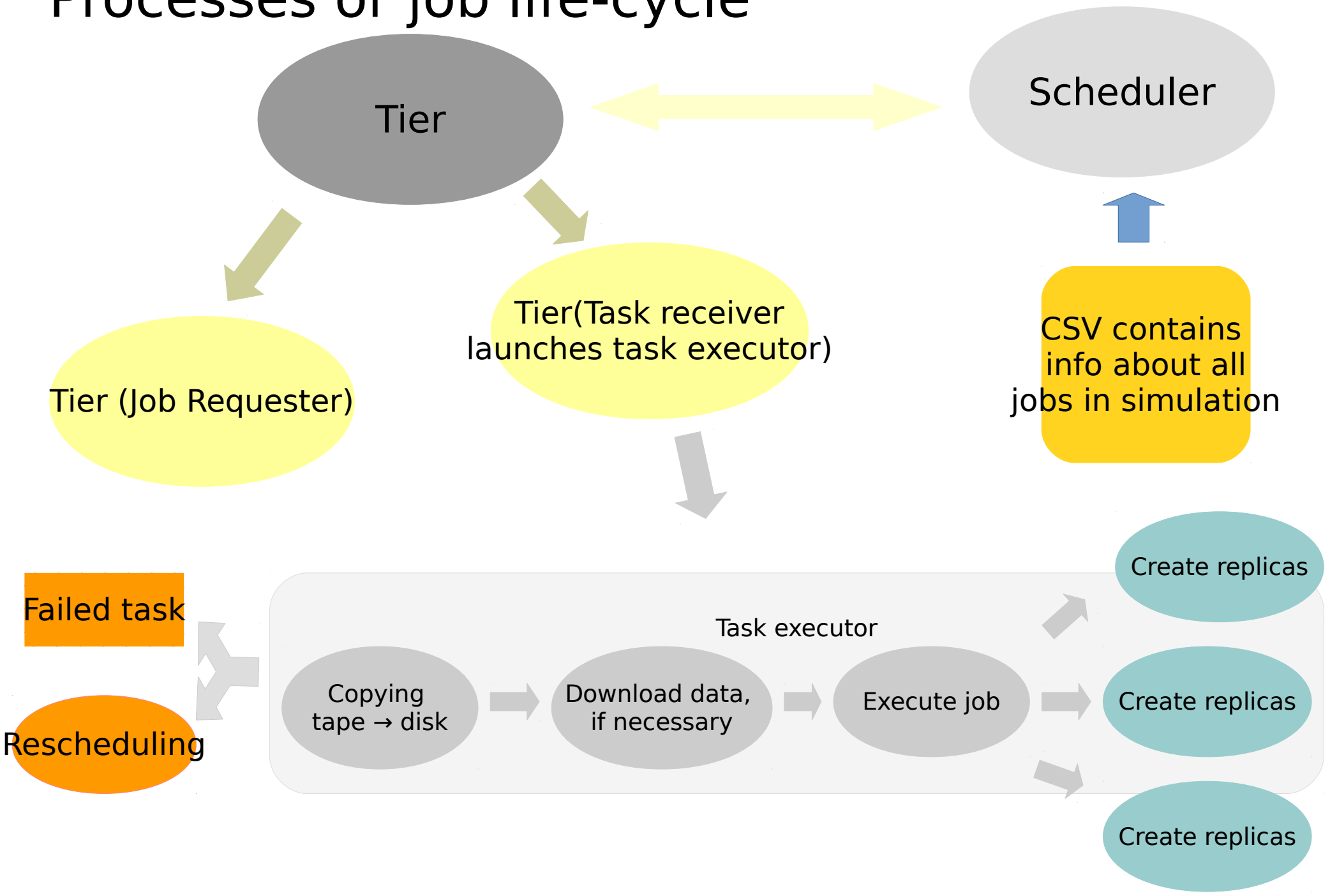
Grid topology

- Tier0
- 7 Tier1
- 16 Tier2s

- Currently known:
  - Tier0-Tier1 links
  - Tier1-Tier1 links
  - Tier0/1/2 storage
  - Tier0/1/2 CPU
- Currently unknown
  - Tier0-Tier2 links
  - Tier1-Tier2 links
  - {Tier2s} belong to Tier1
  - Multi-links

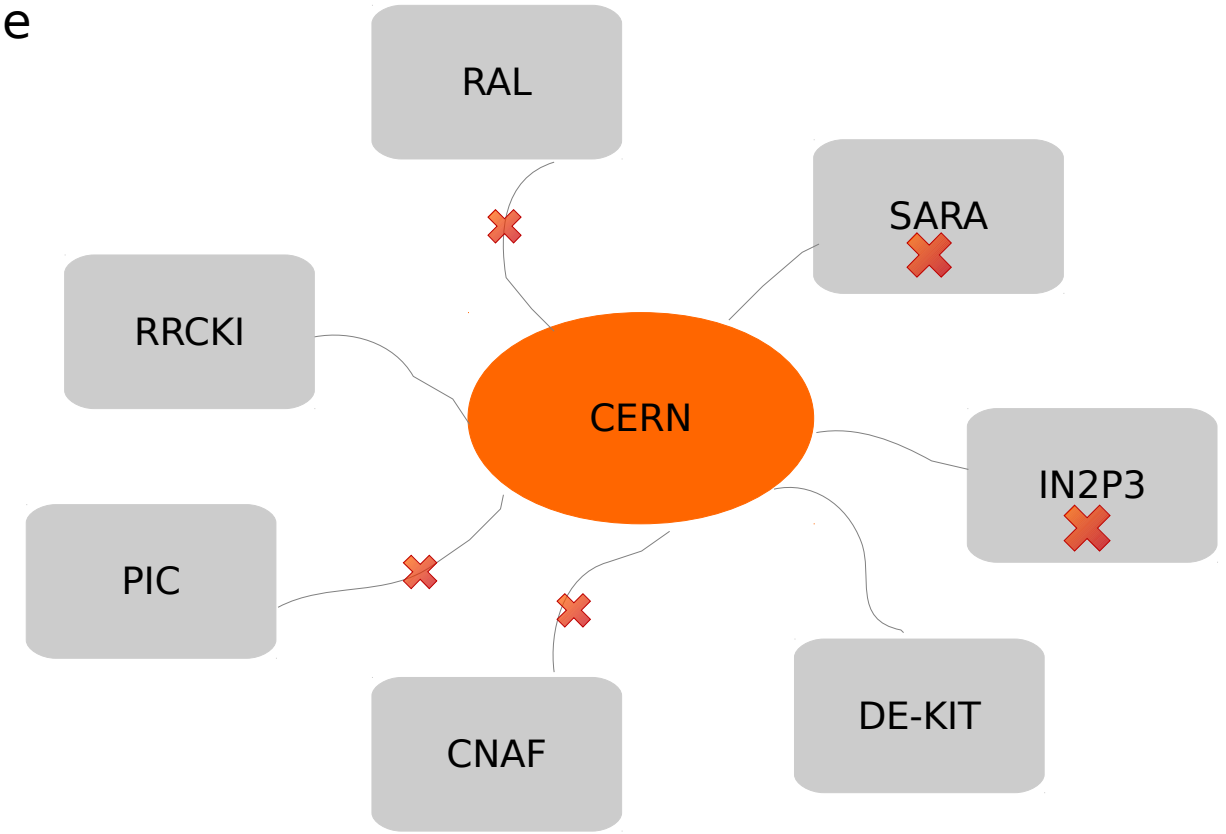


Processes or job life-cycle



Anomalies

- Links
  - Decreasing of bandwidth (CERN-PIC, CERN-RAL)
  - Link break
- Host
  - Core’s break by schedule
- Rescheduling?



Not all links are shown here

Algorithms of scheduling

Simple
- Distributes task by “place” in the queue

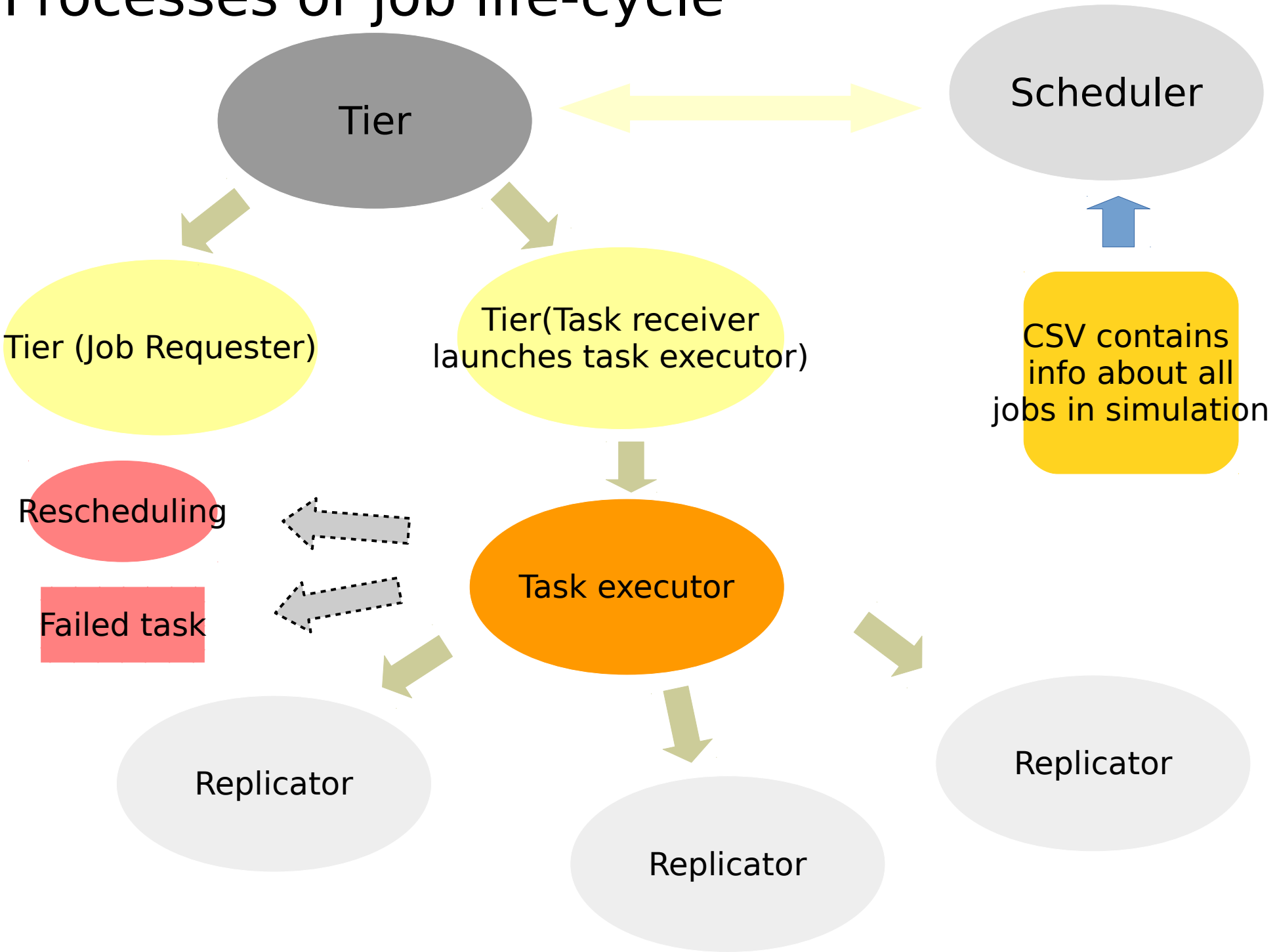
Data Availability Matching (DAM)
- It accounts for the availability of data on the requesting tier. If there are no suitable jobs DAM becomes simple algorithm

Tracing metrics

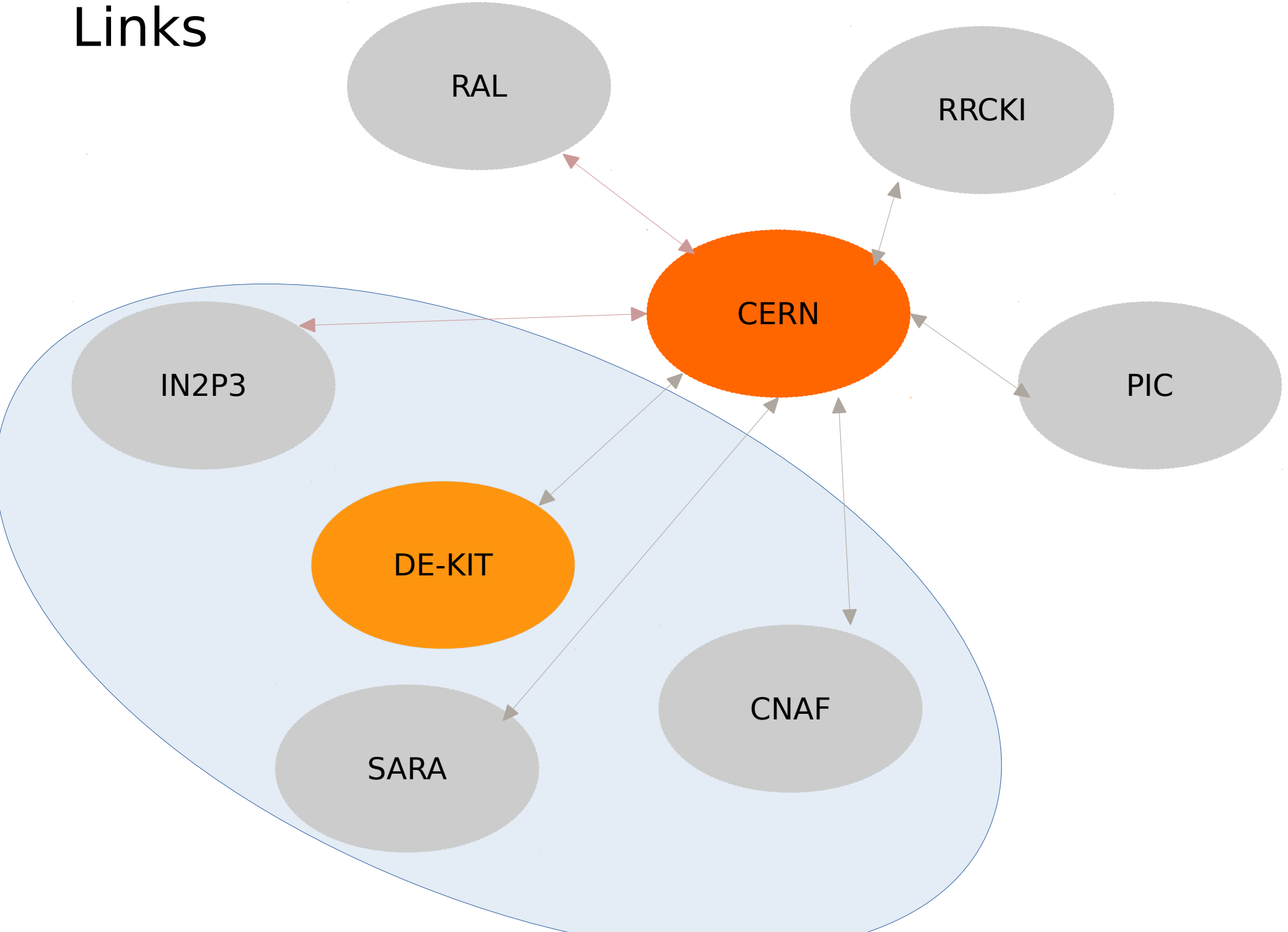
- Link workload
- Number of running cores per site
- Time of job execution
- Time of job scheduling
- Tier efficiency
- Total number of datasets on disks/tapes per site (daily)
- Total occupied space on disks/tapes per site (daily)
- Cumulative input/output data per site (daily)
- Cumulative transferred data
- Number of job failures per site
- Transfer failures per site

Backup

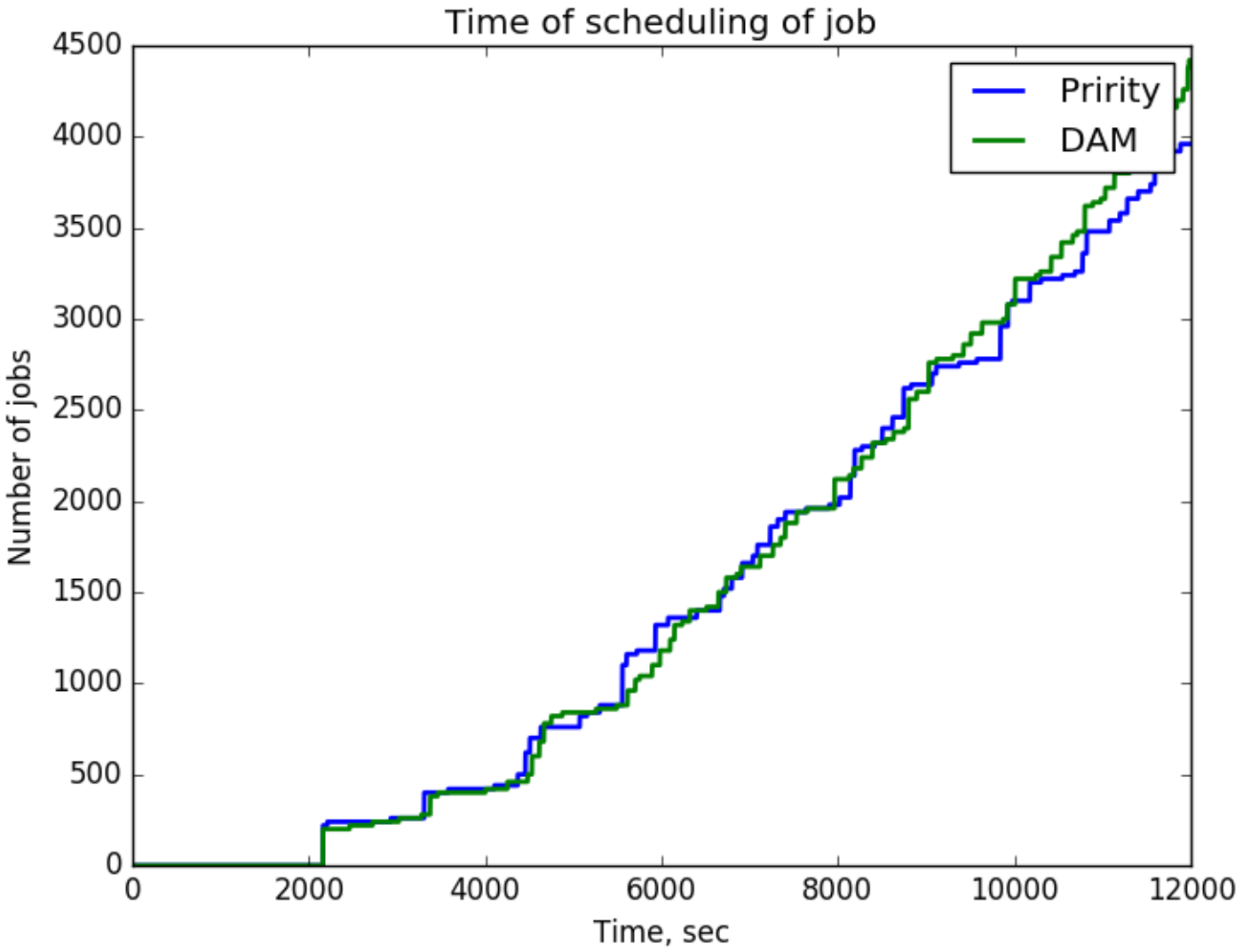
Processes or job life-cycle

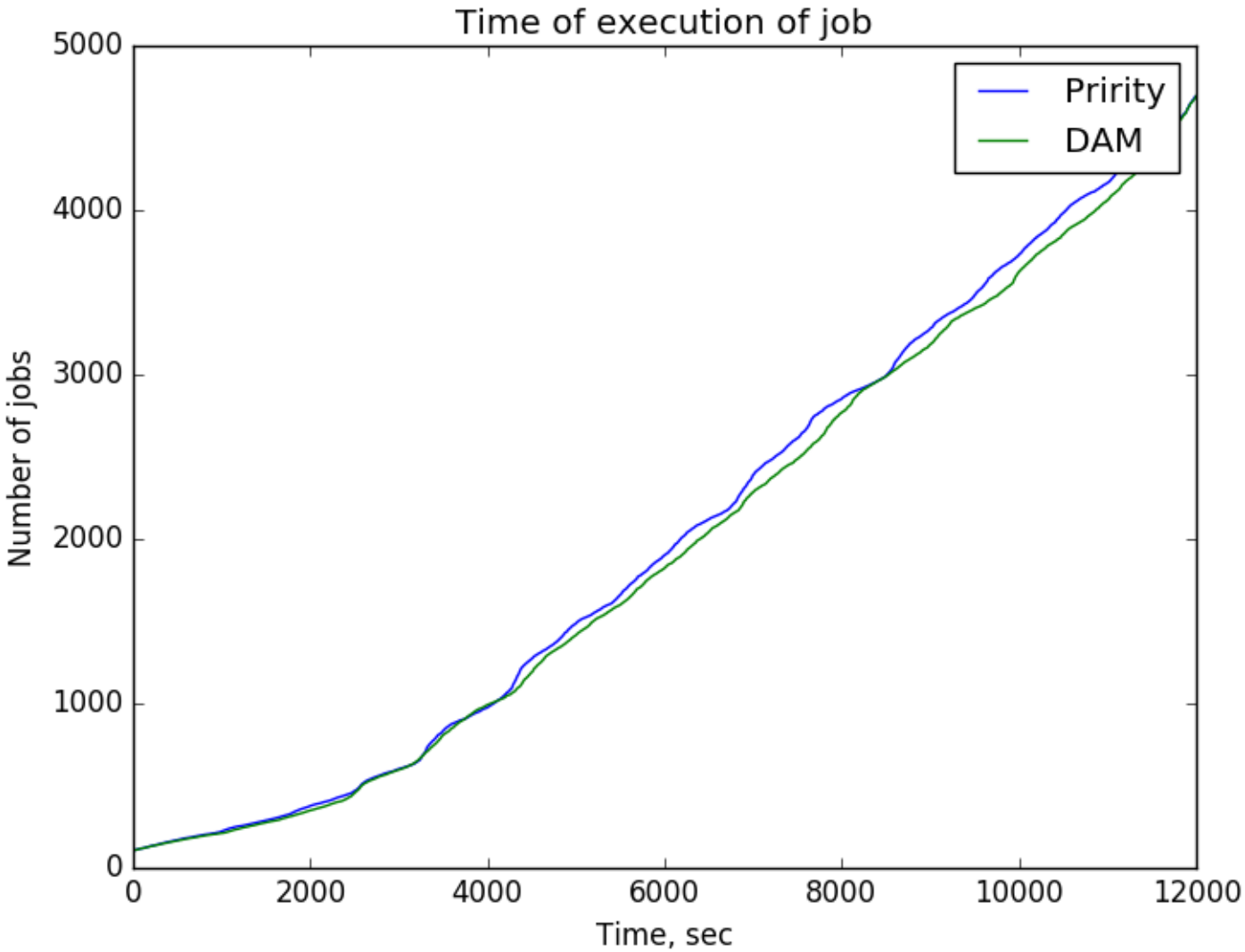


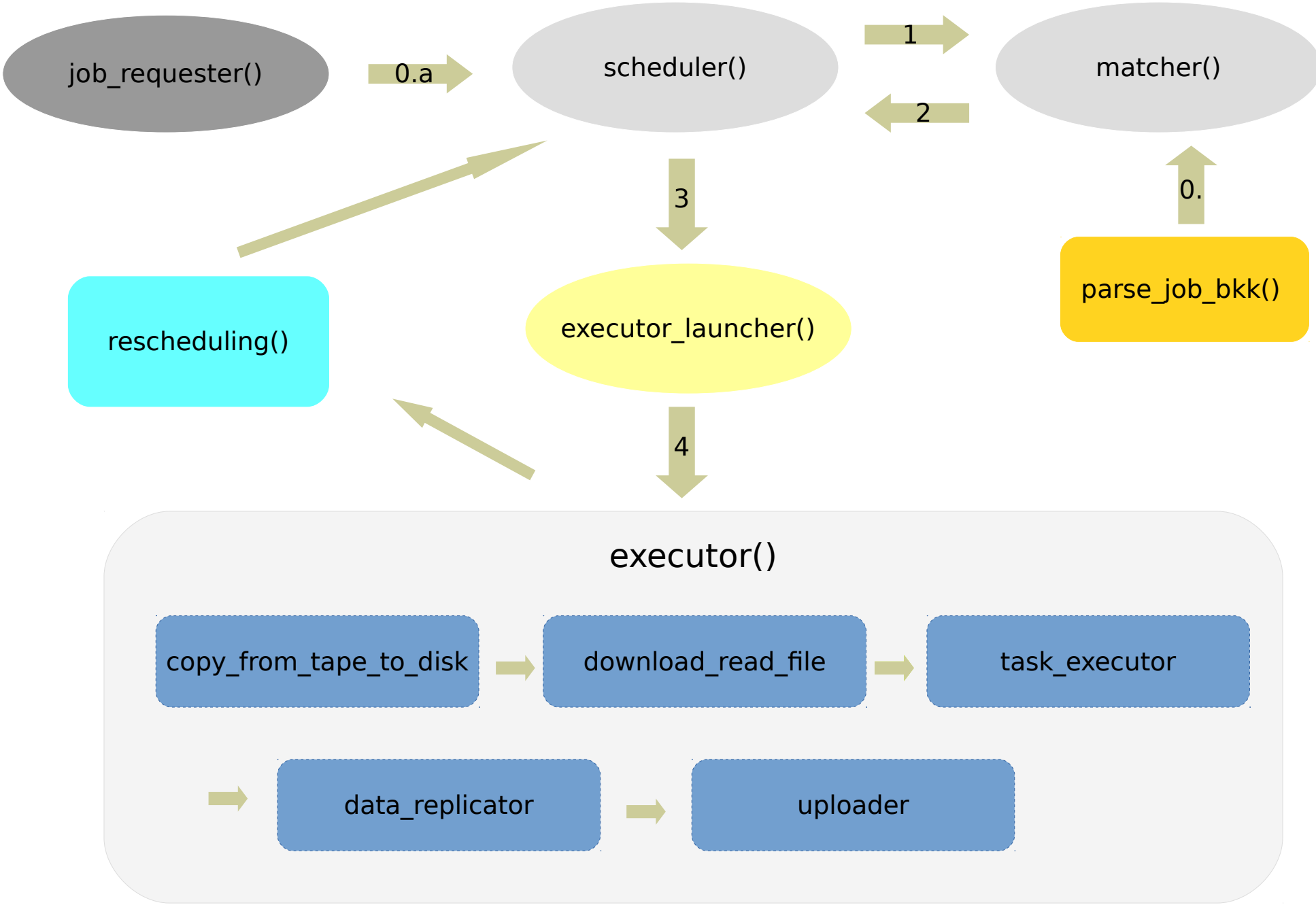
Links



Plots


Time of scheduling of job
Pririty
DAM
Number of jobs
Time, sec
0
500
1000
1500
2000
2500
3000
3500
4000
4500
0
2000
4000
6000
8000
10000
12000


Time of execution of job
Pririty
DAM
Number of jobs
Time, sec
0
1000
2000
3000
4000
5000
0
2000
4000
6000
8000
10000
12000


job_requester()
0.a
scheduler()
1
2
matcher()
0.
parse_job_bkk()
3
executor_launcher()
rescheduling()
4
executor()
copy_from_tape_to_disk
download_read_file
task_executor
data_replicator
uploader