Dynamo

Intelligent Data Management

Benedikt Maier for the Dynamo Team

Dan Abercrombie, Max Goncharov, Benedikt Maier, Sid Narayanan, Christoph Paus

March 27, 2018

First words

Dynamo is ...

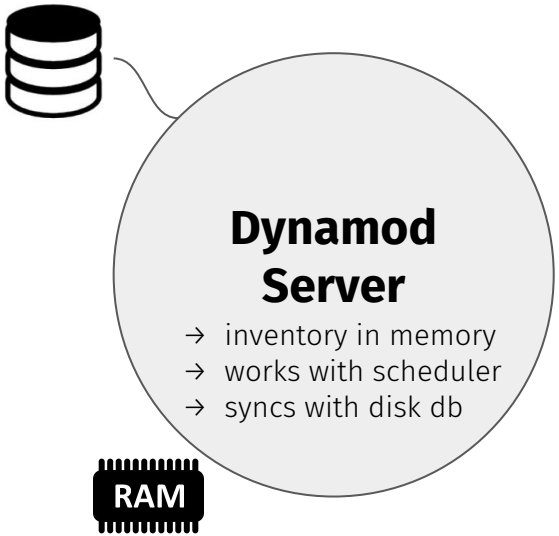
... an intelligent framework that manages data to facilitate easy access to the popular data by researchers,

... a collection of agents that can be scheduled to run in any configurable order. New agents can be easily defined and included into the framework,

... written in python,

... available at https://github.com/SmartDataProjects.

Core element: the inventory



Inventory

Information about placement of all datasets and blocks (and files) on disks and tape → many millions of objects, all sitting in memory

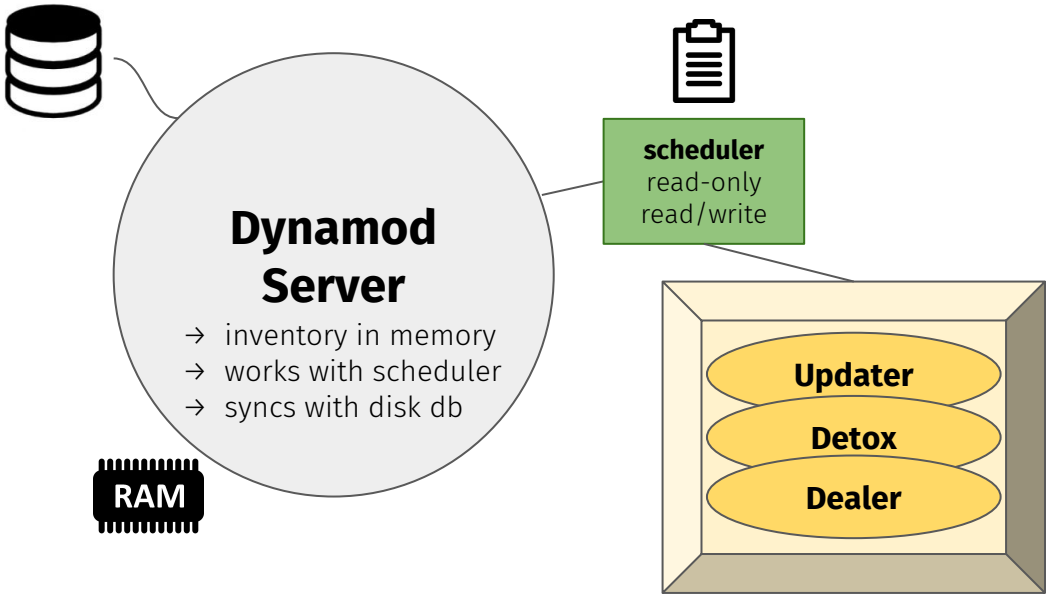
Scheduler

Handles any task that queries or wants to change inventory status (grants read/write access)

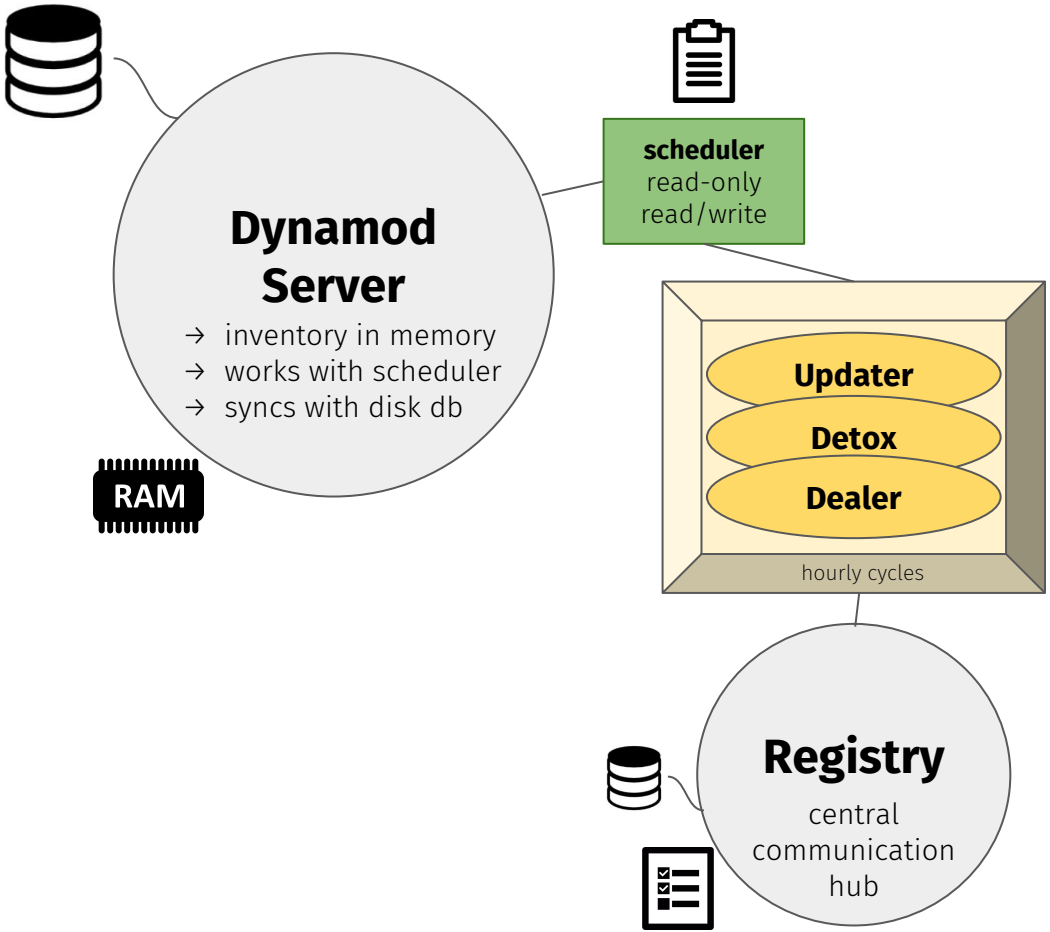
MySQL backend

Whenever a write-request changed inventory status, the change is mirrored to persistent storage db

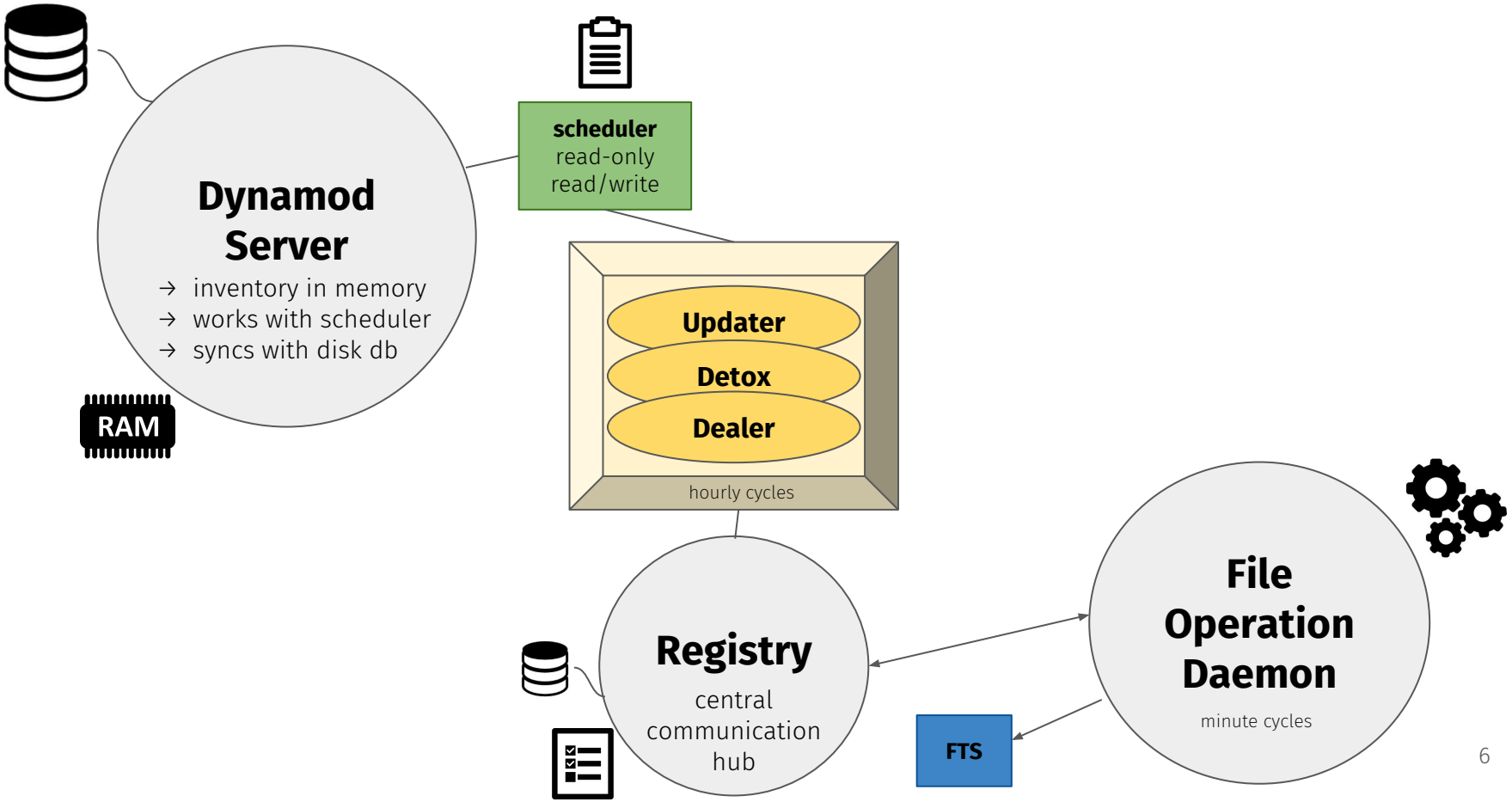
Design


Dynamod Server
→ inventory in memory
→ works with scheduler
→ syncs with disk db
RAM
scheduler
read-only
read/write
Updater
Detox
Dealer

Design


Dynamod Server
→ inventory in memory
→ works with scheduler
→ syncs with disk db
RAM
scheduler
read-only
read/write
Updater
Detox
Dealer
hourly cycles
Registry
central communication hub

Design


Dynamod Server
→ inventory in memory
→ works with scheduler
→ syncs with disk db
RAM
scheduler
read-only
read/write
Updater
Detox
Dealer
hourly cycles
Registry
central communication hub
FTS
File Operation Daemon
minute cycles

Dealer plugins

Undertaker

→ offloads data from broken/unavailable sites

Popularity

→ replicates datasets based on # user requests in global queue

Dealer

Enforcer

→ dataset replication based on custom, non-dynamic rules

Balancer

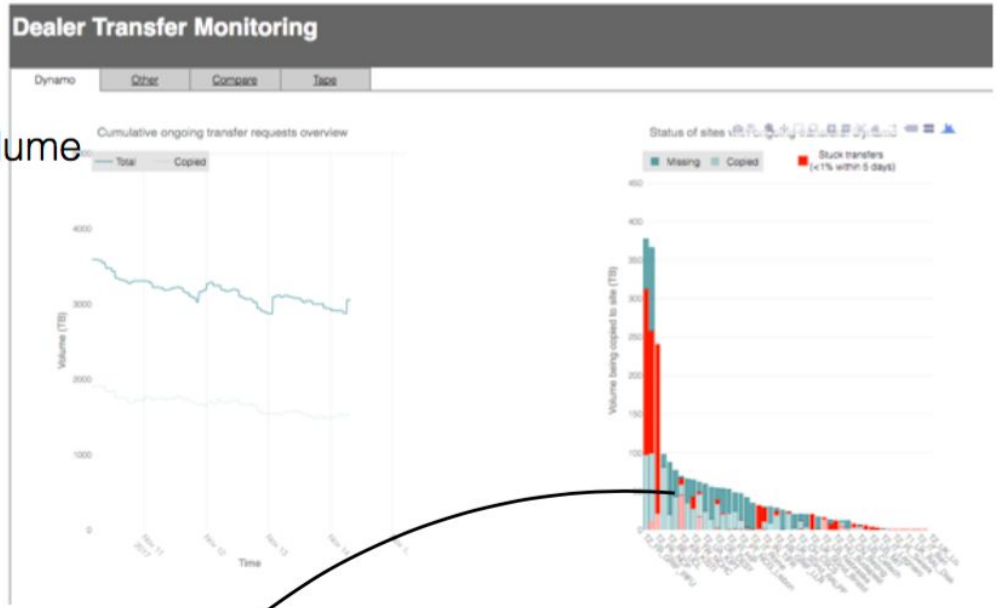
→ balances fraction of protected data across sites

These all make decisions of the type: “Make a replica of dataset X at site Y”

Dealer monitoring

http://dynamo.mit.edu/dynamo/dealermon/main.php

Aggregated copied/missing volume



Individual datasets to sites

Click on a bar

T2_FR_GRIF_IRFU

Click Request ID for corresponding PhEDEx API.
Click Replica Name for corresponding progress graph.
Red cells signify stuck transfers. (<1% copied in past 5 days)
Number of Replicas: 329

Request ID:	Replica Name:	Copied (TB):	Total (TB):
1139612	/SingleMuon/Run2017D-v1/RAW	46.7162	46.7517
1146121	/MuOnia/Run2012D-22Jan2013-v1/AOD	0	19.8384
1146121	/DoubleEG/Run2016B-18Apr2017_ver2-v1/MINIAOD	0	3.4458
1144555	/MinBias_TuneCUETP8M1_13TeV-pythia8/RunIISpring15DR74-NoPURealisticRecodebug_741_p1_mcRun2_Realistic_50ns_v0-v2/GEN-SIM-RECODEBUG	27.6836	44.6587
1144686	/WJetsToLNu_TuneCUETP8M1_13TeV-amcatnloFXFX-pythia8/RunIIFall15DR76-PU25nsData2015v1_76X_mcRun2_asymptotic_v12_ext2-v1/AODSIM	32.8776	39.491
1145655	/HIMinimumBias7/HIRun2015-02May2016-v1/AOD	16.6338	31.4865
1145755	/WJetsToLNu_Pt-600ToInf_TuneCUETP8M1_13TeV-amcatnloFXFX-pythia8/RunIISpring16MiniAODv2-PUSpring16_80X_mcRun2_asymptotic_2016_miniAODv2_v0-v1/MINIAODSIM	0	0.0325
1145755	/DMV_NNPDF30_Axial_Mphi-2500_Mchi-1_gSM-0p25_gDM-1p0_v2_13TeV-powheg/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0.0831	0.0942
1145755	/VBFHToTauTau_M125_13TeV_powheg_pythia8_PythiaFragment_Up/RunIISpring16MiniAODv2-PUSpring16RAWAODSIM_reHLT_80X_mcRun2_asymptotic_v14-v1/MINIAODSIM	0.0033	0.0432
1145755	/QstarToJJ_M_9000_TuneCUETP8M1_13TeV_pythia8/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0.0262	0.0474
1145755	/VBFHHTo2B2G_CV_1_C2V_1_C3_0_13TeV-madgraph/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0.0062	0.0342
1145755	/SeesawTypeIII_SIGMAplusSIGMAminusWW_M-780_13TeV-madgraph-pythia8/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0.002	0.0311
1145755	/SingleMuon/Run2016C-MuAlOverlaps-07Aug17-v1/ALCARECO	0.0534	0.249
1145755	/WJetsToLNu_TuneCUETP8M1_13TeV-madgraphMLM-pythia8/RunIISummer16MiniAODv2-FlatPU28to62HcalNZSRAW_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM	0.0139	0.045
1145755	/HeavyNeutrino_trilepton_M-10_V-0p01_2l_NLO/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0.0005	0.0287
1145755	/WprimeToTB_plusSM_TToLep_M-3100_LH_TuneCUETP8M1_13TeV-comphep-pythia8/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0	0.0628
1145755	/TTGamma_SingleLeptFromT_TuneCUETP8M2T4_13TeV-amcatnlo-fsrdown-pythia8-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v2/MINIAODSIM	0.041	0.2239
1145755	/SinglePhoton_Pt-200/PhaseIITDRSpring17DR-PU140CaloAging4500Ultimate_91X_upgrade2023_realistic_v3-v2/GEN-SIM-RECO	0.0351	0.2197
1145755	/ZH_HToZG_ZToAll_M120_13TeV_powheg_pythia8/RunIISummer16DR80Premix-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/AODSIM	0	0.0317

Enforcer

http://dynamo.mit.edu/dynamo/dealermon/enforcer.php

→ dataset replication based on custom, non-dynamic rules

e.g. US-CMS decision to have one replica of widely used datasets at US sites

Configurable via JSON


```
"usa_miniaod": {
    "datasets": ["/*/*/MINIAOD*"],
    "num_copies": 1,
    "sites": ["T1_US_FNAL_Disk","T2_US_*"]
  }
```

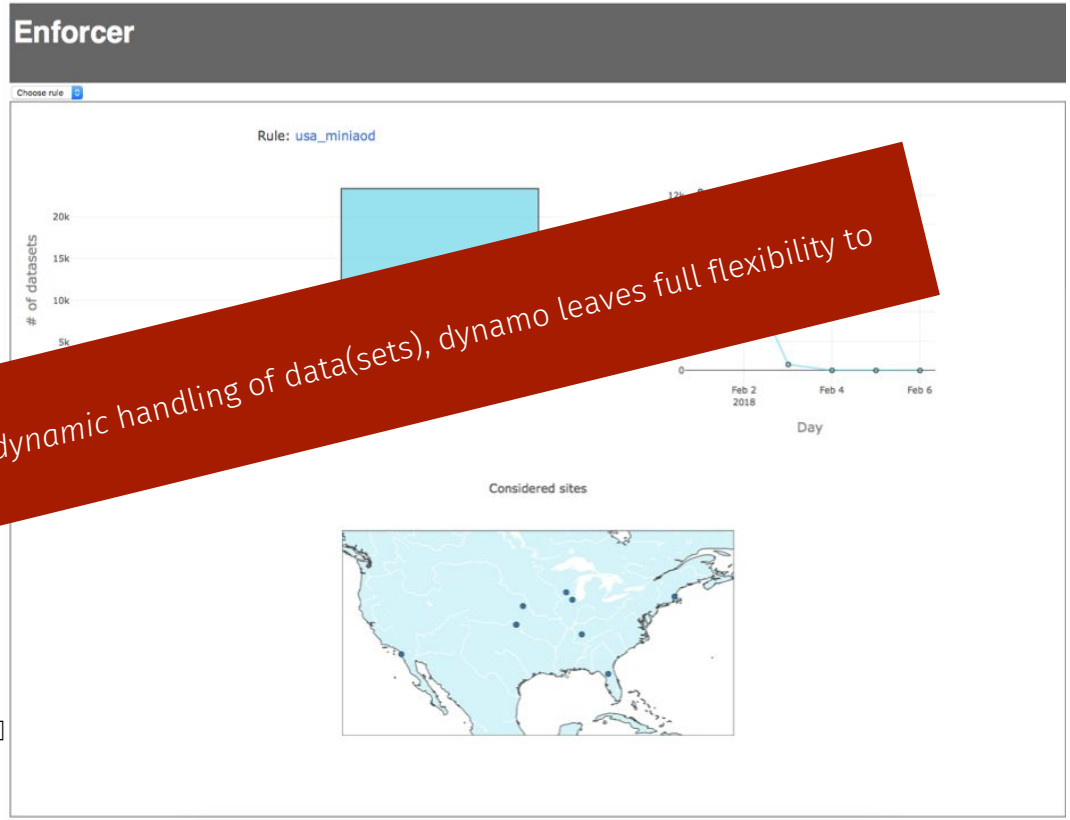
Enforcer

→ dataset replication based on custom, non-dynamic rules

e.g. US-CMS decision to have one replica of widely used datasets at US sites

Configurable via JSON

```
"us                : {
    datasets": ["/*/*/MINIAOD*"],
    "num_copies": 1,
    "sites": ["T1_US_FNAL_Disk","T2_US_*"]
  }
```

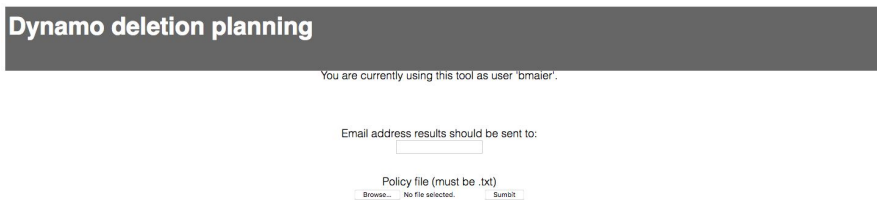


Take-away message:
Despite the underlying paradigm being the dynamic handling of data(sets), dynamo leaves full flexibility to impose custom rules.


```
### Deletion trigger
When site.occupancy > 0.9
Until site.occupancy < 0.85

### Replica protection / deletion policies
Ignore site.name == T*_MSS

ProtectBlock blockreplica.is_locked
...
ProtectBlock blockreplica.is_last_transfer_source
...
Dismiss dataset.usage_rank > 200
...
# Default decision
Dismiss
```



Detox

→ Everything is based on a human-readable policy file. In principle, anything that is not explicitly protected can be deleted.
→ Virtually any DatasetReplica/BlockReplica attribute can be defined/added and used to manage deletions.
→ On top of dynamic deletions, policy files for dedicated deletion campaigns can easily be spelled out (and tested).
→ The entire history of deletions is automatically saved so we can always say why something got deleted where and when.

Design

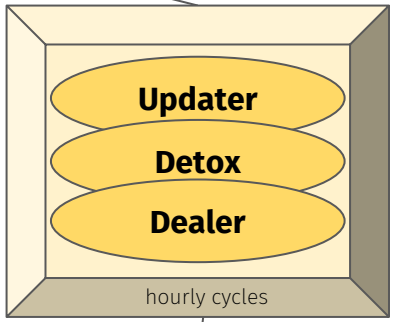
script upload
web interface

command line
analysis

scheduler
read-only
read/write

Dynamod Server

- → inventory in memory
- → works with scheduler
- → syncs with disk db



monitoring

site consistency

Registry

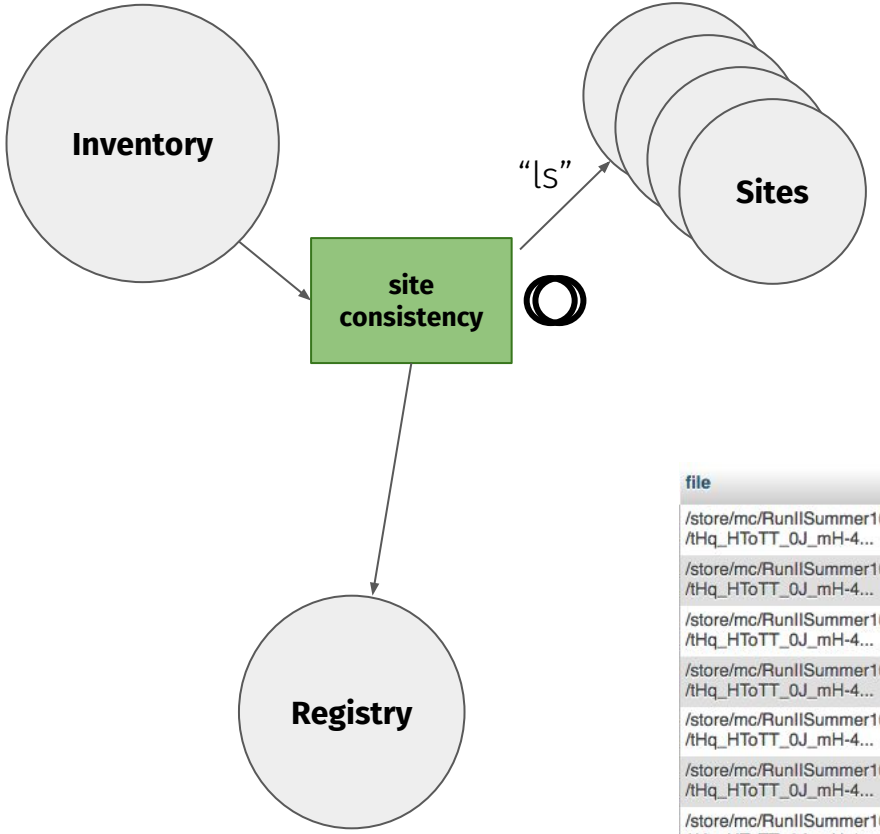
central
communication
hub

File Operation Daemon

minute cycles

FTS

Site consistency



Aligns what should be on the site with what *is* on the sites.

(Before, this was done not very frequently for T0/T1, and only occasionally for T2s)

Important to avoid stuck transfers, problems with user access, or wasted space.

→ catalog obtained from inventory
→ comparing this against results from remotely probing site content via xrdfs (10min - 2hrs per site)

→ Ingests "missing" files to be transferred and "orphan" files to be deleted into registry, also publishes ASCII files in web

file	site_from	site_to	status	created
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T2_FI_HIP	T1_UK_RAL_Disk	new	2018-02-14 22:06:37
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T1_US_FNAL_Disk	T1_UK_RAL_Disk	new	2018-02-14 22:06:37
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T1_FR_CCIN2P3_Disk	T1_UK_RAL_Disk	new	2018-02-14 22:06:37
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T2_UA_KIPT	T1_UK_RAL_Disk	new	2018-02-14 22:06:37
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T2_FI_HIP	T1_UK_RAL_Disk	new	2018-02-14 22:06:37
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T1_US_FNAL_Disk	T1_UK_RAL_Disk	new	2018-02-14 22:06:37
/store/mc/RunIISummer16MiniAODv2 /tHq_HToTT_0J_mH-4...	T1_FR_CCIN2P3_Disk	T1_UK_RAL_Disk	new	2018-02-14 22:06:37

Site consistency

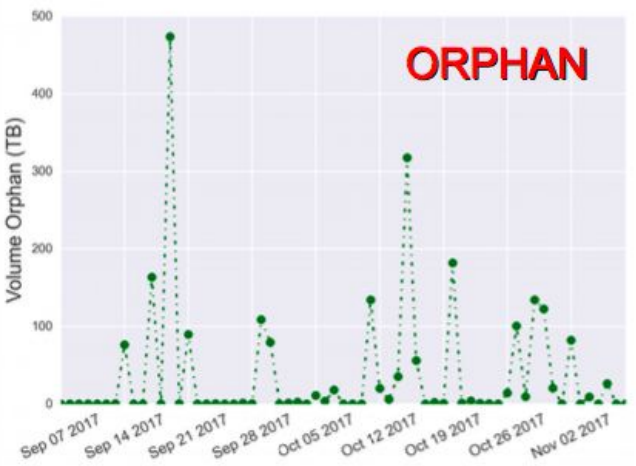
In the roll-out phase last fall, we

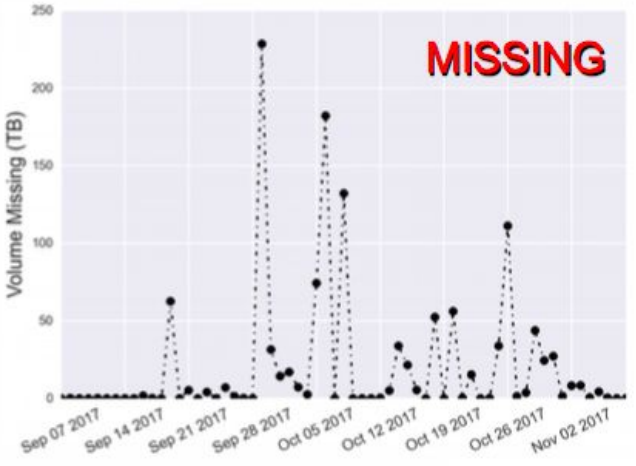
→ deleted O(1 PB) orphan files
→ recovered O(1 PB) missing files
→ removed several million of empty directories that previous deletion processes left behind

Spikes in the right plots appear when we probed a new site for the first time.

Now, orphan/missing files appear occasionally.

Constitutes the very first use case for file transfers fully operated by dynamo.





Consistency Check Summary

Debugged sites	Sites that need debugged or hand checking	All sites

Last Update	Site Name	Time [min]	Number Files	Number Nodes	Unlisted	Empty Nodes	Num Missing	No Source	Size [TB]	Num Orphan	Size [TB]
2018-03-14 22:24:29	T2_TW_NCHC	53	128536	18866	0	10	0 [blocks]	0	0.00	0	0.00
2018-03-14 21:39:20	T2_FR_GRIF_IRFU	68	354694	29003	0	2804	0 [blocks]	0	0.00	0	0.00
2018-03-14 20:49:20	T1_DE_KIT_Disk	47	705434	39063	0	26	0 [blocks]	0	0.00	0	0.00
2018-03-14 19:39:17	T2_AT_Vienna	8	88548	1031	0	13	0 [blocks]	0	0.00	0	0.00
2018-03-14 17:53:17	T2_IT_Rome	22	171489	9468	0	478	4559 [blocks]	1746	12.88	0	0.00
2018-03-14 17:51:59	T2_US_Purdue	81	618567	84345	0	7734	0 [blocks]	0	0.00	0	0.00
2018-03-14 17:07:41	T2_DE_DESY	96	1349167	73689	0	4743	0 [blocks]	0	0.00	0	0.00
2018-03-14 14:04:31	T2_BR_SPRACE	33	273535	22082	0	3590	0 [blocks]	0	0.00	0	0.00
2018-03-14 12:50:56	T2_UK_SGrid_Bristol	20	109198	8117	0	541	0 [blocks]	0	0.00	0	0.00
2018-03-14 11:46:31	T2_PK_NCP	15	17592	1964	0	23	0 [blocks]	0	0.00	1966	2.40
2018-03-14 10:43:04	T2_FR_CCIN2P3	12	97088	7140	0	219	0 [blocks]	0	0.00	0	0.00
2018-03-14 10:39:47	T2_PL_Warsaw	68	85320	19548	0	1798	16464 [blocks]	2039	51.57	0	0.00
2018-03-14 07:55:46	T2_UK_SGrid_RALPP	24	201675	21943	0	1487	0 [blocks]	0	0.00	0	0.00
2018-03-14 07:02:19	T2_US_Florida	31	636861	62861	0	4454	2 [blocks]	2	0.01	0	0.00
2018-03-14 06:11:53	T2_FR_GRIF_LLR	41	228493	22369	0	4695	10 [blocks]	10	0.02	1	0.00

File operations

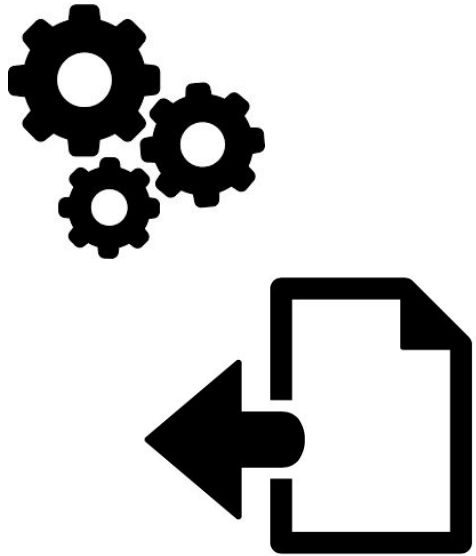
→ Module developed in tandem with Site Consistency

→ Deletion: immediate action with gfal-rm

→ Transfer: form FTS job and submit to CERN FTS

→ Disk-to-disk transfers available since last year

→ Functionality to stage from tape established last month



Design

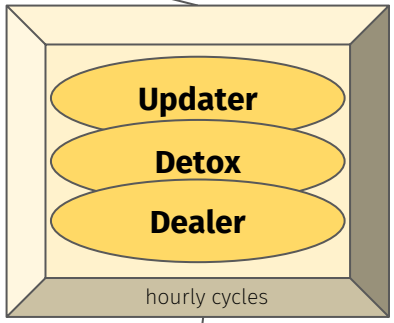
script upload
web interface

command line
analysis

Dynamod Server

- → inventory in memory
- → works with scheduler
- → synchs with disk db

scheduler
read-only
read/write



register data
(production)

monitoring

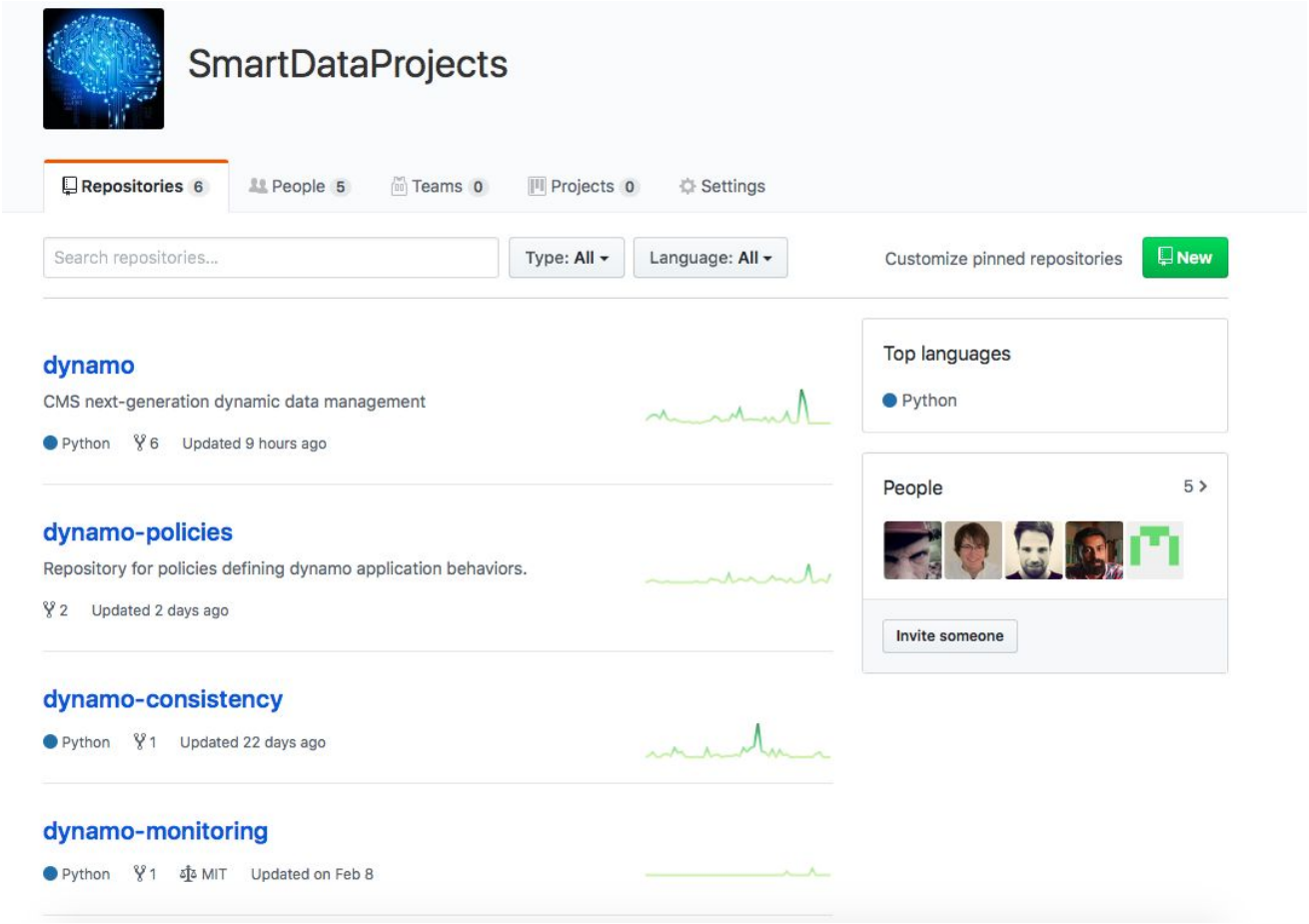
site consistency

Registry
central
communication
hub

File Operation Daemon
minute cycles

FTS

Dynamo ecosystem



Ongoing work & next steps

→ Porting dynamo from bash-script-philosophy to server mode that is able to handle tasks asynchronously was the big achievement late last year.

→ Much more stable

→ Much faster

→ Much easier to interface dynamo with other services

→ Implementation of REST APIs #1 priority to query inventory status and allow any tool to “dock on”

→ Phasing out PhEDex as underlying tool for handling transfers & deletions

→ Working on documentation to make it easy for other experiments to install and use dynamo

→ Continuous integration and unit testing will come soon (right now development model consists of developing and testing on dedicated server and git-reviewing among collaborators)