Status of GoeGrid and the local monitoring system in ATLAS

Gen Kawamura, **Stan Lai, Arnulf Quadt**

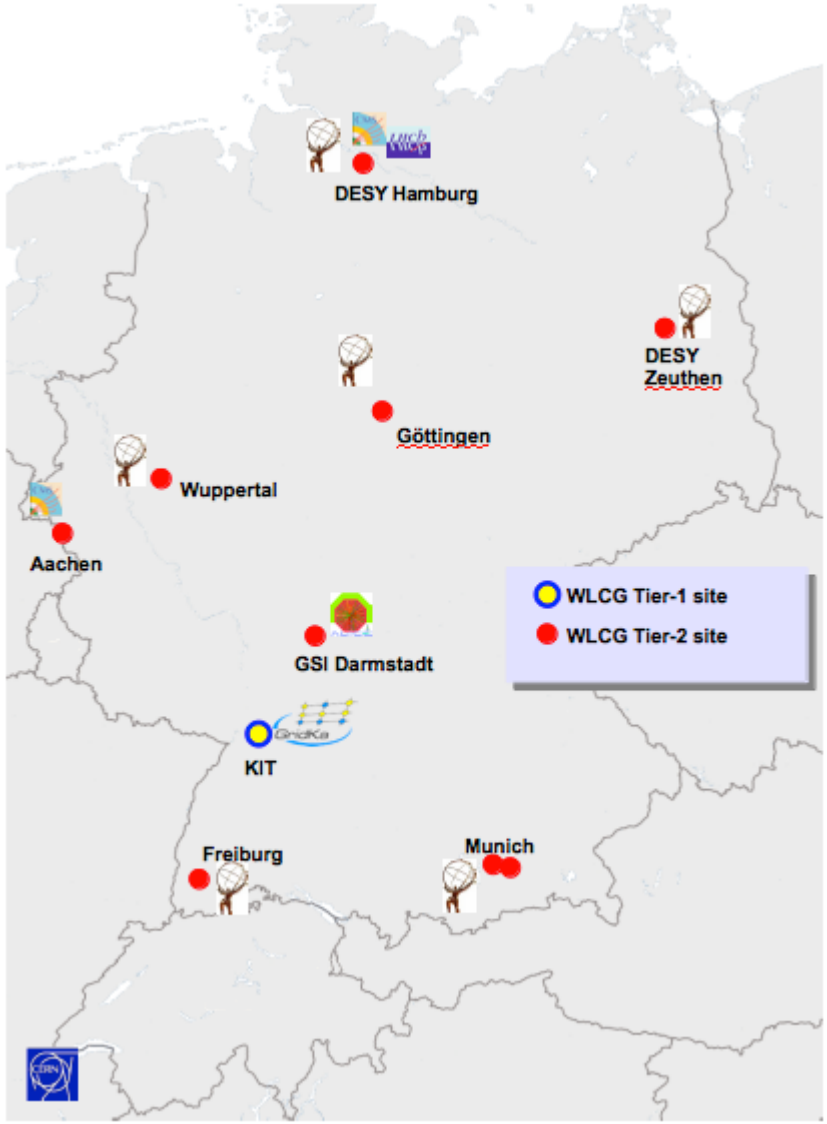
II. Physikalisches Institut, Georg-August-Universität Göttingen

Outline

- Status of GoeGrid Tier-2 site in ATLAS
  - GoeGrid Tier-2 in ATLAS
- Monitoring infrastructure for ATLAS, Tier-1 and GoeGrid

ATLAS-D and GoeGrid

- Our workhoses in ATLAS-D
  - Tier-1: GridKa
  - Tier-2s: U-Göttingen, U-Freiburg, U-München, U-Wuppertal, DESY-HN, DESY ZN, MPI München
  - NAF: DESY



GoeGrid Tier-2 in ATLAS

- CPU job slots 6000, Storage total 3PB
  - For ATLAS in Oct. 2019
    - Pledged / Nominal CPUs: 21k / 47k (HEP-SPEC06)
    - Pledged / Nominal Disks: 1.8 / 2.35 (PB)
- Hardware finance: 500k EUR in 2015-2019
  - Planned: BMBF 150k EUR (2020), Theory 100k (2019)
  - Operated by 2nd Physics (ATLAS)
  - Located at GWDG

GWDG
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

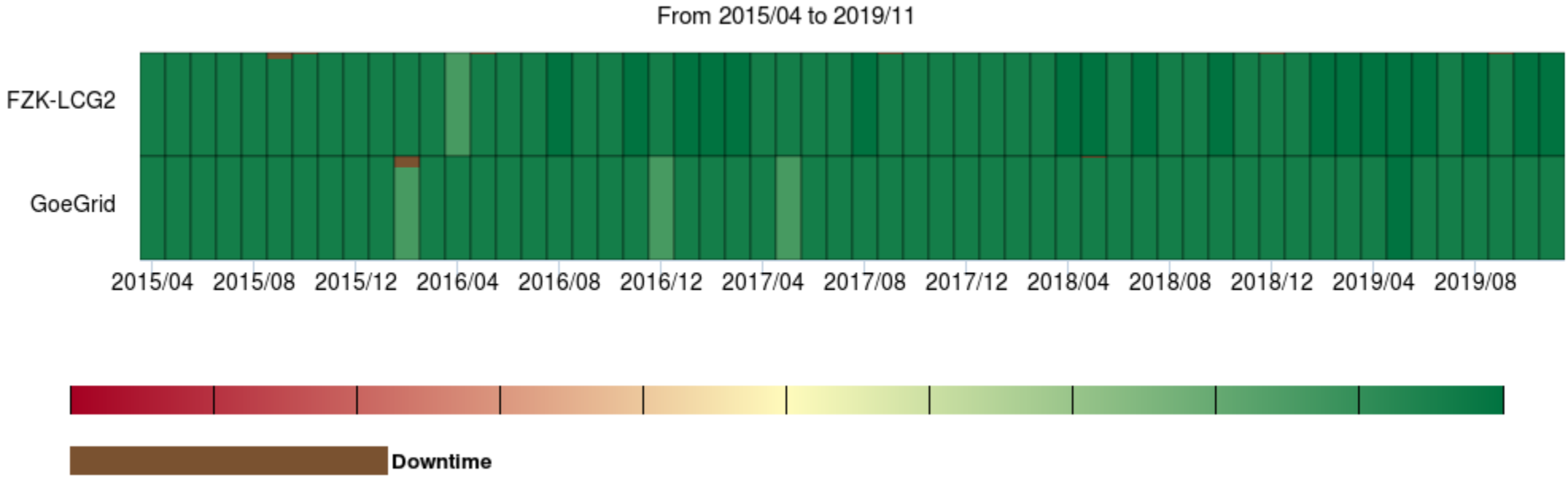
GoeGrid Tier-2 in ATLAS

- GWDG
  - A computing facility providing electricity, network and floor
- Dr. Jürgen Holm
  - Grid computing for faculty

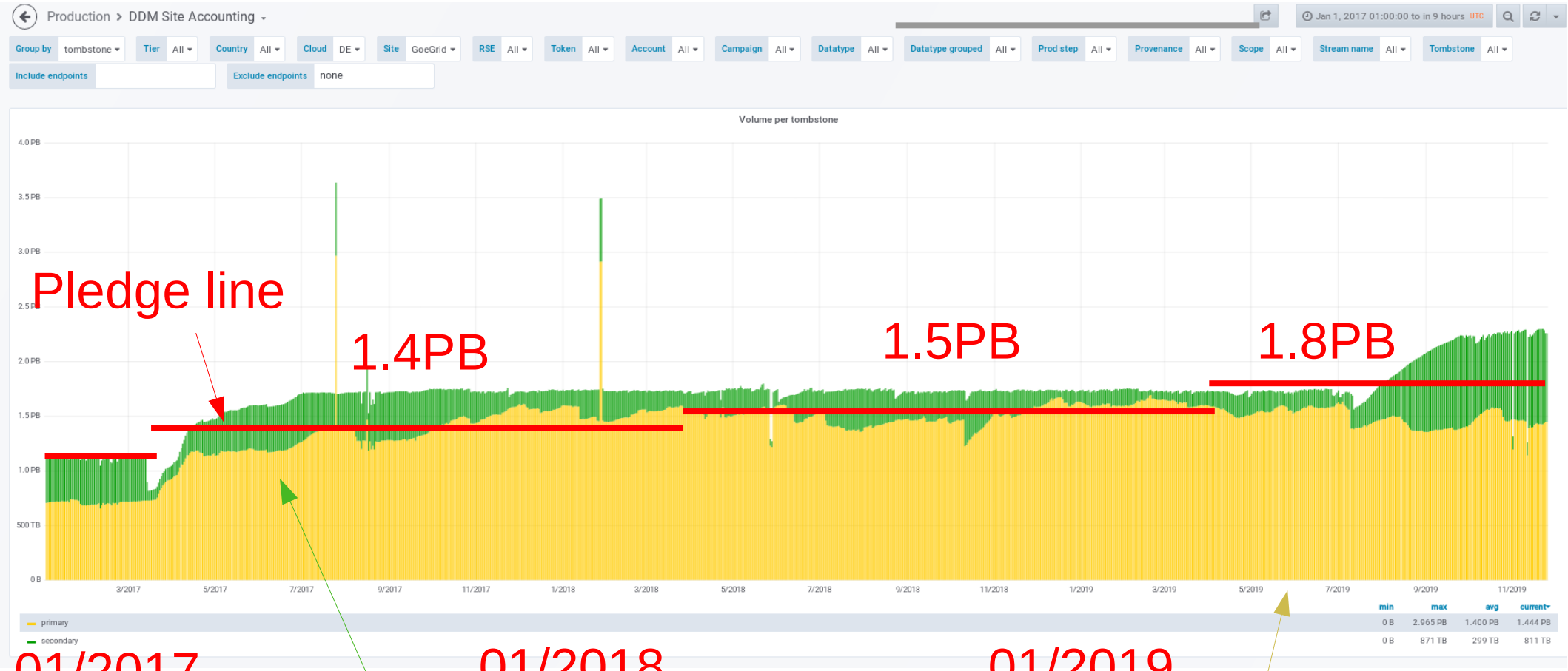
- Tim Ehlers@GWDG
  - Responsible for hardware issues
- Dr. Gen Kawamura
  - Grid services, dCache, ATLAS tasks

GoeGrid Tier-2 in ATLAS

Stable run in Run-2 and Long Shutdown 2 (LS-2) (Apr.2015 – Nov.2019), availability continuously above 95%



Storage usage detailed: Jan.2017 – Nov.2019 (available only after Jan.2017)



Pledge line

1.4PB

1.5PB

1.8PB

01/2017

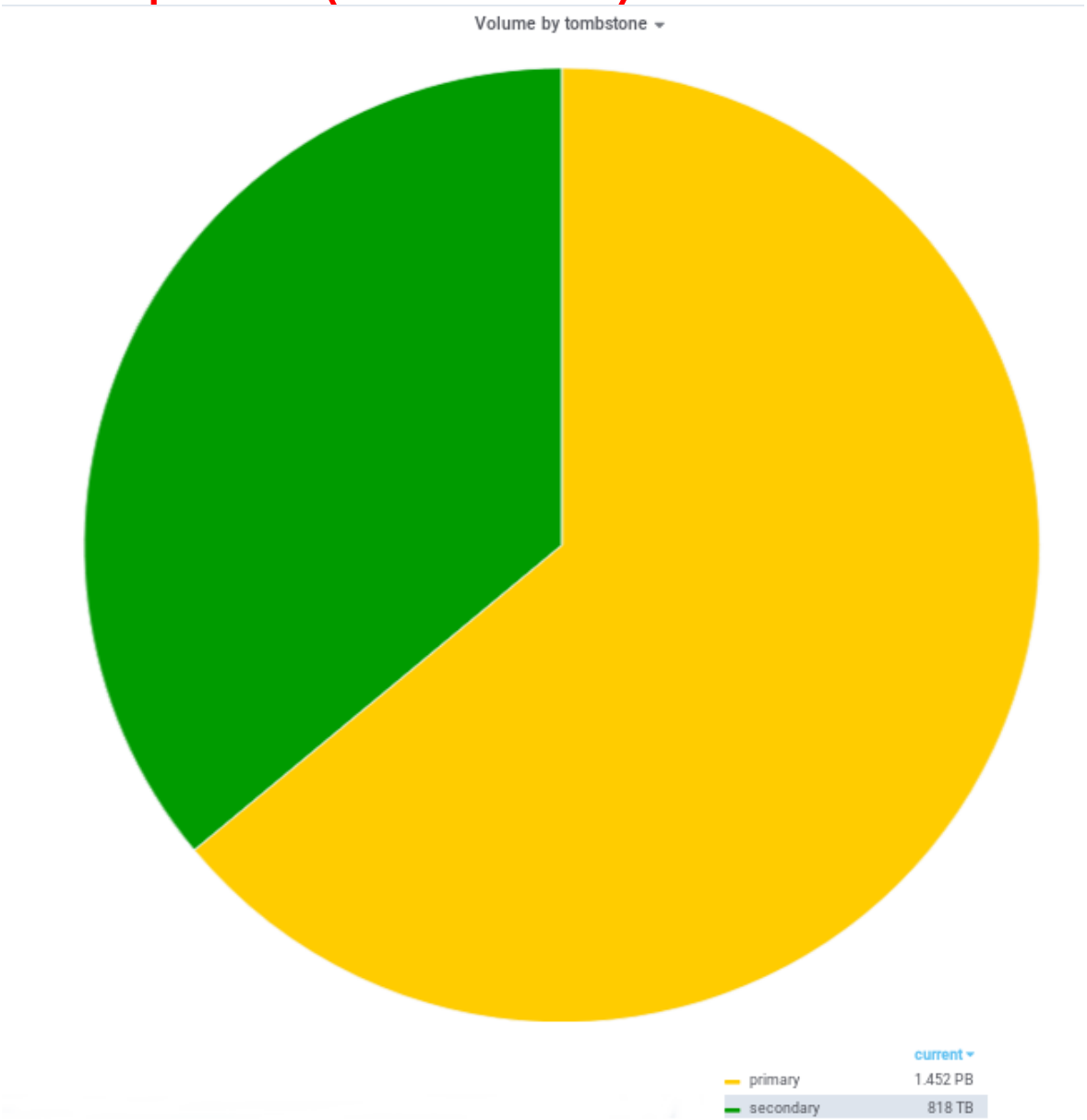
01/2018

01/2019

Primary data

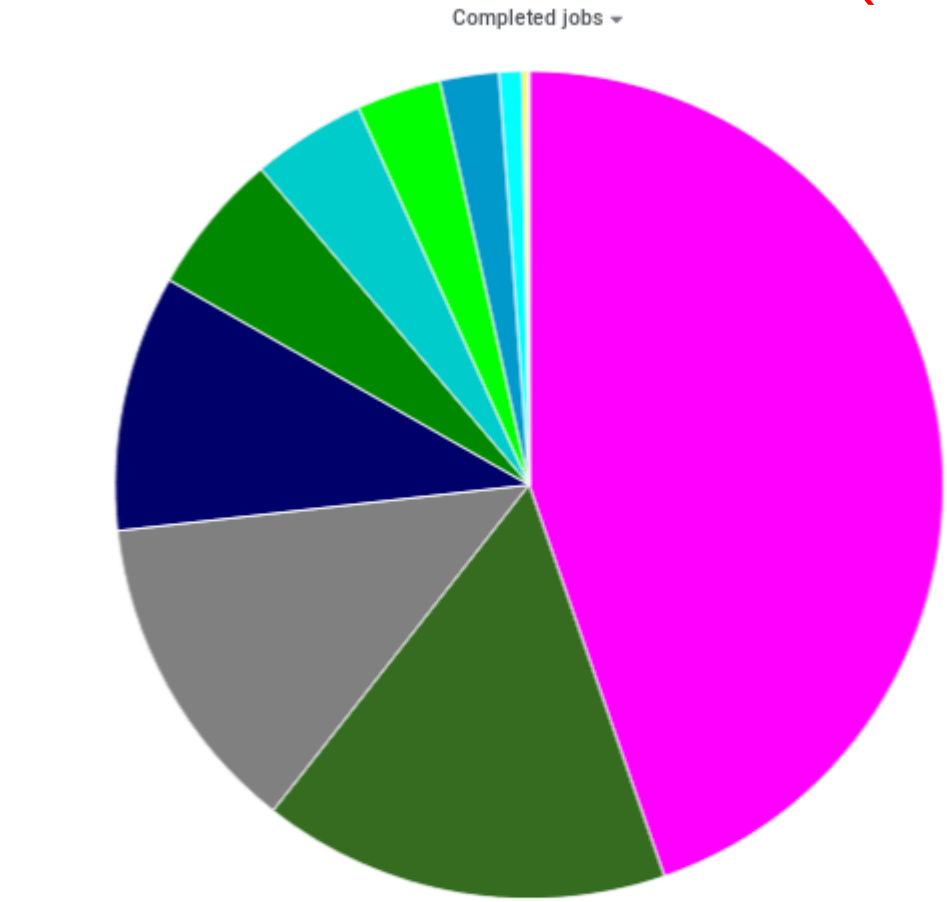
Secondary data

Storage usage snapshot (Nov.2019)



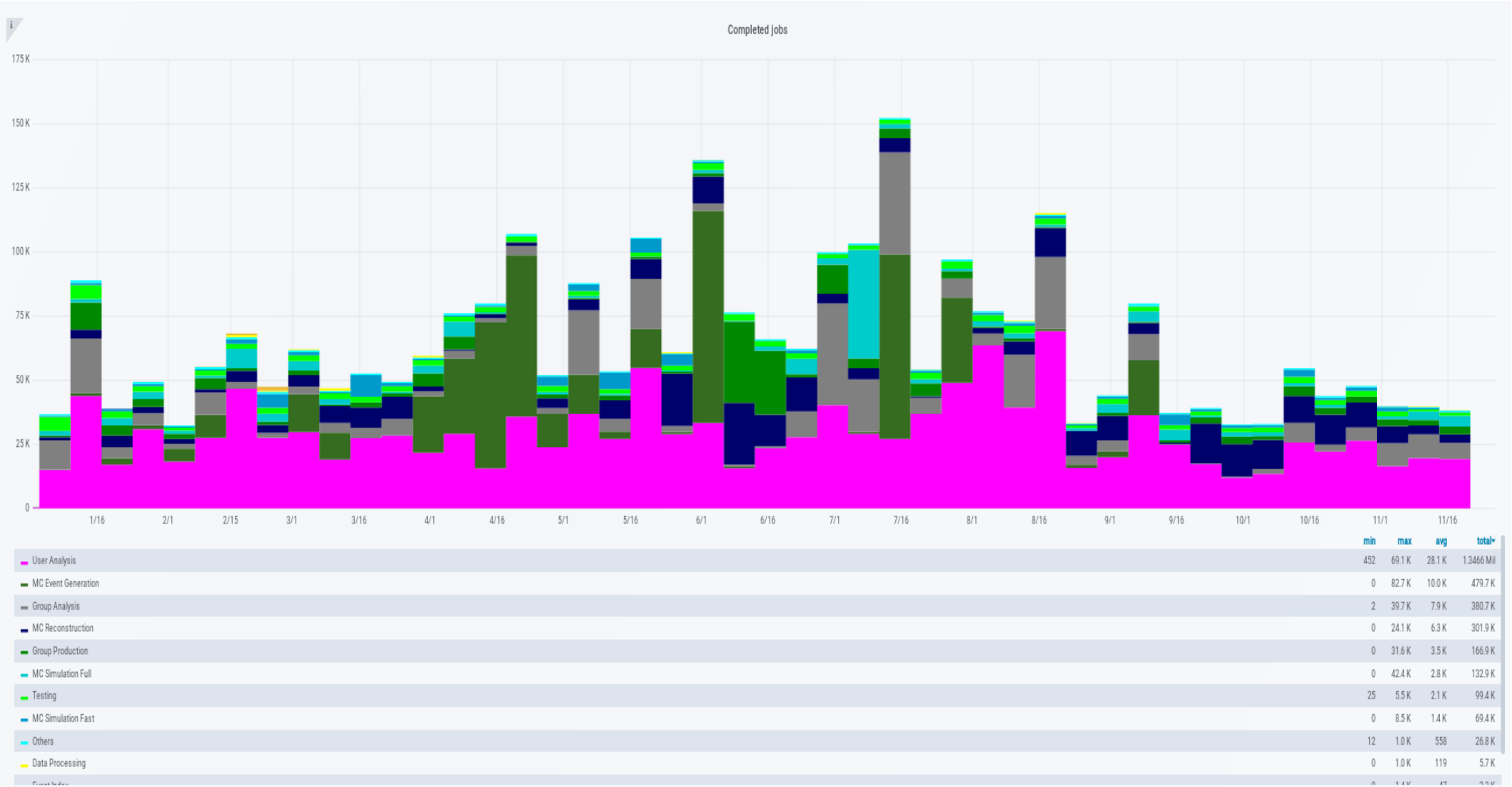
GoeGrid Tier-2 in ATLAS

Job type detailed: Jan.2019 – Nov.2019 (available only after Jan.2019)



	current	percentage
User Analysis	1.349 Mil	45%
MC Event Generation	480 K	16%
Group Analysis	381 K	13%
MC Reconstruction	302 K	10%
Group Production	168 K	6%
MC Simulation Full	133.9 K	4%
Testing	99.5 K	3%
MC Simulation Fast	69.4 K	2%
Others	26.8 K	1%

Job type detailed: Jan.2019 – Nov.2019 (available only after Jan.2019)



	min	max	avg	total
User Analysis	452	69.1 K	28.1 K	1.3466 Mil
MC Event Generation	0	82.7 K	10.0 K	479.7 K
Group Analysis	2	39.7 K	7.9 K	380.7 K
MC Reconstruction	0	24.1 K	6.3 K	301.9 K
Group Production	0	31.6 K	3.5 K	166.9 K
MC Simulation Full	0	42.4 K	2.8 K	132.9 K
Testing	25	5.5 K	2.1 K	99.4 K
MC Simulation Fast	0	8.5 K	1.4 K	69.4 K
Others	12	1.0 K	558	26.8 K
Data Processing	0	1.0 K	119	5.7 K

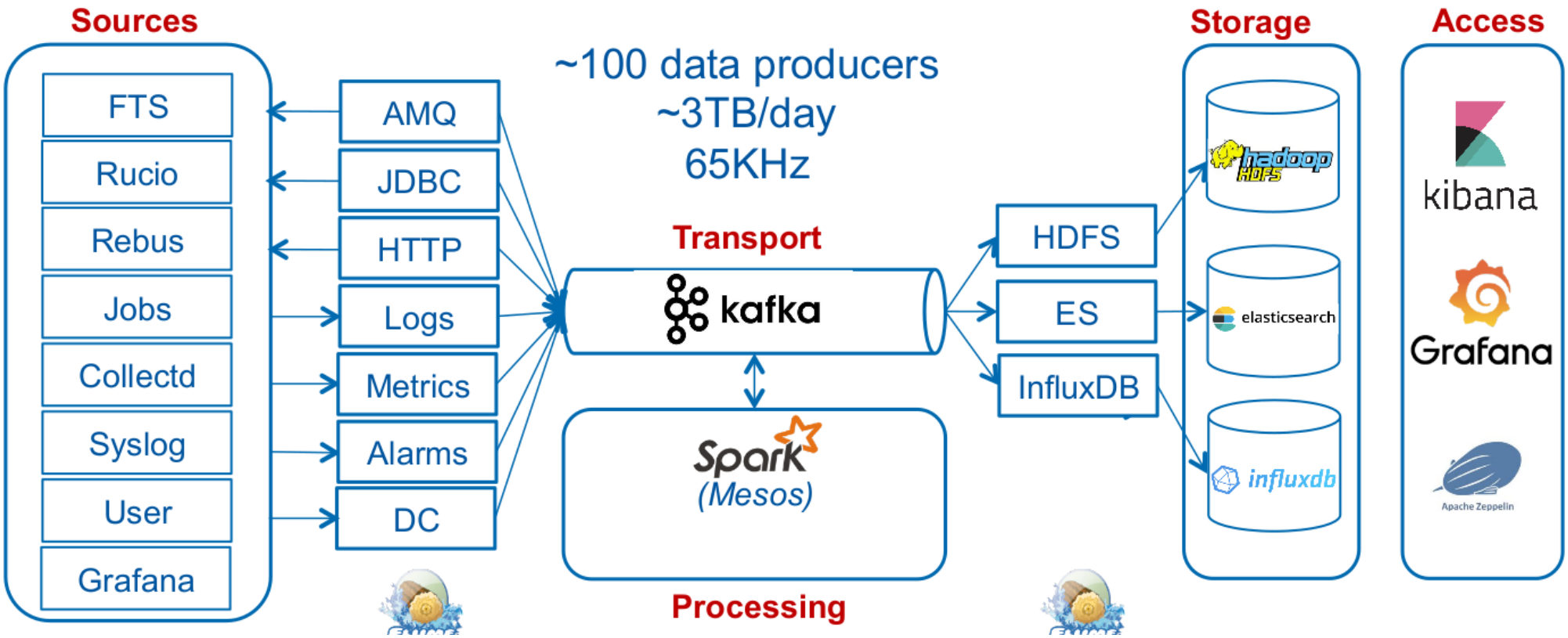
Operation of GoeGrid

- Availability continuously above 95%
  - Stable operation in ATLAS, by ATLAS group in 2nd Physics Institute
  - Successful run in Run-2 and LS-2
  - Above pledged CPU/Storage
  - Migrated CentOS7 WNs with Singularity
  - Ongoing migration: HTCondor and ARC
- Present Middleware: dCache, CREAM-CEs, monitored by HappyFace mobile application and Kibana Elasticsearch framework

Motivation of monitoring/analytics

- Motivation
  - ATLAS central monitoring/Analytics provides overviews of DAQ/Computing on Kibana/Grafana
  - Replacing SSB, DDM, Job, Accounting Dashboards of central monitoring systems
  - Tier-1 FZK monitor jobs/transfers on Grafana
- For each relatively small computing site such as GoeGrid
  - Templating standard monitoring system in DE, e.g. for dCache + HTCondor sites
  - Local logs of a site → sensitive information
  - Use-case scenarios
    - Performance tuning, trouble shooting, error detection, many alarms

Monitoring Infrastructure



- A large infrastructure at CERN
- Alarms service developed at CERN
- Dashboards for Control rooms and experts in ATLAS

Why Grafana?

- Great community support
- Mixed data sources
- User delegated control
- Templating
- Nice UI
- Easy to extend

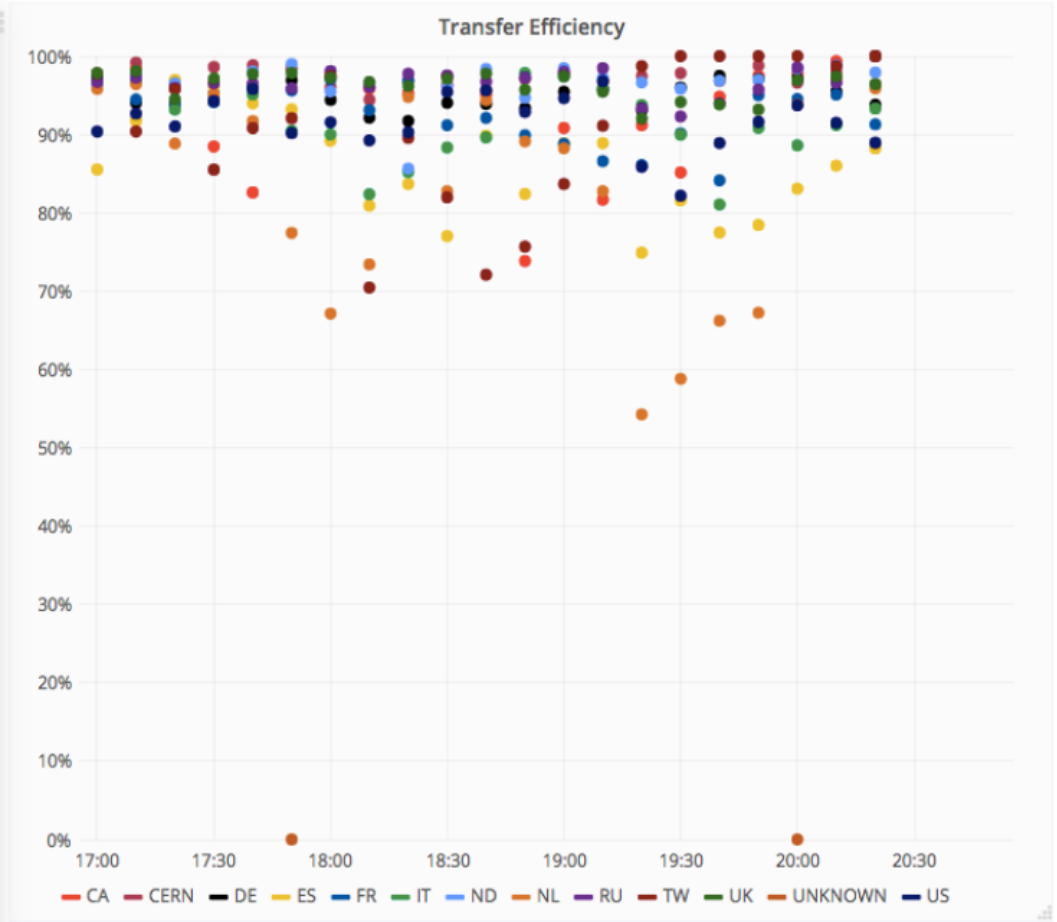
Central Monitoring of ATLAS Computing

Group by: dst_cloud | Binning: auto

Activity: Analysis Input + Data Brokering + Data Consolidation + Data Rebalancing + Deletion + Express + Functional Test + Production Input + Production Output + Recovery + T0 Export + T0 Tape + User Subscriptions + default + Staging

Source country: All | Source site: All | Destination country: All | Destination site: All | Filters: + | Matrix Columns: dst_cloud | Matrix Rows: src_cloud

Transfers



Efficiency

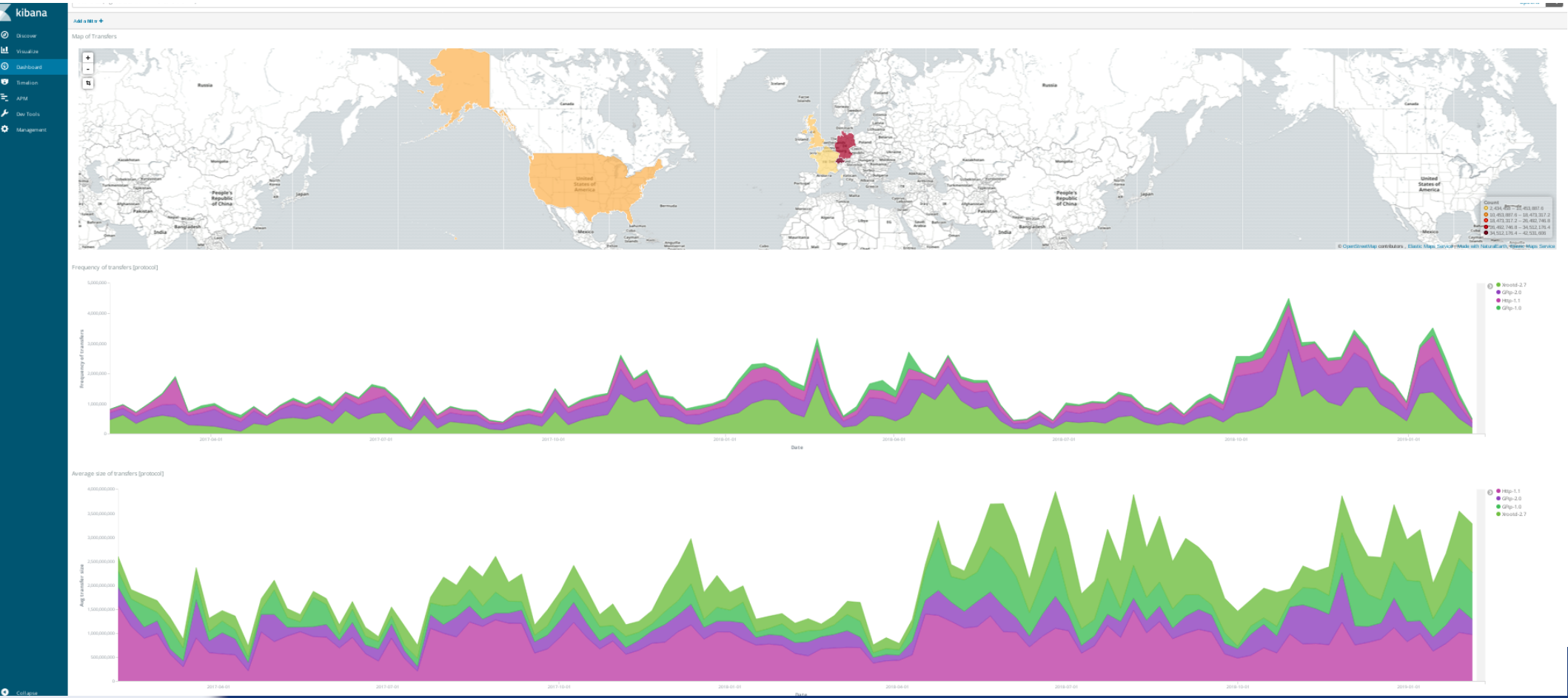
	CA	CERN	DE	ES	FR	IT	ND	NL	RU	TW	UK	UNK
CA	81%	87%	81%	99%	96%	93%	92%	95%	100%	100%	87%	-
CERN	96%	86%	95%	79%	94%	94%	93%	97%	100%	95%	93%	0%
DE	90%	97%	94%	67%	83%	73%	97%	44%	85%	90%	96%	0%
ES	100%	93%	98%	97%	95%	99%	99%	31%	100%	100%	98%	-
FR	100%	99%	95%	91%	96%	96%	98%	98%	100%	95%	100%	-
IT	45%	99%	98%	76%	93%	97%	99%	99%	100%	100%	97%	-
ND	99%	100%	99%	92%	97%	97%	94%	100%	100%	100%	99%	-
NL	100%	100%	93%	96%	96%	96%	100%	39%	100%	100%	92%	-
RU	100%	100%	94%	89%	93%	97%	100%	100%	100%	100%	99%	-
TW	92%	98%	77%	92%	69%	86%	100%	100%	68%	-	89%	-
UK	99%	100%	96%	95%	94%	95%	97%	88%	96%	53%	97%	-
UNKNOWN	100%	100%	84%	100%	99%	97%	100%	63%	100%	100%	99%	-
US	95%	99%	97%	93%	95%	98%	96%	86%	98%	100%	91%	-

Standardized ELK on Docker

- Used for GoeGrid
- Elasticsearch + Kibana (ELK)
- The ELK framework is standardized on Docker
- Shared the Code/Template on GitHub
- Easy to start the monitoring system and to send log data
- No tuning for each ATLAS site

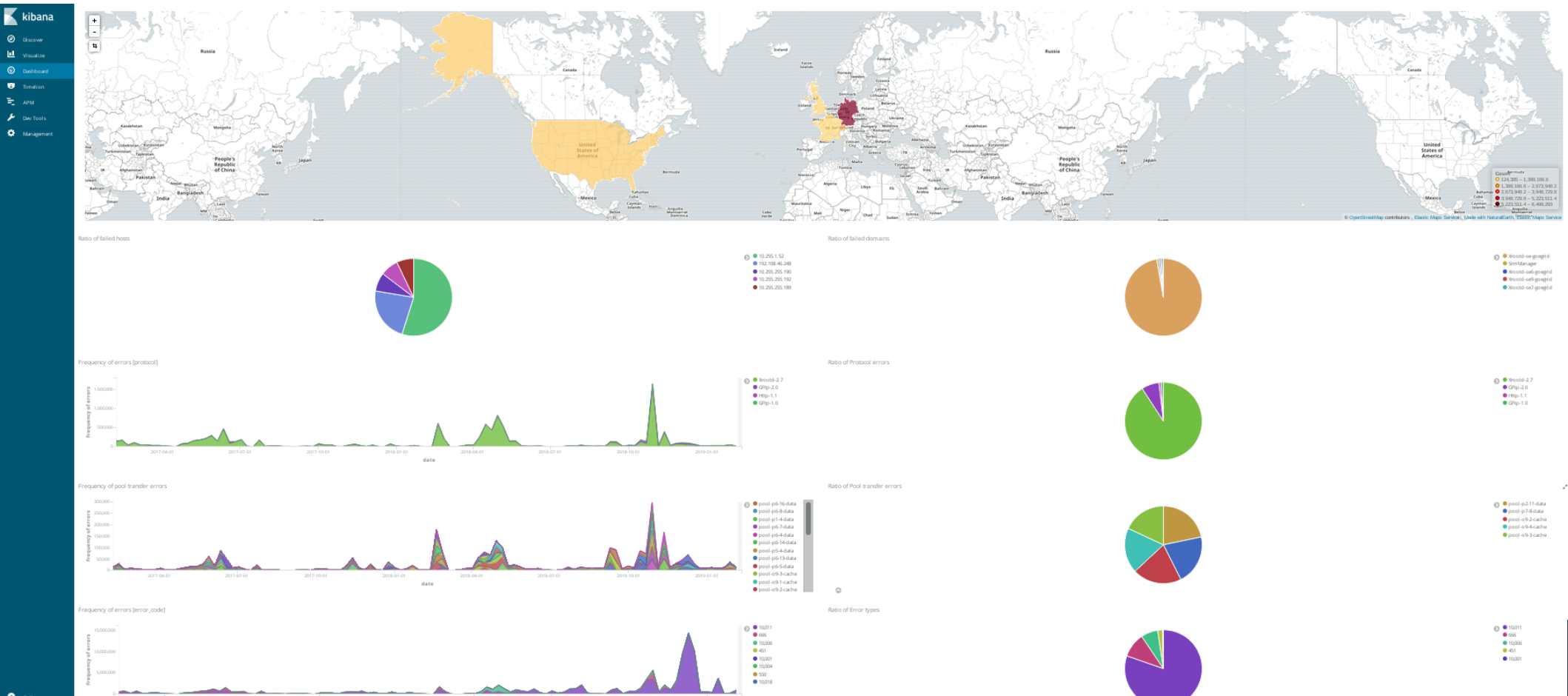
Quick overview on GoeGrid Kibana UI

- *Test Instance*
  - Standard (range: 2 years)



Quick overview on GoeGrid Kibana UI

- *Test Instance*
  - Errors (range: 2 years): “error_code” !=0, “pool” and “remote_host” exist

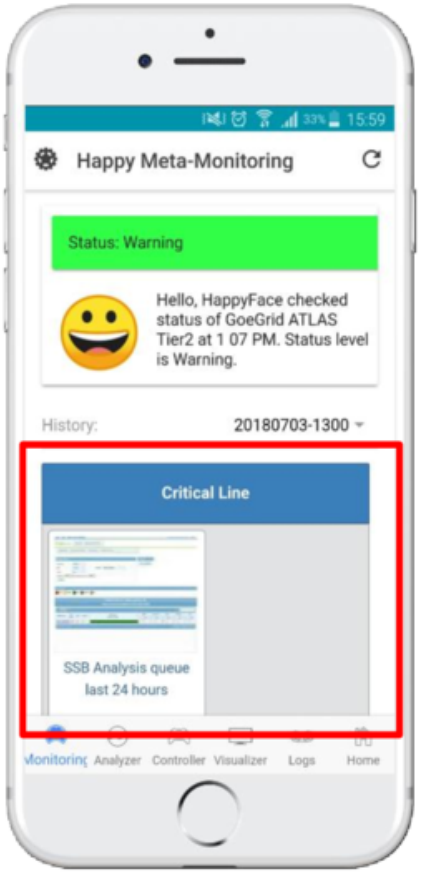



Canada
United States of America
Mexico
Bermuda
Bahamas
Cuba
Cayman Islands
Haiti
Belize
Anguilla
Iceland
Faroe Islands
Finland
Norway
Sweden
Estonia
Latvia
Lithuania
Denmark
Ireland
Belarus
Poland
Czech Republic
Ukraine
Hungary
Moldova
Romania
Slovenia
Serbia
Bulgaria
Abkhazia
Armenia
Andorra
Vatican City
Albania
Greece
Portugal
Malta
Tunisia
Cyprus
Lebanon
Iraq
Morocco
Israel
Kuwait
Algeria
Libya
EG
Saudi Arabia
Mauritania
Niger
Mali
Chad
Eritrea
Yemen

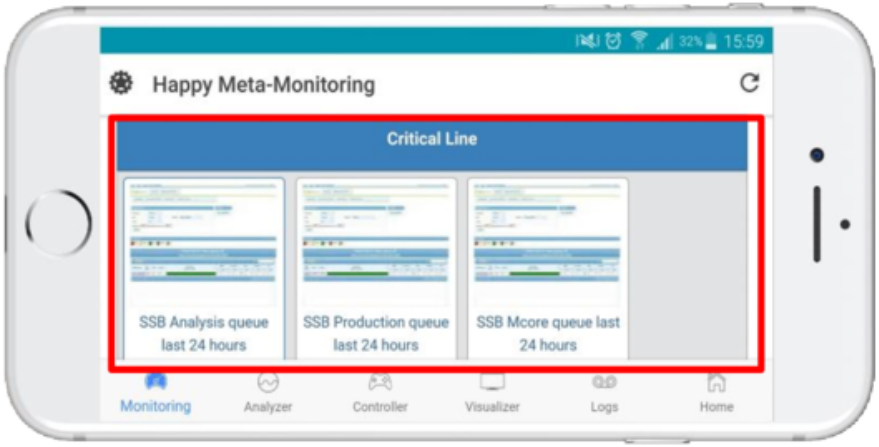

Canada
United States of America
Mexico
Bermuda
Bahamas
Cuba
Cayman Islands
Haiti
Belize
Anguilla
Aruba
Trinidad and Tobago
Iceland
Faroe Islands
Finland
Norway
Sweden
Estonia
Latvia
Lithuania
Denmark
Ireland
Belarus
Poland
Czech Republic
Ukraine
Hungary
Moldova
Romania
Slovenia
Serbia
Bulgaria
Abkhazia
Andorra
Vatican City
Albania
Greece
Portugal
Malta
Tunisia
Cyprus
Lebanon
Iraq
Morocco
Israel
Kuwait
Algeria
Libya
EG
Saudi Arabia
Mauritania
Niger
Mali
Chad
Sudan
Eritrea
Yemen
Cabo Verde
Gambia
Burkina Faso
Guinea
Sierra Leone
Ivory Coast
Togo
Nigeria
South Sudan
Somalia

And, HappyFace mobile application

Graphical user interface: The monitoring tab



Module section:
Displays the screenshot and the name of every module, grouped by their severity.
A tab on the image enlarges it.



Installing the HappyFace application

- Google Play Store

- iTunes App Store

Conclusion and outlook

- Operating a large Uni Tier-2 site, GoeGrid
- Monitored by HappyFace mobile application and Monitoring/Analytics framework on Docker
  - Kibana/Elasticsearch for dCache/PBS → Grafana/Elasticsearch for dCache/HTCondor
  - Geo-plot module, a simple failure detection
- HappyFace Smartphone application
  - Backend is HappyFace on Docker
  - Available in iTunes and Google stores

Thank you for your attention

Herzlichen Dank
für Ihre
Aufmerksamkeit !

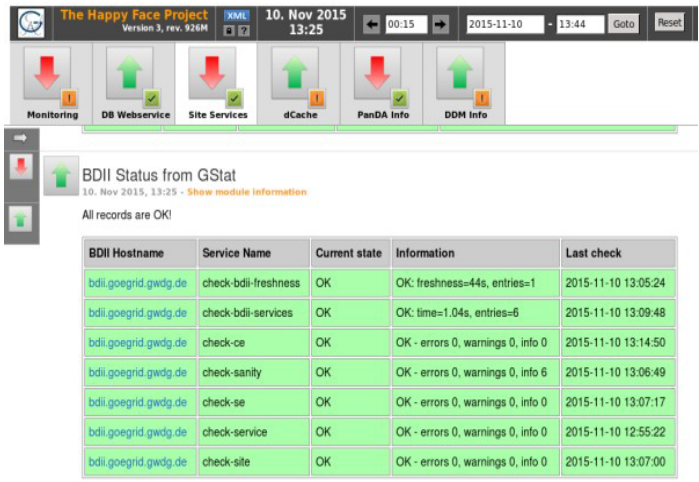
Backup

Backup and links

- HappyFace projects on Github
  - https://github.com/HappyFaceMonitoring
  - https://github.com/HappyFaceGoettingen
  - Gantry: a wrapper command
    - https://github.com/HappyFaceGoettingen/mad-gantry
  - Module and Mobile application
    - https://github.com/HappyFaceGoettingen/HappyFace-MadMask

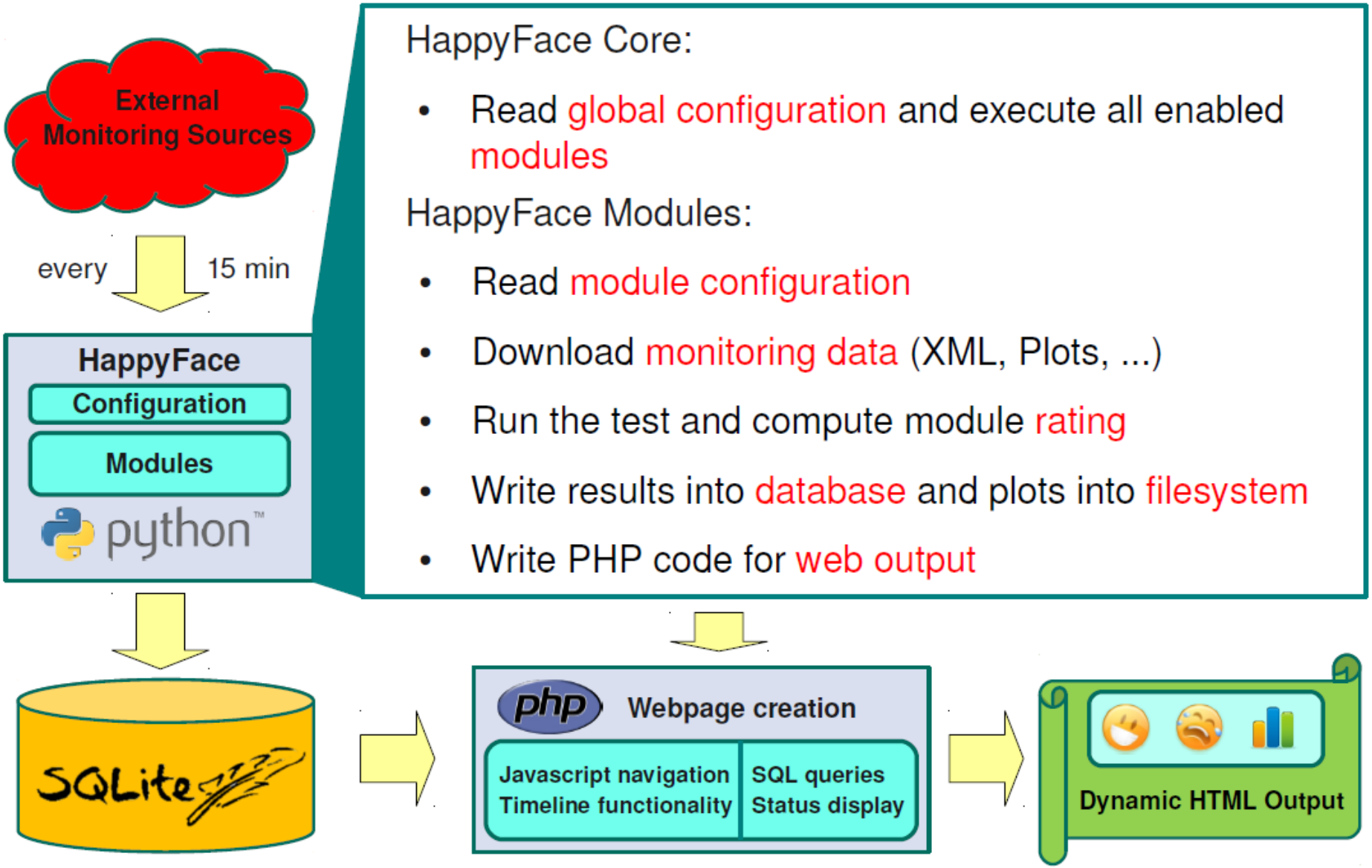
HappyFace meta-monitoring system

- What is HappyFace?
  - Essentials
    - A meta-monitoring tool for users/shifters and site admins
  - Developed
    - The core system by KIT (current version 3.0)
    - Joint efforts of D-CMS and ATLAS-D, collaboration among KIT, Aachen and Göttingen
  - Functionality
    - Meta-monitoring system
    - Monitoring crawler, parser, formatter
    - Historical function with DB backend
    - Web service interface



HappyFace meta-monitoring system

The HappyFace Project Architecture



HappyFace Core:

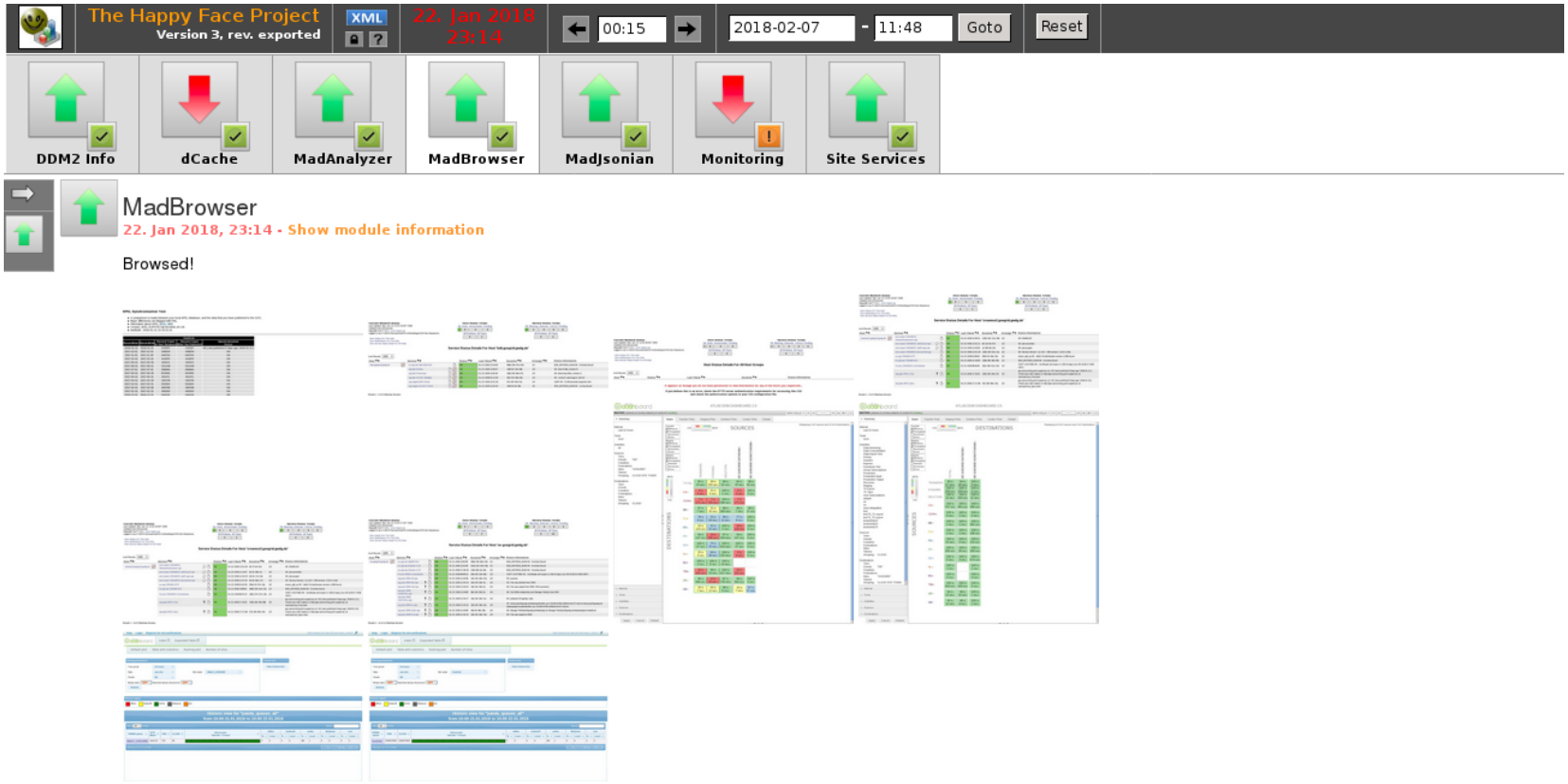
- Read global configuration and execute all enabled modules

HappyFace Modules:

- Read module configuration
- Download monitoring data (XML, Plots, ...)
- Run the test and compute module rating
- Write results into database and plots into filesystem
- Write PHP code for web output

Dockerized instance (web view)

http://141.5.108.30:10200/category



Dockerized instance (mobile view)

http://141.5.108.30:20200