

---

**The 9th DUW**

**Development  
plans**



---

Federico Stagni

---

- v6r22 (v6r22-pre1 out) [prod in few weeks]
  - **New:** Web 4.0
    - based on ExtJS6 (previous: based on ExtJS4)
  - **New:** extension: OAuthDIRAC for OAuth2 authentication capabilities
    - For SSO and EGI check-in (see pres from Andrei earlier)
  - New PilotManagerHandler service
  - Suspended status for VOMS users now taken into account
- v7r0 [prod this year, if everything goes well]
  - **New:** ProductionSystem (see pres from Luisa on Tuesday)
  - DIRACOS fully replaces Externals + IcgBundle
  - WMS JobsParameters table: on ElasticSearch (with flag)
  - M2Crypto instead of pyGSI (with env variable)
  - Pilot3 as default version



# DIRAC

THE INTERWARE

On a longer term  
(core + structural  
changes)

The DIRAC Core and Framework has been developed ~10 years ago, and now Lots of stuff in the DIRAC framework is available/maintained elsewhere

- gLogger → python logging [**DONE**][in prod]
  - and plugins on the shelf!
- pyGSI → M2Crypto [**DONE**][in v7r0 (testing)]
- dips → https [**IN PROGRESS**]
  - see later

Assumption:

increase of one order of magnitude

- Services: **~OK** if more services
  - → that “~” is meaningful!
- Agents: **KO**

- **HW scalability: more (micro)services**
    - Supposing a good load balancer, this is not necessarily bad
      - self-tuning system?
        - enter in the beautiful world of the **orchestrators**...
        - services are OK, but agents not
        - since when LHCb tested Mesos, k8s won the battle
          - not there yet
          - *any progress/test from anyone (not LHCb)?*
  - For today's implementation of agents:
    - Parallelization is hard
    - Multiple instances may not be possible
      - Lots of work anyway
- NOT scalable

- MQs are in DIRAC
  - For failover purposes
  - Consumers as DIRAC components → [RFC](#)
- Push, not pull
- We can replace several agents with Consumers
  - and also (especially?) executors
  - Agents, executors, consumers as a single component?
  - ... what about trying with this guy?
    - <http://python-rq.org/>
    - a nice project...

What I was  
proposing 1  
year ago...



RQ: Redis  
Queue

- MySQL
  - we can run on 5.6 and 5.7
  - MySQL 8 is a reality
    - adjustments needed?
      - DBs made with SQLAlchemy are compatible. Others?
- Elasticsearch usage growing
  - Monitoring system
    - for WMSAccounting
    - for internal monitoring
  - WMS (JobParameters from v7r0)
  - Would you be against of ES becoming a requirement?



- Use HTTPS to allow usage of standard libraries
  - HTTPS is widely used
    - Lower risk of security breach
    - Easier to get help
    - Easier to understand what the protocol really do underneath
    - Many conventions already exists to send datas (like *JSON* or *multipart/form-data*)
    - Frameworks already exists in python 2&3 for server-side (Tornado) and client side (requests)
  - For developers it may be easier to modify the code

PR from Luis Martin (stagiaire in LHCb):

<https://github.com/DIRACGrid/DIRAC/pull/3815>

- new client and server speaking HTTPs, with Tornado
  - tested and documented
  - integration postponed, for now
    - will anyway need to rely on a Tornado fork, because of M2Crypto

when DIRAC will speak fully HTTP(s):

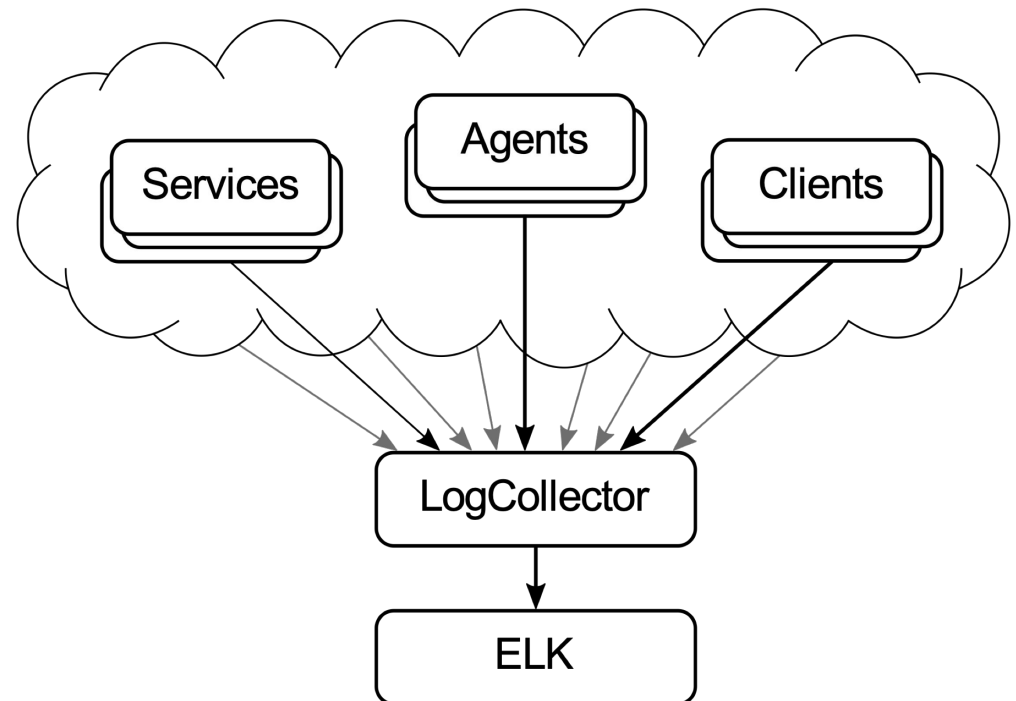
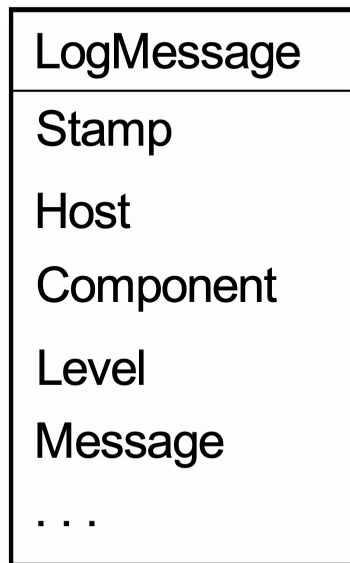
- the core of WebApp does not need to exist, as the core of DIRAC would serve just the same purpose
- handlers can be migrated, or better disappear
- meaning WebApp = JS only

- We don't have target release, yet
  - won't be v7r0
    - so we will be in at least 2020 (for something production level)
- we have been polishing the code for long time now
  - we applied python-modernize on some part of the code
- wide, deep, testing is fundamental
  - e.g. enabled pycodestyle, pylint --py3k on integration branch
  - new (completely new) code that goes to integration should be coded as python3 code, and made backward compatible
  - *almost* all the dependencies that we will rely on are already python3 compatible
    - and some stopped patching for python 2.7 (e.g. tornado)
- Pilot will probably move first
  - ...but the pilot will need to also be able to start with 2.6, 2.7, and 3.x

# A log collector in Go

- Evaluate the Go programming language performance
  - LogCollectorBackend (python)
  - LogCollector (Go)
  - Messages (JSON)

From  
Christophe  
Meessen

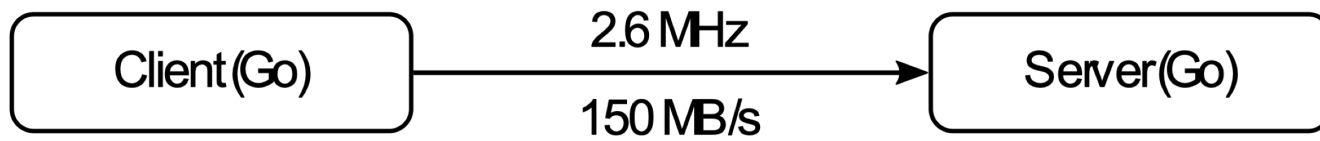


# Go: bandwidth upper limit ?

---

- Simple client server sending JSON or Binary messages
  - Binary: 58 bytes/msg, 150MB/s, 2.6MHz
  - JSON: 115 bytes/msg, 43MB/s, 350kHz
  - Binary saturate gigabit network
  - Limited by CPU, not I/O
  - See: <https://github.com/chmike/go-dmon>, branch: fullSpeed

From  
Christophe  
Meessen



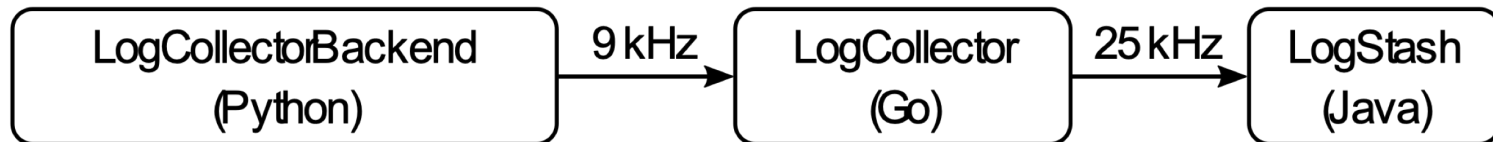
# LogCollectorBacked and logCollector

---

## □ LogCollector:

- LogCollectorBackend (Python): output max 9kHz
- logCollector (Go): output max 90kHz
- logStash (Java): input max 25kHz
- LogBackend: <https://github.com/chmike/LogCollectorClient>
- LogCollector: <https://github.com/chmike/logCollector>

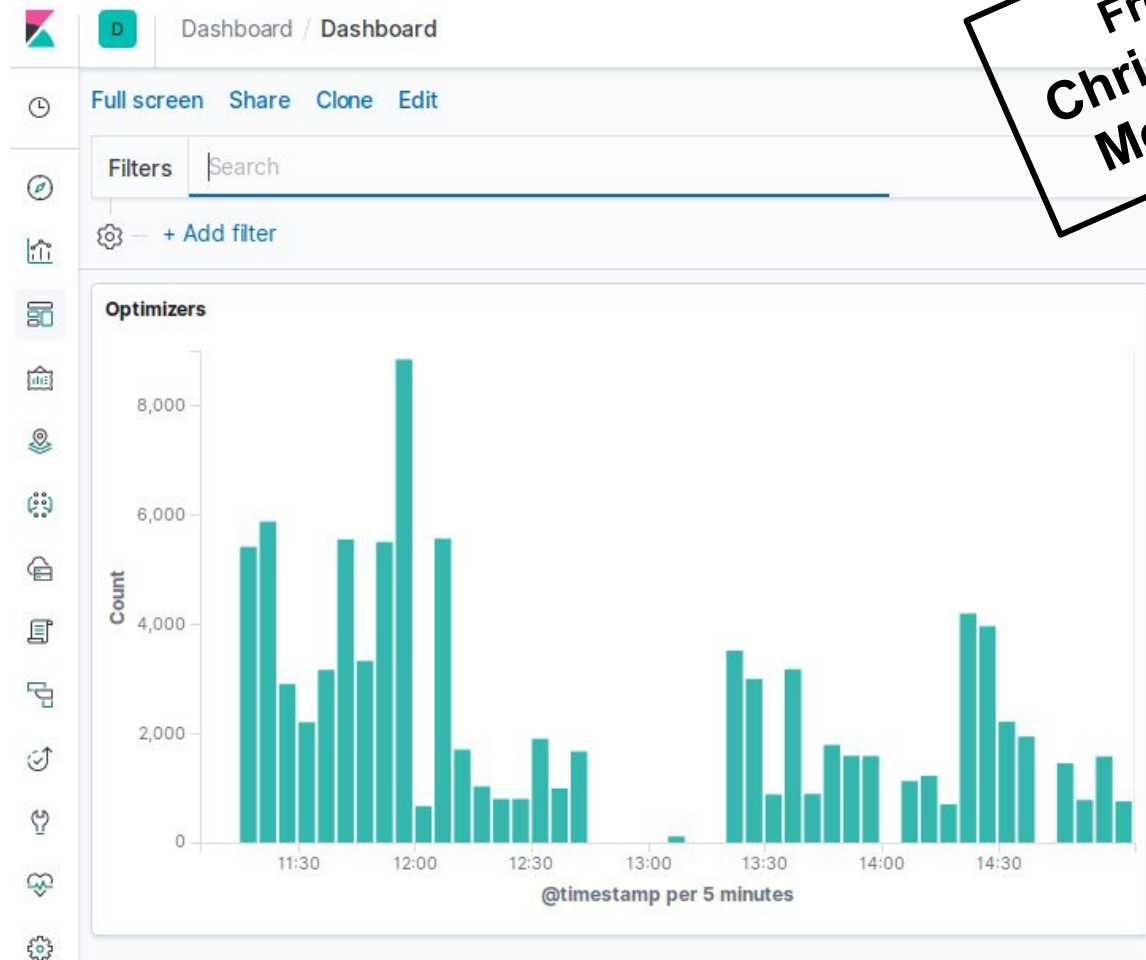
From  
Christophe  
Meessen



# Test with France-Grille

- Kibana:
  - 41 clients
  - Optimizers

From  
Christophe  
Meessen





Maybe not obvious, but the tests, certification, integration process is a daily work.

- Constant improvements.
- We use Travis, Jenkins, gitlab-CI...
  - and the certification setup

# Summary of main developments

<u>Development</u>	<u>Production level</u>	<u>Caveats/notes</u>
New WebApp, based on ExtJS6	Now (v6r22)	
OAuth2 support	~Now (v6r22) (with extension)	Will move to use native DIRAC HTTPs, when deployed
DIRACOS	~Now	Can be installed with dirac-install since v6r21, will be default from v7r0
ProductionSystem	v7r0 (2019)	
Pilot3 logging	2019 (?)	
DIPs → HTTPs	2020	<ul style="list-style-type: none"> <li>• M2Crypto (v7r0)</li> <li>• Patched Tornado</li> </ul>
Python 3	2020	DIRAC dependencies
DIRAC→Rucio bridge	? [t b det here]	

## Other developments

<u>Development</u>	<u>Production level</u>	<u>Caveats/notes</u>
Gitlab-CI for “matrix style” integration tests	2019	CERN summer student work
JobParameters on ES	v7r0 (2019)	...needs ES (for who wants it)
Multi-VO RSS	v7r1 (?)	
MySQL 8, ES 7	?	
DIRAC for HPC edge nodes	?	
Package+run with Docker+orchestrators	?	ATM not the biggest priority, but mostly missing personpower



interested? have a student? just get in touch!

we can provide projects!

## BACKUP Slides

- 
- Are we hitting it too much?
    - every RPC call goes through it
  - Memory cache?
    - object cache?
  - Database?
    - Redis, MemcacheDB, pickleDB, ...?
    -

- The DIRAC executor framework was an attempt of a scalable queueing system
- With an MQ system the concept of “executor” stays
- A consumer can spawn a (process)pool of executors
  - and we can have several running
- Agents can become a way to trigger executors at regular times
  - ...does it remind you the RequestExecutingAgent?
    - but you can have several “triggers” -- consumers

---

Whatever we do, now and in the future:

- Need to keep a running system working, with continuity
  - whatever we decide, DIRAC won't impose you to deploy a parallel DIRAC
- Little manpower
  - Need to focus on important aspects