SPINOSO Vincenzo (INFN-BARI on behalf of CMS and ReCaS Collaborations) – MISSIATO Massimiliano (INFN-BARI on behalf of the CMS Collaboration)

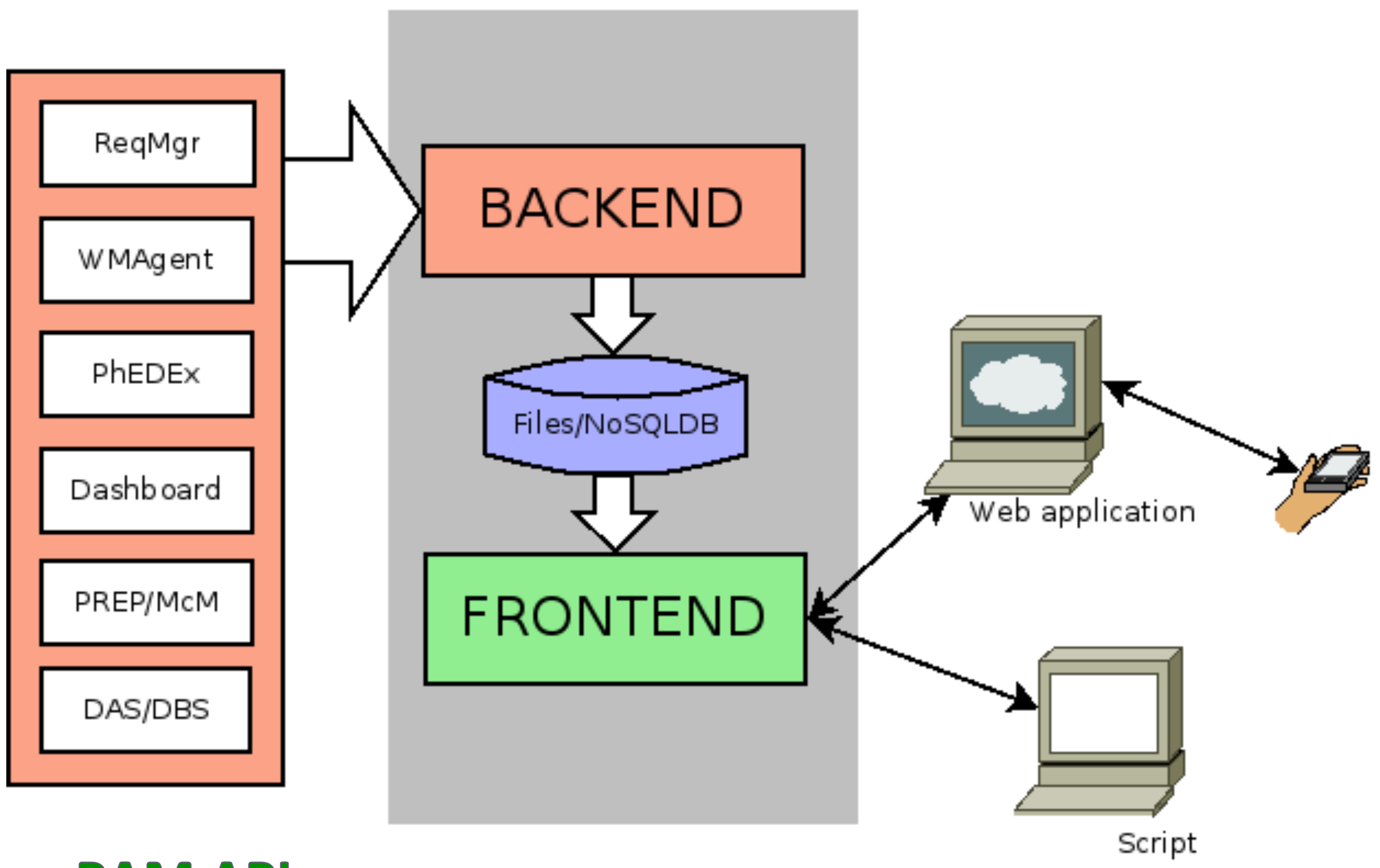
A flexible monitoring infrastructure for the simulation requests

“Running and monitoring simulations usually involves several different aspects of the entire workflow: the configuration of the job, the site issues, the software deployment at the site, the file catalogue, the transfers of the simulated data. In addition, the final product of the simulation is often the result of several sequential steps. This project tries a different approach to monitoring the simulation requests. All the necessary data are collected from the central services which lead the submission of the requests and the data management, and stored by a backend into a NoSQL-based data cache; those data can be queried through a Web Service interface, which returns JSON responses, and allows users, sites, physics groups to easily create their own web frontend, aggregating only the needed information. As an example, it will be shown how it is possible to monitor the CMS services (ReqMgr, DAS/DBS, PhEDEx) using a central backend and multiple customized cross-language frontends.”

WHY ANOTHER MONITORING

Applications to **automate** the **preparation** of the simulation requests (PREP/McM$_1$) and their **execution** (ReqMgr$_2$, WMAgent) have recently taken the challenge of producing more than 10B events per year.
However:

- monitoring systems are often "computing-oriented”;
- they publish separated monitoring interfaces
- although the CMS central services publish *web services*, the generic CMS user is not (and is not supposed to be) skilled enough to interact with them.



WHAT PAM IS

PAM (Production Aggregated Monitoring) represents an *abstraction layer on top of CMS central services*; in particular PAM:

- is **easy to use**: it enables the generic CMS user to access an aggregated view of the simulation requests in a very simple way (no complicated APIs to learn);
- reads data from the central services and store them on a **local cache**, in order to protect the central services from continuous (scripted) user queries;
- exports data using **JSON**, which makes information interchange immediate in all the most used programming languages;
- is **flexible**: it's based on flat files, NoSQL databases (CouchDB$_3$), and a very simple python library, in order to allow fast modifications to the code and the data schemas;
- is **robust**: it doesn't rely on complex dependencies, it just needs a standard Python >=2.6 environment;
- is **automatic**: intervention on the code/configuration is needed only if major changes happen on the infrastructure;

WHAT PAM IS NOT

PAM is not a web monitoring interface; instead, it's a data source where users can get their info (using a cron job for instance) and arrange them as they need/like (web pages, simple script stdout…)

PAM API

PAM frontend exports a RESTFUL API which allows to ask for
- **group of requests** (per type, status, ID, PrimaryDataset)
- **single requests details** (ID, PrimaryDataset)
- **chains** (namely, which requests have input (output) dataset equal to the output (input) of a given one)

http://pam.cern.ch/pamws.py/Resource?parameters=values

CHAINS OF REQUESTS

PAM uses a special graph-based algorithm, which is **general**, **exhaustive** and **fast**, in order to:

- match requests which are “connected” each other through a given dataset (the output dataset of a request is the input of the next request)
- return a JSON list of chained requests
- the user can read how they are *really* connected each other (not how they *should*)
    - PAM detects issues (for instance, requests writing to the same output dataset or reading from the same dataset)
    - PAM shows if requests need “attention” (waiting for definition, having already enough events), helping with decreasing latencies

“WHAT DO I NEED TO USE PAM?”

- A regular CERN account on LXPLUS
- A regular X509 proxy certificate (VOMS-proxy)

[1] Poster #152, J.R. Vlimant, "MCM: The Evolution of PREP. The CMS tool for Monte-Carlo Request Management"
[2] M Cinquilli et al 2012, "The CMS workload management system", J. Phys.: Conf. Ser. 396 032113, doi:10.1088/1742-6596/396/3/032113
[3] Apache CouchDB, http://couchdb.apache.org
[4] Valentin Kuznetsov et al., "The CMS data aggregation system", ICCS 2010, doi:10.1016/j.procs.2010.04.172