

Using DODAS as deployment manager for smart caching of CMS data management system

Tracoli Mirco on behalf of CMS collaboration and DODAS team



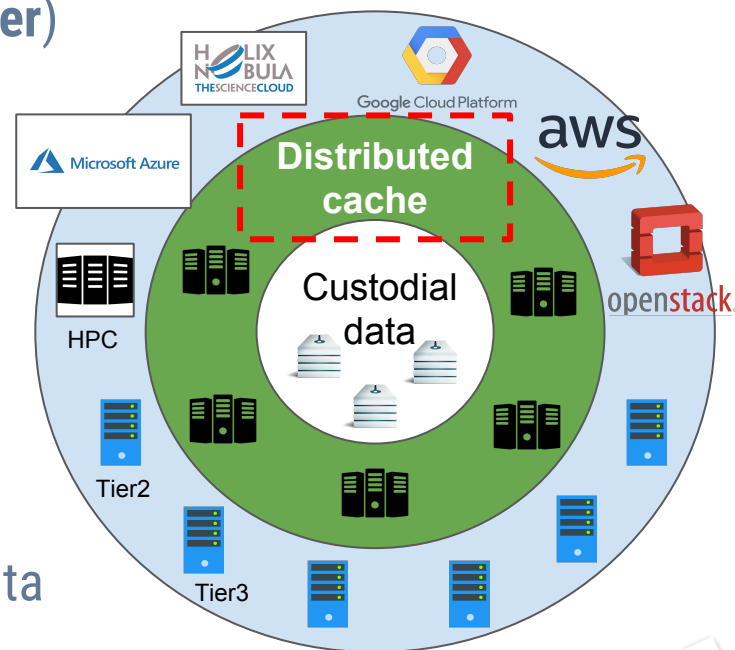
ACAT 2019 - Saas Fee, Switzerland

- Introduction to **CMS data** cache in the context of future **WLCG Data Lake**
- **Intelligent data-cache** operations
 - **Machine Learning** based strategy
- **DODAS** as enabling technology **for Machine Learning as a Service**
 - **Architecture** and key features
- **Proof-of-Concept** workflow
 - From raw data reduction and ML model training to inference
 - **Integration** with the cache middleware
- Conclusions

Data cache at CMS



- The **current CMS Data Management model** has a **meshed hierarchical centrally managed storages** at computing sites (**Tier**)
- **WLCG** towards a **data-lake model**:
 - **Fewer world-wide centers** with custodial data
 - **heterogeneous** set of **resources accessing** custodial data **remotely**
- A key element of future data lake will be the **data cache layer** that aims to:
 - make remote access to data more efficient
 - mitigate the amount of request to custodial data



Smart cache management: why



A **smart cache layer management** will improve the computing model with:

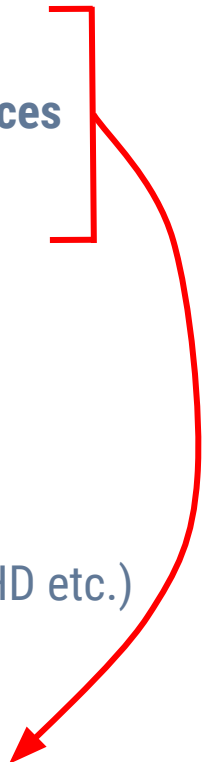
- **Enhanced CPU efficiency**
 - Thanks to I/O latency reduction of remote access
- **Reduced required disk space**
 - **Smart data pre-placement** on cache
 - Optimized data eviction
 - Use of diskless resources (Cloud and HPC)
- **Lowered operational costs:**
 - Leveraging real time routing and caching decision

Foresee the possibility to **dynamically deploy cache systems on opportunistic sites**.

Our strategy to enable smart caching

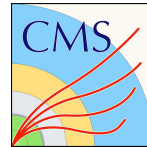


- **Training with Machine Learning techniques**
 - **Over historical data** (~40Gb) about usage of **CMS computing resources**
 - Using specific information such as **data popularity**
- Collecting **real-time information to Improve training**:
 - Network status, Network topology, Workflow type...
- **Achieve a solution** that:
 - Enables **autonomous management** of cache content
 - Manages **real-time Quality of Service** (dynamic routing, SSD over HHD etc.)
 - **Selects the best route** for the data



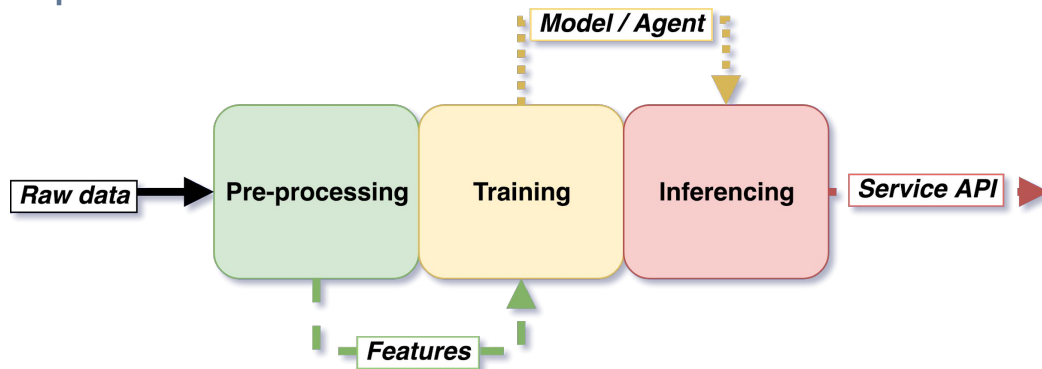
This presentation will **focus on the workflow environment implementation.**

Workflow overview



The environment is composed by independent modules:

- Pre-processing
- Training
- Inferencing (end-user service)



Each module provides a **specific service** for the **ML toolchain**. The aim is on a highly generic implementation of these **building blocks**. **DODAS** (see *later*) **allows this** achievement creating a platform for this model:

- On “**any**” **cloud** provider with a minimal effort
- Enabling **self-healing** and **scale-up** capabilities

DODAS as enabling technology



Dynamic On Demand Analysis Service is an **Open source project** for creating analysis **container based** clusters on-demand on **any cloud infrastructure** (details on next slide) with almost **zero effort**:



- just a simple configuration file with an end-to-end deployment in ~15 min.

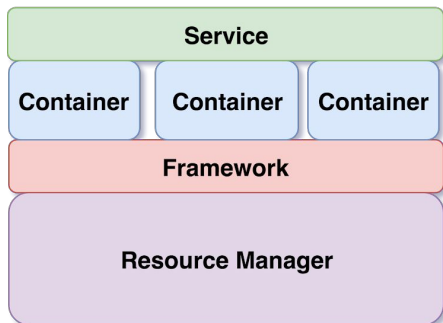
DODAS provides **container-based** solutions to instantiate:

- **Clusters for Big Data tasks:**
 - **Hadoop** cluster
 - **Spark** cluster
 - Generic **ML frameworks** (Including both Training and inference)
- HTCondor batch system as a service

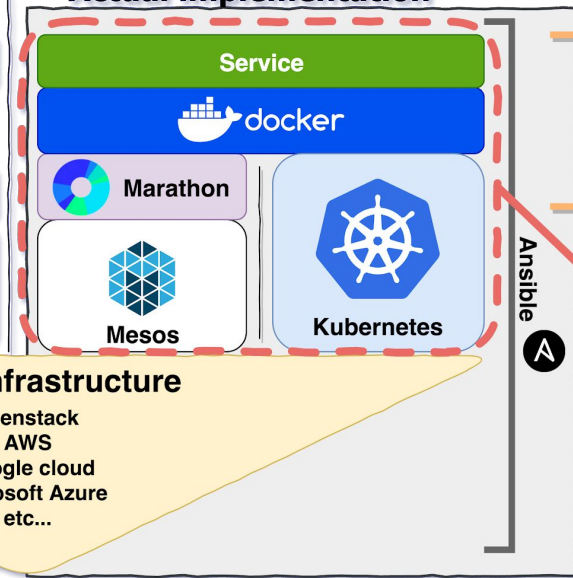
DODAS - Architecture



Model



Actual implementation



Cloud Infrastructure



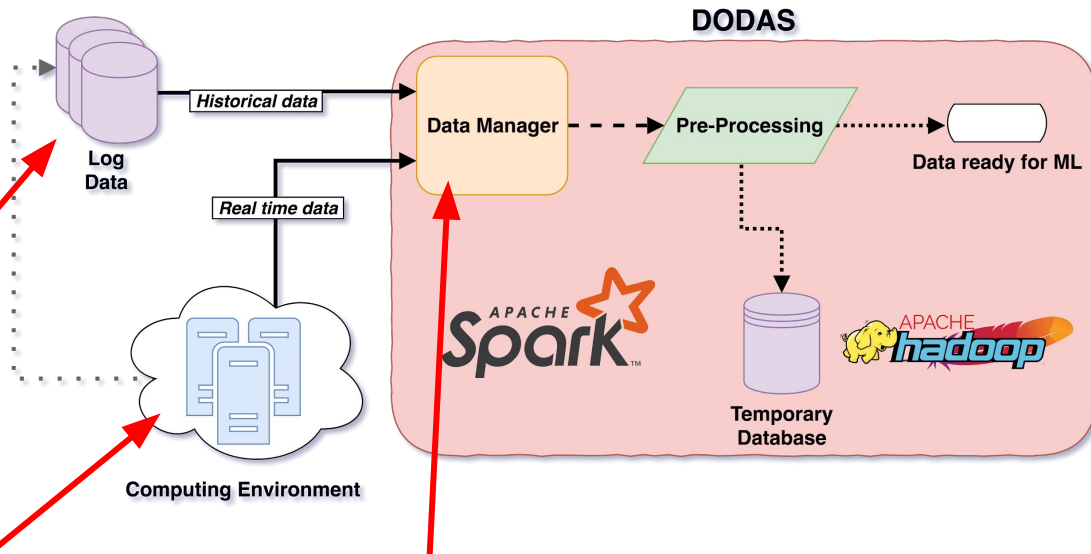
DODAS will be used to implement a **smart cache decision service** because it allows to **compose automatically the blocks** of the toolchain.

Auth

Data flow and Pre-processing

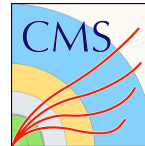


- The **CMS available logs** are the **key** to the success of the model development
- A **Primary data** source is historical data of infrastructure utilization:
 - **Data logs** are in JSON format, **stored** in a **Hadoop** file system and **serialized** using **Avro**.
- The **Secondary data** source are **real-time information**
 - Info of hardware, clusters, network and the cache system (content and status)
 - Streaming information feed

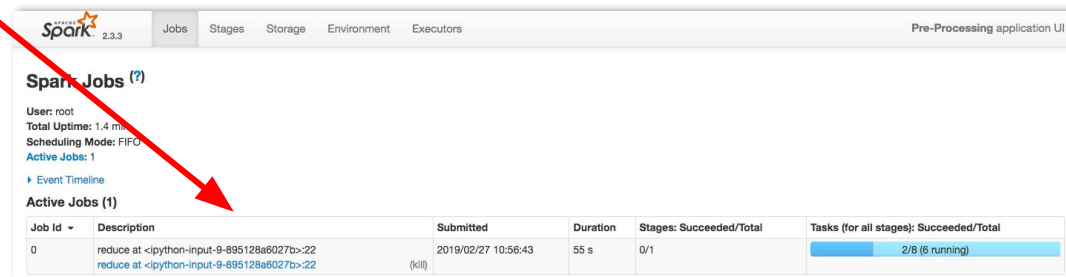


- The **Data Manager** can be used by end-users to **pre-fetch data** into DODAS environment or to **get a stream** of data in real-time.

Pre-processing step



- **Spark** is a part of DODAS deployment and end-users have access to it when DODAS is up and running
 - Technicalities are transparently handled by DODAS



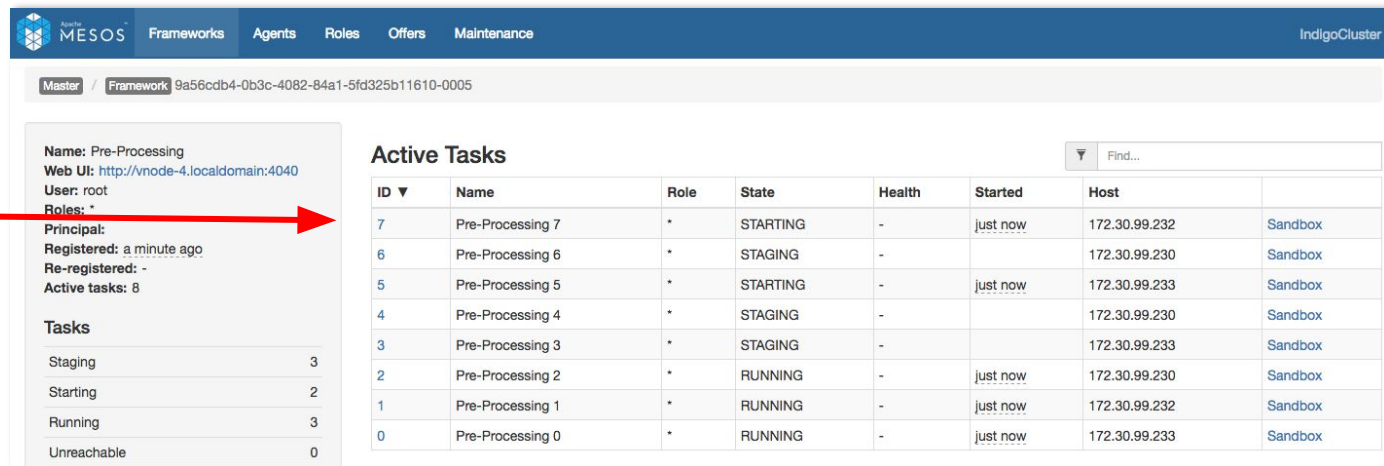
Spark 2.3.3 Jobs

User: root
Total Uptime: 1.4 min
Scheduling Mode: FIFO
Active Jobs: 1
Event Timeline

Active Jobs (1)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	reduce at <python-input-9-895128a6027b>:22 reduce at <python-input-9-895128a6027b>:22 (kill)	2019/02/27 10:56:43	55 s	0/1	2/8 (6 running)

- The service is **completely transparent** to the user, **Mesos** will manage the Spark's job.



Mesos Frameworks Agents Roles Offers Maintenance

Master / Framework 9a56cdb4-0b3c-4082-84a1-5fd325b11610-0005

Name: Pre-Processing
Web UI: http://vnode-4.localdomain:4040
User: root
Roles: *
Principal:
Registered: a minute ago
Re-registered: -
Active tasks: 8

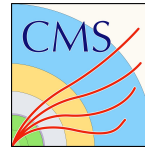
Tasks

Task	Count
Staging	3
Starting	2
Running	3
Unreachable	0

Active Tasks

ID	Name	Role	State	Health	Started	Host	
7	Pre-Processing 7	*	STARTING	-	just now	172.30.99.232	Sandbox
6	Pre-Processing 6	*	STAGING	-		172.30.99.230	Sandbox
5	Pre-Processing 5	*	STARTING	-	just now	172.30.99.233	Sandbox
4	Pre-Processing 4	*	STAGING	-		172.30.99.230	Sandbox
3	Pre-Processing 3	*	STAGING	-		172.30.99.233	Sandbox
2	Pre-Processing 2	*	RUNNING	-	just now	172.30.99.230	Sandbox
1	Pre-Processing 1	*	RUNNING	-	just now	172.30.99.232	Sandbox
0	Pre-Processing 0	*	RUNNING	-	just now	172.30.99.233	Sandbox

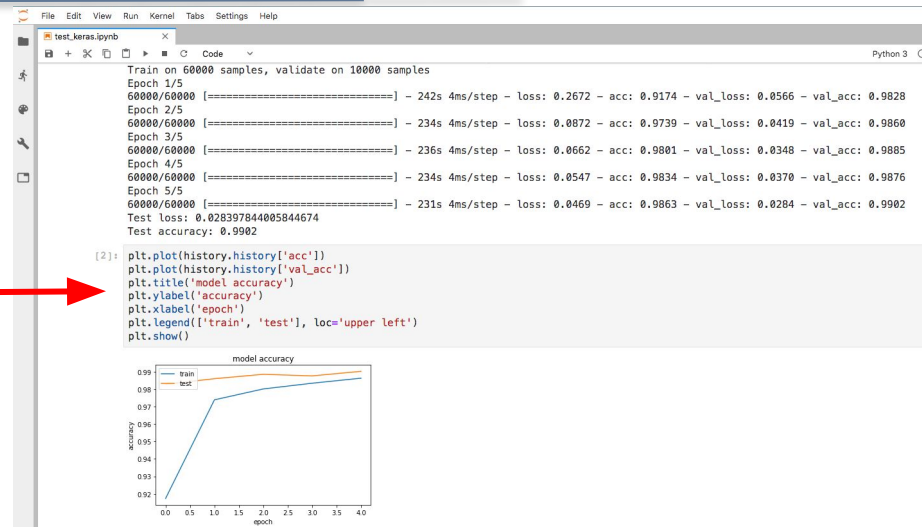
Training models over reduced data



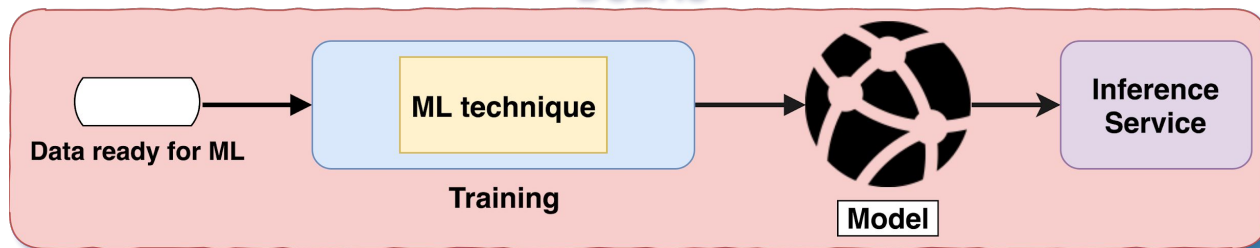
- Reduced data are automatically available for training ML models
- The developed environment is ready with the most used **ML frameworks**:
 - **Jupyter, Keras and TensorFlow**
 - Highly **customizable**: e.g. **Intel BigDL framework** has been added to use **alongside Spark** for the **training** phase.

The **output** of this phase is a **model** to use in the inference step.

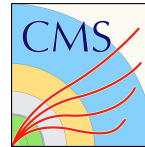
Trained model is **automatically loaded** into the **inference service**.



DODAS

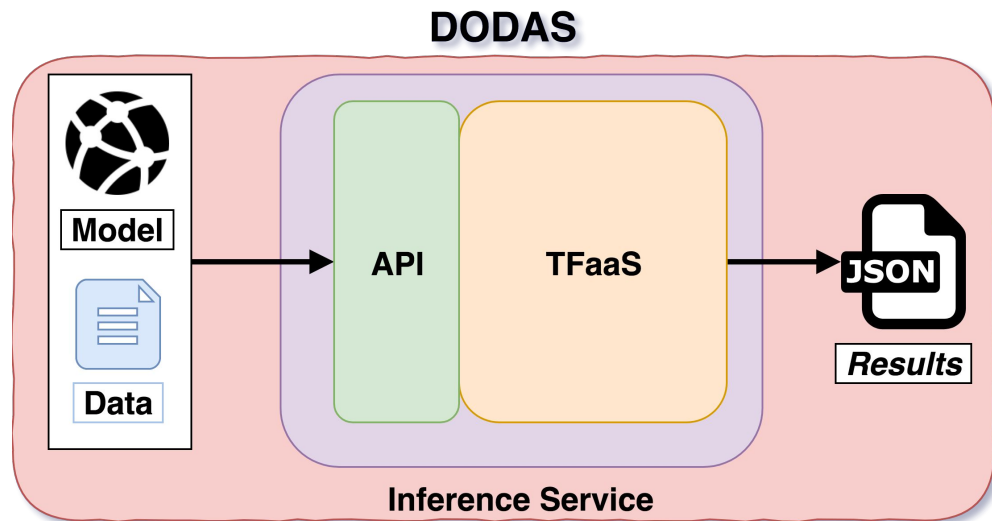


Performing Inference



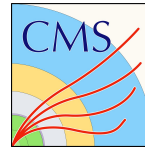
The inference service is implemented using the **CMS TFaaS**, embedded in DODAS. It is a **Software as a Service** based on **TensorFlow framework** for Machine Learning and exposes an **API** through the **HTTP** protocol:

- **/models**: to view existing models on TFaaS server
- **/json**: to serve TF model predictions in JSON data-format
- **/upload**: to push a model to TFaaS server
- **/delete**: to delete your model



DOI: Valentin Kuznetsov. (2018, July 9). vkuznet/TFaaS: First public version (Version v01.00.06). Zenodo.
<http://doi.org/10.5281/zenodo.1308049>

Inferencing with TFaaS



Call the model: `curl -X POST http://tfaas/json -d @data.json -H "Accept: application/json" -H "Content-Type: application/json"`

Result:

```
{"labels":[{"label":"a","probability":1}, {"label":"b","probability":2.815438e-8}, {"label":"c","probability":4.65911e-18}]}
```



SUPPORTED MODELS

TFaaS built around TensorFlow libraries and therefore will support any TF model you'll upload to it. The model should be uploaded in ProtoBuffer (.pb) data-format along with model parameters.

COMPAT

It is possible to upload a model in TensorFlow SavedModel format. Please follow the steps below:

- 1 Download
- 2 Save your model
- 3 Convert

Existing models

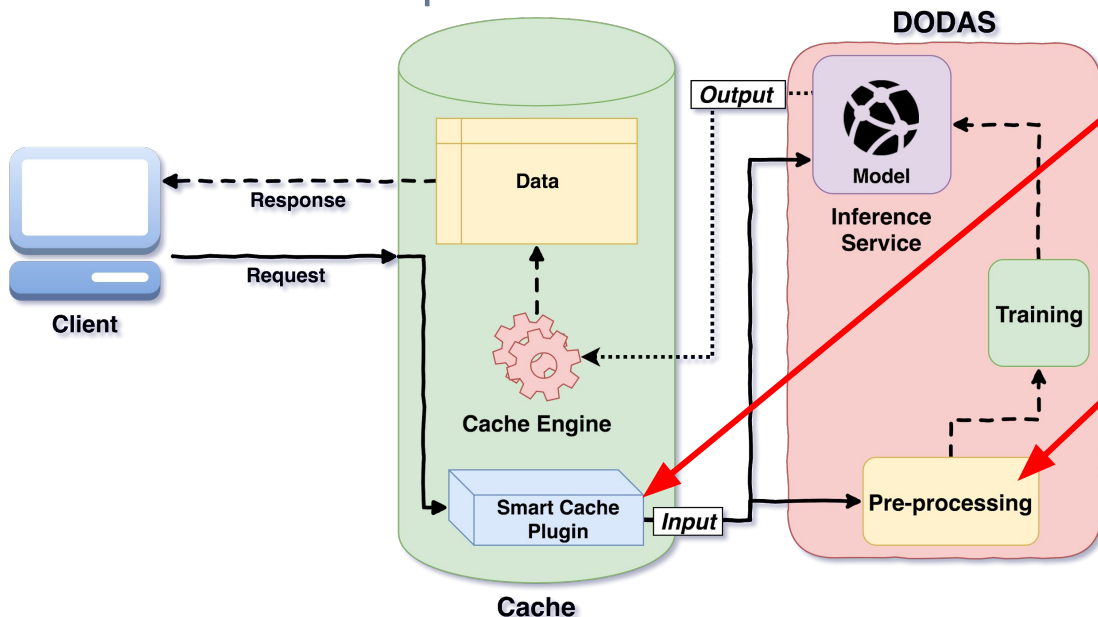
- name: MyModel
- model: [model.pb](#), [graph view](#)
- labels: [labels.txt](#)
- description:
- timestamp: 2019-02-27 20:37:39 +048197028 +0100 CET m=+16963.822615362

The screenshot shows the MARATHON Applications dashboard. A red box highlights the 'tfaas' application in the list, which is in a 'Running' state. A red arrow points from the 'Existing models' section to this application.

STATUS	NAME	CPU	MEMORY	STATUS	RUNNING INSTANCES	HEALTH
Running	jupyter-proxy	0.1	32 MiB	Running	1 of 1	...
Deploying	spark-bastion	2.0	1 GiB	Running	1 of 1	...
Suspended	spark-proxy	0.1	32 MiB	Running	1 of 1	...
Delayed	spark-shuffle-service	0.3	192 MiB	Running	3 of 3	...
Waiting	spark-tunnel	0.0	0 B	Suspended	0 of 0	...
HEALTHY	tfaas	1.0	512 MiB	Running	1 of 1	...
Unhealthy						
Unknown						

Integration with Data Cache

- The plan is to **extend the XRootD cache** (XCache) with a specific **plugin** which queries against the developed **AI Service**
 - The TFaaS endpoint



Runtime information are used to **continue** the **training** of the **model**

- A **proof-of-concept implementation** to enable **smart data cache at CMS** has been shown
 - The first tests of full workflow are promising
 - **Research and develop a model** for the proposed problem
 - **Study the performances**
 - **Benchmark** the model also through **simulation**
- Usage of **DODAS** as technology to Abstract underlying infrastructure, scalability, automation and self-healing
- The DODAS based **smart decision service is completely generic**
 - **Customizable** and thus **reusable** for similar use cases

End of presentation

“Cloud is about how you do computing, not where you do computing.”

Paul Maritz, CEO of VMware

Backup

DODAS - Overview

