C ontrol system based on
H ighly
A bstracted and
O pen
S tructure

!CHAOS: a new software infrastructure to realize the controls of future accelerators and large experiments

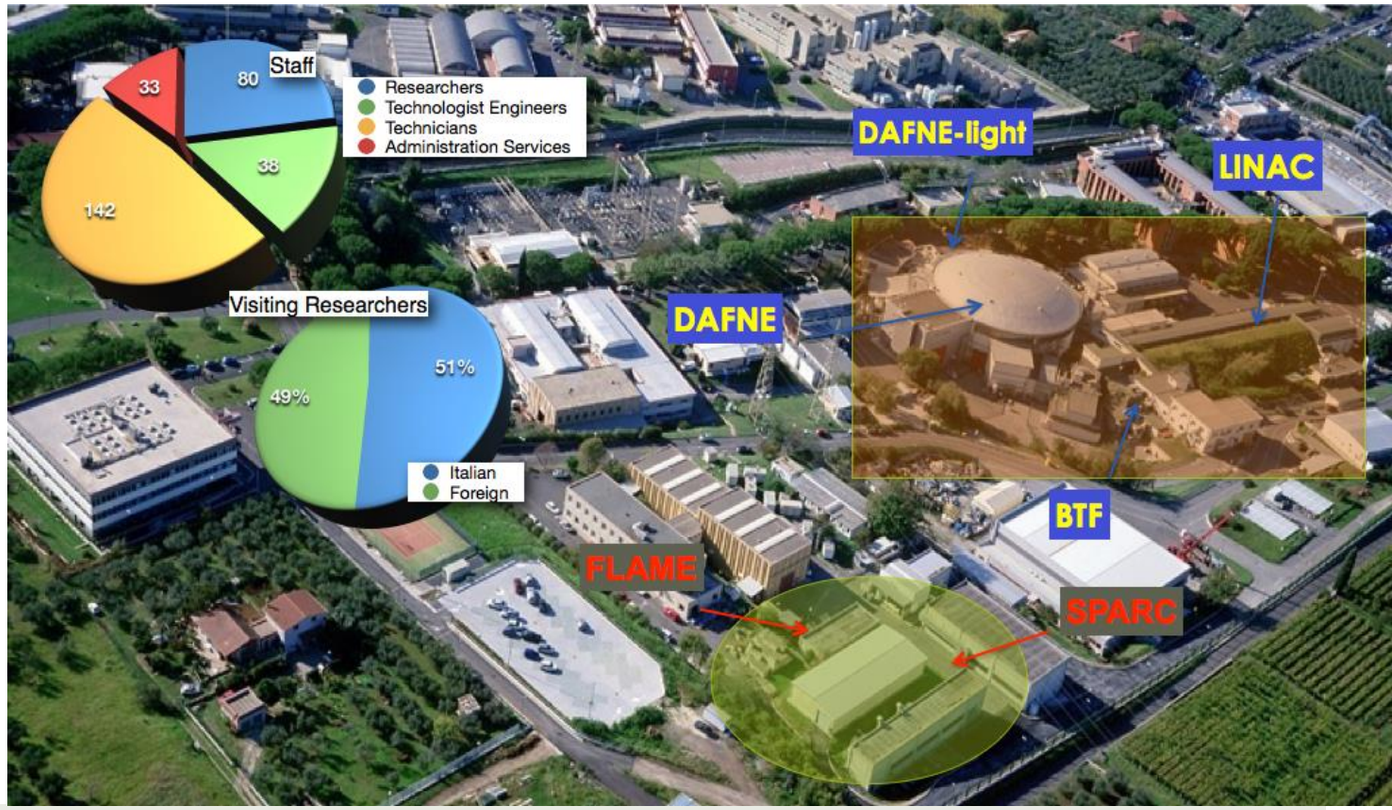
https://chaosframework.atlassian.net/wiki/display/DOC/General+View

Andrea Michelotti for the control group

Outline

- !CHAOS architecture objectives
- !CHAOS Overview
- !CHAOS as “Control as a (cloud) Service”
- !CHAOS TestBed @INFN-LNF
- conclusion

1961	AdA	Frascati	Italy
1964	VEPP2	Novosibirsk	URSS
1965	ACO	Orsay	France
1969	ADONE	Frascati	Italy
1971	CEA	Cambridge	USA
1972	SPEAR	Stanford	USA
1974	DORIS	Hamburg	Germany
1975	VEPP-2M	Novosibirsk	URSS
1977	VEPP-3	Novosibirsk	URSS
1978	VEPP-4	Novosibirsk	URSS
1978	PETRA	Hamburg	Germany
1979	CESR	Cornell	USA
1980	PEP	Stanford	USA
1981	Sp-pbarS	CERN	Switzerland
1982	p-pbar	Fermilab	USA
1987	TEVATRON	Fermilab	USA
1989	SLC	Stanford	USA
1989	BEPC	Beijing	China
1989	LEP ,	CERN,	Switzerland
1992	HERA	Hamburg	Germany
1994	VEPP-4M	Novosibirsk	Russia
1999	DAΦNE	Frascati	Italy
1999	KEKB	Tsukuba	Japan
2000	RHIC	Brookhaven	USA
2003	VEPP-2000	Novosibirsk	Russia
2008	BEPCII	Beijing	China
2009	LHC	CERN	Switzerland



Why a new control architecture?

- New IT technologies that handle big data throughput, developed for web based applications, can be successfully applied

- New accelerator infrastructures opportunities

- Capitalize know-how from DAϕNE /SPARC INFN-LNF experiences (control system based on LabVIEW®)

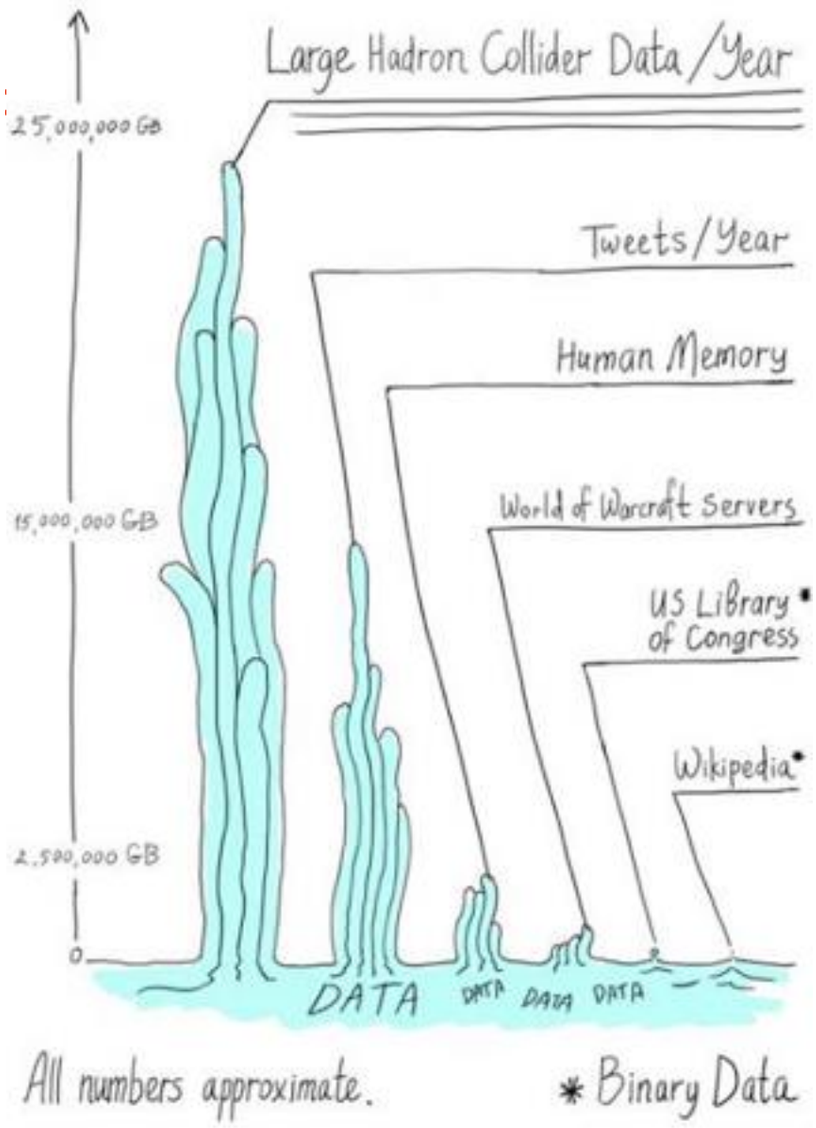
- Availability of DAϕNE /SPARC test facilities

- INFN interest to standardization and new technology development

Objectives of the project

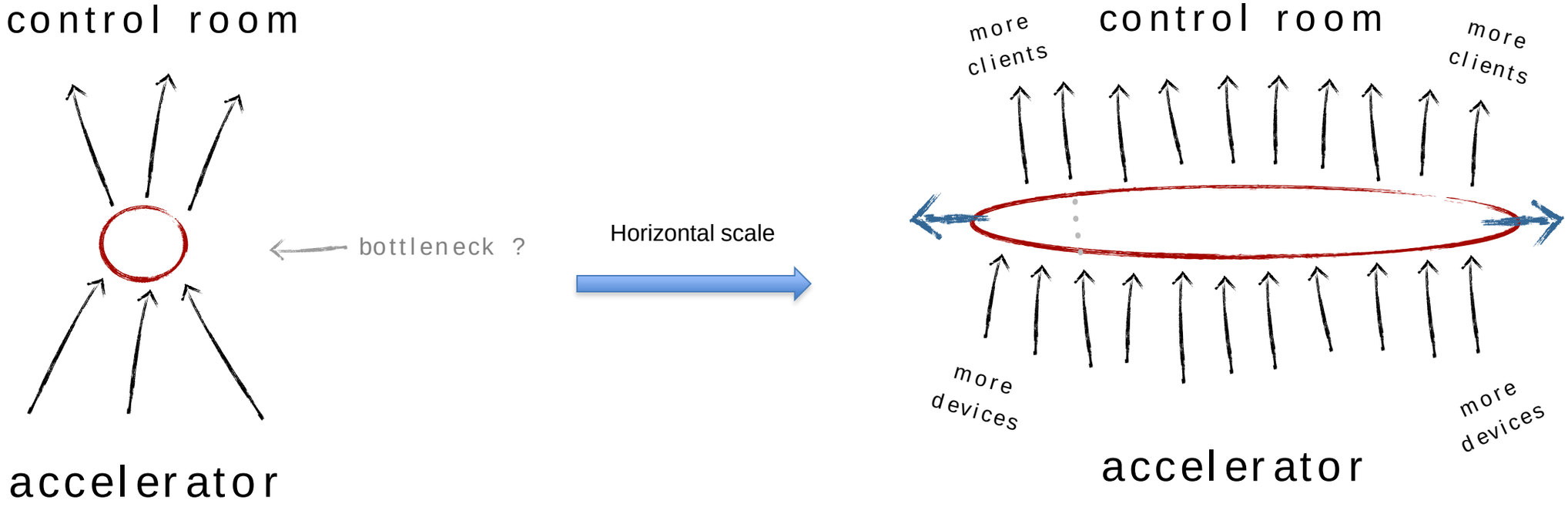
design a new open source controls architecture in order to achieve:

- intrinsic **scalability** ensuring high **throughput** data acquisition
- intrinsic **reliability** ensuring controls system fault tolerance
- minimize **maintainability and extensibility** efforts
- support “plug & play” **connectivity** functionality
- **uniformity** and **standardization** of data



!CHAOS Overview

!CHAOS has been designed to widely scale and to manage large installations with different needs in terms of storage and data throughput



!CHAOS Overview

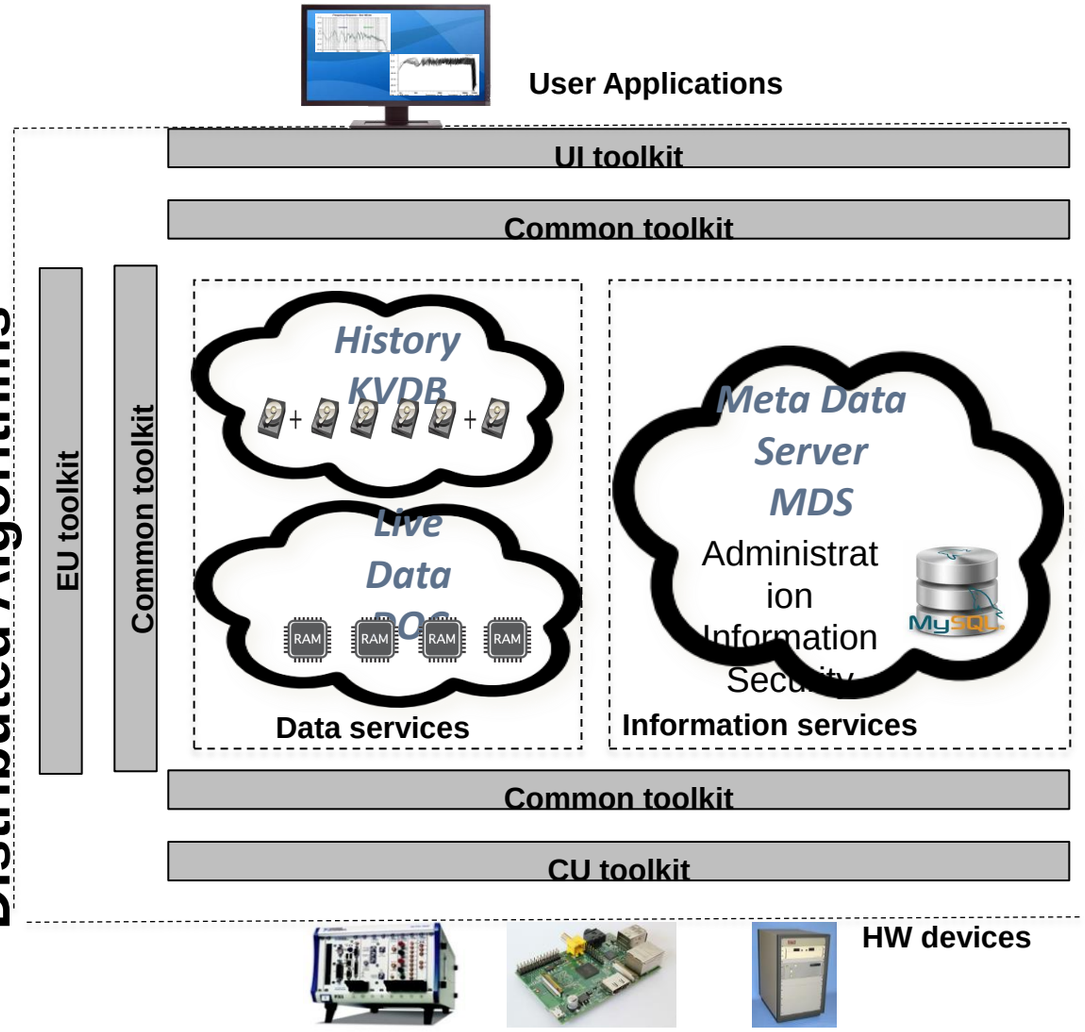
Chaos can be view as a collection of toolkits and services to build a full control and DAQ system

- **CUToolkit** tools for developing **device drivers** nodes
- **EUToolkit** tools for developing distributed **control algorithms** nodes
- **UIToolkit** tools for developing **user applications** nodes
- **CommonToolkit** is the common layer that offers the common communication services among CU,EU,UI nodes and abstracts the interface versus **data** and **information** services

Services:

- **MDS** MetaDataServer provides management, information and security services of !CHAOS nodes. For instance it manages the configurations of the HW devices (CUs).

- **LiveData** realizes a distributed cache (on RAM)
- **HistoryData** realizes a distributed permanent storage (on DISK)



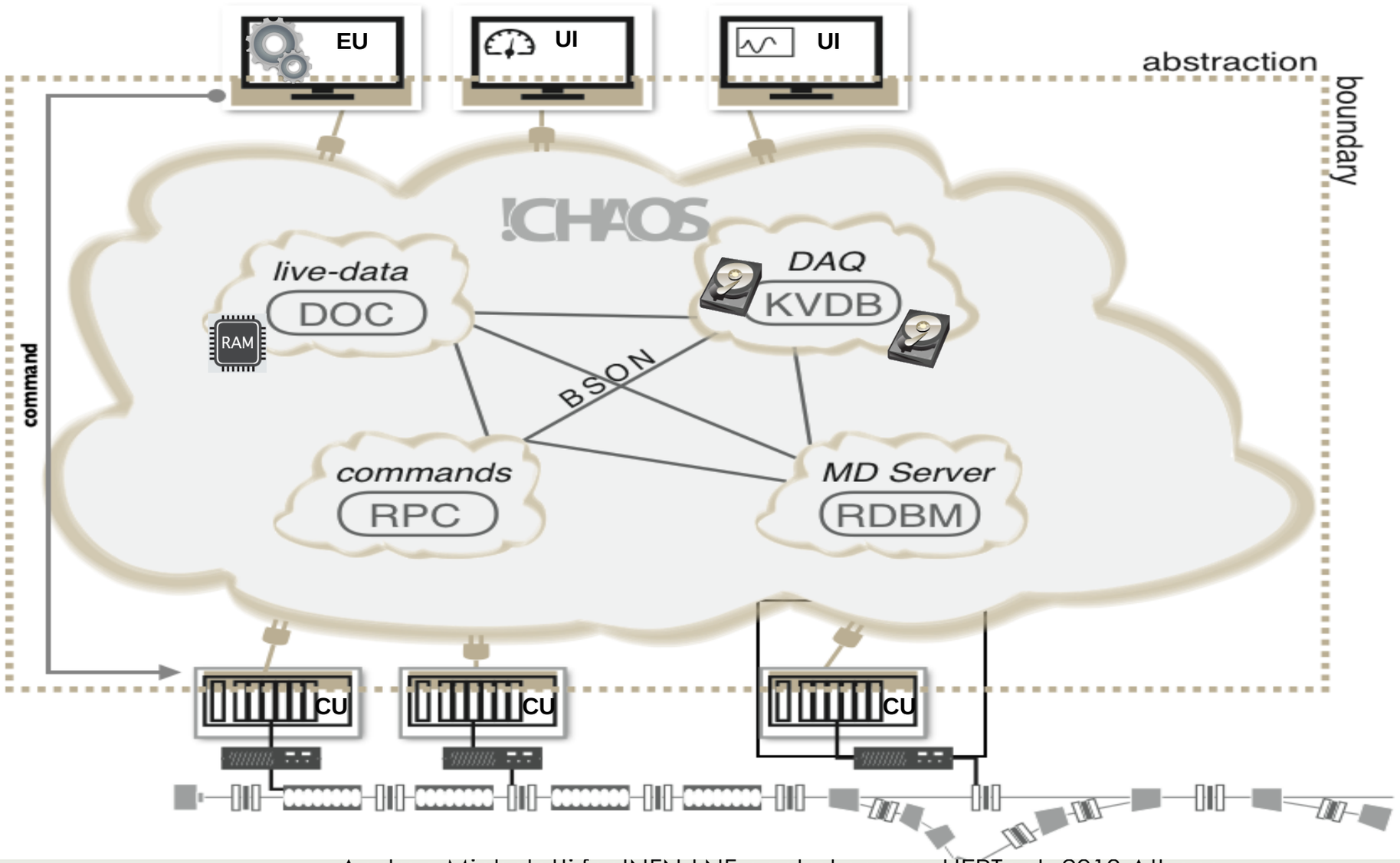
!CHAOS Overview: Keywords and Ingredients

- Scalability, Performance and Reliability

  Extensive use of distributed **Key Value DB** (for History) and **Distributed Object Cache** (for Live Data).

- Maintainability and Extensibility

  Extensive use of Abstractions (C++), use of **BSON** for data description and serialization.



!CHAOS Overview
Distributed Object Cache (Live Data)

Memory distributed caching allows to span data over multiple servers so that you can grow in size and in transactional capacity, but still gives you a logical view of a single cache.

- Allows high-performance caching of data (using RAM)
- dynamical *keys* re-distribution allows automatic failover by redirecting to other servers the load of failed one.
- Horizontal Scalability is also guaranteed by definition

!CHAOS Overview
KeyValueDB (History Data)

A KeyValueDB (sometimes referred as Non relational DB), provides a highly optimized mechanism for storage and data retrieval thanks to it's simpler structure respect to relational databases

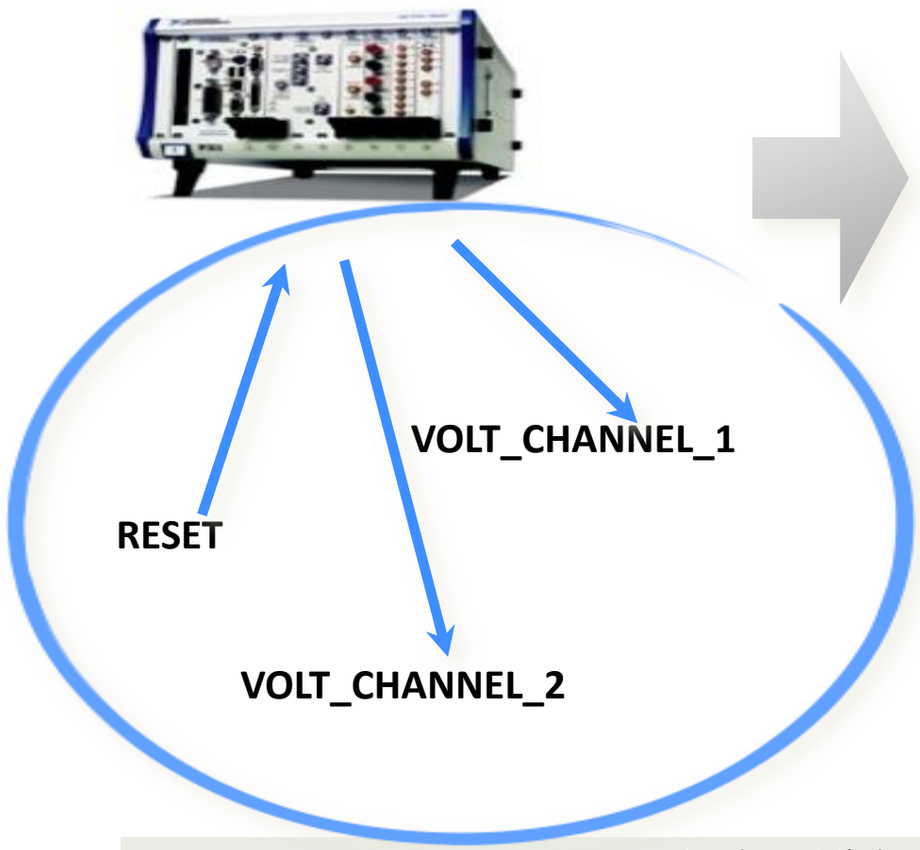
- Scalability and load balancing by sharding (horizontal scaling)
- Fast I/O because of simpler structure (don't use tables)
- Schema-less (high availability, flexibility)
- Fast parallel searches on cluster nodes by using map-reduce

!CHAOS Overview
BSON(serialization)

Binary encoded serialization of JSON (javascript object notation)

!Chaos use BSON for dataset resource description and for RPC messages (serialization/deserialization) between nodes.

- Key/Value representation, well suited for KVDB and DOC
- Fast because binary optimized
- It is schema-less which gives flexibility and availability



Device DATASET
name:VOLT_CHANNEL_1
type:VOLT32
flow:output

name:VOLT_CHANNEL_2
type:VOLT32
flow:output

name: RESET
type: BYTE
flow: input

The BSON representation

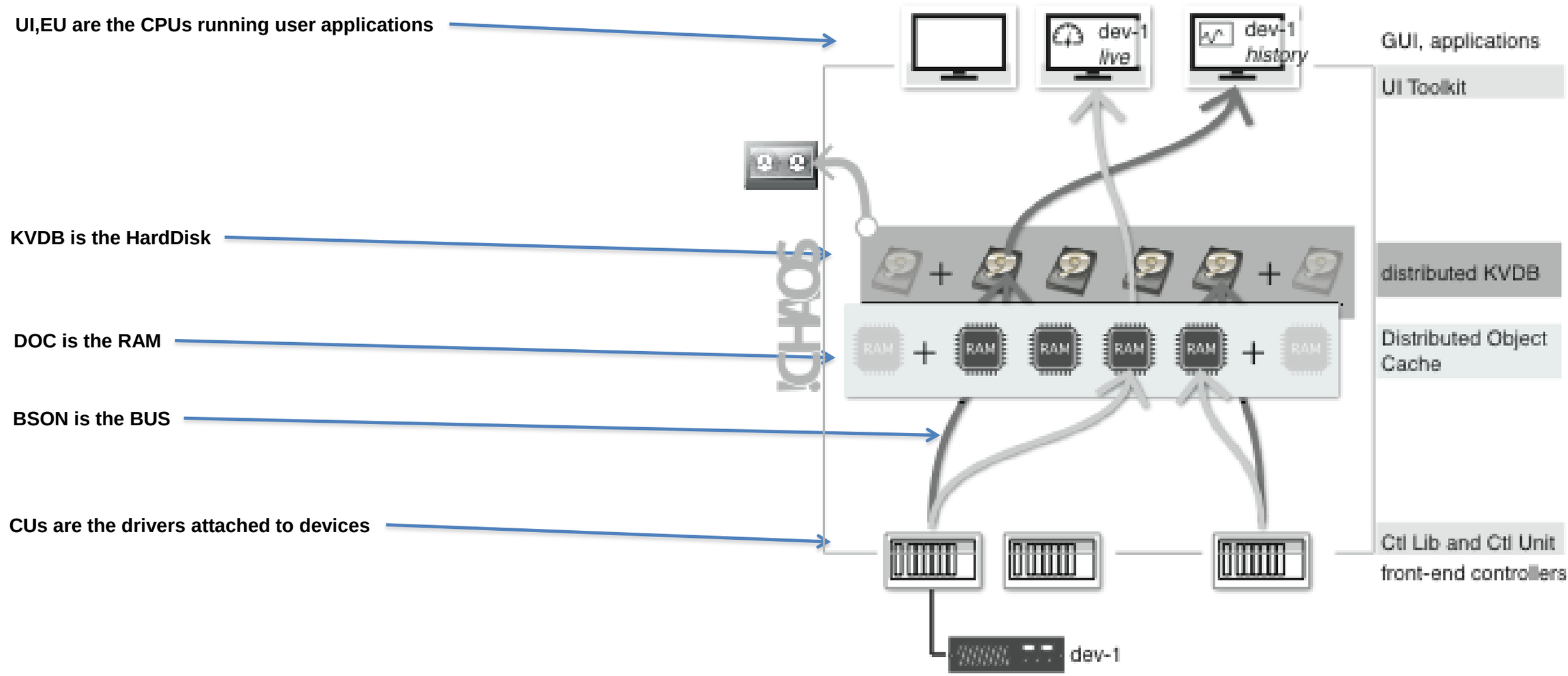
```
{“device_id”: DEVICE_ID
 “device_ds”: {
  "name"        : "VOLT_CHANNEL_1",
  "desc"        : "Output volt",
  "attr_dir"    : 1,
  "type"        : 4
                }

....

}
```

!CHAOS Overview
Another way to look at !Chaos

!CHAOS can be view as a distributed computer

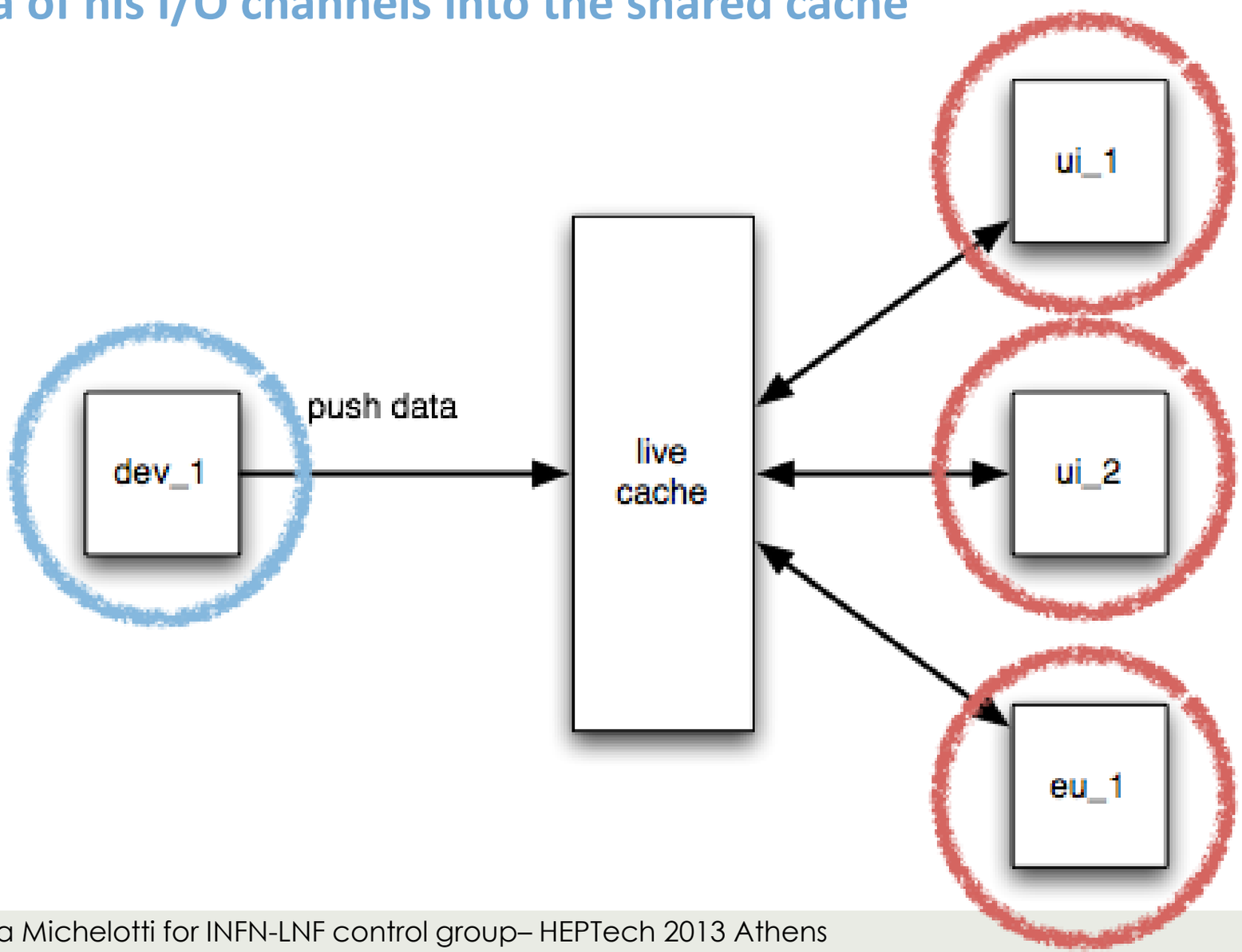


!CHAOS Overview
Data Scaling and Reliability

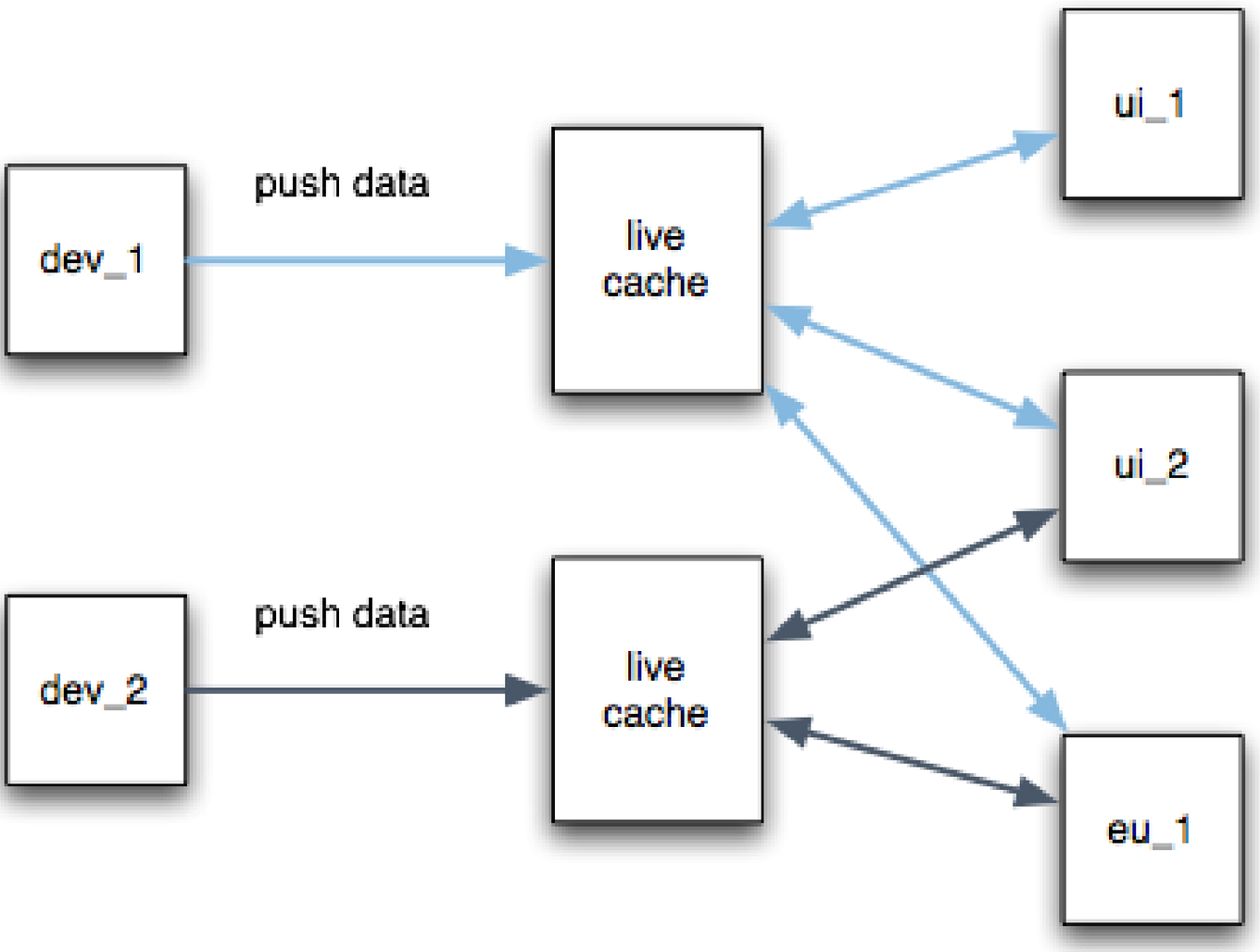
1. every node can read data from the central cache at any time[polling]

2. the nodes can register to central cache for receive update[push]

Every instrument pushes the data of his I/O channels into the shared cache

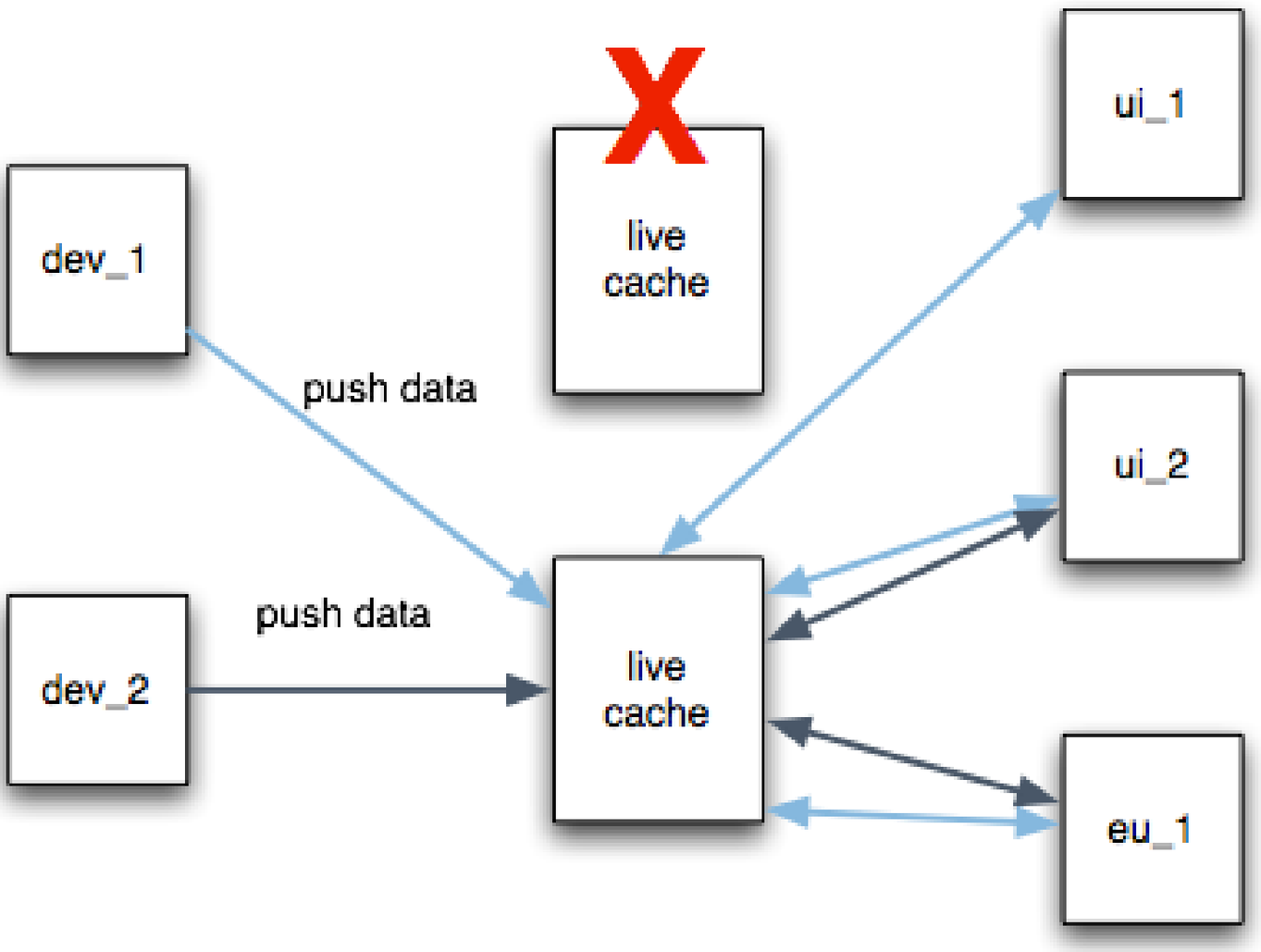


!CHAOS Overview
Data Scaling and Reliability



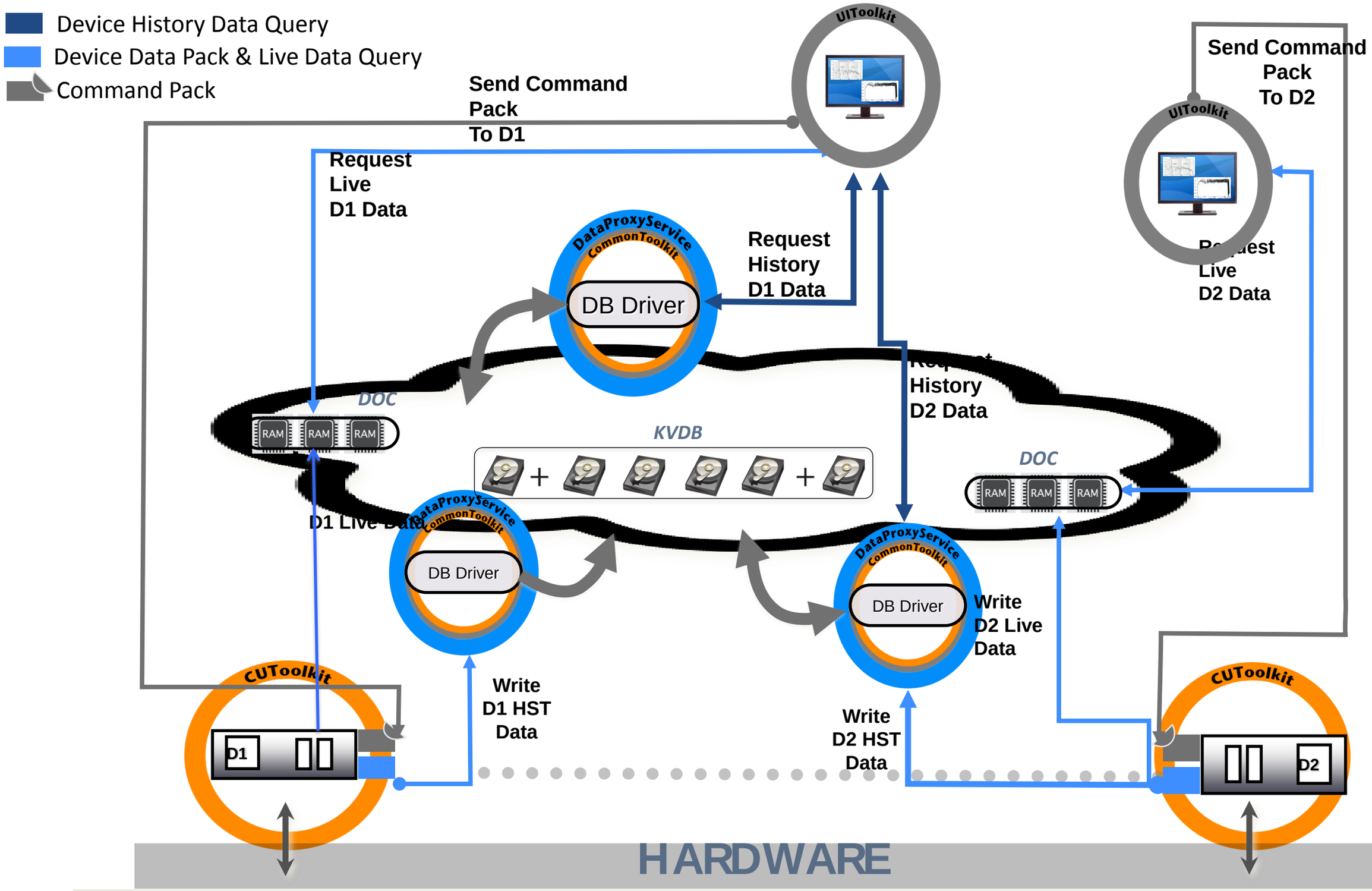
All data of all instruments are distributed across many cache server

!CHAOS Overview
Data Scaling and Reliability

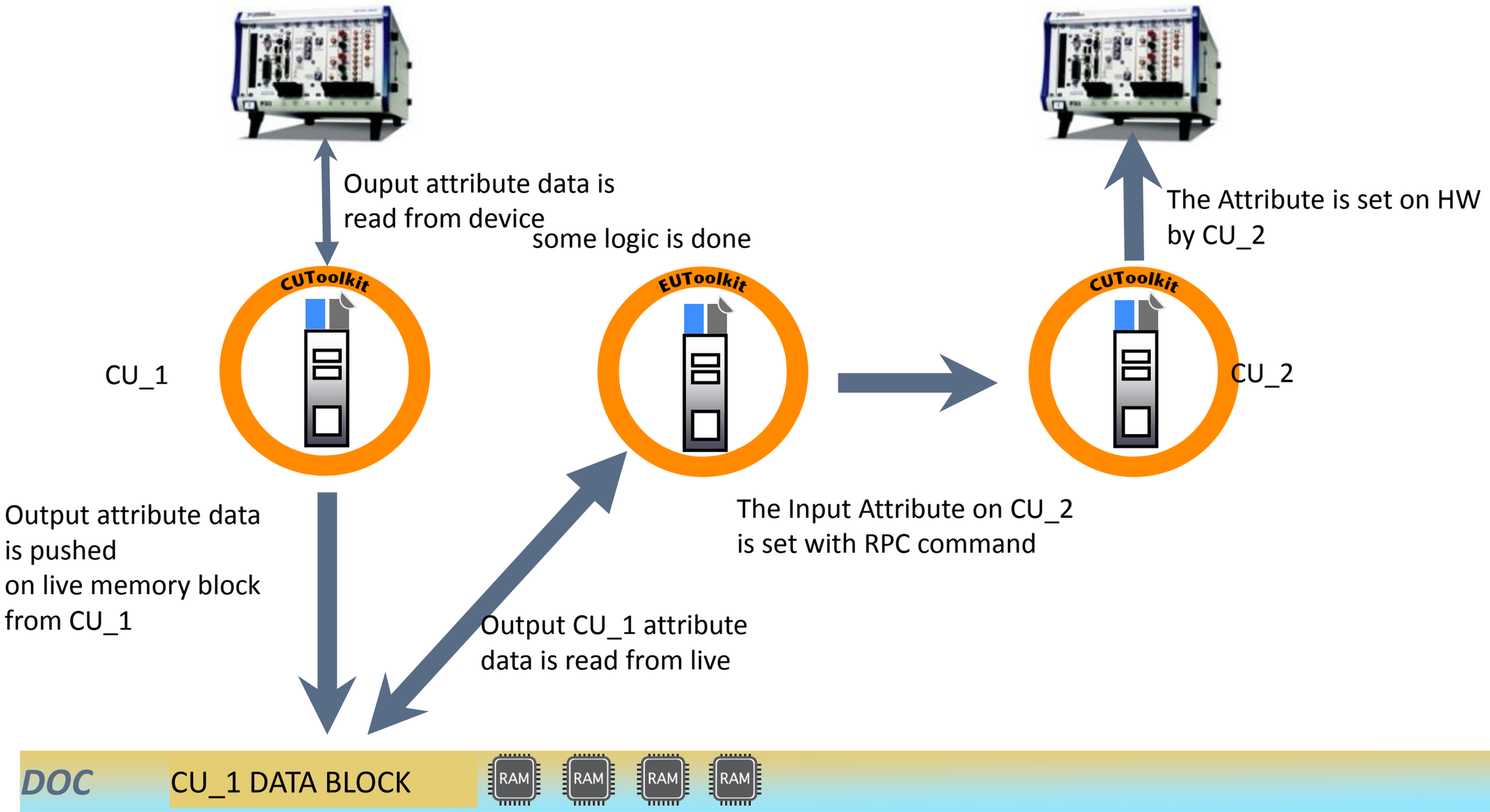


If one server goes down, all client that push's, or read, data on that server, use another server, with highest priority

!CHAOS Overview
Data Flow example

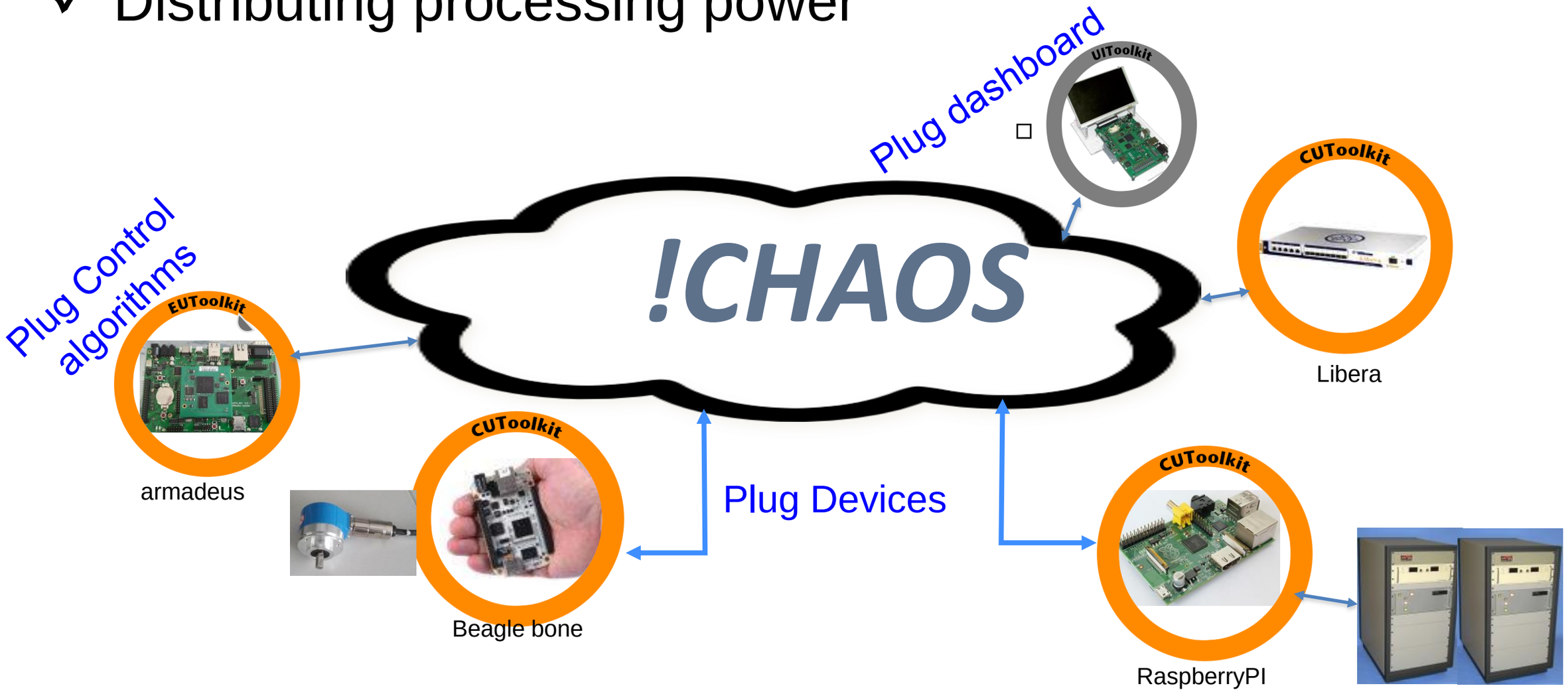


!CHAOS Overview
Data Flow example



!CHAOS Overview
Embedding !Chaos

- **!Chaos is well suited to be put on embedded systems**
  - ✓ Reducing overall cost
  - ✓ Reducing the number of different protocols to manage
  - ✓ Distributing processing power

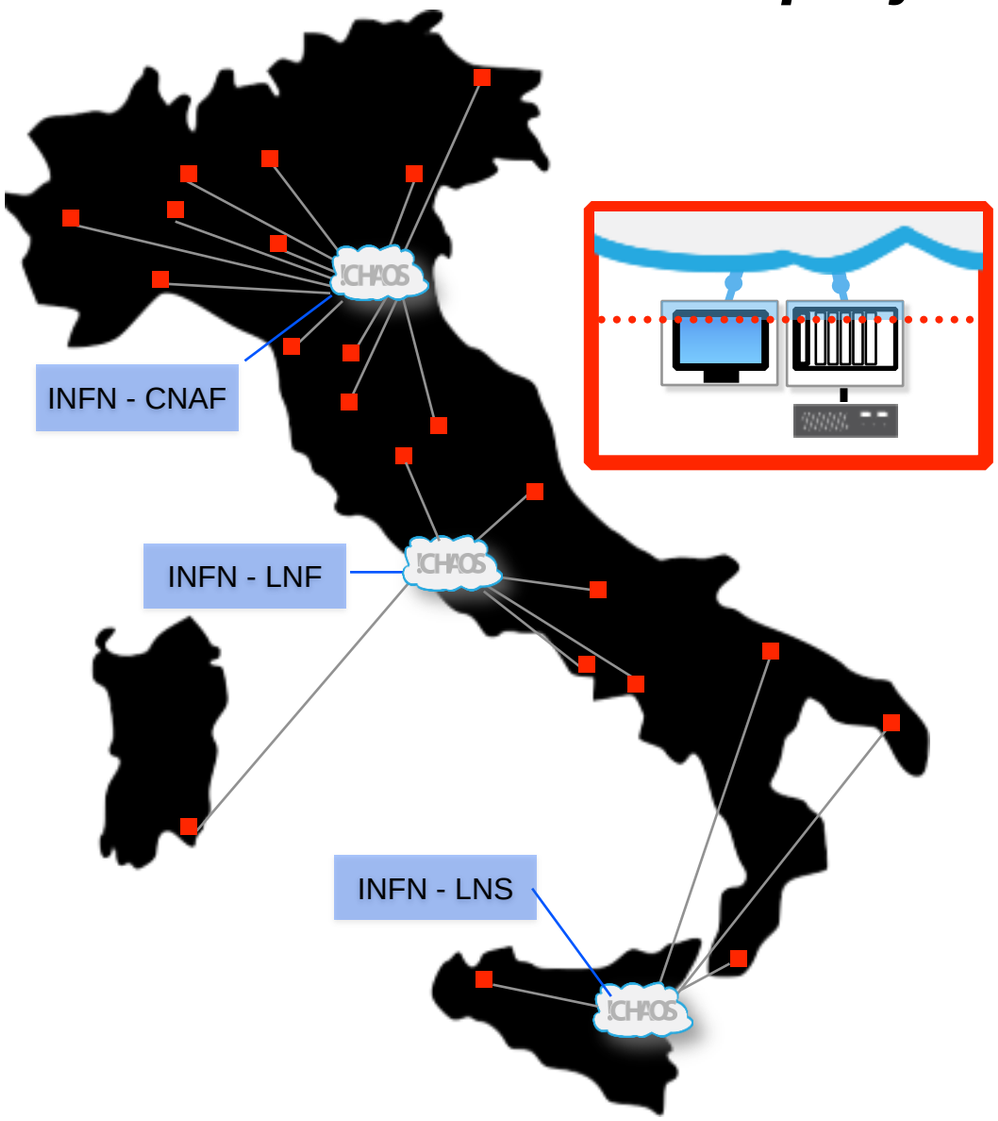


!CHAOS Overview
Framework Status

- CU toolkit prototype ready
- UI toolkit prototype ready
- MDS java simple prototype ready, but we want to rewrite in C++
- Live data prototype ready
- EU toolkit under development
- History data under development

“Control as a (cloud) Service”

R&D Cloud project just financed by Italian Ministry of Research



INFN-LNF (Laboratori Nazionali di Frascati)
INFN-TV (Sezione di Tor Vergata)
INFN-PG (Sezione di Perugia)
INFN-CNAF (Centro Nazionale Tecnolgie Informatiche)
INFN-PD (Padova)
INFN-LNS (Laboratori Nazionali di Catania)
National Instruments (NI)

ADF
Solaris

!CHAOS as infrastructure at **national** level which realizes a cloud of services distributed and shared over the LAN/WAN, which allows the **monitoring and control** of a large number of device/intelligent components and provides processing and archiving resources.

National Instruments collaboration

!CHAOS inside LV

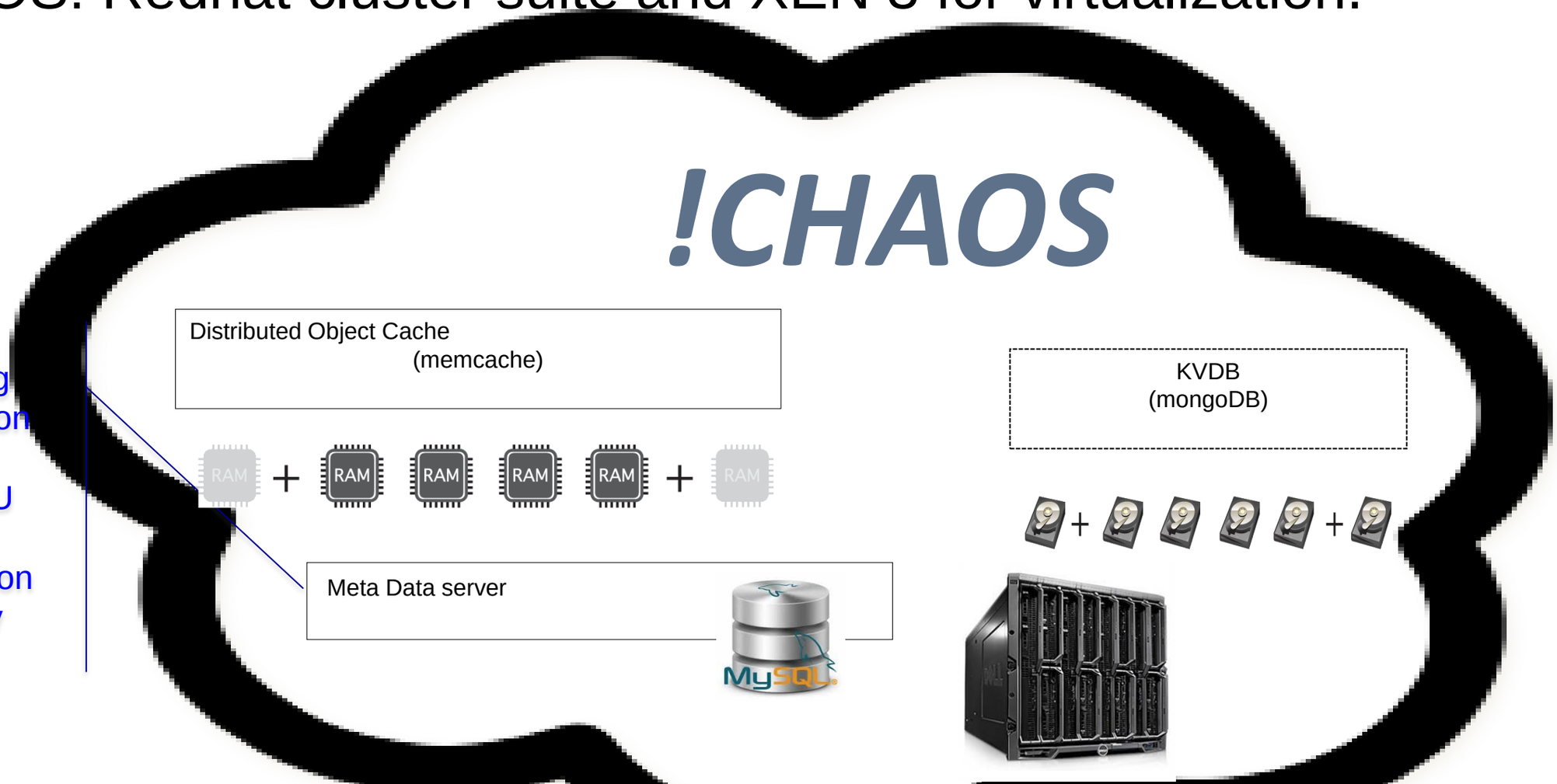
- To easily build Graphical interface
- Reuse existing lot of legacy LV code developed by INFN

LV inside !CHAOS

- To build drivers in LV
- Reuse existing legacy LV code developed by INFN

Work in Progress
Test Bed HW/SW @INFN-LNF

- !Chaos Cloud hosted on a DELL blade system with 9 blade machines
- OS: Redhat cluster suite and XEN 3 for virtualization.



Information Node
CU publishing
CU configuration manager
Sematic of CU data
Node information
Node Activity
Security
…

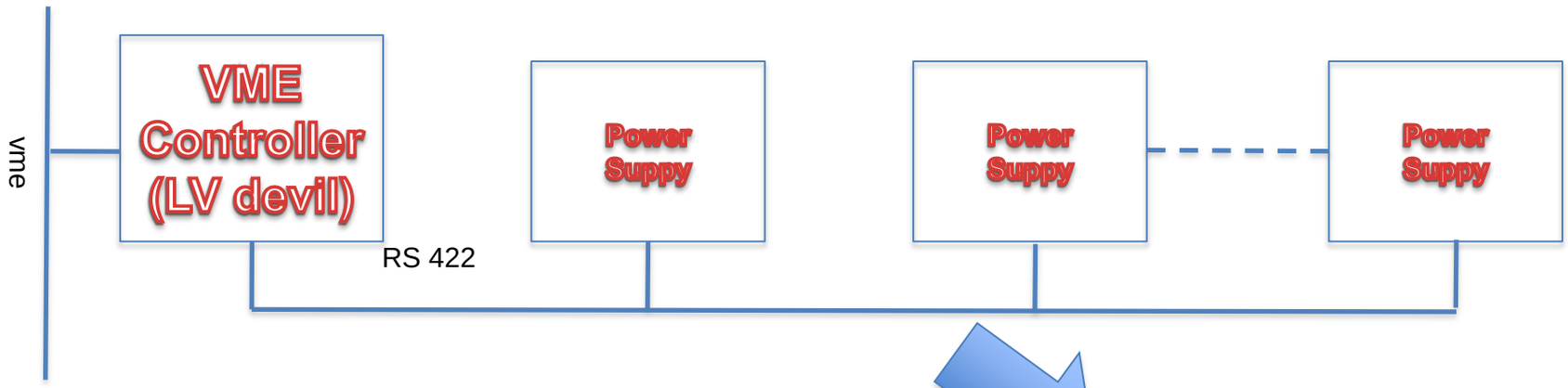
Work in Progress
FIRST !CHAOS RUN
15 Nov 2013 @INFN-LNF: BTF

!CHAOS replaced for one day traditional control on the transfer line.

The beam accelerated by the linac can be deviated trought the highlighted transfer line to an experimental wall: Daϕne BTF.

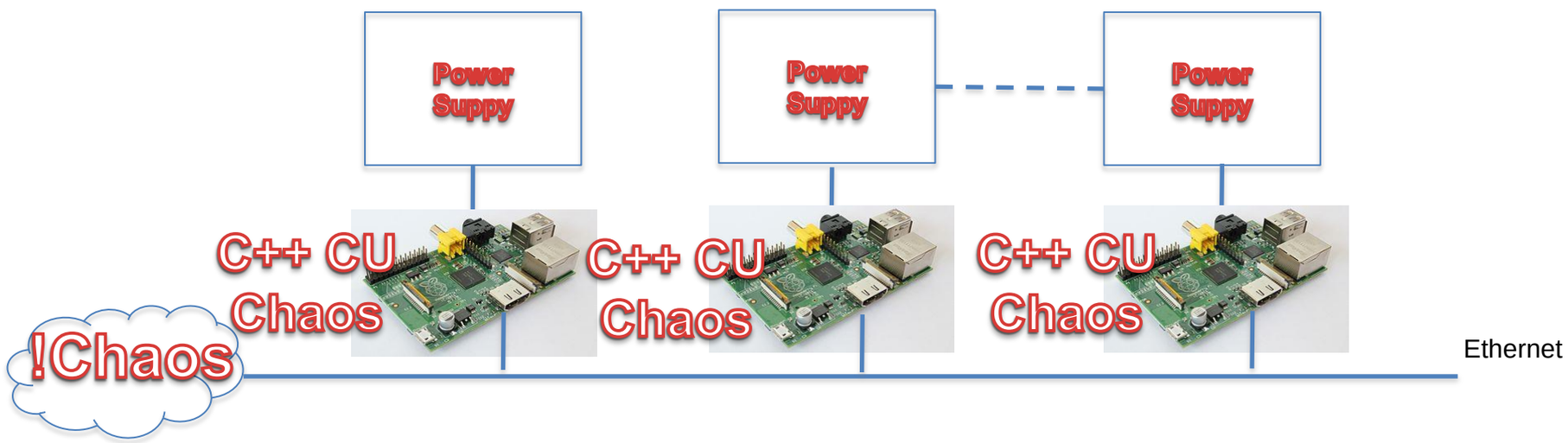
Work in Progress
Test Bed HW/SW @INFN-LNF

Control of power supplies through !CHAOS and cheap embedded controllers. Replacing dated and expensive VME HW



Today

End 2013



Conclusions

Preliminary studies show that cutting-edge Internet Technologies such as DOC and non relational DB can be profitably used also in a control and DAQ system.

!CHAOS incorporates these emerging technologies but it is not locked to a particular implementation.

!CHAOS is by construction a scalable and extensible control system infrastructure, providing the services for communication, data archiving, timing and information.

Thanks for the Time

!CHAOS Group

Claudio Bisegni,

Luciano Catani, Giampiero Di Pirro, Luca Foggetta, Riccardo Gargana, Matteo Mara, Giovanni Mazzitelli, Andrea Michelotti, Antonello Paoletti, Stecchi Alessandro, Federico Zani

thesis's

Flaminio Antonucci (TV), Andrea Capozzi (TV), Francesco Iesu (Cagliari Informatica),