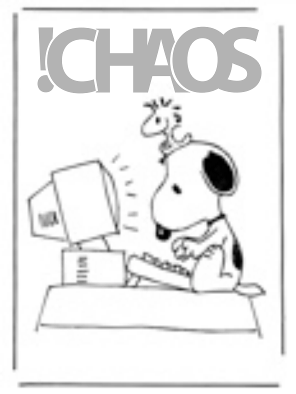
ACCELERATOR COMPUTING INFRASTRUCTURE & CONTROLS R&D INTRODUCTION



R. Amendola (INFN Roma TV), **C. Bisegni** (INFN-LNF), **S. Calabrò** (LAL & INFN-LNF), **L. Catani** (INFN Roma TV), **P. Ciuffetti** (INFN-LNF), **G. Di Pirro** (INFN-LNF), **L. Foggetta** (LAL & INFN-LNF), **G. Mazzitelli** (INFN-LNF), **A. Stecchi** (INFN-LNF), **F. Zani** (INFN Roma TV)
in collaboration with:
LNF computing services, LNF Accelerator Division Controls and Diagnostic Services, Padova timing group

ACCELERATOR COMPUTING INFRASTRUCTURE

design, develop and maintain a computing infrastructure with the following purpose: implementation and maintenance of an **Electronics Management Data System** (EMDS) dedicated to the storing and presentation of all project documents, cads, etc; implementation and maintenance of a **Project Management Data System** (PDS) in order to efficiently allocate and monitors efforts and costs; develop a common infrastructure and **tools with the experiment** in order to share and correlate data; implementation and maintenance of **accelerator simulation code** FARM/TIER2 share; implementation and maintenance of **servers and services** needed for the **accelerator controls**

SOFTWARE INFRASTRUCTURE, CONTROL SYSTEMS

design and implementation of the **controls system**; development and implementation of the **drivers**, and interface with accelerators device; development and implementation of the **user interface** and **high level** accelerator softwares; development and implementation accelerator infrastructure interface to **monitor and control subsystems** device like PLC, field bus, etc (electrical, fluid, etc installations); design and develop accelerator **simulation code interface** and controls systems in order to permit an easy and standardized data flow; implementation and development of an accelerator **logbook and trouble ticketing** system in order to monitors, store and allows statistics on accelerator devices and subsystems; design and develop **web tools for public** and private data presentation and correlation, **online analysis, and monitoring**.

USERS INFRASTRUCTURE, REMOTE CONTROL ROOM

The infrastructure previously introduced (hardware and software) requires to develop **identification and security** tools and the implementation of **collaborating tools** for the community participating to the project.
In the mean time, the international community interested in the development of the accelerator, push also to foreseen a **Remote Control Room** in order to permit and guarantee participation in the operation and high efficiency in diagnostics and fault solution

ACCELERATOR COMPUTING FARM

- has been installed a computer **FARM** dedicated to accelerators simulation & calculation code

- 5/16 slot rack equipped with blede 2 processor Intel Xeon X5660, 64 bit esa-core,2,80 GHz, 48 GB RAM, FiberChannel, GigabitEthernet dual.

  - simulation and calculation code: HFSS, GdFidL, MatLab, Mathematica, OPERA, ORCAD, inventor, FLUKA, GEANT, MCNPX, ANSYS

  - Controls R&D: Labview, memcache, mongoDB, etc

- hardware has been installed in April 2011. FARM configuration under the LNF computing infrastructure is going on. Software installation and configuration are also started.

SUPER**B** TIER2 STARTUP@LNF

- a VO for **SuperB** is starting at LNF thanks to a collaboration with **ATLAS@LNF TIER2** resources and personnels and LNF computing service and infrastructure

- the share will be addressed to **accelerator and experiment** purposes: MDI and backgrounds, CMAD for e-cloud and IBS simulation, dynamic aperture calculation, etc as well as fast, full and GARFIELD simulation

BACKGROUND...

- Frascati and Tor Vergata group have a **long experience in design, develop and implementation** of innovative controls systems. in ’90 for DAFNE the first PC and LABVIEW based controls systems has been successfully developed and operated making a braking trough in the concepts of controls

- experience and knowhow have continued on SPARC and contributed in others accelerators in the world and are **available for a new challenging business**

- the two Frascati running accelerators, DAFNE and SPARC, offer a *natural gym* to **study** a new and innovative control system, to develop the core, to test the critical parts and the software and hardware needs

- Frascati and Tor Vergata offers unique **infrastructures** and a large amount of computing **resources** interested in the project

- the project has been presented in INFN CSN5 and triggered the private company *National Instruments* to collaborate in the development

NATIONAL INSTRUMENTS

CONTROLS SUPER**B** R&D

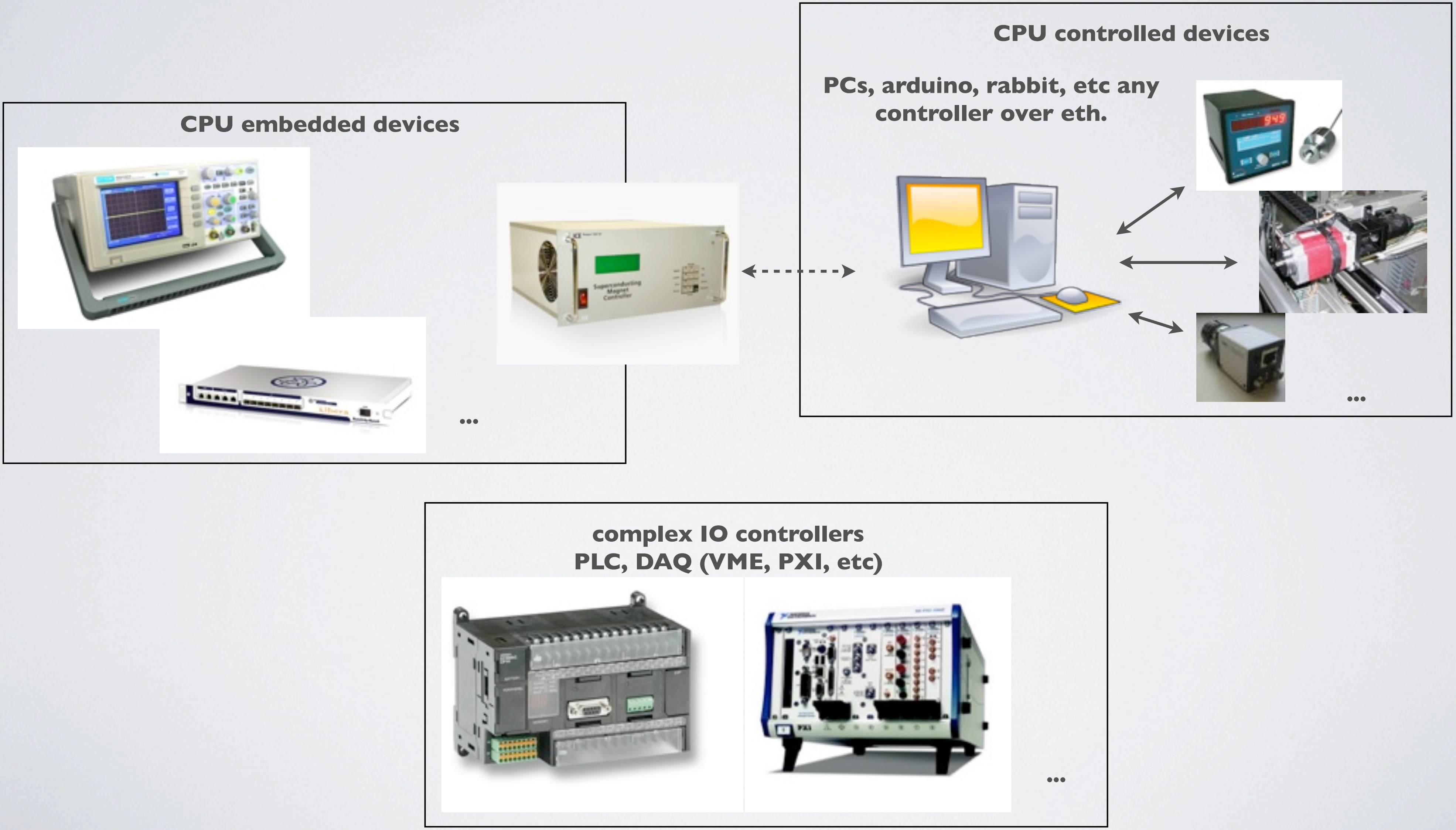
- follow today **software trends** dominated by web technologies and services where ***large database*** are used and very ***high throughput*** is needed on the largest and robust available data bus: ***ethernet***
- be free to implement **any kind of devices** reducing the hardware dependence and development time
- exploit the availability of many **programable cpu embedded devices**
- be able **controls** and where needed **acquire** data with performance limited only by hardware availability

→ move from polling to pushing based system
introducing new different feature to be exploited

IDEA...

- design a system where use the knowhow and tools coming from large amount data handling like in google, facebook, etc that means **no relational DB** where store live and history data with very high performance.
- all devices are completely independent and auto-configuring directly (semantics and syntax) in a **metadata server** allowing easy and fast data retrievement
- development on any different software and hardware platform
- to produce a **Control System Library** permitting to reduce the development needs only to the core part connected to the specific hardware device

EMBEDDED / CONTROLLED DEVICES



CPU embedded devices

...

CPU controlled devices

PCs, arduino, rabbit, etc any controller over eth.

...

complex IO controllers
PLC, DAQ (VME, PXI, etc)

...

DATA CATEGORIES AND THROUGHPUTS

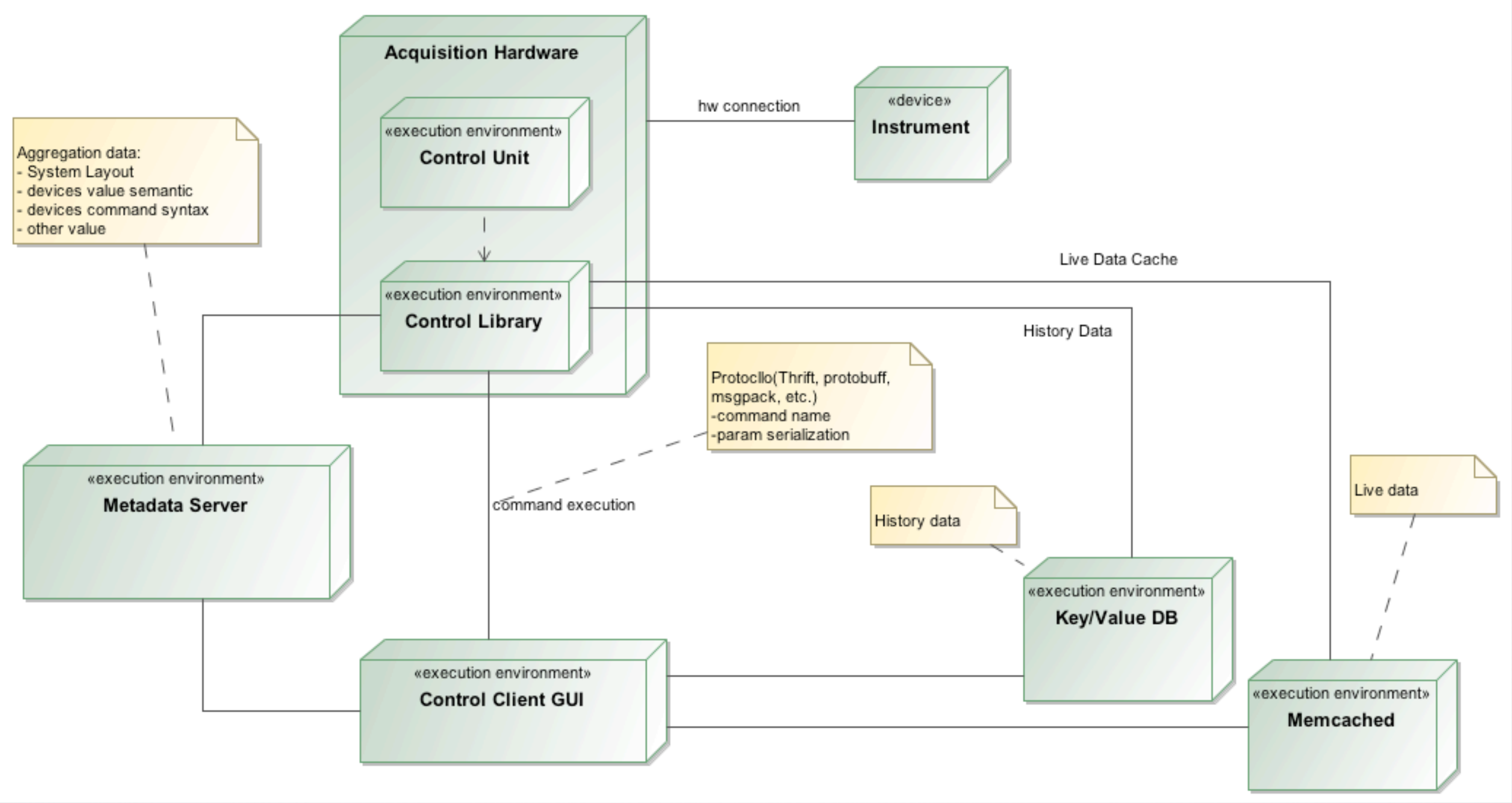
- data can be divided essentially in three different type:
  - **slow data** (a few bytes @ Hz)
    - eg: magnets, vacuum, temperature, etc
  - **fast data** (Kbytes of bytes @ kHz)
    - eg: BPM, beam lost monitor, luminosity monitor, synchronized bump, etc

→ continuos data

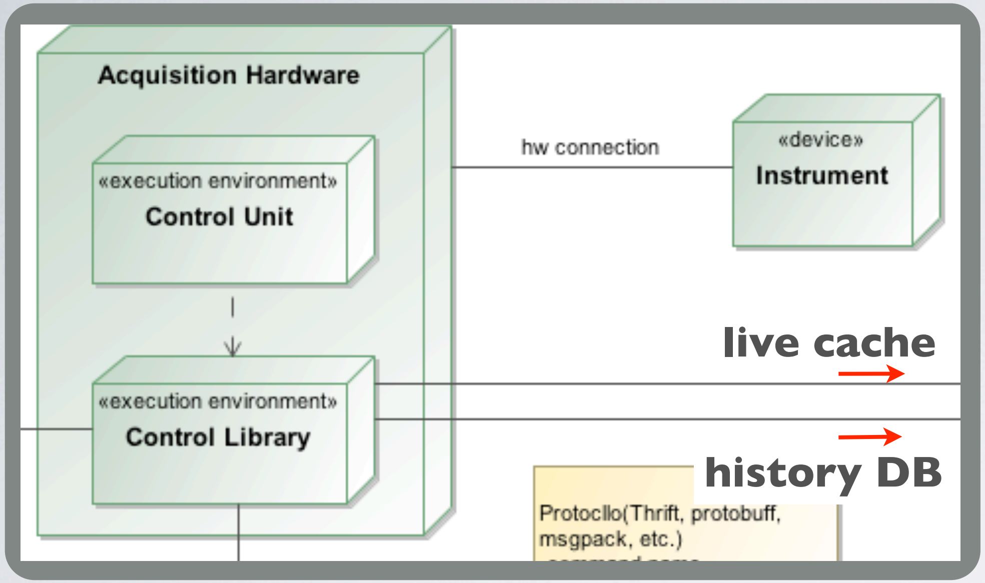
- **very fast data** (Mbytes @ GHz)
  - eg: BPM single pass, scope, RF, etc

⇢ data bursted, limited by hardware and software dead time

SYSTEM DATA FLOW



FRONT END



- **CPU embedded devices**
- **CPU controlled devices**
- **Complex IO controllers PLC, DAQ (VME, PXI, etc)**

The **Control Unit** (*CU*) is the user software (*driver*) to be interfaced with the **Control Library** (CUCL) a multi task process that provides:

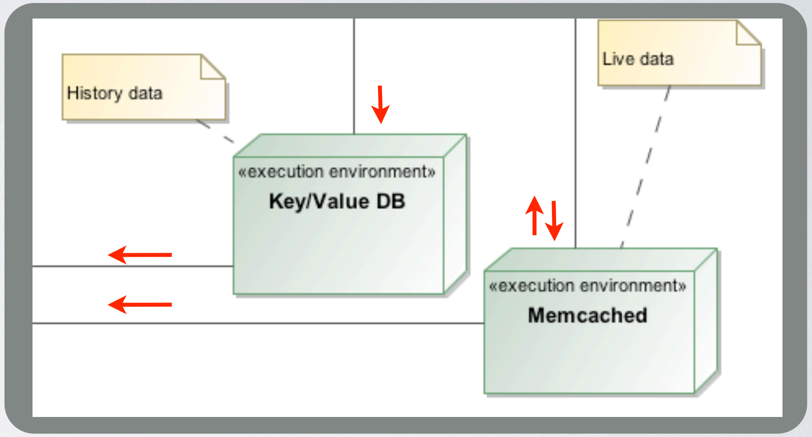
- to handle input (*command*) and output (*readout*) data;
- to initialize and configure data flow (type, frequency, etc)

the **front end** gets device configurations from the **meta data server** where in mean time it auto-configure all data semantics and syntax

LIVE CACHE AND HISTORY DB

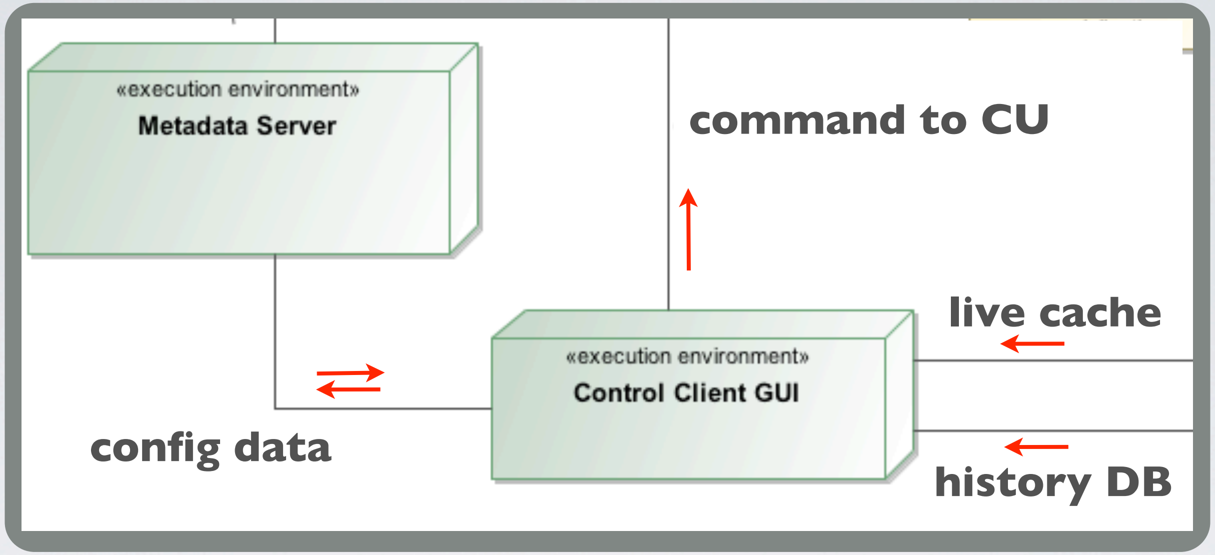
Data acquired by CU (*cu clock*) are updated in two **no relational DB** (*key/value*):

- **live-cache** (*live clock*)
- **history** (*history clock*)



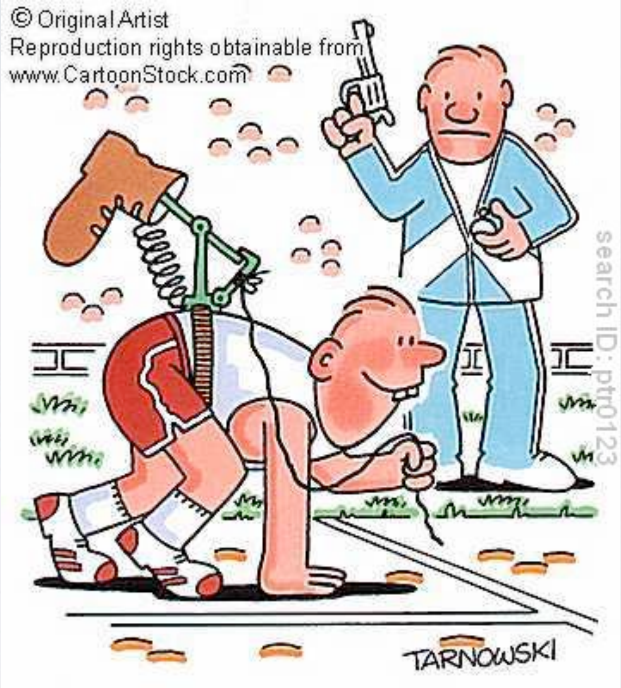
for bought the solutions, candidates under tests are two free open-source software: **MongoDB** - from "humongous" - is a scalable, high-performance, open source, document-oriented database & **Memcached** a free & open source, high-performance, distributed memory object caching system

USER INTERFACE TOOLKIT



The **User Interface Toolkit** (UITK) retrieves all configuration information to access data and control devices from the **Metadata Server** previously updated by front end; The Graphic User Interface provides the live and archived data representation and correlation

TIMING



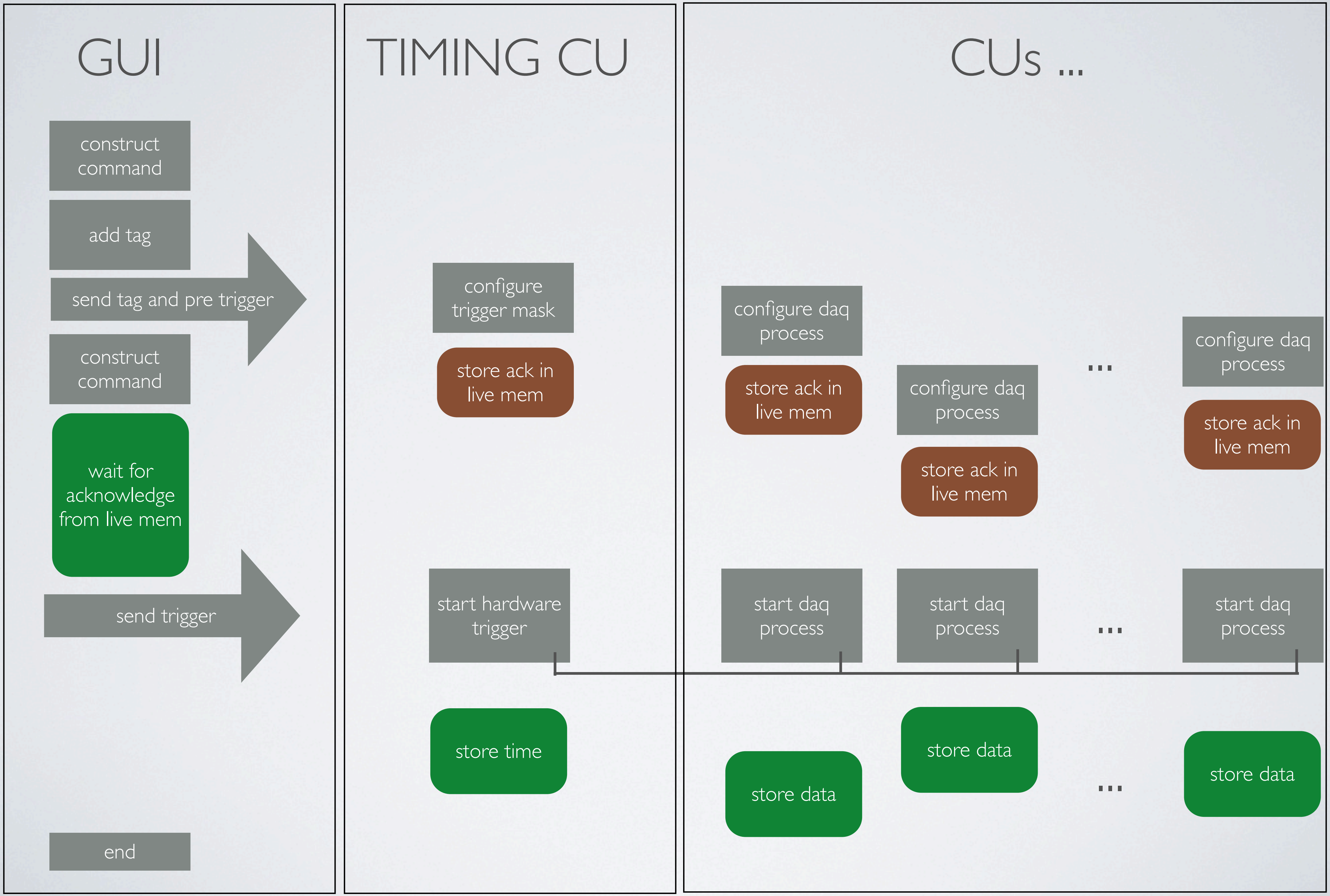
- **TAG** events data with **µs** precision
- **synchronize** (jitter) data with **ps** precision
- allows maximum **repetition rate** with a minimum dead time respect to accelerator event determinate by injection frequency **100 Hz**

real timing requirements are not jet fully defined...

SIDC DATA TIMING

- **TIMESTAMP** of any controller/device is **synchronized** (NTP/PPT or custom solution/controller)
- a timing system distribute and provide hardware trigger (TTL/NIM) to any different controller/device needs a timing accuracy greater then milliseconds
- **PRE TRIGGER** command mask configure controllers/devices to execute a specific task and pre configure the **timing controller** to dispatch a specific mask to the controllers/devices.
- any pre trigger mask is flagged with a specific **timing TAG**
- **TRIGGER** command to timing controller latch time stamp and send hardware trigger to controllers/devices
- data from controllers/devices and timing controller are **updated** with their own duty cycle in the live/history data

DAQ TIMING FLOW



GUI

construct command

add tag

send tag and pre trigger

construct command

wait for acknowledge from live mem

send trigger

end

TIMING CU

configure trigger mask

store ack in live mem

start hardware trigger

store time

CUs ...

configure daq process

store ack in live mem

configure daq process

store ack in live mem

...

configure daq process

store ack in live mem

start daq process

start daq process

...

start daq process

store data

store data

...

store data

!CHAOS PURPOSE

define a new SidC topology with the following futures:

- **redundancy** of all its parts; intrinsic **scalability**; **no point of failure**; hardware (device to be controlled) **hot-integration** and **auto configuration**
- integration in the SidC structure (library) of **triggered DAQ** operation mode
- based on a **distributed object caching** for real-time data access (Live Database)
- based on **no relational** database-oriented archiving data (History Database)
- abstraction of the structural components of the control to **reduce dependence on the particular HW and SW**, allowing for an extreme adaptability
- compatibility with **commercial standard** and custom components and any **future developments**

more on howto in the next talk...

CONCLUSION

- **Controls R&D**, based on this new concepts and knowhow of INFN accelerator personnels, **started** - see next talk - and we are open to match and integrate new and different ideas, experiment peoples feedback and to collaborate on the common task

- Accelerators **computing infrastructure is under design**, and need to be soon interface with the experiment and integrated with diagnostics, timing and other scientific hints and requirements

- We are working to **widen the community** of people interested in developing the accelerator computing infrastructure, codes, controls and diagnostic, drivers, etc.