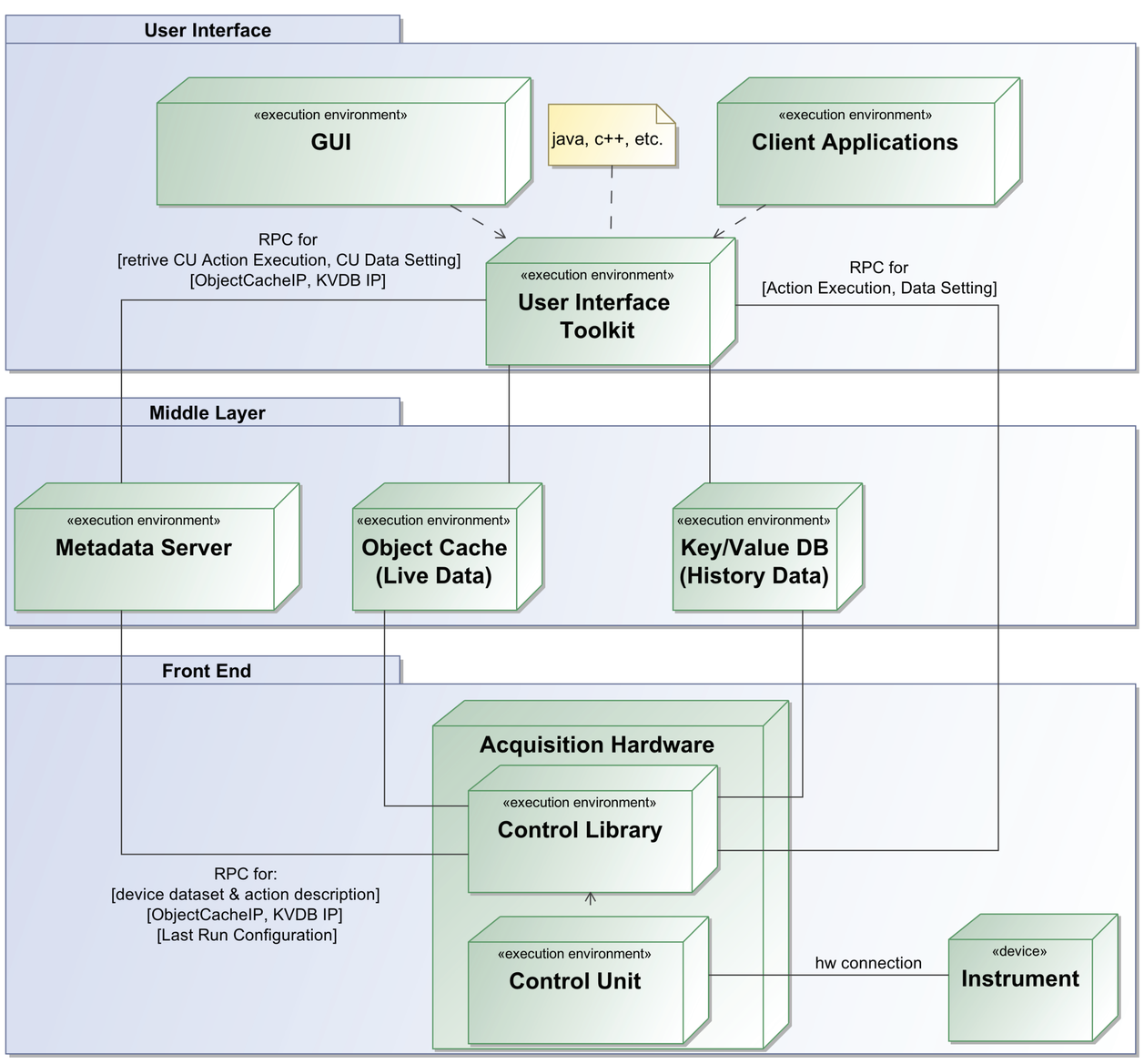
# HIGH PERFORMANCE WEB APPLICATIONS FOR PARTICLE ACCELERATOR CONTROL SYSTEMS

*G. Mazzitelli, C. Bisegni, P. Ciuffetti, G. Di Pirro, A. Stecchi - INFN/LNF, Frascati (RM), Italy;*
*S. Calabrò, L. Foggetta - LAL-CNRS, Orsay, France & INFN/LNF, Frascati (RM), Italy; L. Catani, F. Zani - INFN-Roma Tor Vergata, Roma, Italy*

**The integration of *web technologies* and applications has been one of the major trends for the development of new services for Control Systems (CS) of particle accelerators and large experimental apparatuses. Nowadays, high performance web technologies exhibit some features that would allow their deeper integration in a CS and their employment in developing CS' core components.**
**We present a preliminary investigations of a new paradigm for a particle accelerators CS, named *!CHAOS*, and the associated machine data acquisition system based on a synergic combination of network distributed cache memory and a non-relational key/value database.**

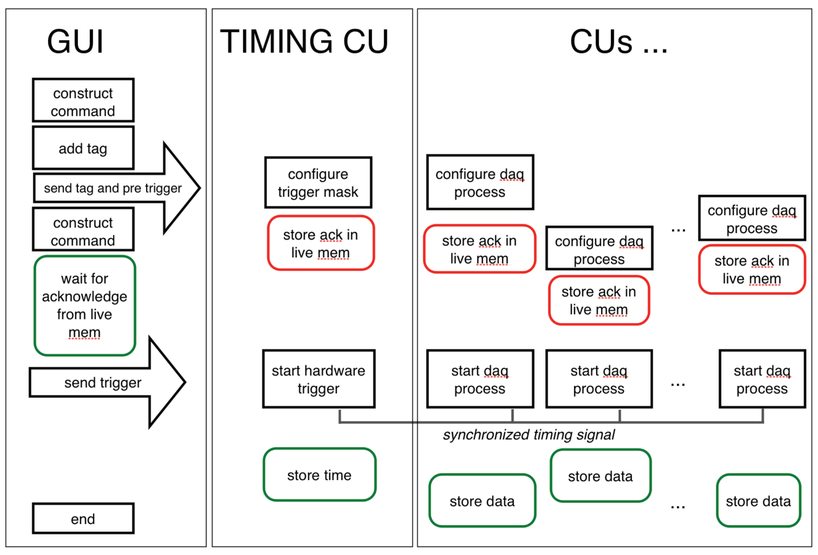



The aim of **!CHAOS** (**C**ontrol **S**ystem based on **H**ighly **A**bstracted **O**perating **S**tructure – but not a mess!) is to provide a solution that naturally allows:

- redundancy of all its parts
- intrinsic scalability
- minimization of points of failure
- hardware hot-integration and auto configuration.

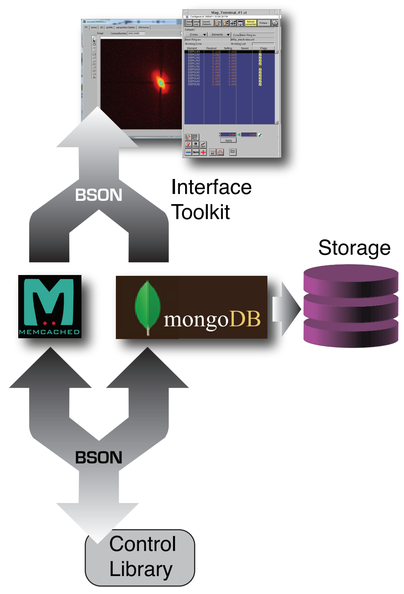
!CHAOS, basically developed in C++, employs distributed **object caching** for real-time data access (Live Database) and a **key-value database** for data archiving (History Database), continuously filled by data **pushed** from acquisition hardware.
A **Control Library** (CL) completely manages data and commands flow, the control processes and the devices configuration. The device's programmer is only asked to develop the driver for the specific controlled hardware, called **Control Unit**, that is the only part that instances the !CHAOS abstraction. The CL also provides the **syntax** and **semantics** for dataset and command to the **Metadata Server** that allows the correct information retrievement.
The CL takes care of the data **serialization**, communication with databases, handling of system's and client's commands and standard services of CS.
**User Interface Toolkit** provides the client applications (display panels, measurement applications, etc.) with the interface to the CS framework for retrieving configuration information from **Metadata Server**, accessing live data (from cache db) and archived data (from history db), sending commands to devices, etc

**!CHAOS** is designed in such a way to **accommodate any kind of devices** to reduce the hardware dependence and to drop the development time by exploiting the availability of many devices with embedded programmable CPU.




Furthermore, the CS has to be able to **control** and - where needed - to **acquire** data with e performance limited only by the hardware capability. This is guaranteed by the possibility to easily **trigger** the hardware and synchronize operations among distributed components.

Data serialization strategy adopted is **BSON**, a binary-encoded JSON (JavaScript Object Notation) documents, optimized for fast storage performance.
Two **open-source** software, wich allow scalability and redundancy, are currently under tests as candidates: **Memcached** for the live data object caching and **mongoDB** for the history key-value database




The Control Systems, presently driving the DAΦNE and SPARC accelerators in the National Laboratory of Frascati (INFN) have been used for testing one of the core components of !CHAOS: the live-data caching. At DAΦNE the front-end controllers acquire the data from the devices and continuously update - through fiber optics - a central common VME address space, which is also accessible from the user's applications. For our tests, the routines used to update the devices' set values have been easily modified in order to write into a memcached server.
The preliminary tests performed in a real environment and on real elements under control in the two accelerators, have confirmed that the performance of a no-relational database resident on RAM is practically limited only by Ethernet bandwith. The systems load is very low, while redundancy and scalability allows being confident on the behaviour for a larger accelerator complex such as the SuperB.

NATIONAL INSTRUMENTS LabVIEW

A first release of the CL has been developed on Linux and OS X Operative Systems and it is also under integration with the LabVIEW environment.

The !CHAOS project is part of **SuperB R&D**, an unique opportunity for breakthrough innovation in particle accelerators technologies.

*We would like to thank the Padova group of M. Bellato et al, M. Serio and A. Drago who are starting to collaborate on the timing issues related to the SuperB controls and accelerator synchronization, as well as the LNF computing centre for the support provided in system administration issues and networks analysis and implementation. M. Biagini and P. Raimondi for the encouraging start up support; M Giabbai for the editing review of the paper.*