

Control system based on a
Highly
Abstracted and
Open
Structure



!CHAOS is a **INFN** experiment aimed to defining and validating a new paradigm for distributed control systems (DCS) and fast data acquisition (DAQ) for particle accelerators and experimental apparatuses.

Indeed, its flexibility allows a much wider spectrum of applications.

Goals:

- design an innovative control system based on the **newest software technologies**
- define a **new services topology** for controlling devices and data acquisition
- define a **new transfer mode for live data** (on-line display and control of devices), as alternative to client/server typically used in the DCS
- **high scalability** for each core service;
- **high abstraction** of devices control software;

Dip. Matematica
Roma TV



industrial
partners



INFN-Padova

SLAC

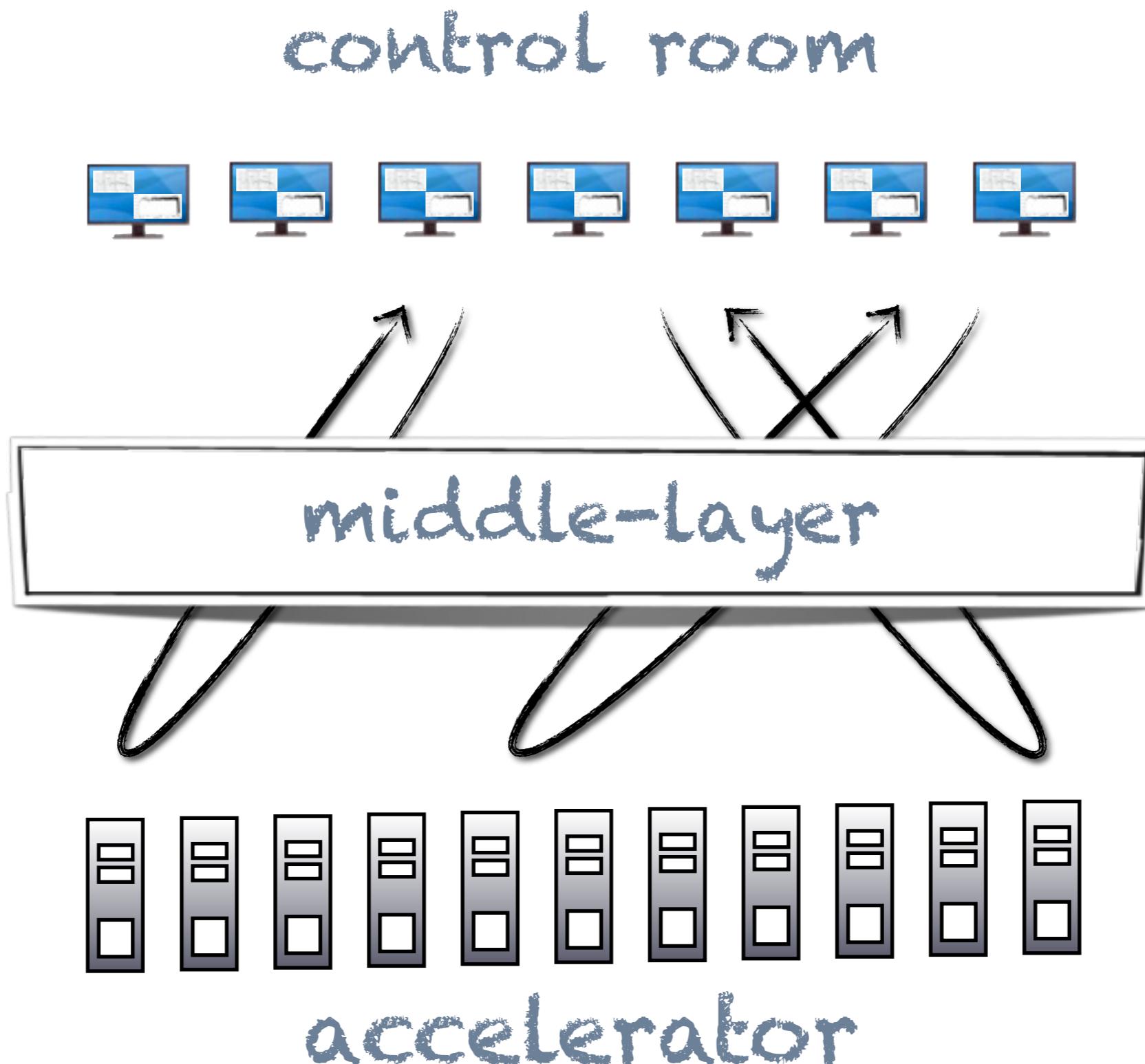


Bari Politecnico

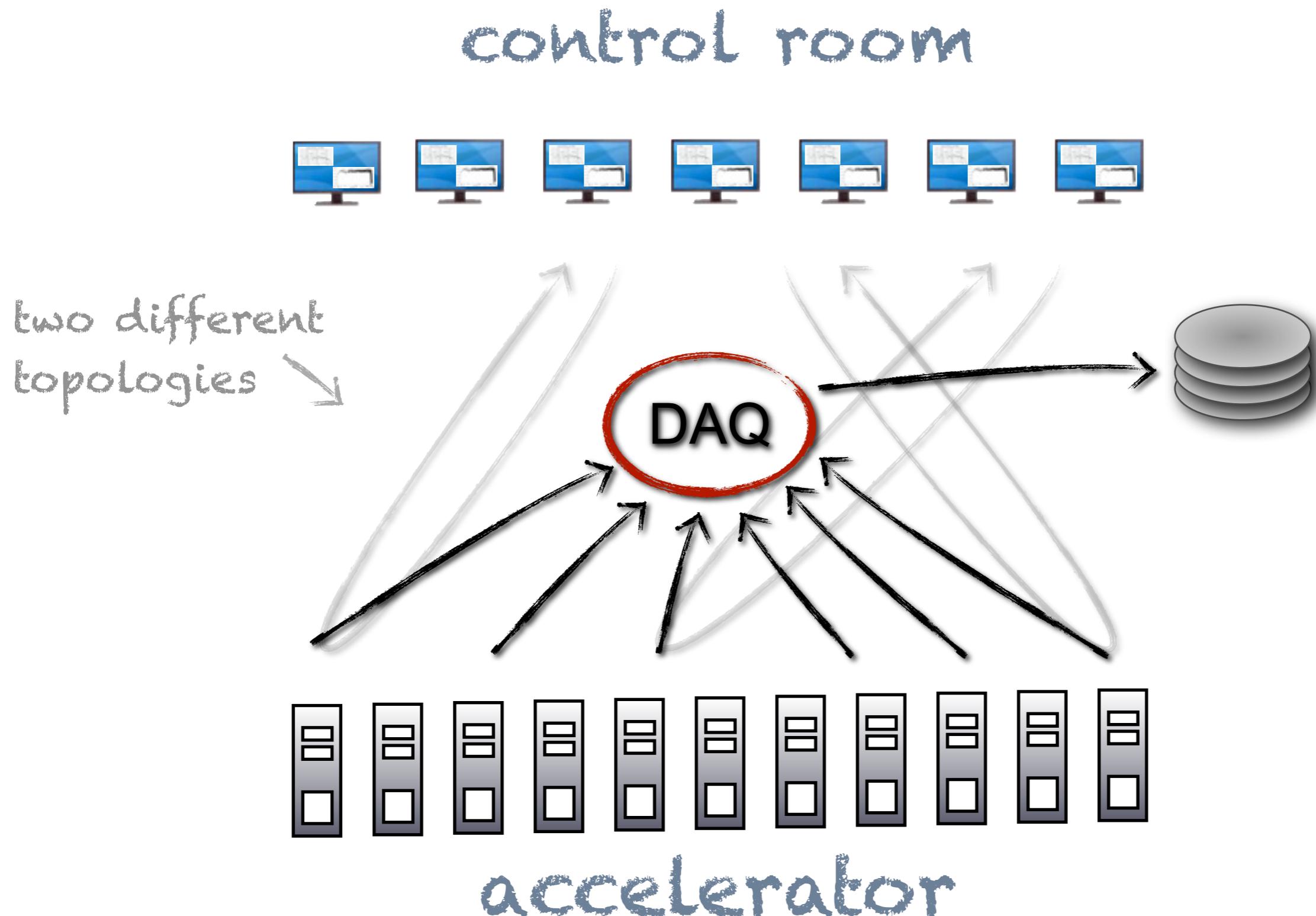
Bari Università & INFN



“the standard model”

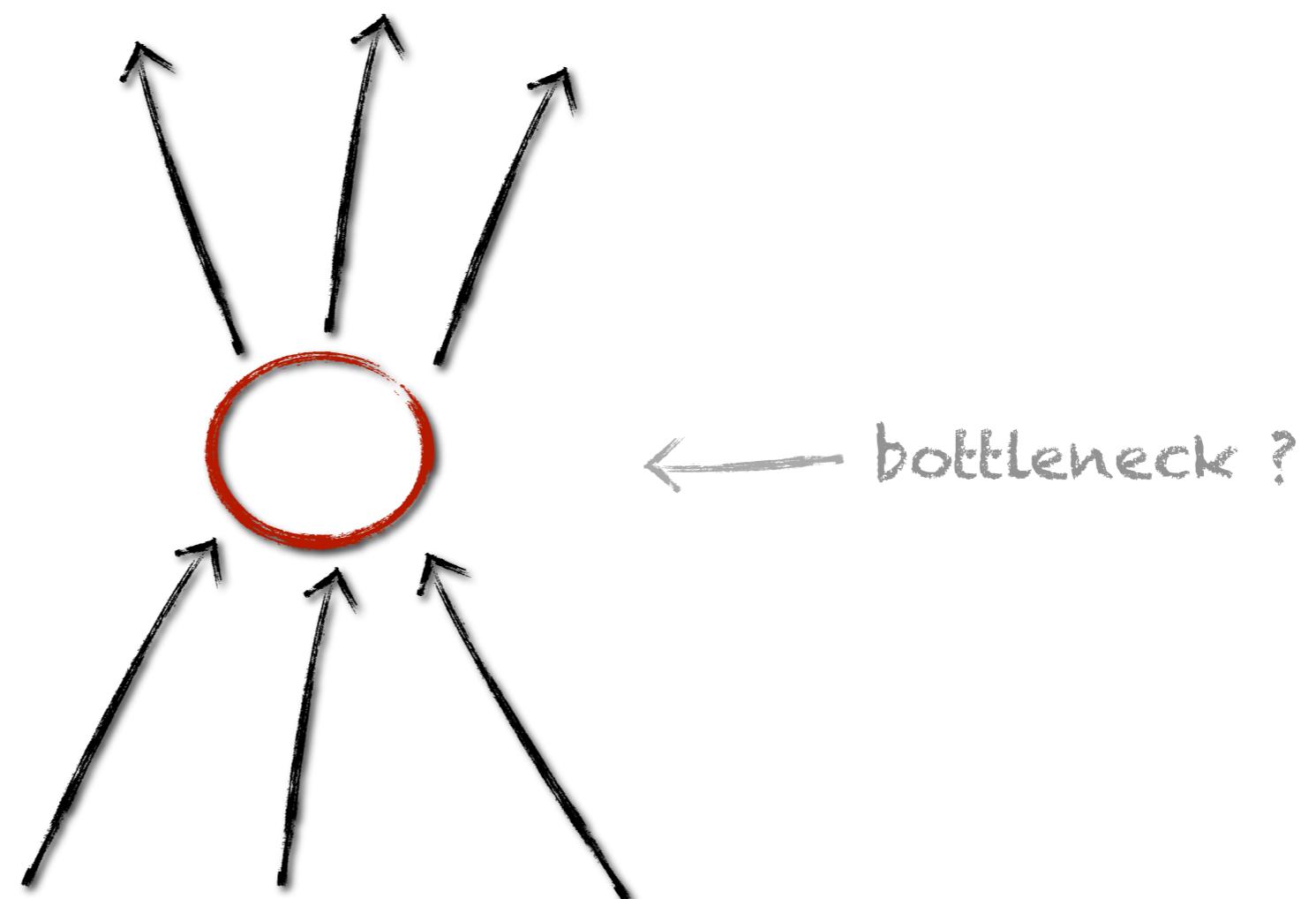


“the standard model”



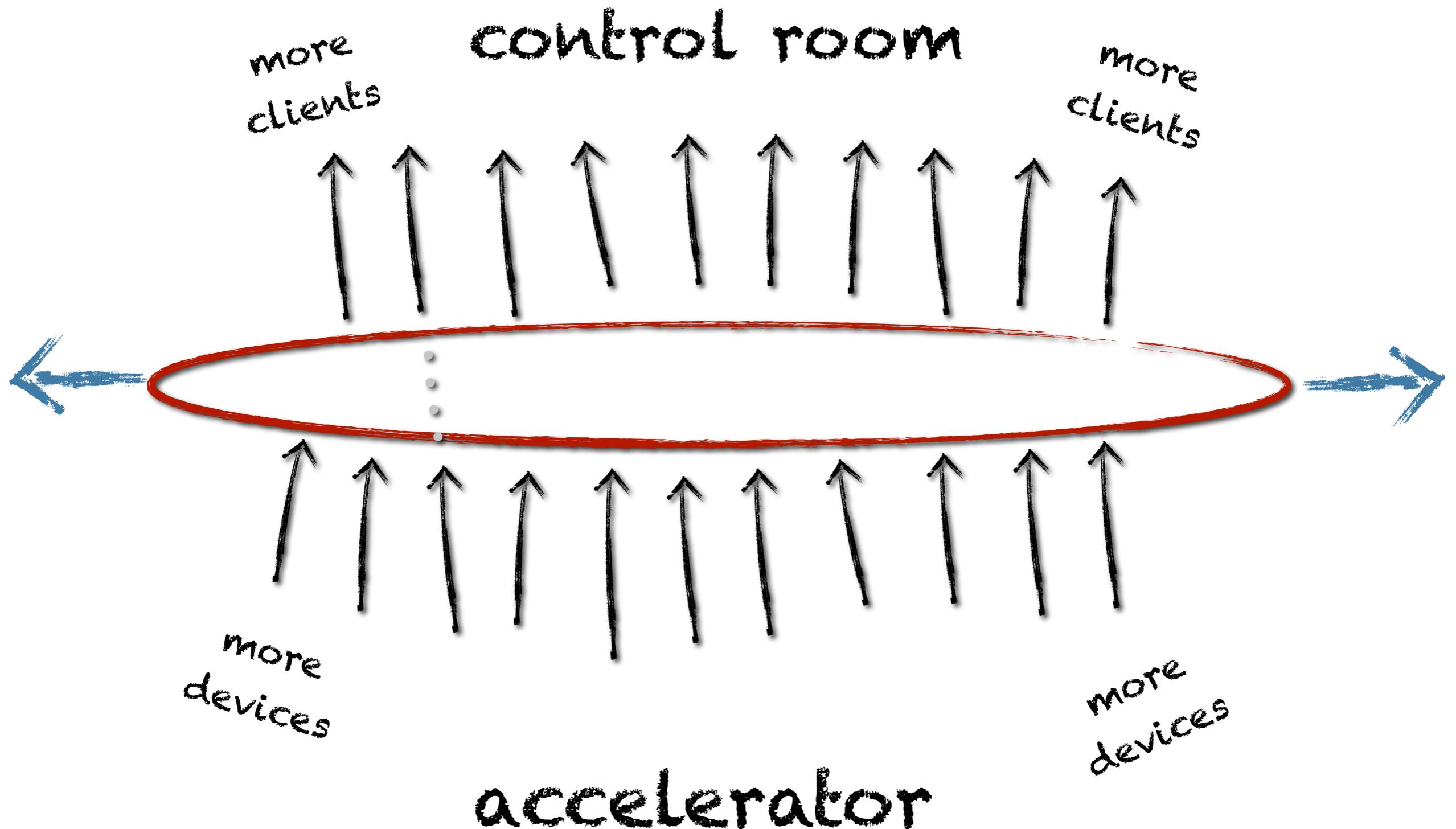
the new paradigm

control room

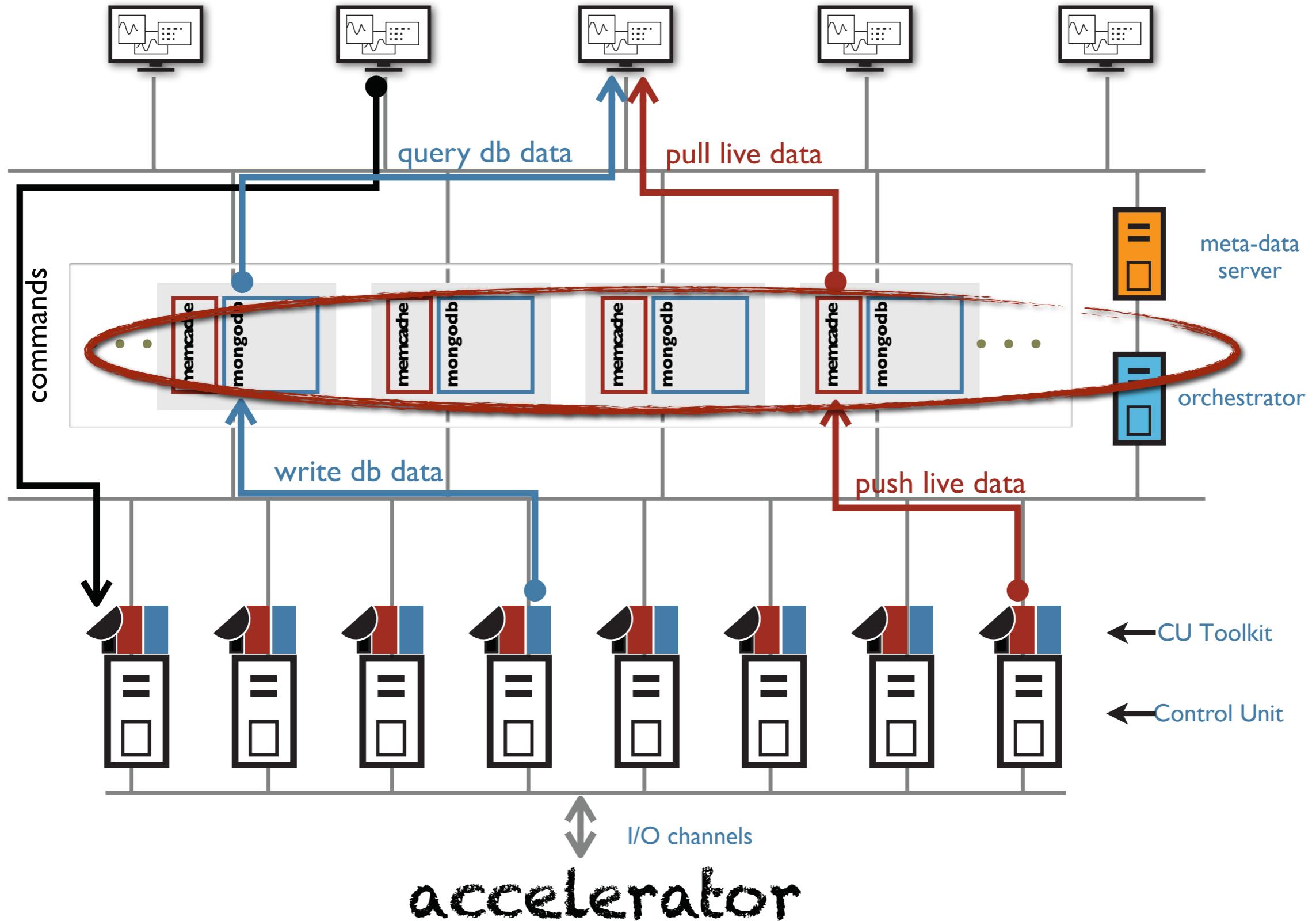


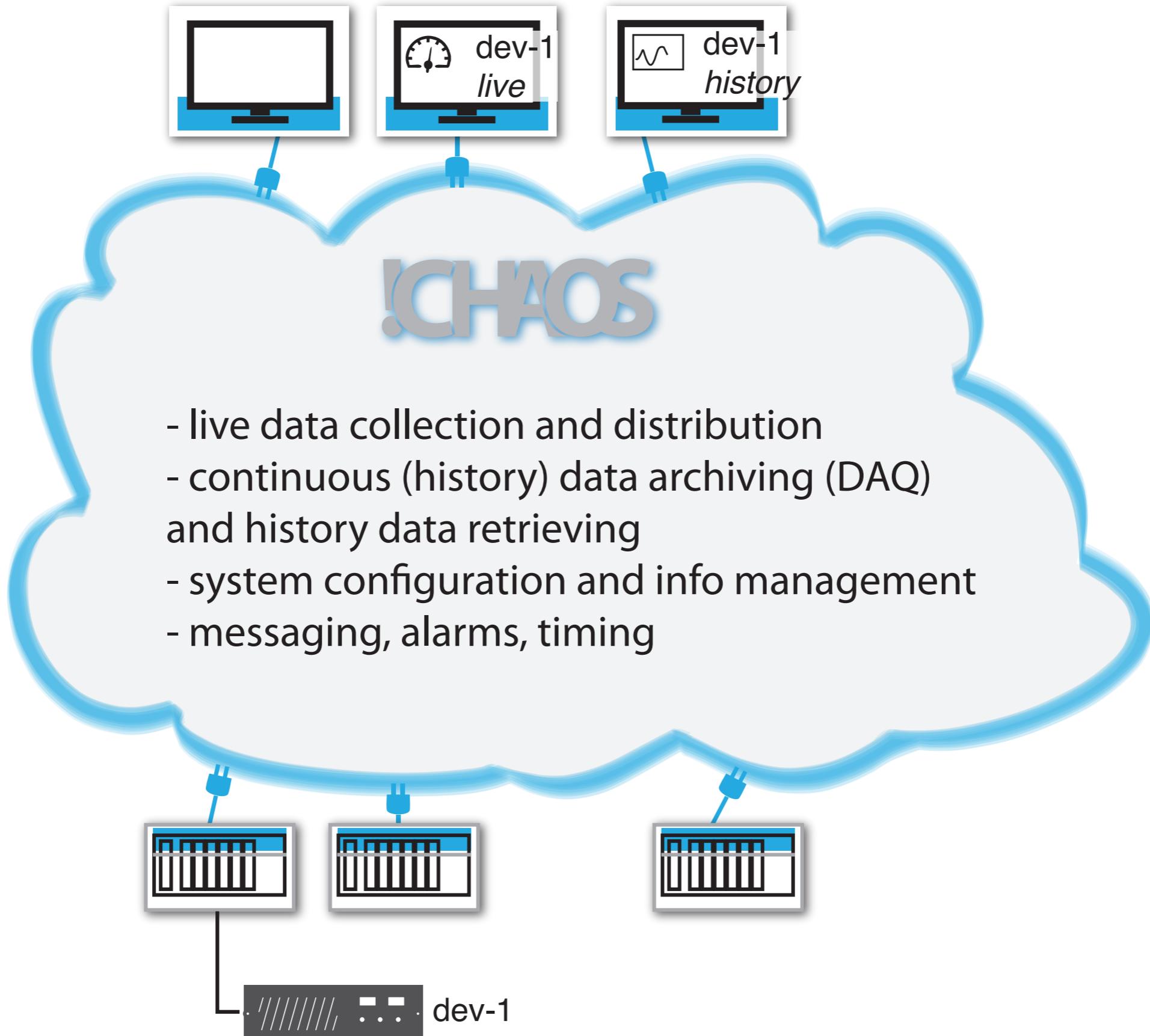
accelerator

the new paradigm: scalability eliminates bottleneck

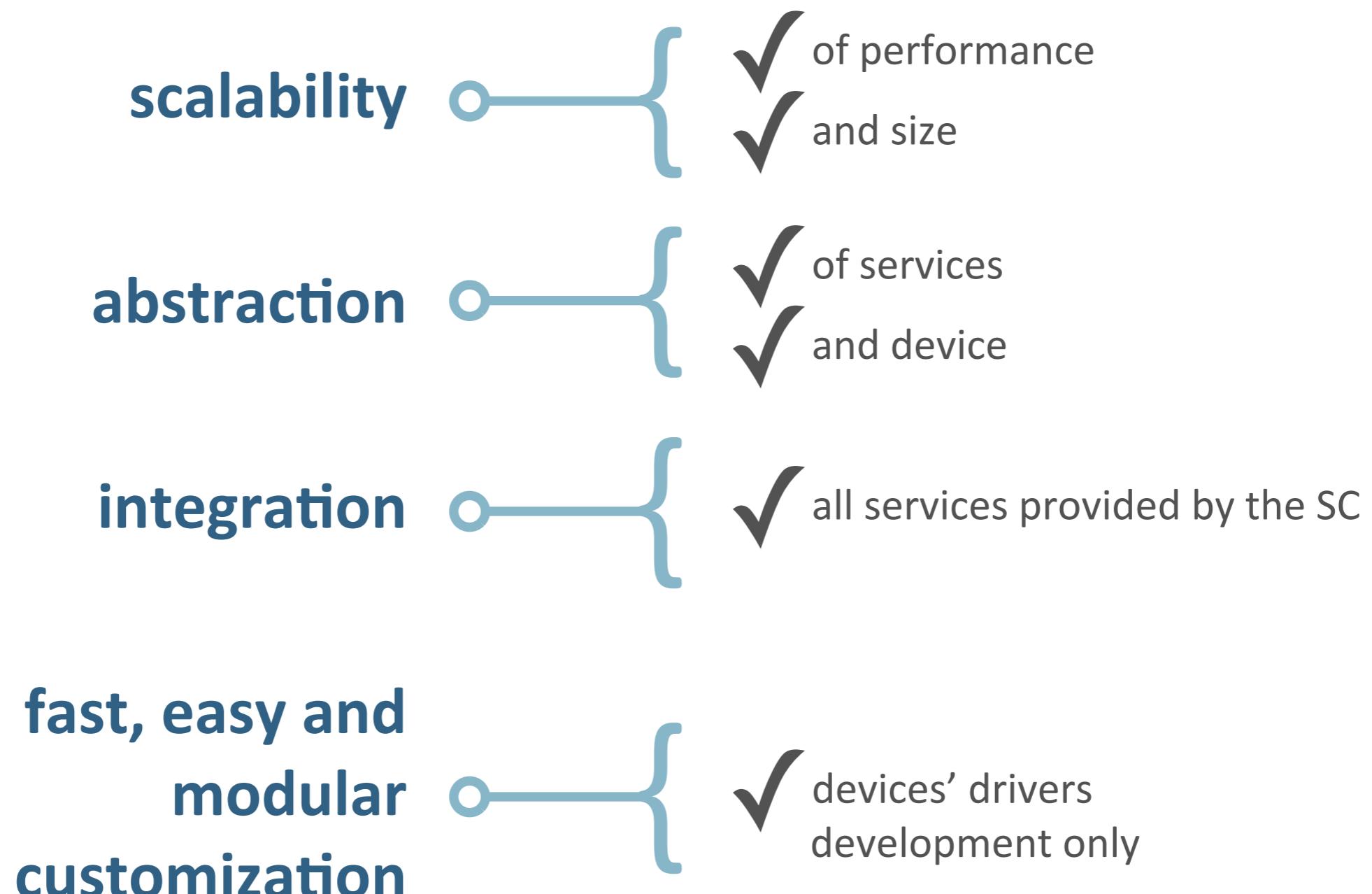


Control System topology

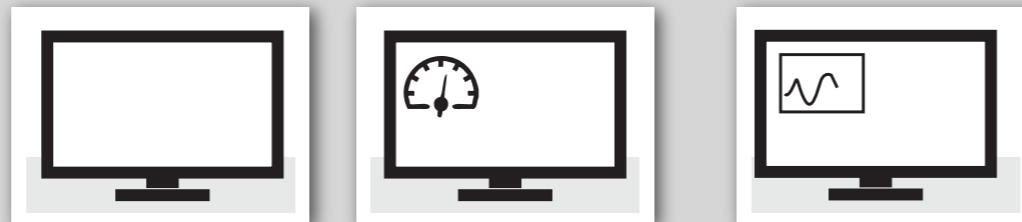




!CHAOS keywords



client



- ✓ request system information
- ✓ display or analyze *live* data
- ✓ run users' applications (measurements)
- ✓ request changes of device settings (commands)

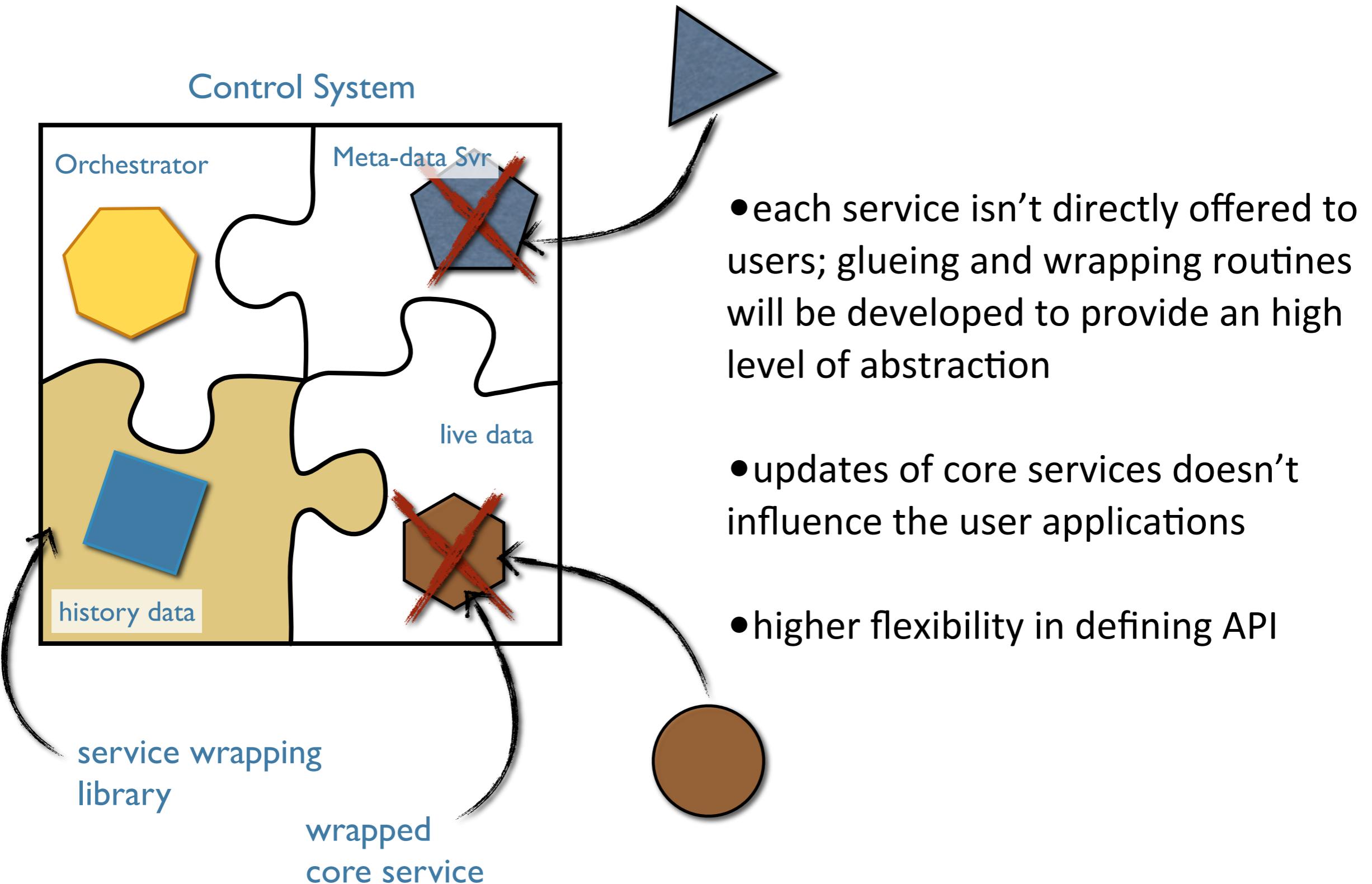
server

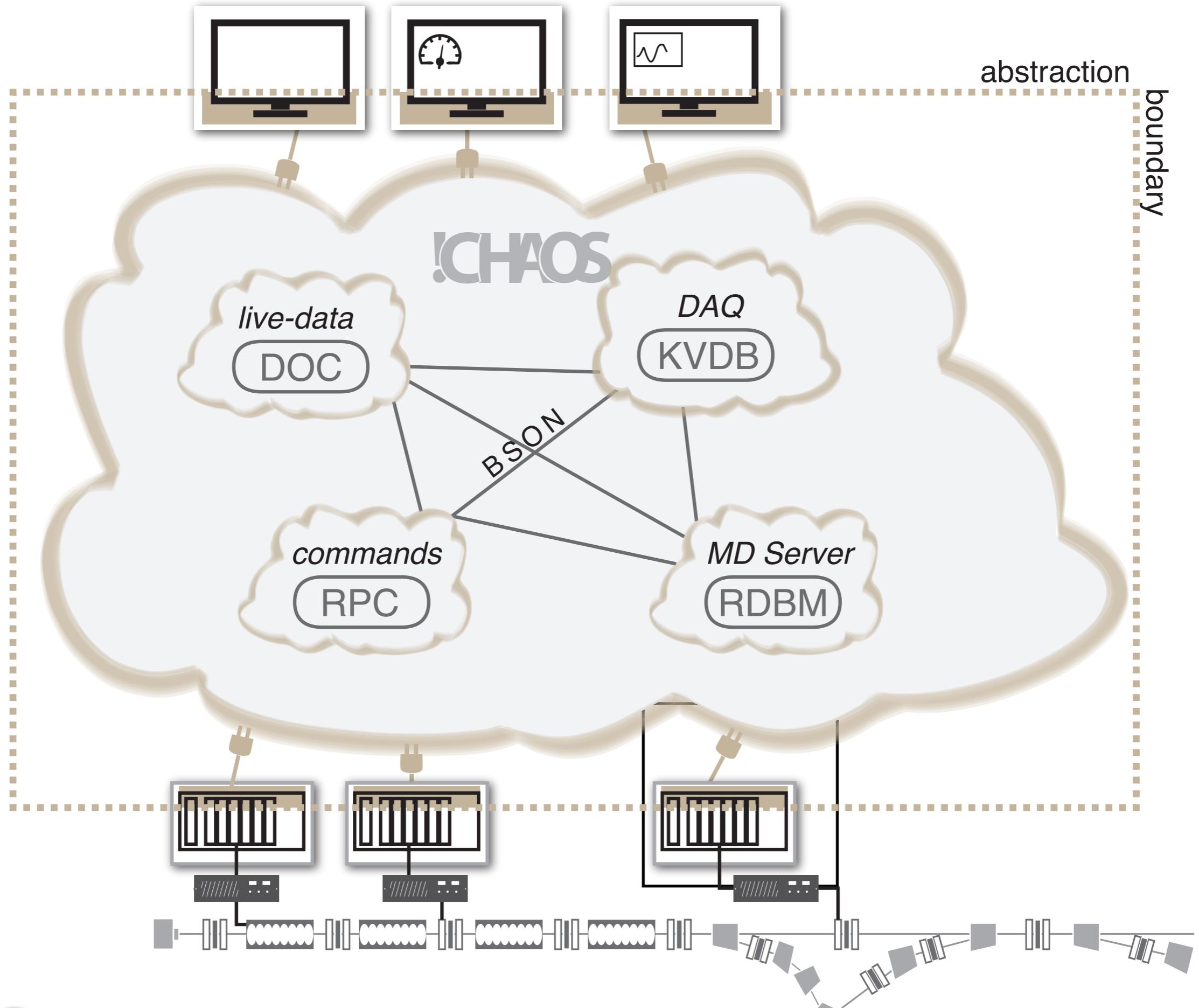


· / / / / / / ... dev-1

- ✓ provide (or receive) list of device to control
- ✓ execute device specific control loop
- ✓ provide read-out data
- ✓ change device settings (commands)

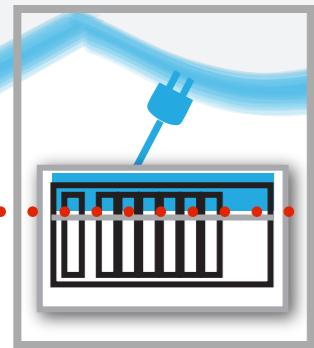
Abstraction of components





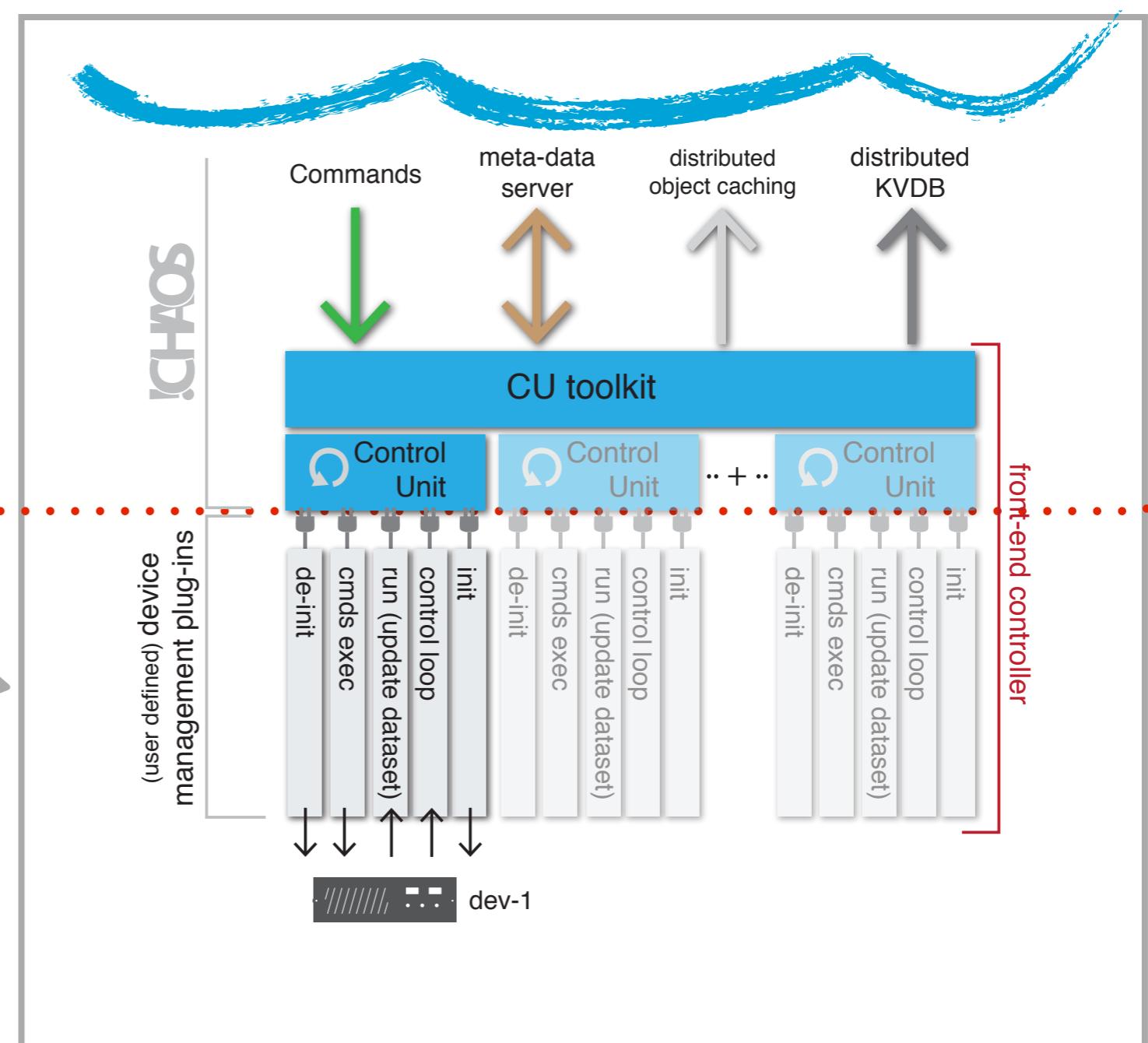
!CHAOS
abstraction boundaries

distribution
data archiving (DAQ)
logging
and info management
logging

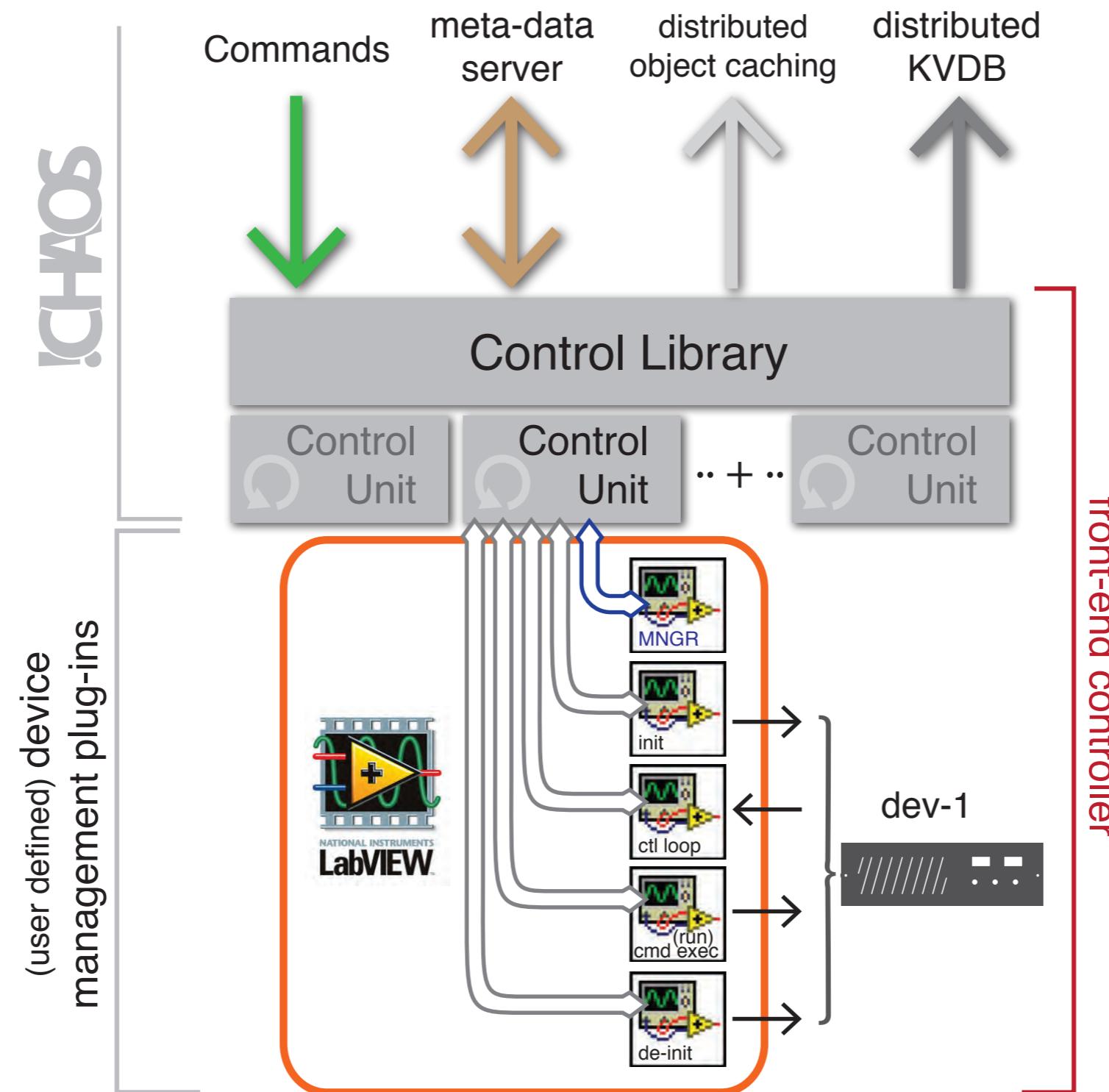


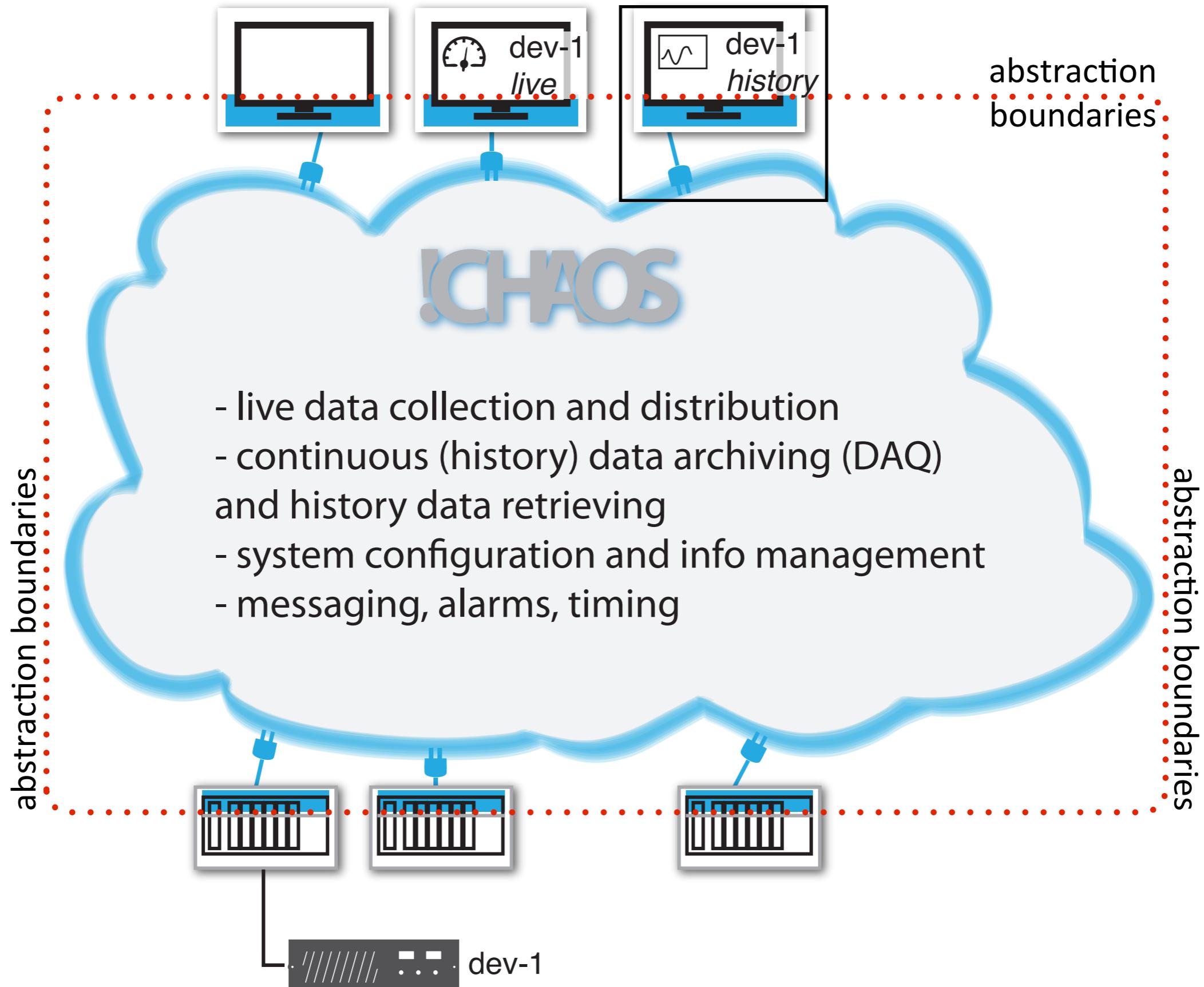
abstraction boundaries

CU abstraction

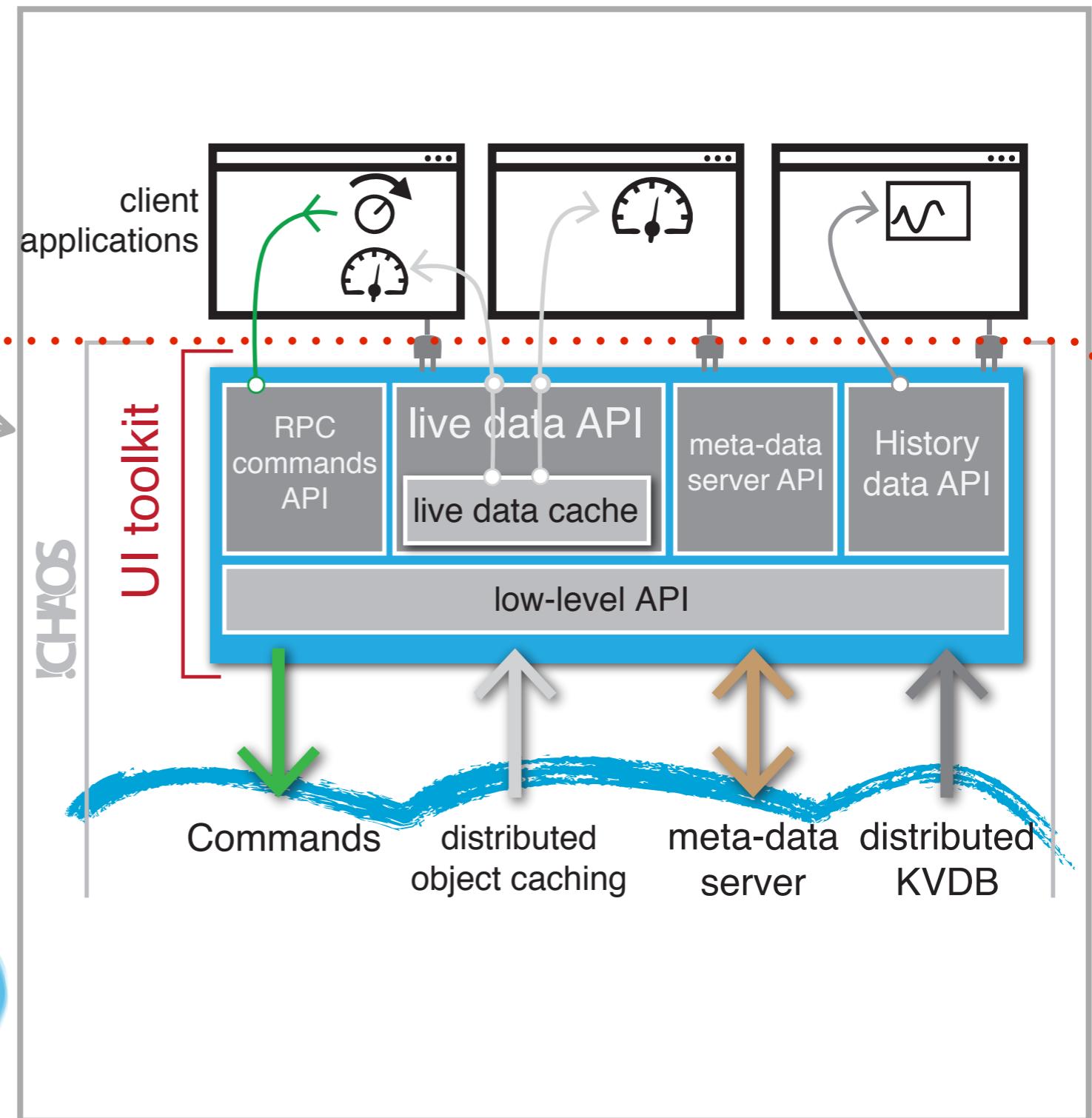
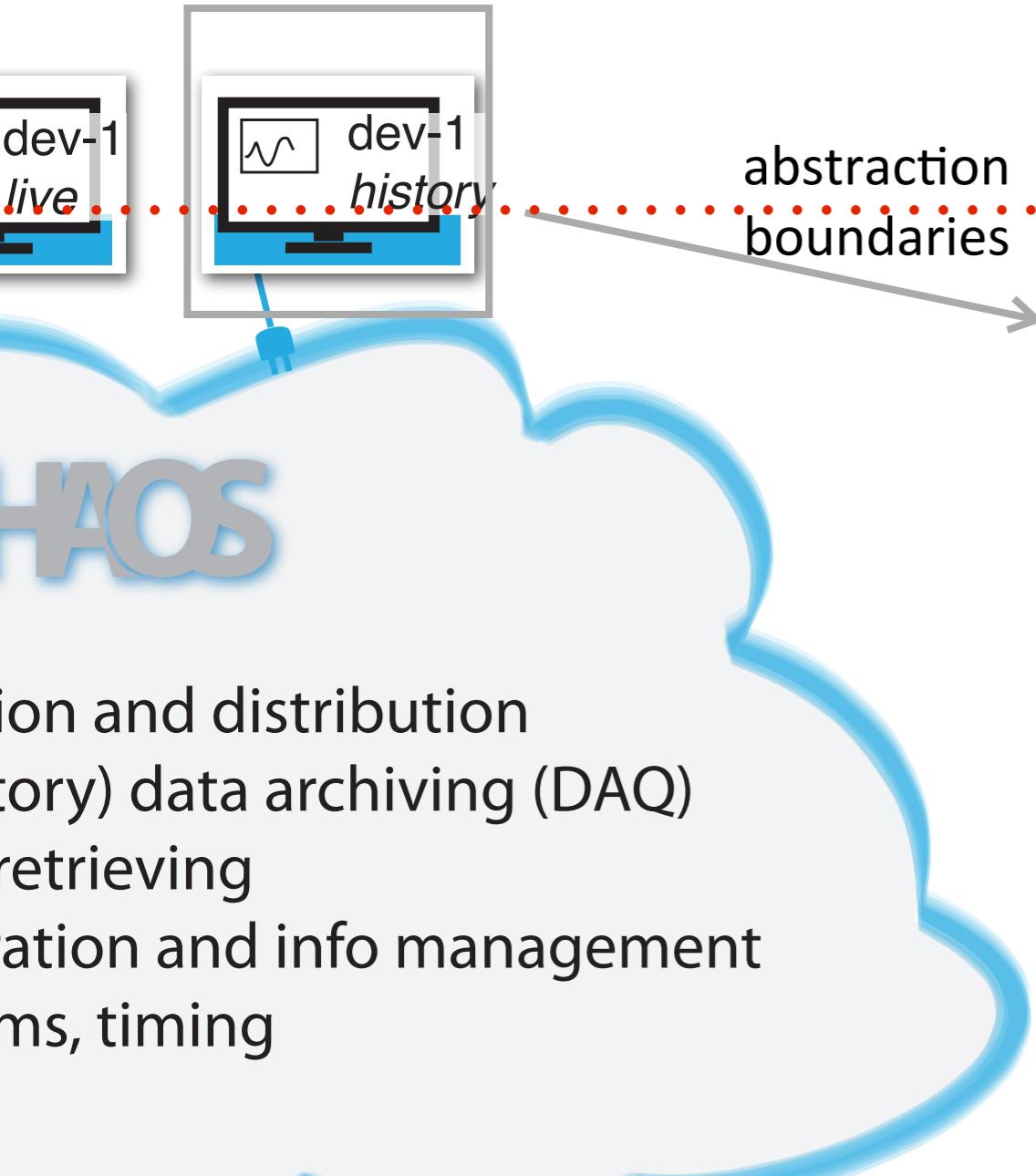


LabVIEW-CHAOS integration (a very basic solution)

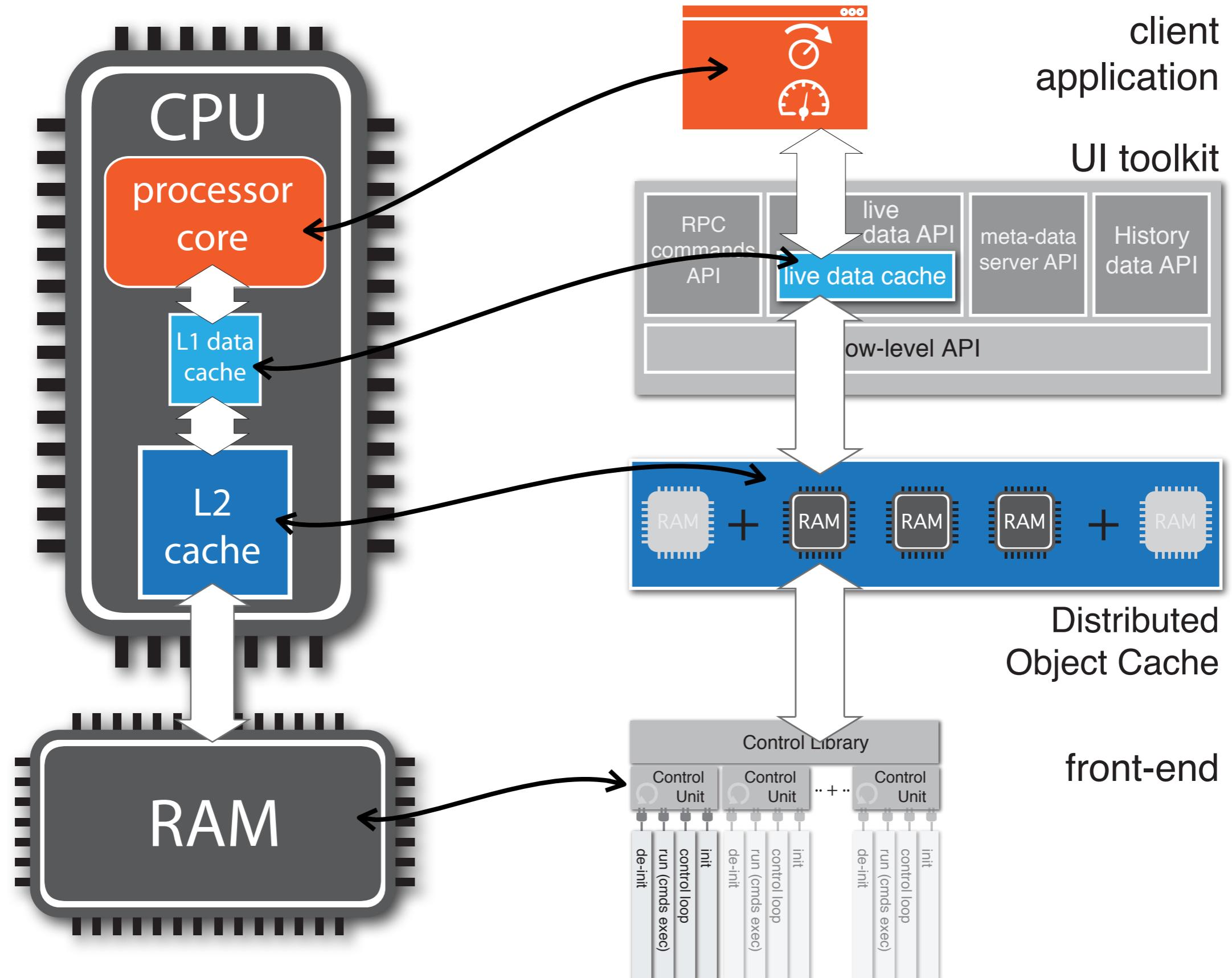




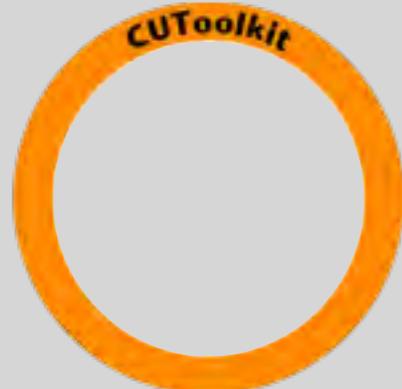
client abstraction



are we simply caching everywhere ?



!CHAOS framework layers



CUToolkit, abstract the !CHAOS resources to the device drivers developers.



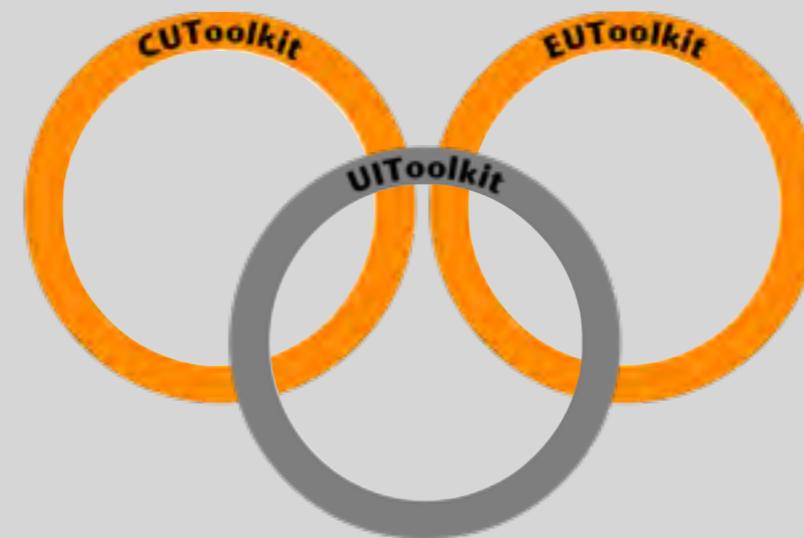
EUToolkit tools for developing distributed control node or computing nodes



UIToolkit tools for developing client application that accesses !CHAOS resource

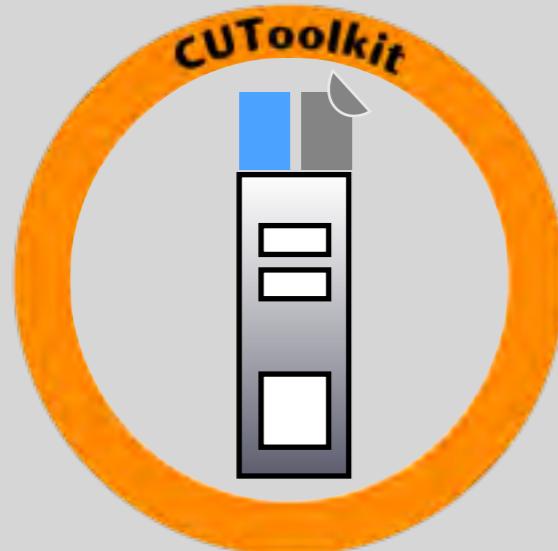


!CHAOS framework layers



These three layers represent the
!CHAOS Framework

!CHAOS Node & Service



Control Unit, a piece of software developed on CUToolkit implementing the device drivers



Execution Unit, is a the specialized software implementing slow-controls or computing algorithms.

-  ChaosQL Data Pack Channel
-  Chaos Command Pack Channel

!CHAOS Node & Service

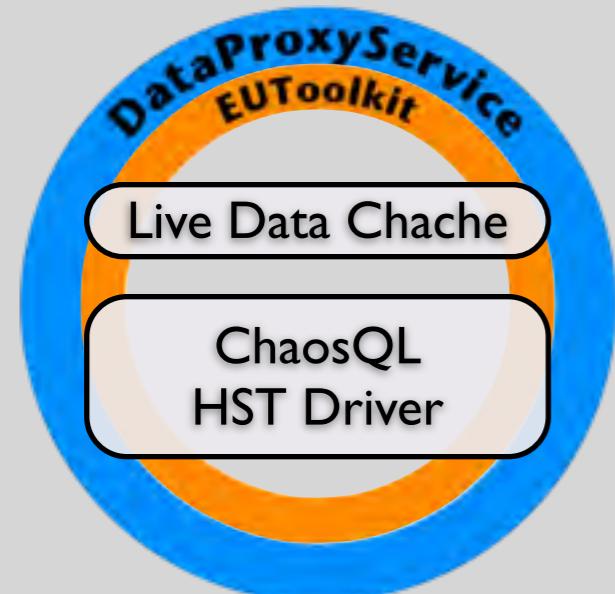


The Chaos Control GUI is based on the UI Toolkit for accessing !CHAOS resources. The UI Toolkit is also used by control panels/client applications developers to make their custom application



Metadata Server, stores and keeps track of all information about devices' DataSets and Commands, CU addresses and Hardware topology. It's developed on UI Toolkit software layer.

!CHAOS Node & Service

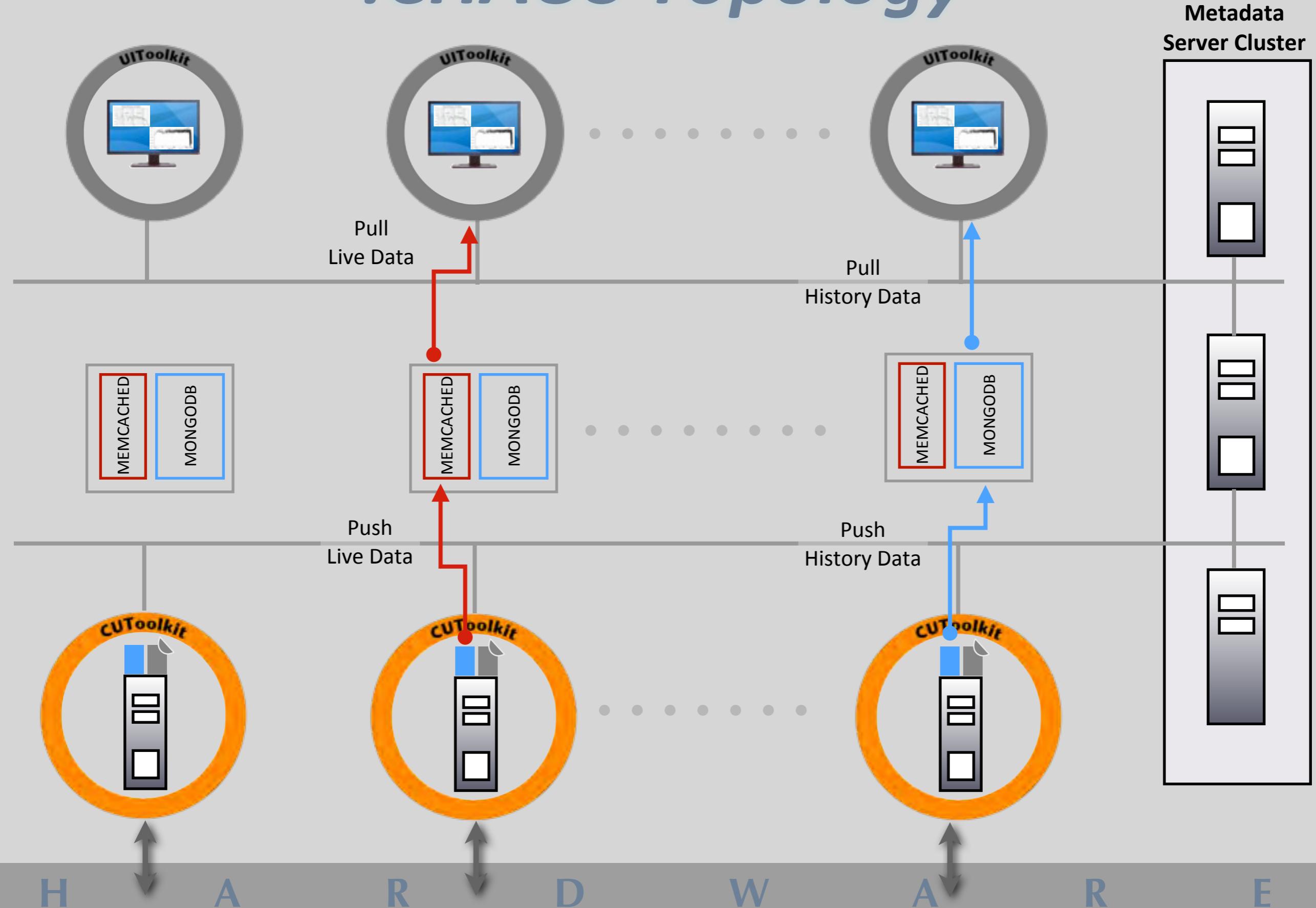


Data Proxy Service, is a scalable service that implement a common proxy for the Live and History data services. It includes memcached and the drivers for implementing ChaosQL for storing or querying history data



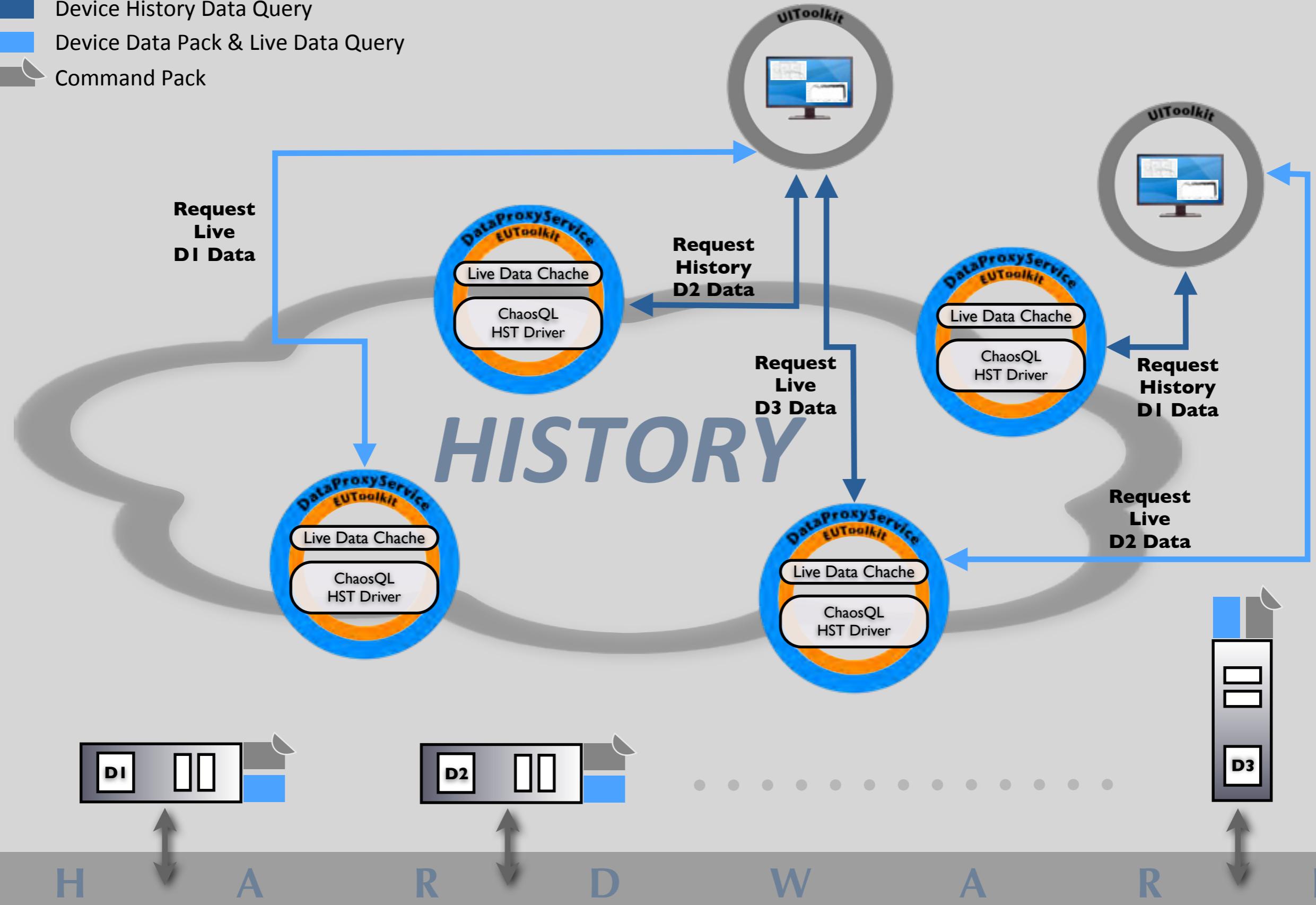
This is the History Storage Cloud, that can be accessed by means of the Data Service Proxy.

!CHAOS Topology



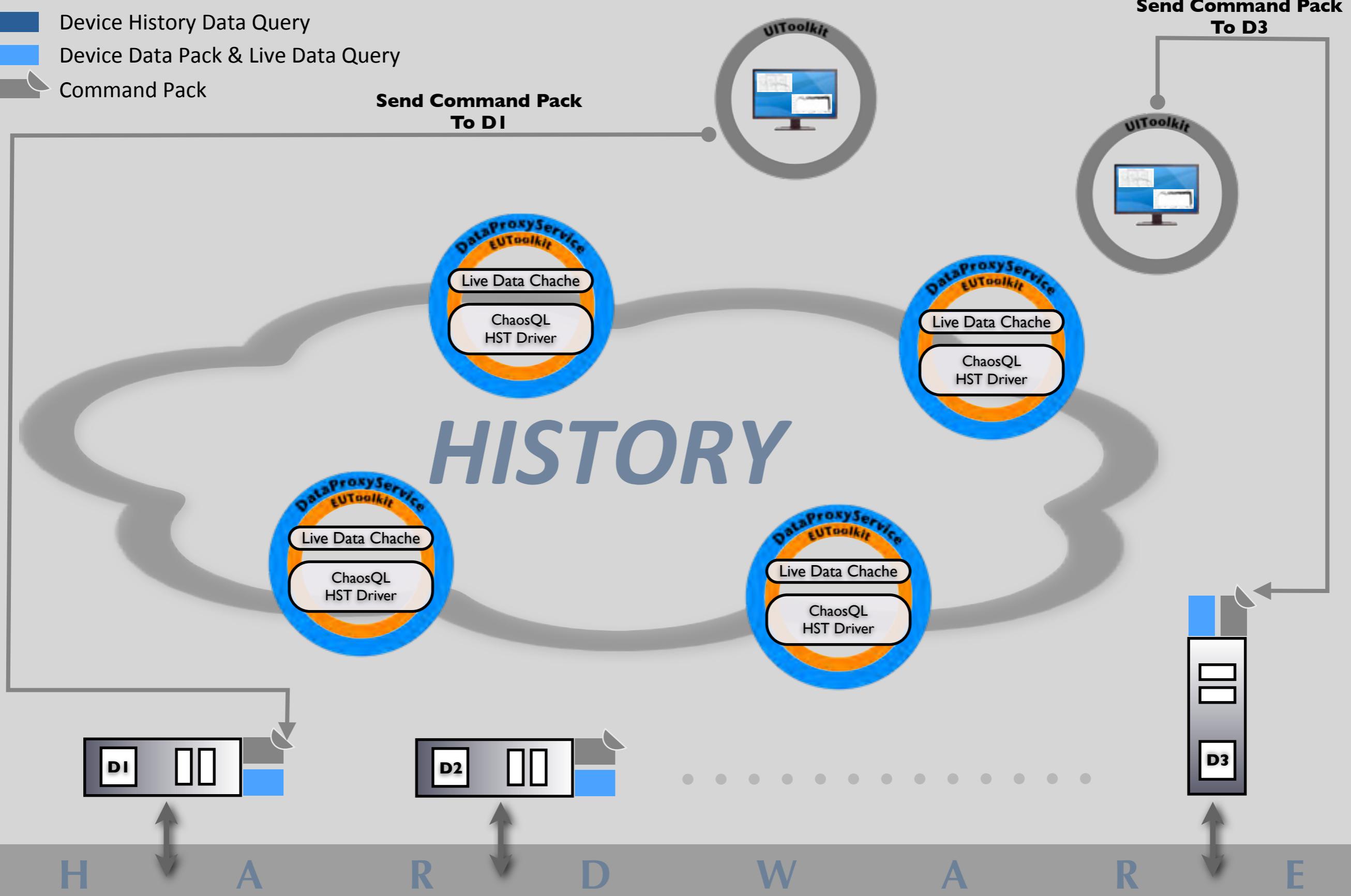
!CHAOS Topology

- Device History Data Query
- Device Data Pack & Live Data Query
- Command Pack



!CHAOS Topology

- Device History Data Query
- Device Data Pack & Live Data Query
- Command Pack



Meta-data Server

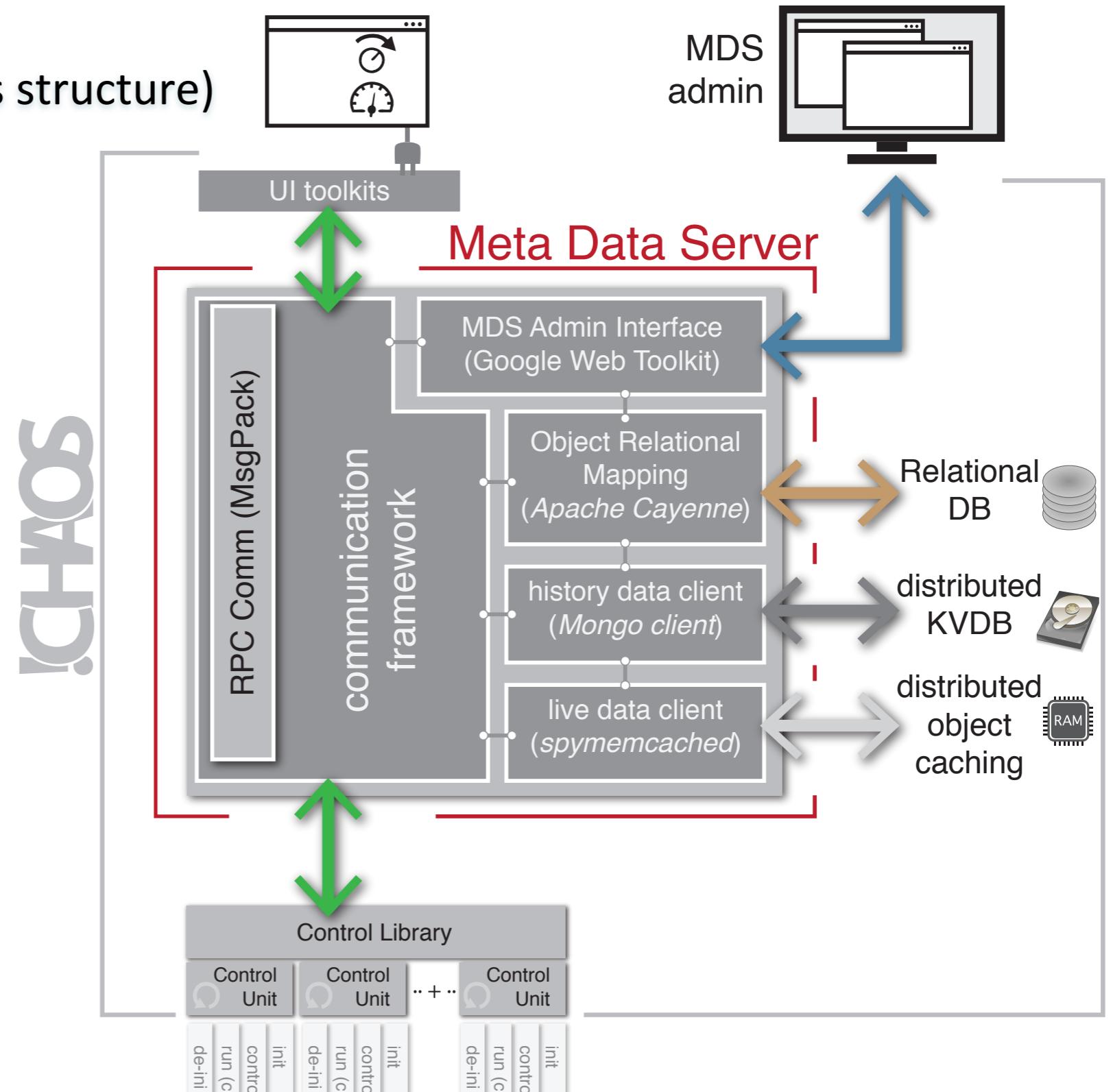
- CU configuration manager

(e.g. managing of pushing data rate)

- Semantic of data (e.g. db records structure)

- Command's list and semantic

- Naming service



core services

history db

The screenshot shows the MongoDB homepage. At the top is the MongoDB logo with the text '(name: "Mongo", type: "db")'. Below it is a navigation bar with links: Home, Getting Started, Development, Production, Support, and Community. Under 'Getting Started', there are links to Quickstart, Downloads, Tutorial, Features, and SQL to Mongo Mapping Chart. Under 'Development', there are links to Manual, C/C++/C#/C&/.NET, ColdFusion, Erlang, Factor, Java, Javascript, PHP, Python, Ruby, and Perl. Under 'Production', there are links to Production Notes, Replication, Sharding, Security, and Backup. Under 'Support', there are links to Forum, IRC, Bug tracker, Commercial support, Training, Consulting, and Hosting. Under 'Community', there are links to Blog, Articles, Twitter, Forum, Facebook, LinkedIn, Job Board, and User groups: NY, SF, DC, Chicago, London, Stockholm and more.

The screenshot shows the MessagePack Project homepage. The main heading is 'What's MessagePack?'. Below it is a section titled 'Serialization' with text about binary-based efficient object serialization. It includes a code snippet in C-like syntax:

```
msgpack* msgpack_pack_int(msgpack* msg, int32_t val) {
    msg->type = MSGPACK_TYPE_UINT;
    msg->u.uint = val;
    return msg;
}
```

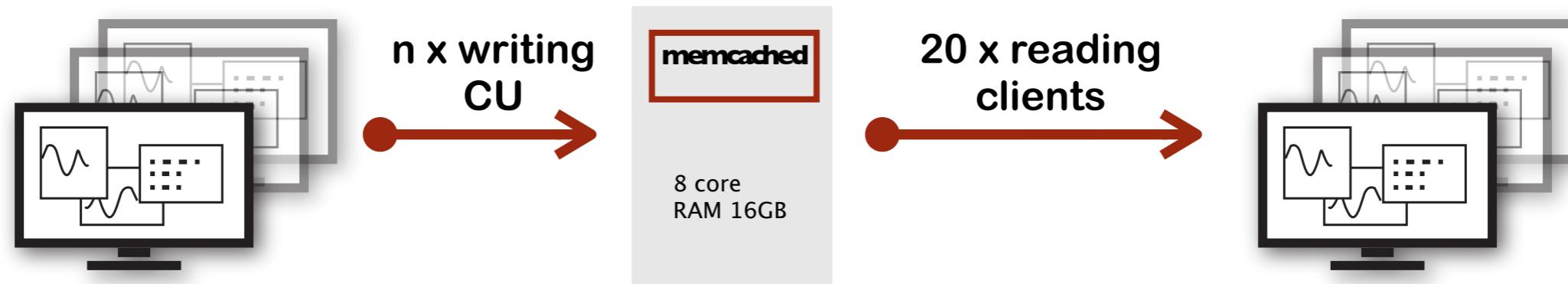
The screenshot shows the Memcached homepage. The main heading is 'What is Memcached?'. Below it is a section with text about Memcached being a free & open source, high-performance, distributed memory object caching system. It includes a code snippet in C-like syntax:

```
memcached_return memcached_int(memcached_st *memcached, uint32_t val) {
    memcached_return ret;
    ret = memcached_int32(memcached, val);
    return ret;
}
```

On the right side, there is a sidebar titled 'Memcached Users' with a list of websites that use Memcached, including LiveJournal, Wikipedia, Flickr, Bebo, Twitter, Typepad, Yellowbot, YouTube, Digg, WordPress.com, Craigslist, and Mixi. There is also a 'Quick Example' section with a 'Cache Results' button and a 'Play with telnet' button.

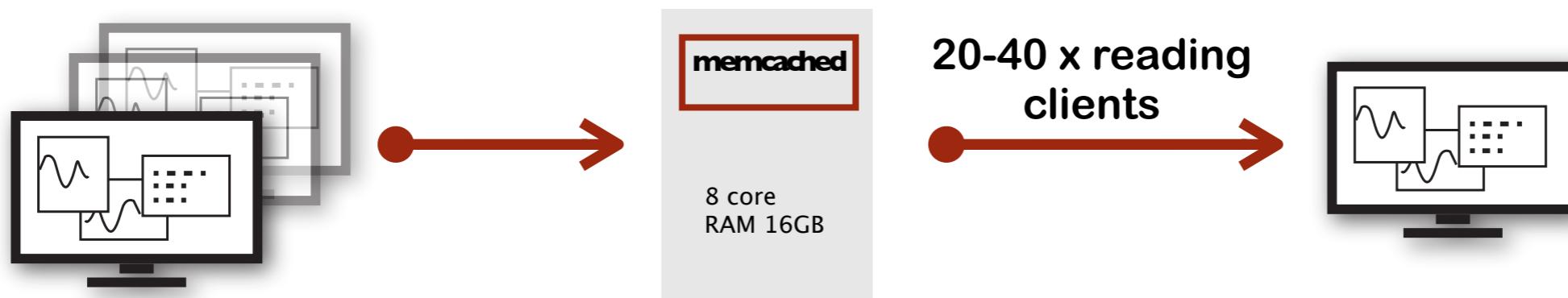
live data

test #3.1



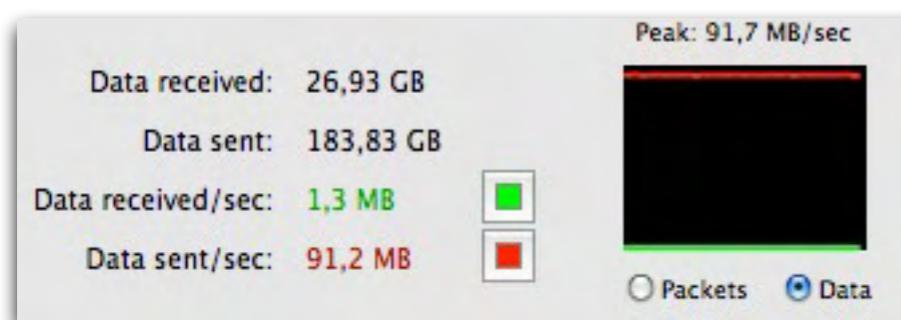
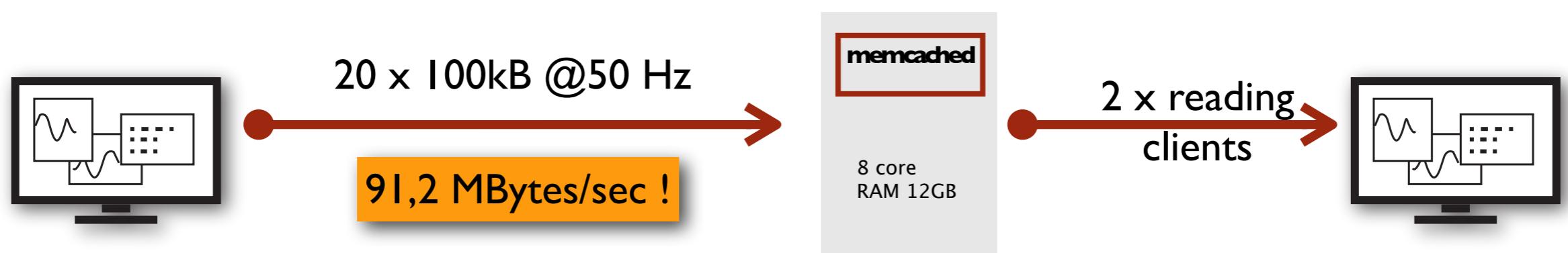
writing every... (msec)	#CU (Write)	#clients (Read)	#servers	#processes/ server	CPU load (%)
20	60	20	1	1	3-5
20	80	20	1	1	4-6
20	80	20	2	1	2-3
50	60	20	1	1	1-3
50	80	20	2	1	0-2
100	60	20	1	1	?
100	80	20	2	1	?

test #3.2



writing every... (msec)	#CU (Write)	#clients (Read)	#servers	#processes/ server	CPU load (%)
20	80	20	1	4 (1 per core)	2-3
20	80	40	1	4 (1 per core)	2-3
		40	1	4 (1 per core)	0

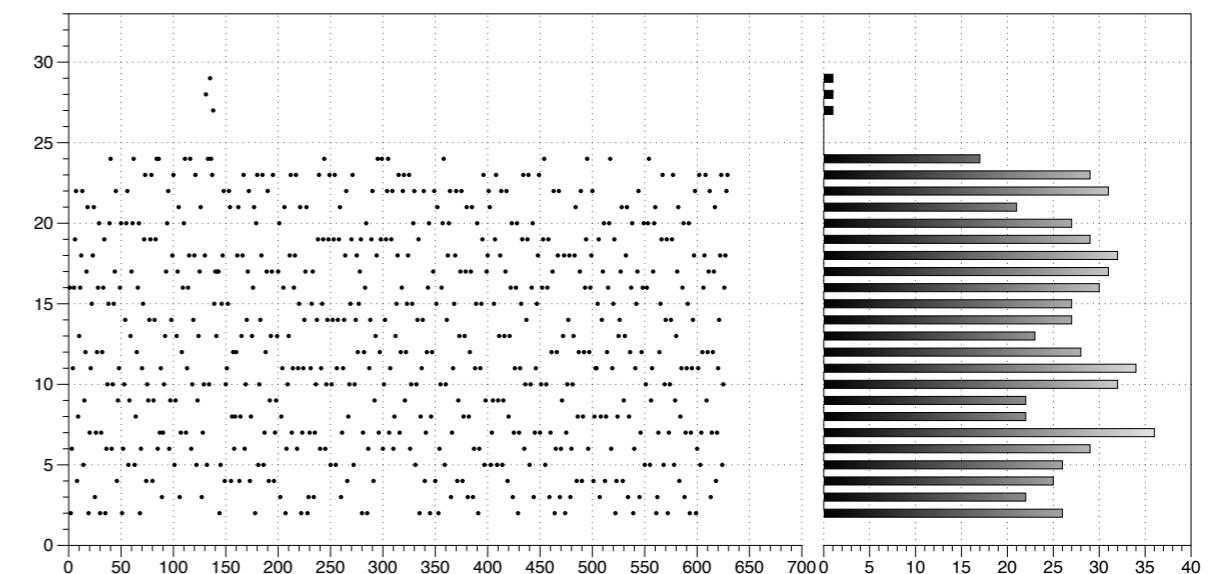
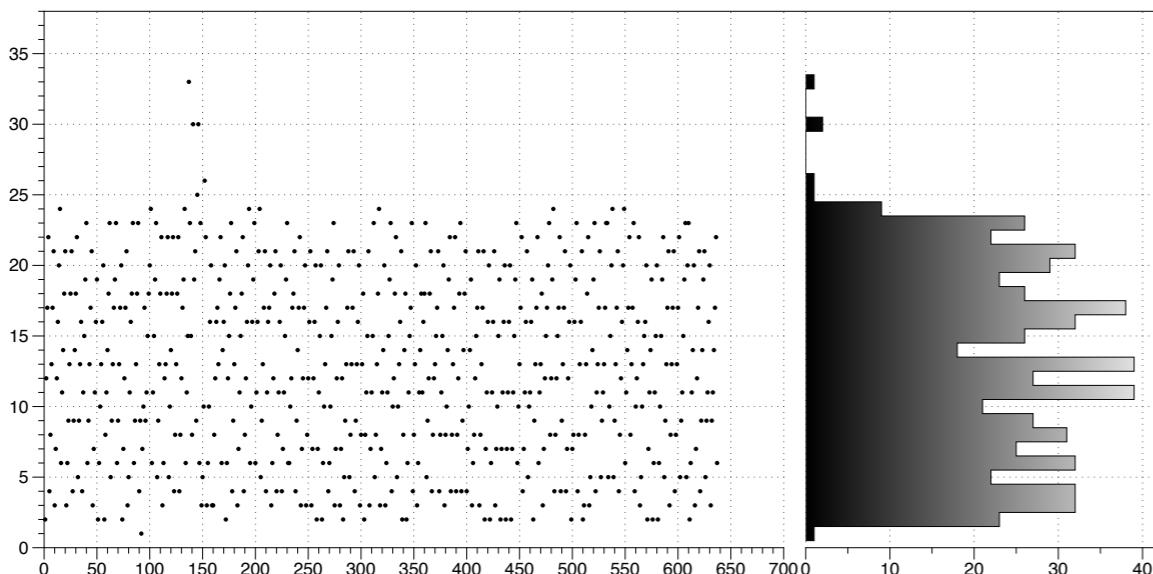
test #4



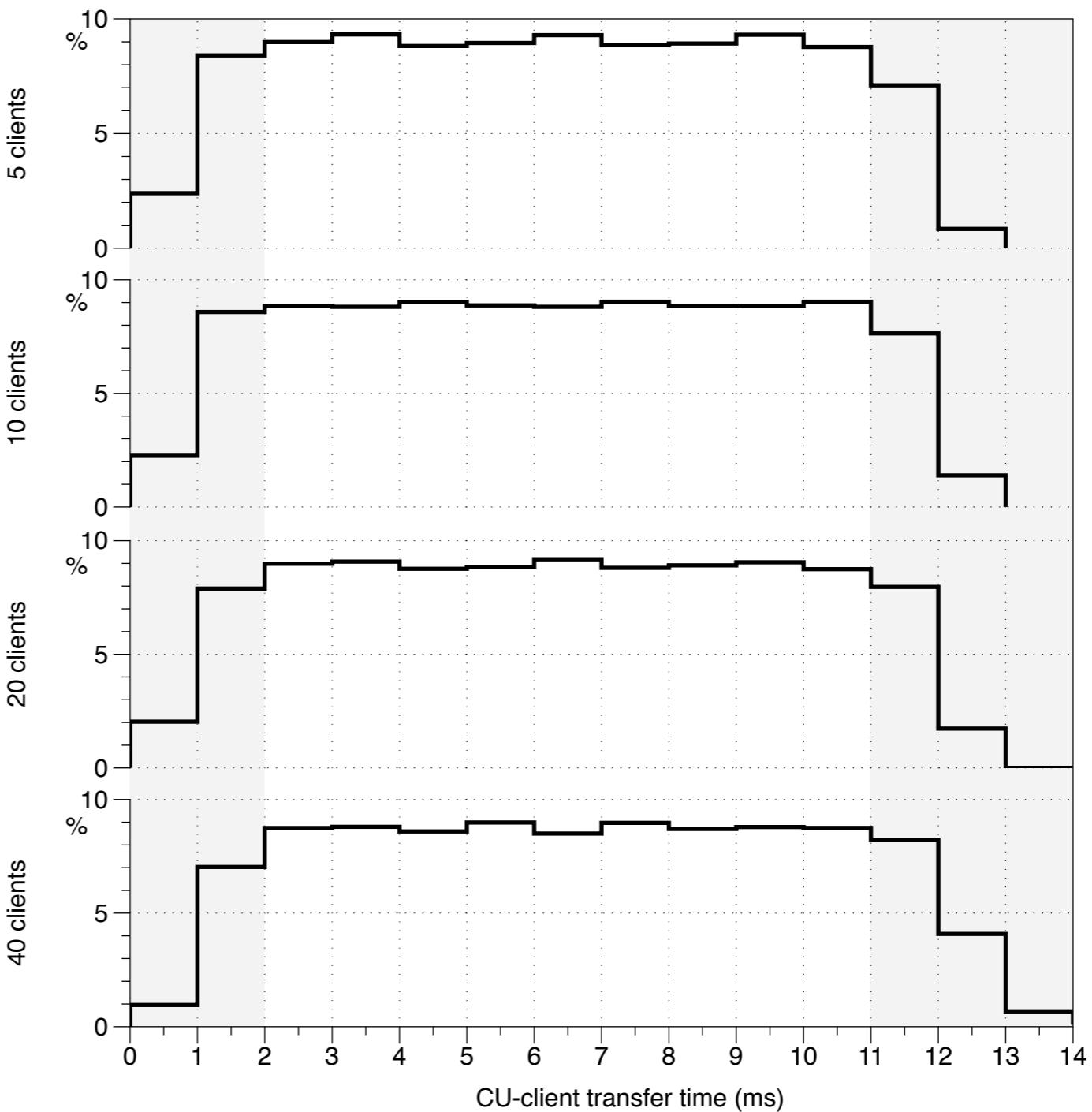
s4_hardware1_w20_m20_buff100000_rd10.log

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	MEM	TIME+	COMMAND
28059	dbuser	15	0	72236	10m	616	6	11.0	0.1	3:50.82	memcached
28066	dbuser	15	0	129m	5688	628	6	11.0	0.0	3:09.89	memcached
28052	dbuser	15	0	69812	8024	612	6	7.0	0.0	2:13.86	memcached
28074	dbuser	15	0	67568	5816	616	6	4.0	0.0	1:29.09	memcached

s4_hardware1_w20_m20_buff100000_rd12.log

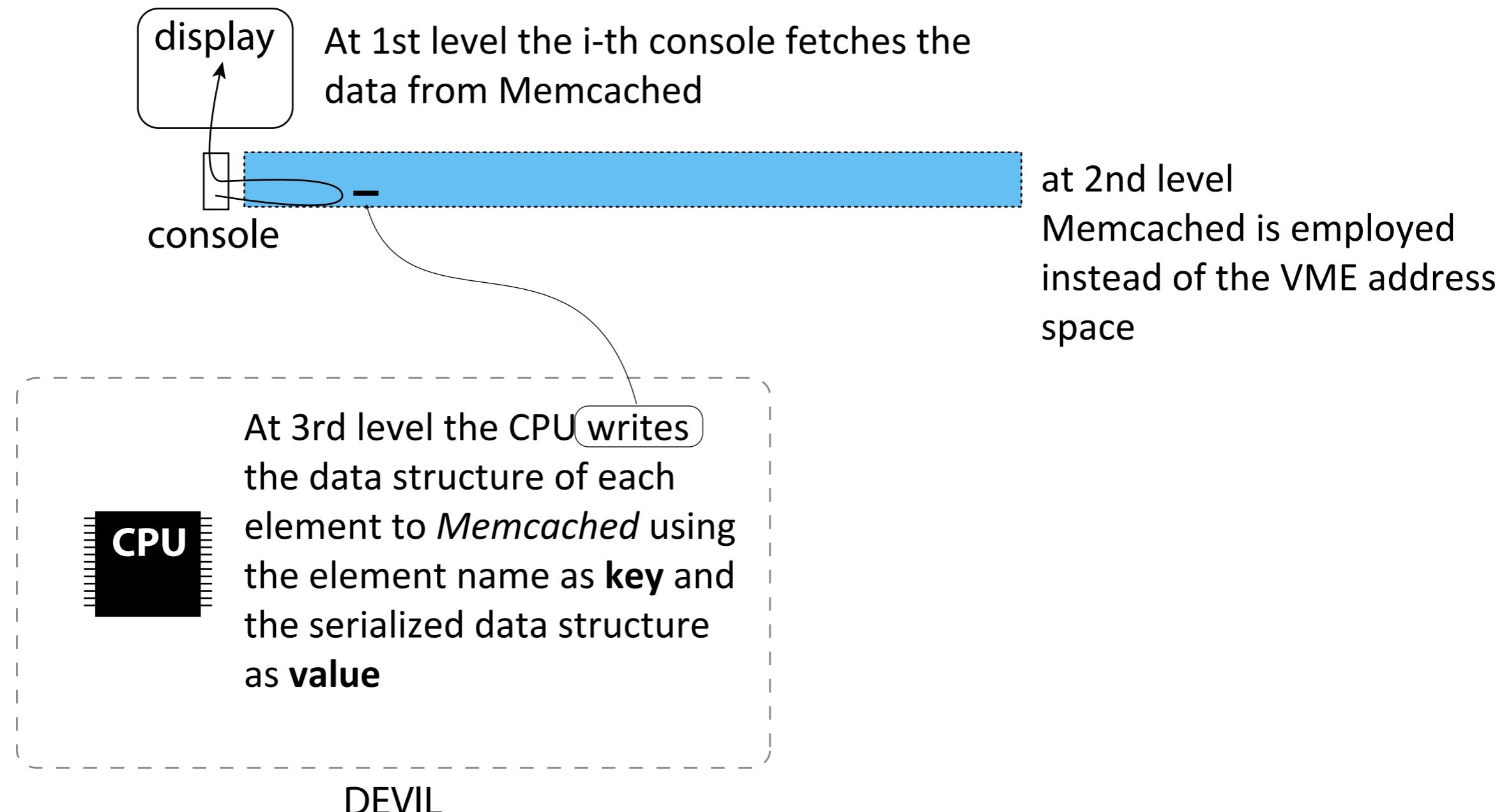


Measured transfer time between front-end CU and a client application via DOC for a different number of concurrent clients reading the same key/value being continuously updated by the CU.



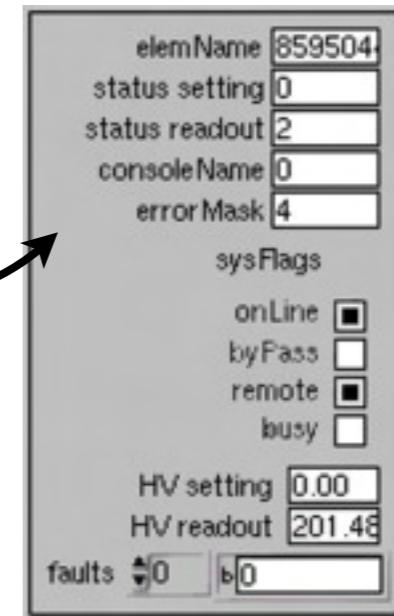
tests at DAFNE

The idea: replace the 2nd level VME address space with the *Memcached* associative memory



Preliminary measurements

data size: 64 bytes for packet read



fetch frequency ~ 100 Hz

with no dependency on the number of fetching consoles
(up to 7 in our test)

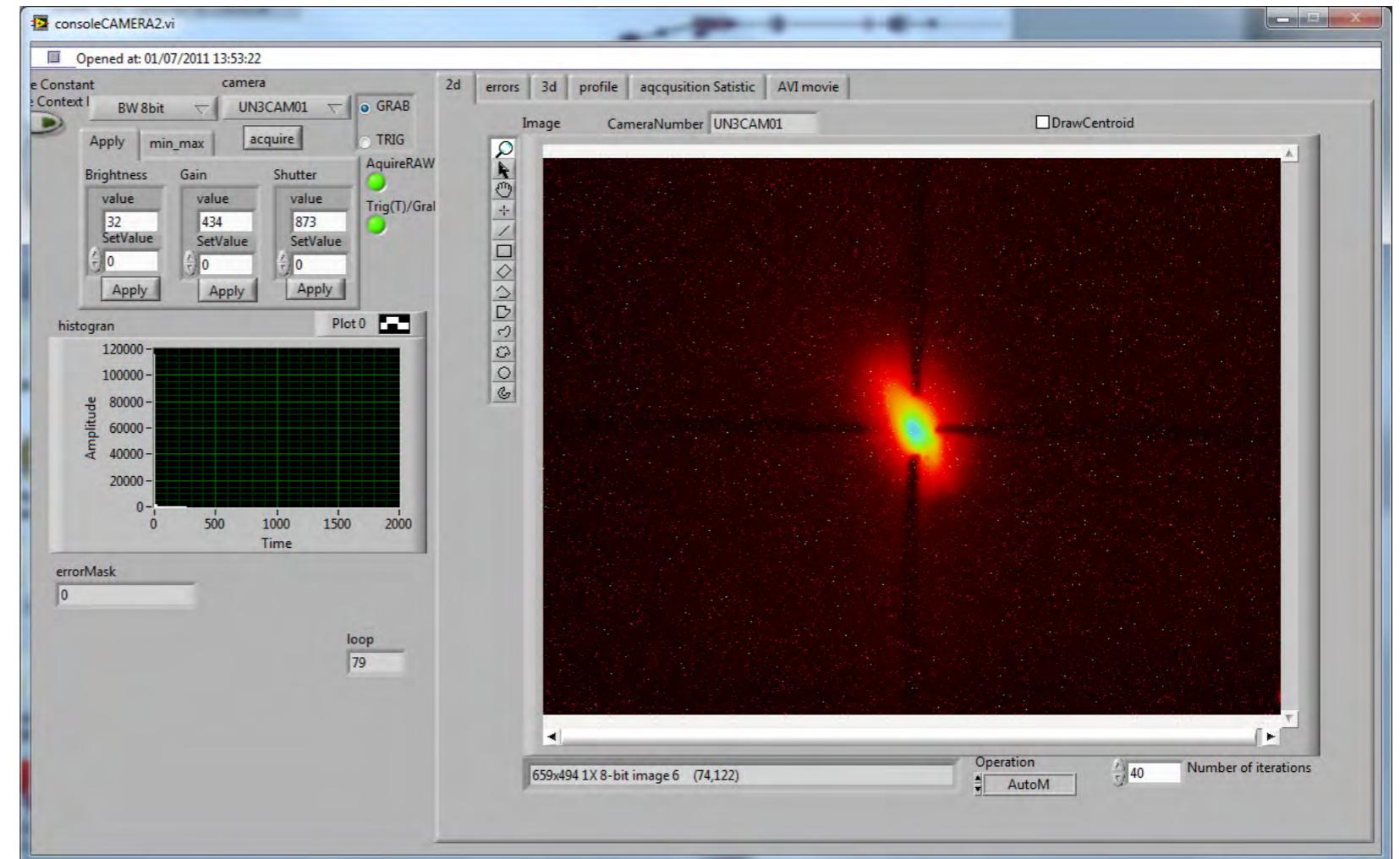
Memcached server load (measured with the *top* command)

CPU: 0.3% - 0.7% memory: ~ 0.1%

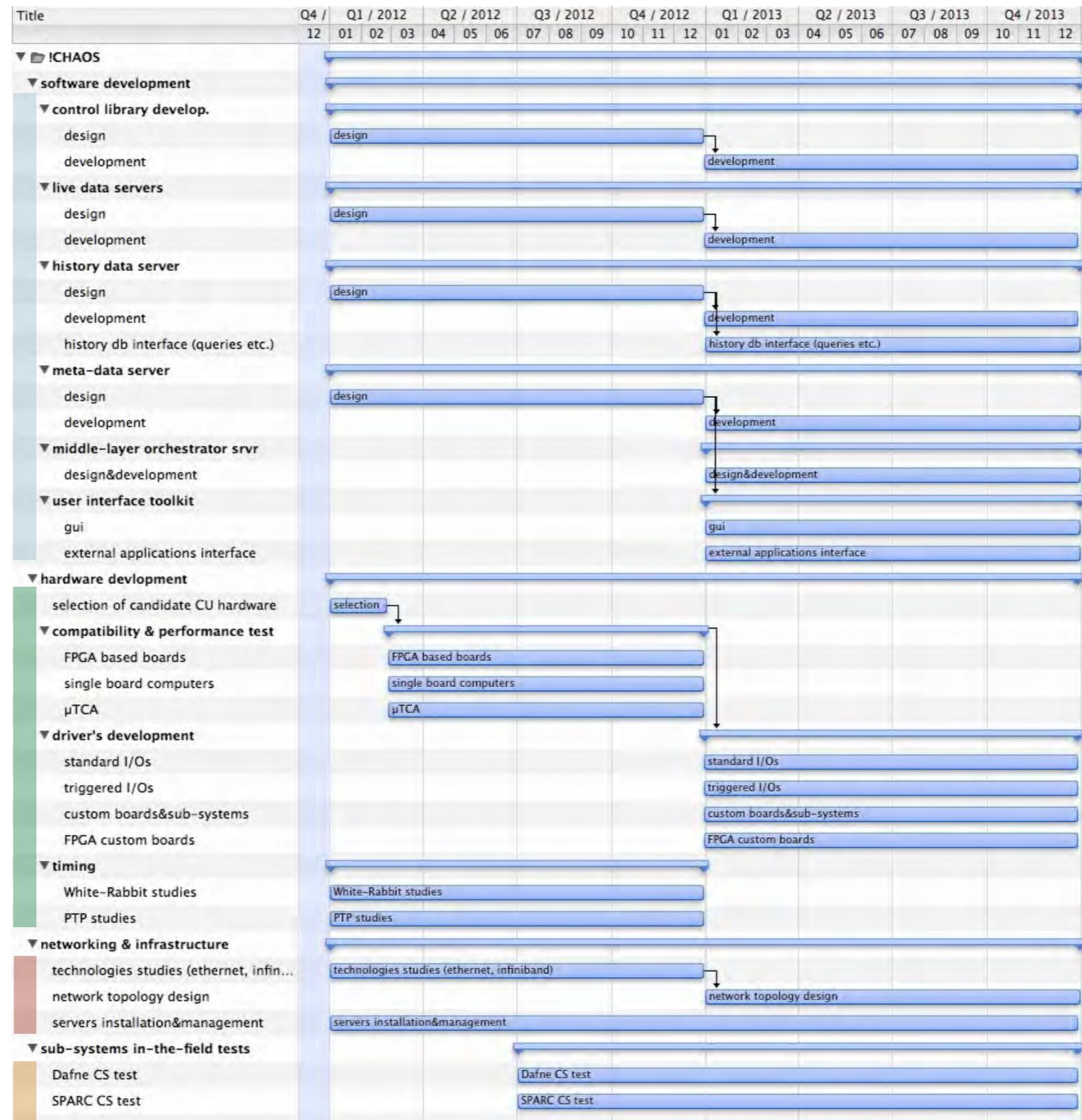
tests at SPARC

Memcached has been used for storing the beam image from a digital camera of beam diagnostic.

- network: Ethernet @1 Gbps
- image size: 640x480
 - @8 bit = **300 kB**
 - @16 bit = **600 kB**



measured fetch frequency: **8bit ~ 25 Hz, 16bit ~ 13 Hz**; same transfer rate for all consoles fetching images from memcached (up to 4 in our test)



!CHAOS contact persons



2nd SuperB Collaboration Meeting @ INFN-LNF

13-16 December 2011

INFN-LNF

[Home](#) > [Timetable](#) > Session details 

Parallel III: Acc III - Controls

Phone number: +39 06 6228 8548

or http://server10.infn.it/video/index.php?page=telephone_numbers

Meeting ID: 1430

Place: INFN-LNF <!-- ID_UTENTE=483 -->
Via E. Fermi, 40
00044 Frascati (RM)
Italy

Room: Aula Div Acc

Dates: Wednesday 14 December 2011 09:00

Conveners: [Catani, Luciano](#)

[Contribution List](#) [Time Table](#)

Wednesday, 14 December 2011

[36] **[!CHAOS general introduction](#)**
by Alessandro STECCHI (LNF)
(Aula Div Acc: 09:00 - 09:10)

[37] **[!CHAOS general architecture](#)**
by Mr. Claudio BISEGNI (LNF)
(Aula Div Acc: 09:10 - 09:35)

[38] **[!CHAOS first implementation of the Control Unit](#)**
by Luca Gennaro FOGGETTA (LNF)
(Aula Div Acc: 09:35 - 10:00)

Monday 12 December 2011
from **15:00** to **18:00**
Europe/Rome
at **INFN-LNF <!--**
ID_UTENTE=483 --> (
Accelerator Division meeting
room, Bldg 2)
support:
giovanni.mazzitelli@lnf.infn.it

!CHAOS Mini Workshop

Description: !CHAOS Mini Workshop, Frascati **12th November 2011 / 15pm - 18pm.**

An half day mini workshop will be organized by the !CHAOS (Control system based on Highly Abstracted Operating Structure) group before the start of the [Second SuperB Collaboration Meeting](#).

The !CHAOS project [1] [2] aiming at the development of an innovative concept of control system for particle accelerators and experimental apparatus, but suited also for medium and small size distributed laboratory systems looking for high performance data acquisition and archiving.

The mini-workshop will be particularly addressed to the italian community of developers and users of control and/or data acquisition systems, especially those interested in collaborating on the preparation of the proposal for SuperB project's control system.



Material: [live broadcast](#) [live broadcast info](#)

[Monday 12 December 2011](#) |

Monday 12 December 2011

[top↑](#)

- 15:00 Introduction (10')
- 15:10 !CHAOS Architecture (30')
- 15:40 !CHAOS Developments (20')
- 16:00 !CHAOS LabView integration (20')
- 16:20 Archiving and querying historical data on key/value databases (20')
- 16:40 System configuration and meta-data server (20')
- 17:00 Discussion (40')