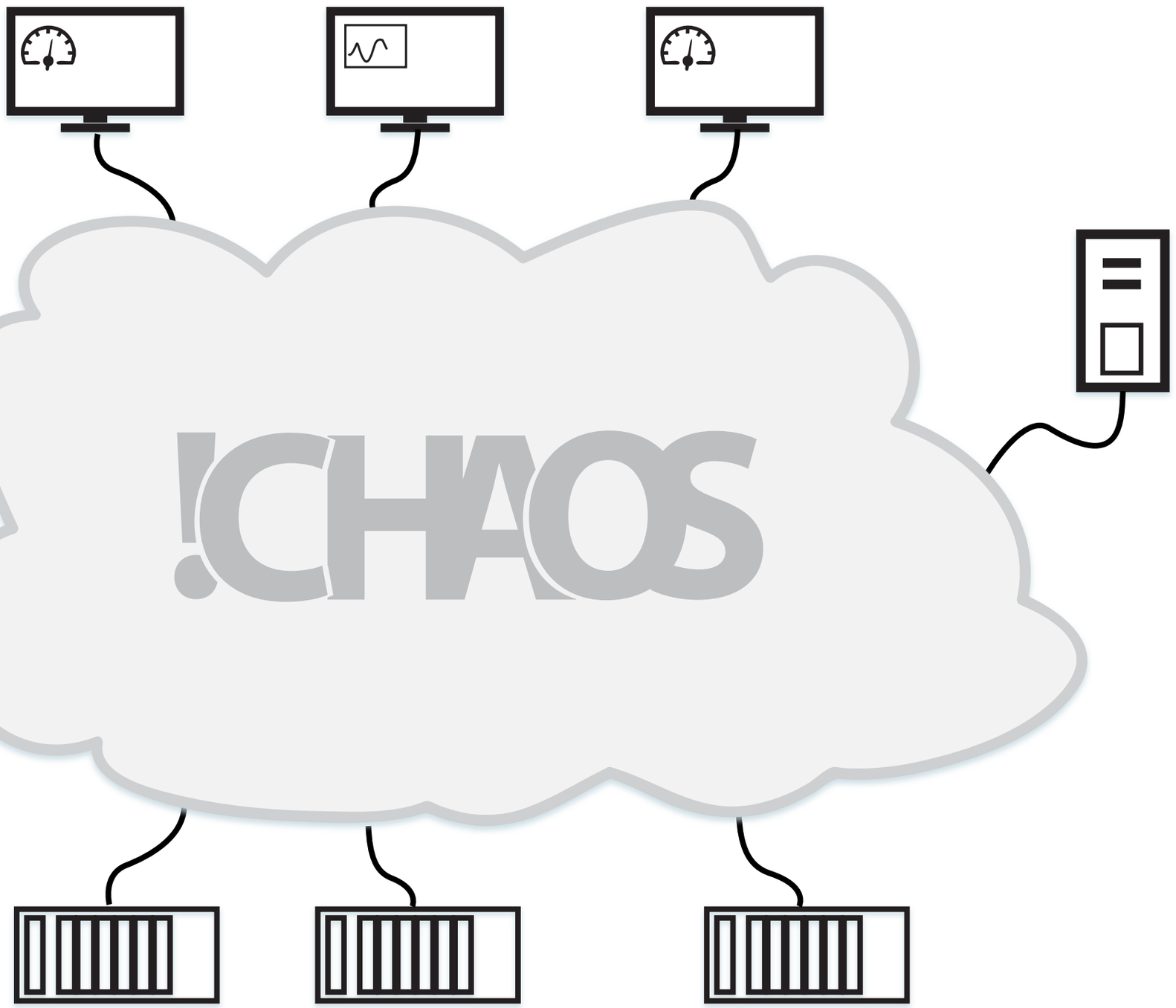
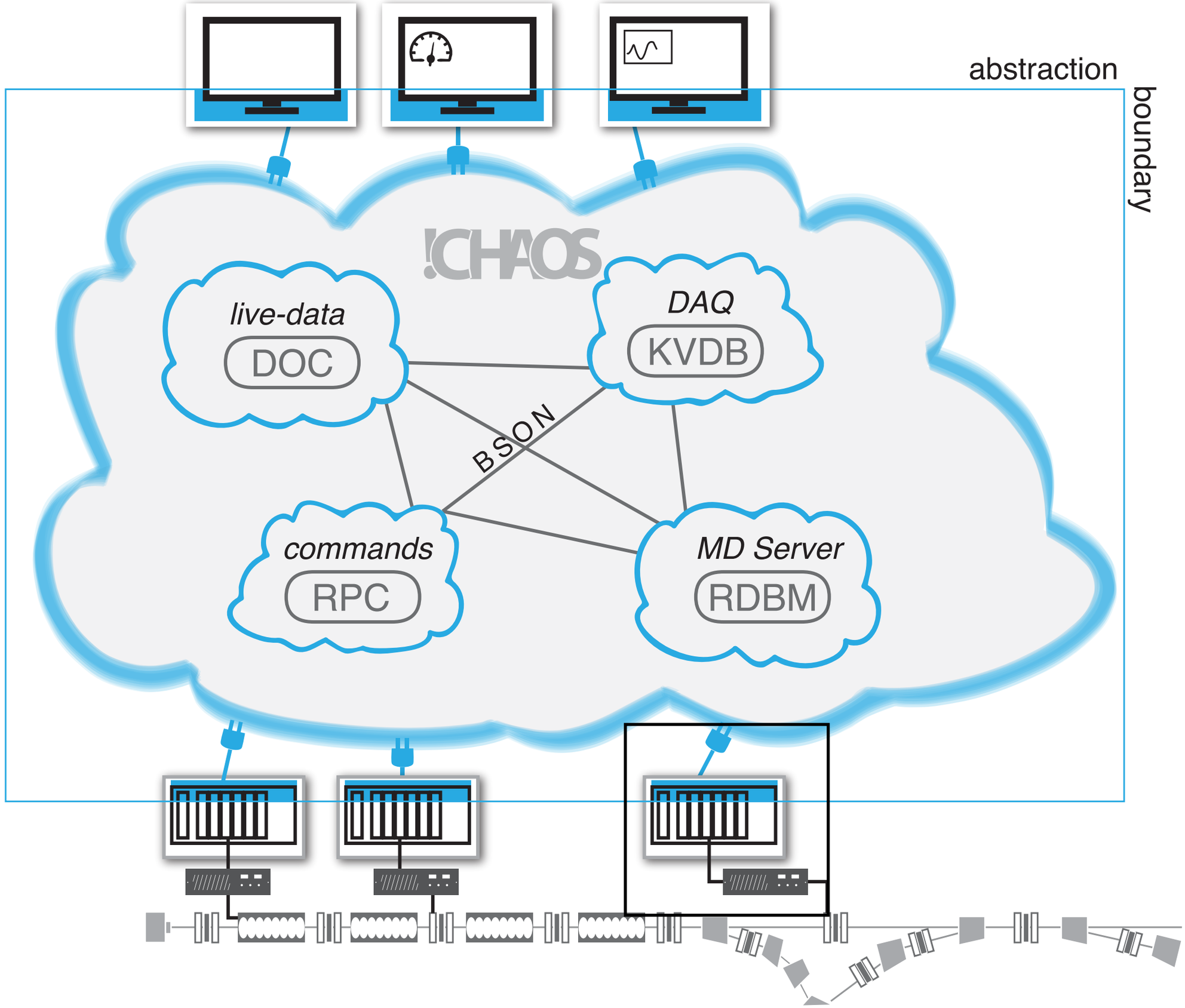
Status of the project
until April 2012 and outlook


!CHAOS


abstraction
boundary
!CHAOS
live-data
DOC
DAQ
KVDB
BSON
commands
RPC
MD Server
RDBM

Users Application

!C clients gateway | LV clients gateway | gateway to other scripting language

User Interface Toolkit

Common Toolkit

distributed Live Data service

distributed History Data service

commands RPC service

Meta data server

abstraction boundaries

Common Toolkit

CU Toolkit

Control Unit

CU modules (config)

CU modules (init)

CU modules (set)

CU modules (de-init)

CU modules (stop)

Control Unit

CU-LV gateway

CU modules (config)

CU modules (init)

CU modules (set)

CU modules (de-init)

CU modules (stop)

I/O Unit

I/O modules

I/O modules

I/O modules

Exec Unit

EU modules

EU modules

EU modules

data consumer

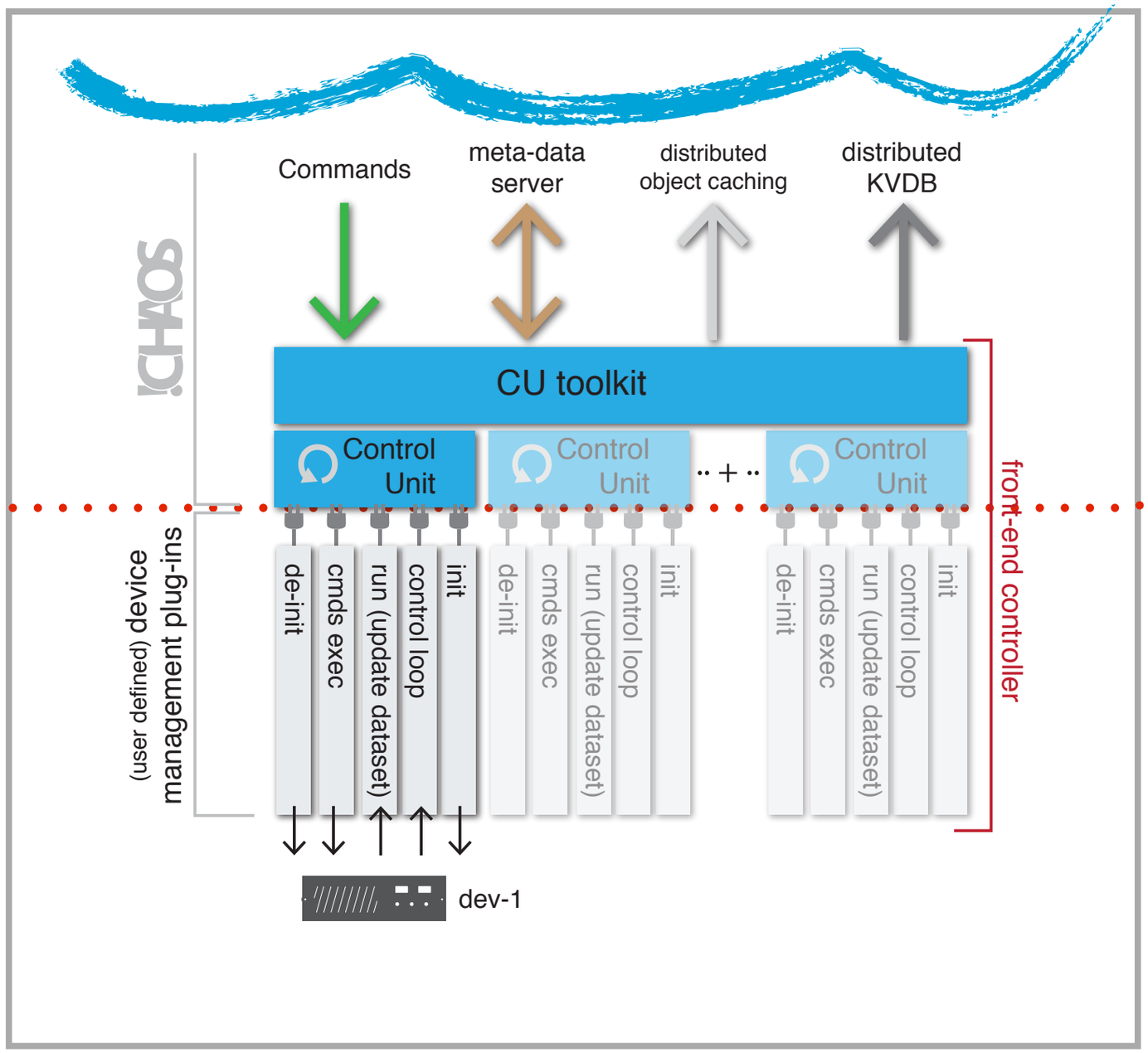
!CHAOS infrastructure

data producer

CU abstraction

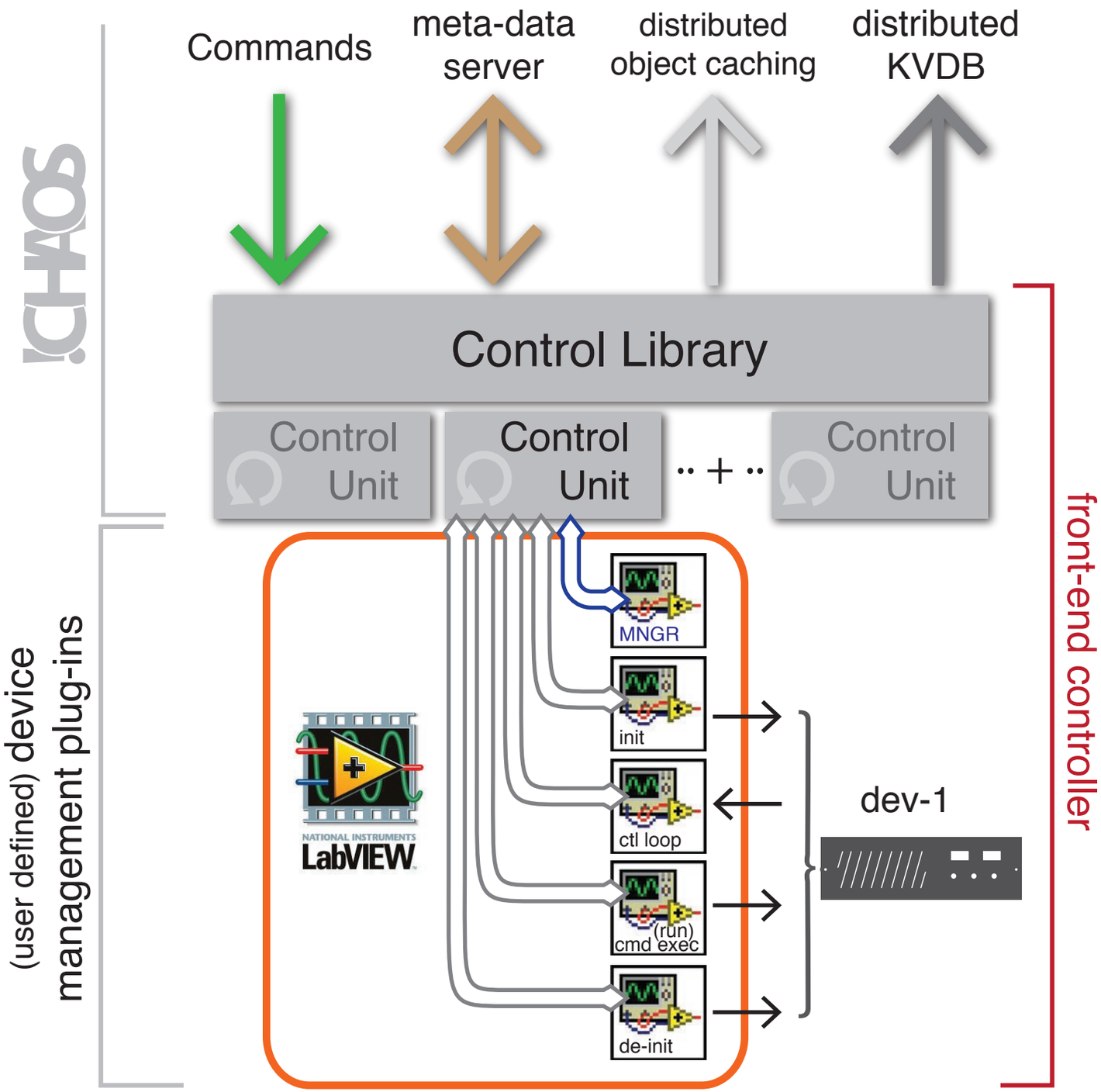
d distribution
ata archiving (DAQ)
ing
and info management
ning

abstraction boundaries

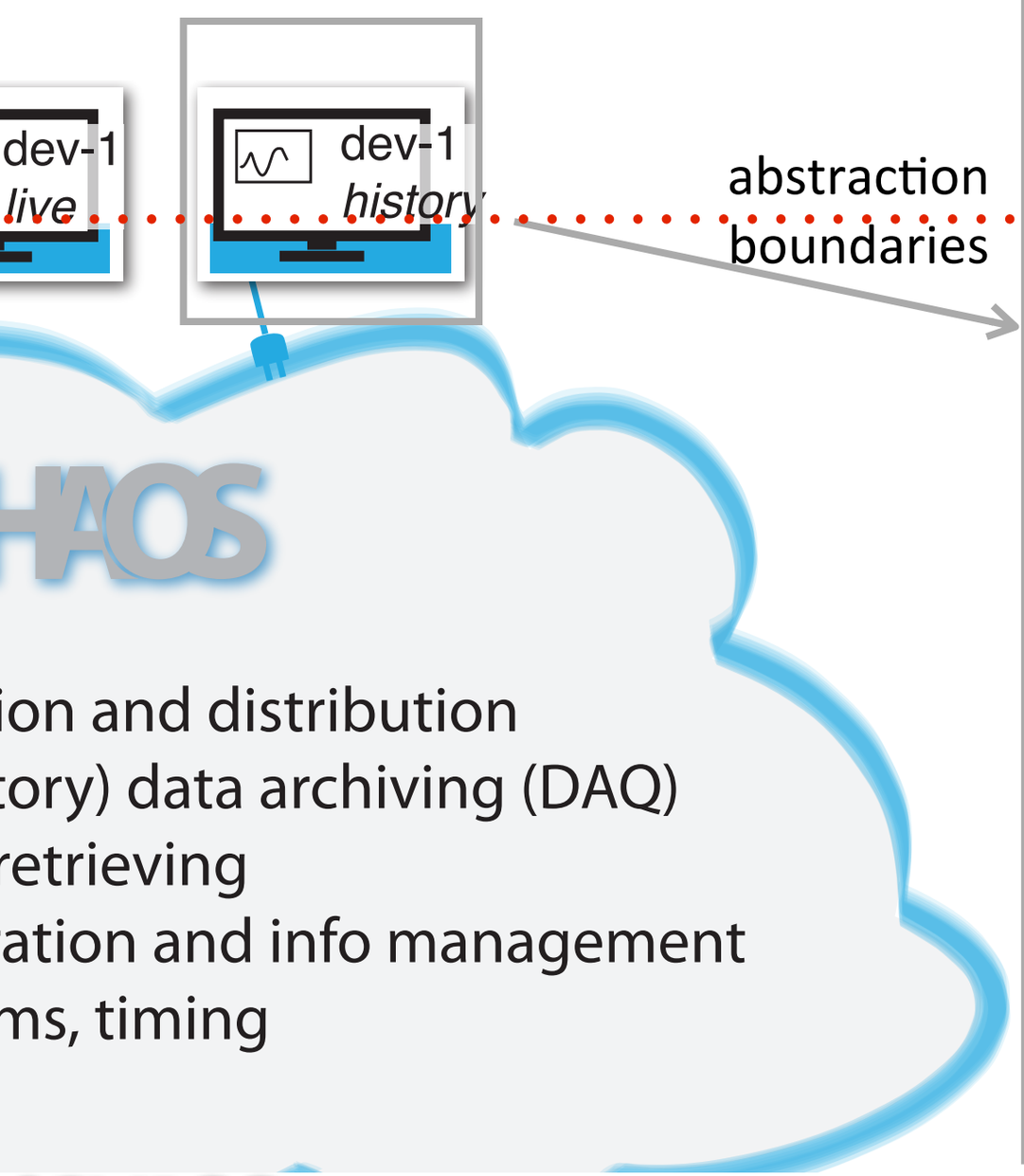


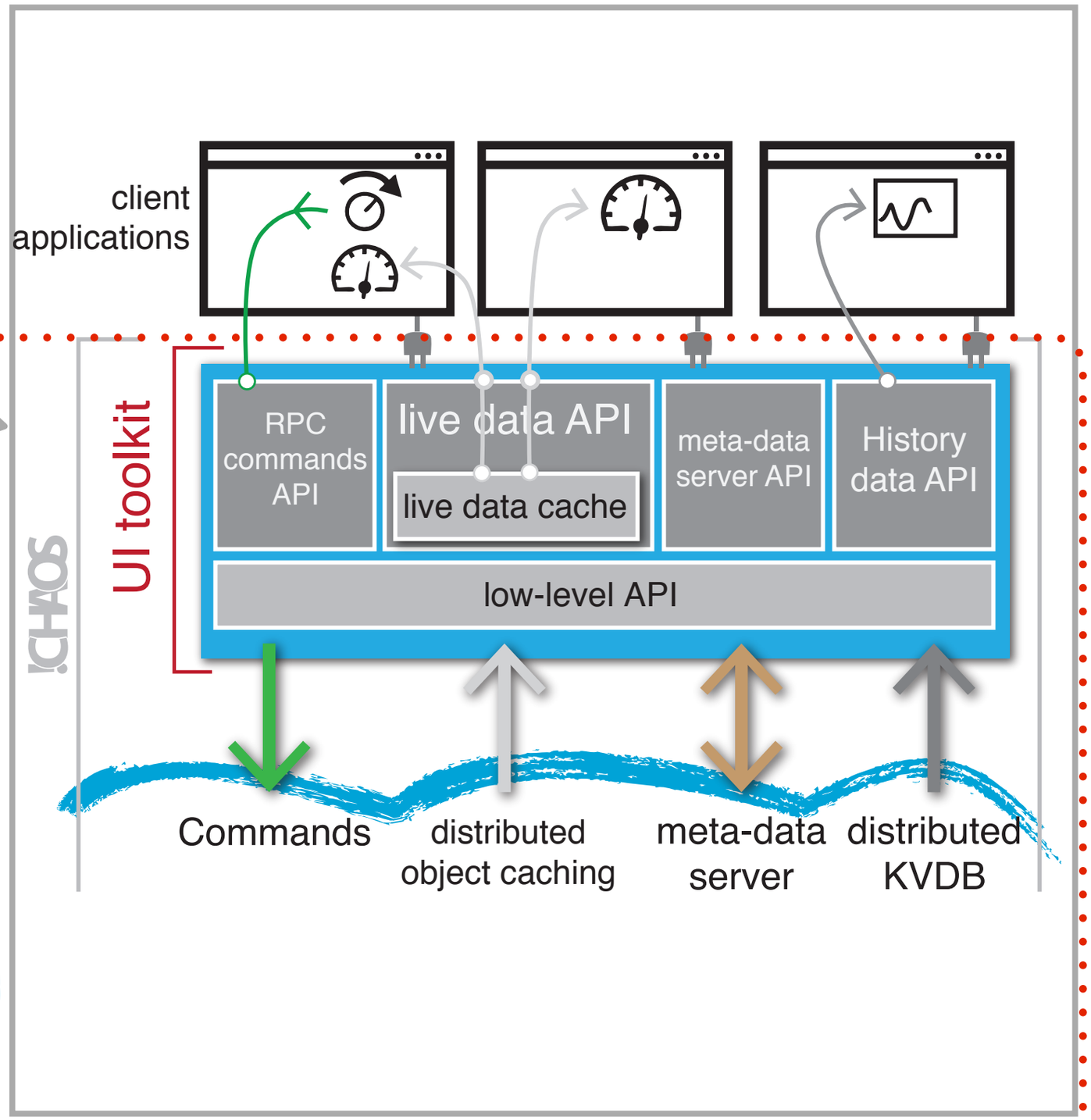
LabVIEW-CHAOS integration

(a very basic solution)



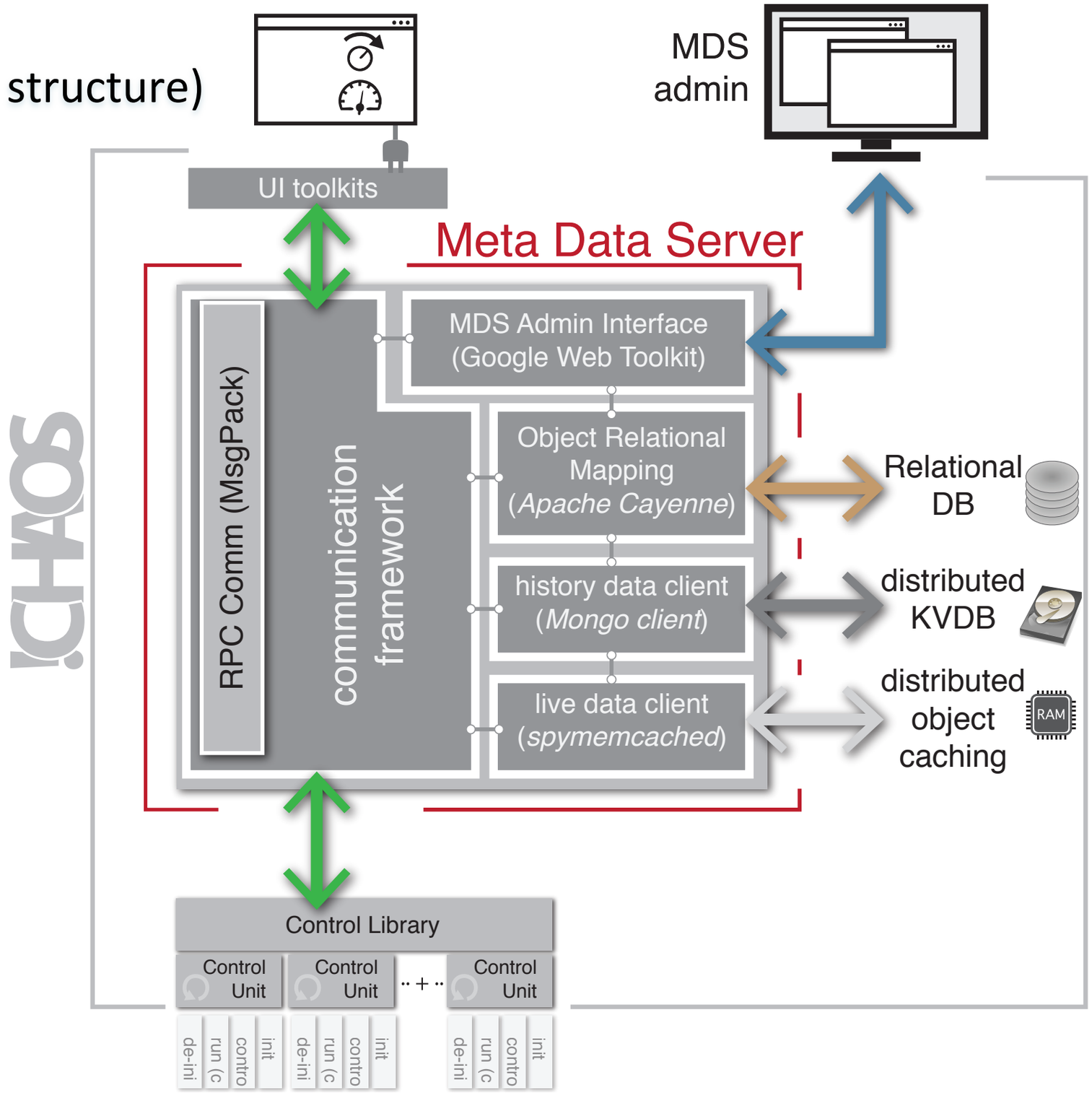
client abstraction



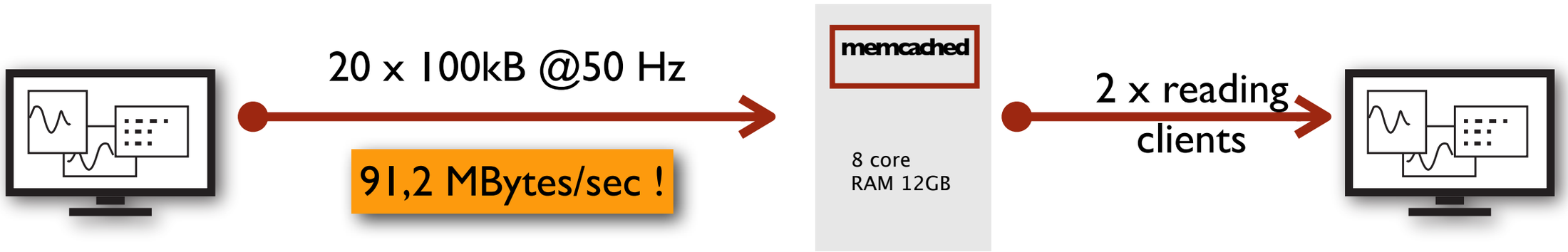


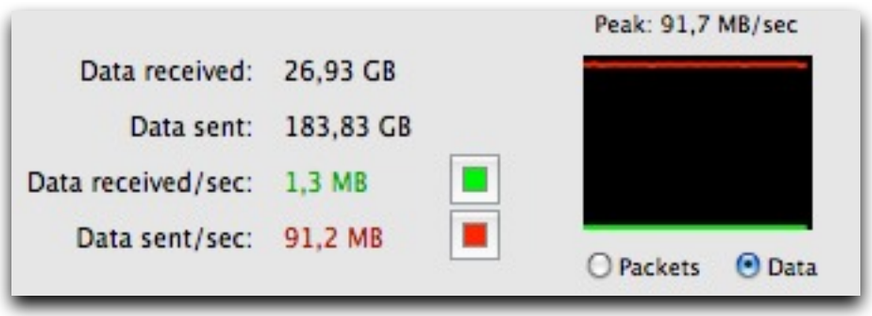
Meta-data Server

- 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



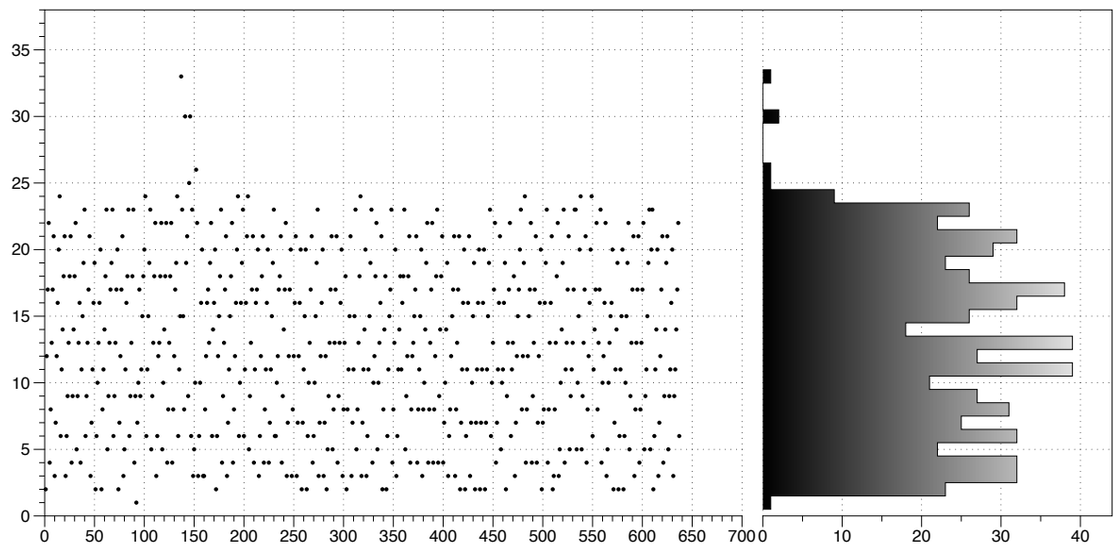
test #4



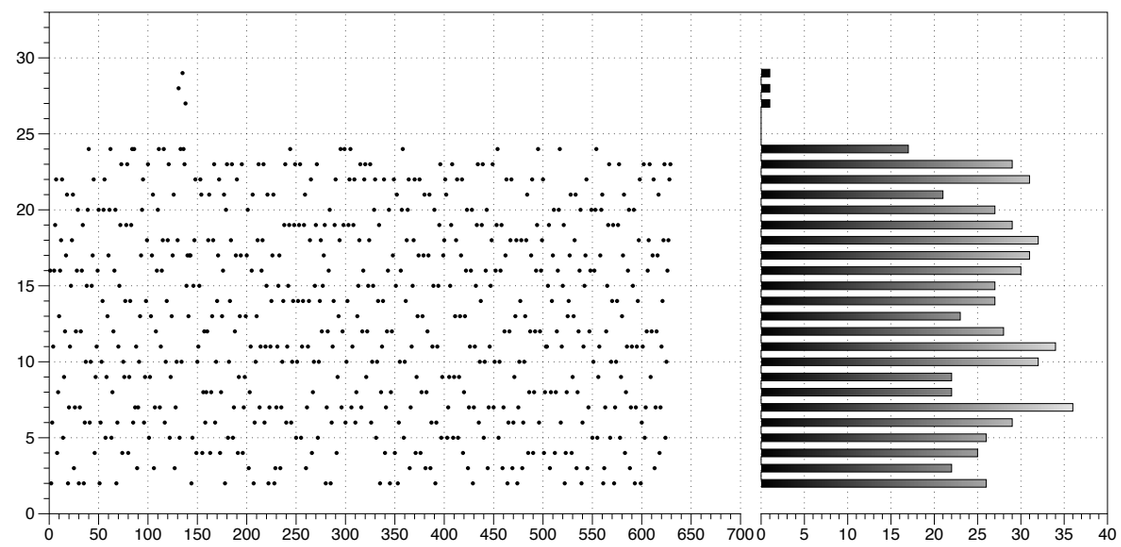


PID	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND
28059	dbuser	15	0	72236	10m	616	11.0	0.1	3:50.82	memcached
28066	dbuser	15	0	129m	5688	628	11.0	0.0	3:09.89	memcached
28052	dbuser	15	0	69812	8024	612	7.0	0.0	2:13.86	memcached
28074	dbuser	15	0	67568	5816	616	4.0	0.0	1:29.09	memcached

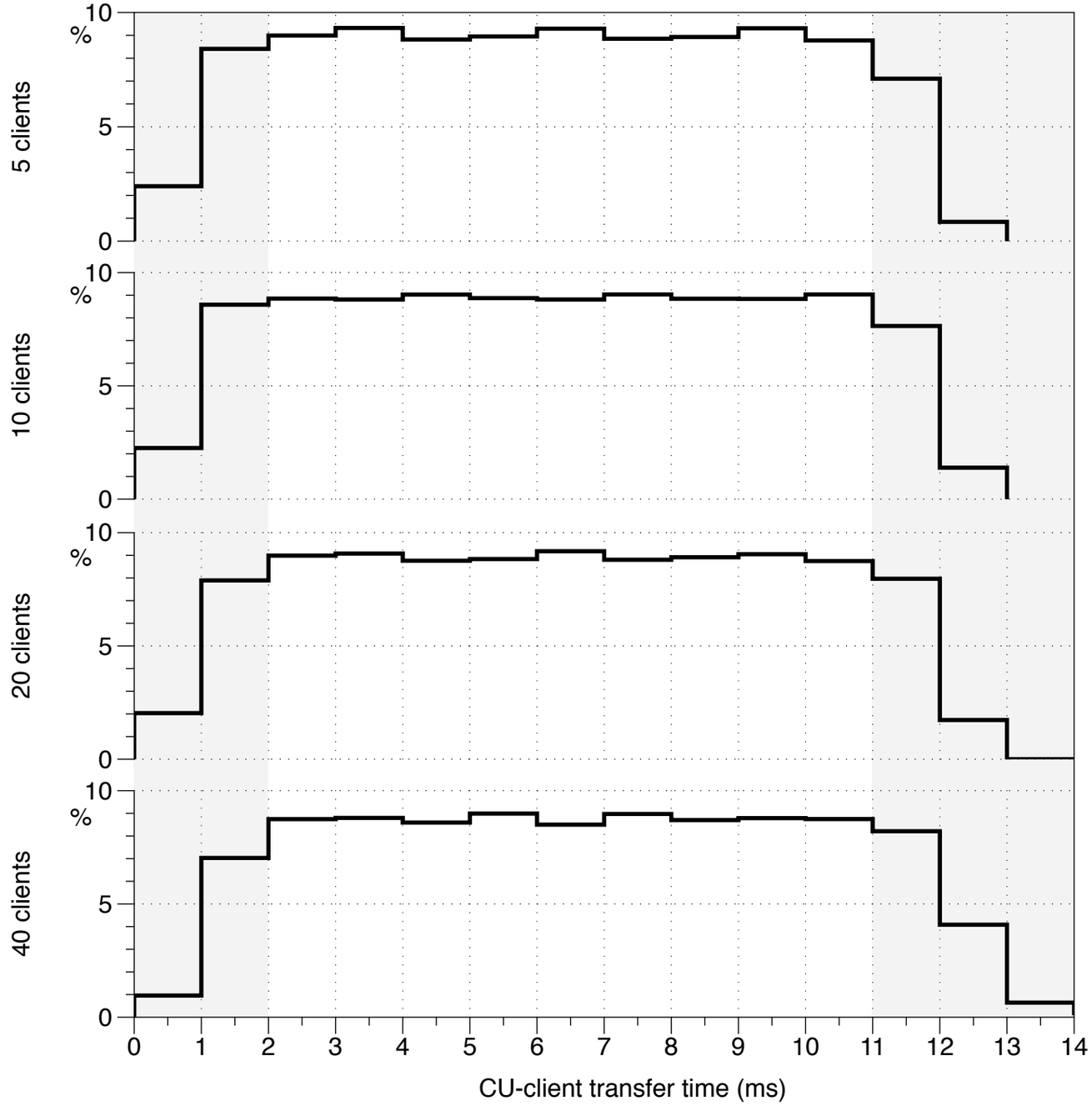
s4_hardware1_w20_m20_buff100000_rd10.log



s4_hardware1_w20_m20_buff100000_rd12.log

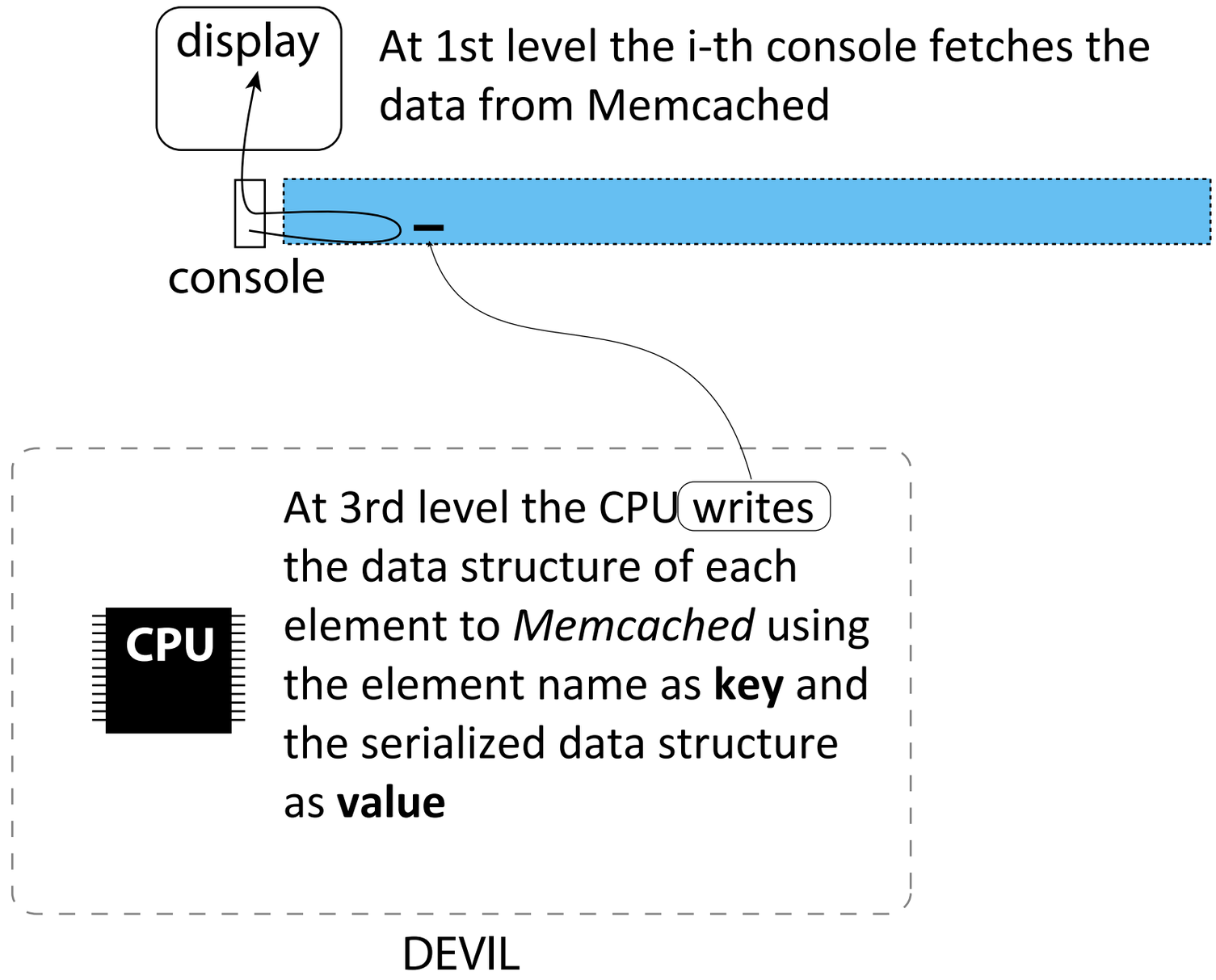


Measured transfer time between front-end CU and a client application via DOC for a different number of concurrent clients reading the same key/value being continuously updated by the CU.



tests at DAFNE

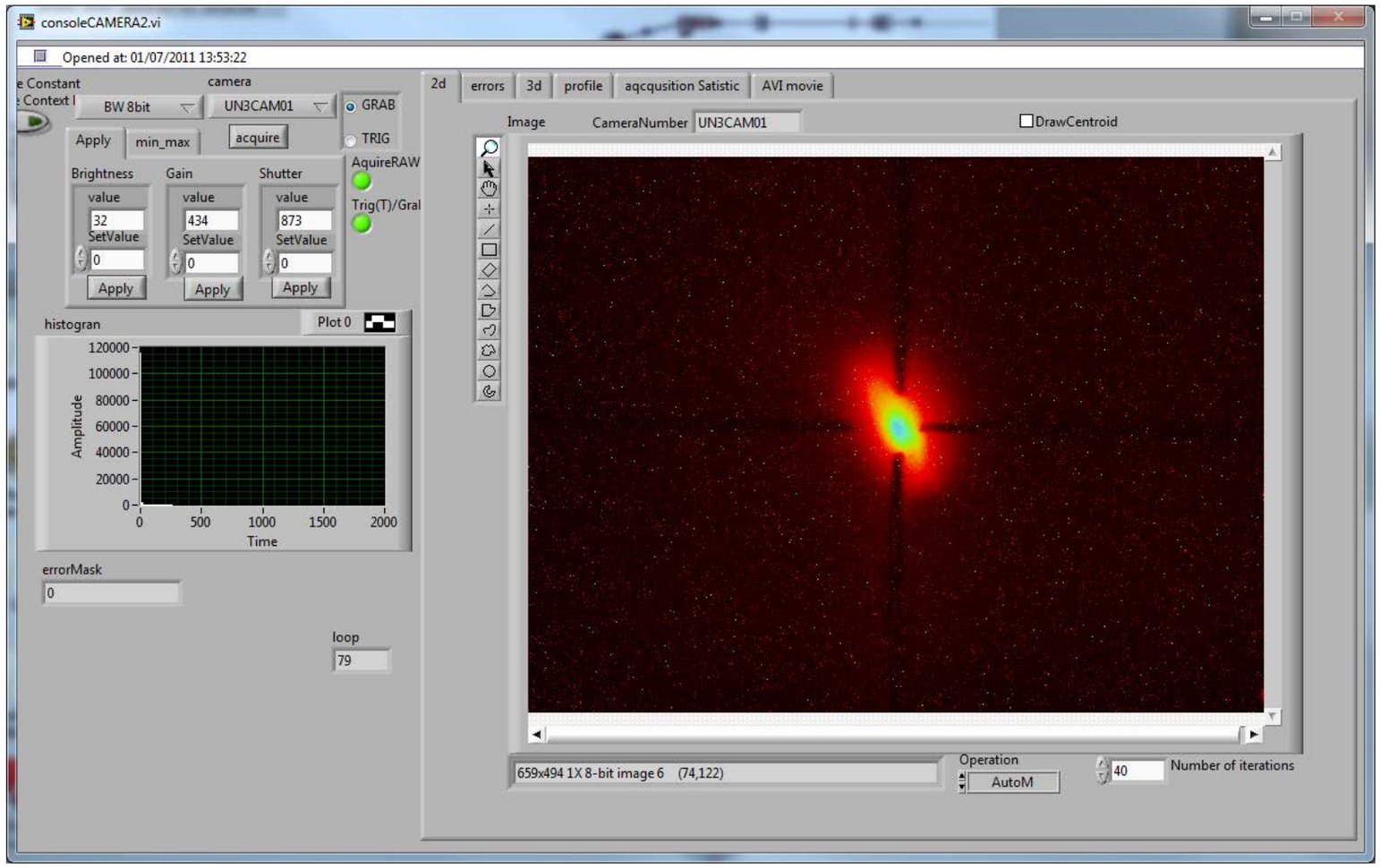
The idea: replace the 2nd level VME address space with the *Memcached* associative memory



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)

ready

>50% ready

Users Application

!C clients gateway

LV clients gateway

gateway to other scripting language

User Interface Toolkit

Common Toolkit

distributed Live Data service

distributed History Data service

commands RPC service

Meta data server

data consumer

!CHAOS infrastructure

abstraction boundaries

Common Toolkit

CU Toolkit

Control Unit

CU modules (config)

CU modules (init)

CU modules (set)

CU modules (de-init)

CU modules (stop)

Control Unit

CU-LV gateway

CU modules (config)

CU modules (init)

CU modules (set)

CU modules (de-init)

CU modules (stop)

I/O Unit

I/O modules

I/O modules

I/O modules

Exec Unit

EU modules

EU modules

EU modules

data producer

man-power (so far)

estimated man power on the !CHAOS code development
and on the project overall

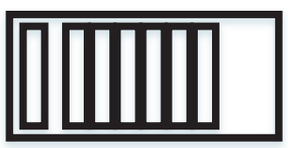
C. Bisegni	1.0 (1.2)
L. Catani	0.3 (0.6)
P. Ciuffetti	0.2 (0.2)
G. Di Pirro	0.1 (0.4)
L. Foggetta	0.4 (0.7)
G. Mazzitelli	0.2 (0.6)
A. Stecchi	0.3 (0.5)
F. Zani	0.3 (0.4)
D. Di Giovenale	0.3 (0.3)*[1.0 FTE for 3 months]*
totale	**3.0 man/year**

development #1

!C User Interface:	*prototype ready - completed 60%*
UI Toolkit:	*prototype ready - completed 40%*
LV Clients gtwy:	*prototype in preparation (shlib for LV) completed 70%*

!CHAOS infrastructure

Live Data:	*single-instance (no-distributed) - completed 90%*
Live Data:	*distributed and scalable service - completed 0%*
RPC service:	*prototype ready (msgpack) - completed 80%*
Meta Data server:	*MDS simulator ready; MDS under development completed 30%*



Common Toolkit:	*prototype ready - completed 80%*
CU Toolkit:	*prototype ready - completed 80%*
CU modules:	*prototype ready (C++) - completed 80%*
CU-LV Gtway & modules:	*prototypes almost ready completed 80%*

development #2

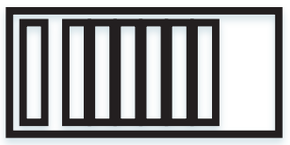
Gateway to scripting: to be *defined according to users' needs*
languages *completed 0%*

!CHAOS infrastructure

Live Data: *distributed and scalable service - completed 0%*

scalability & availability: *test facility in preparation*

History Data service: *performance test ongoing (UniTV)*



Execution Unit: *concept ready - completed 10%*

I/O Unit: *concept ready - completed 5%*

man-power needed for completing the !CHAOS infrastructure in two years

!CHAOS Group	~3.0
SuperB	2.0
others/SupB	~1.0
total (man/year)	**~6.0**

+ some support for travelling