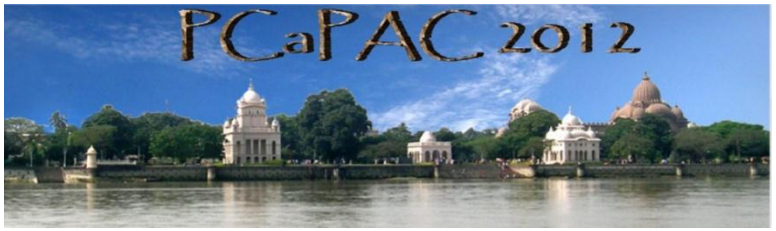
Using Memcahed as real-time database in the SPARC Control System

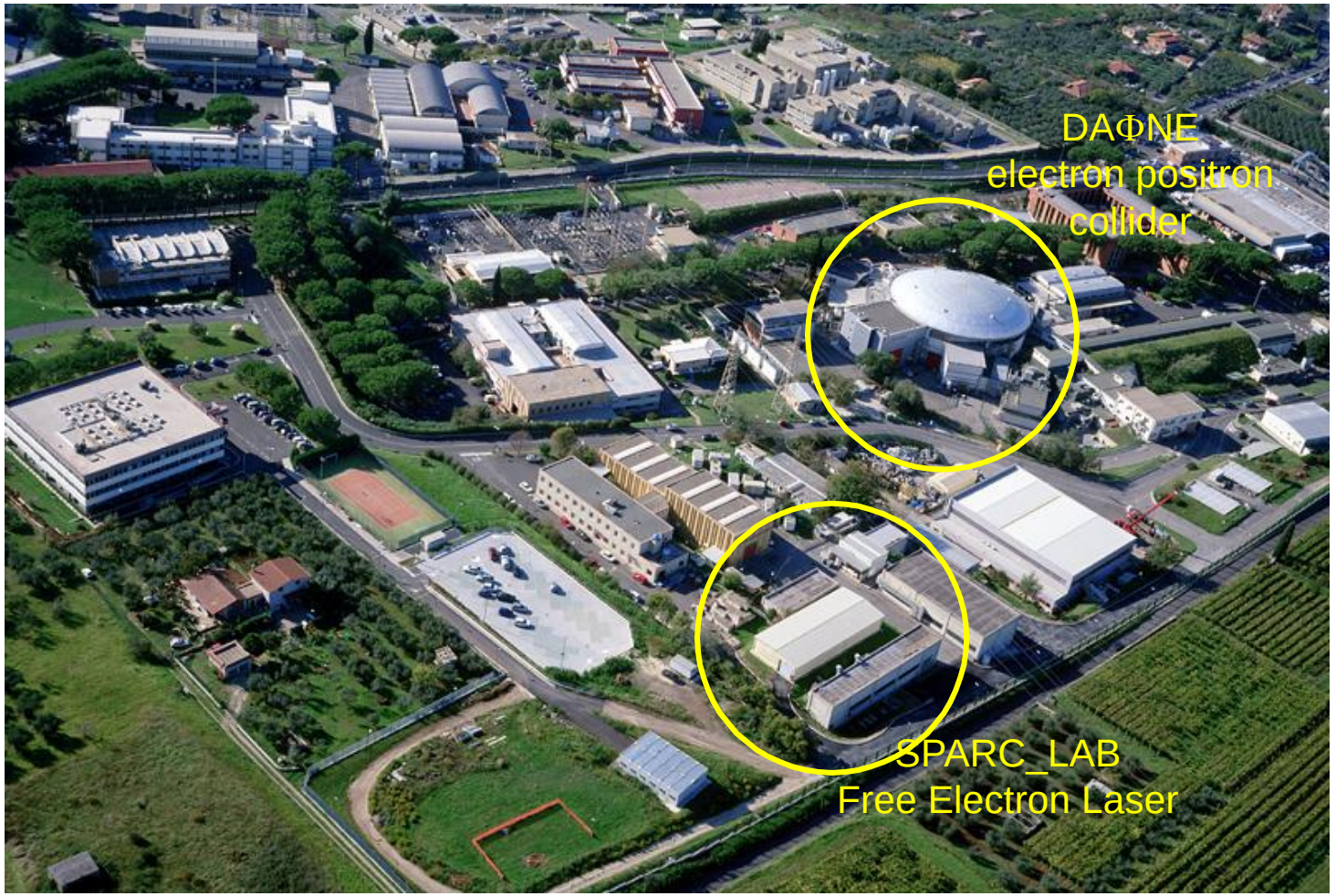
G. Di Pirro,, E. Pace, INFN LNF, Italy



INFN Istituto Nazionale di Fisica Nucleare
Laboratori Nazionali di Frascati

SPARC_LAB

The Frascati Laboratory



The SPARC_LAB Facility

SPARC_LAB Sources for Plasma Accelerators and Radiation Compton with Lasers And Beams

SPARC_LAB *is a facility based on the* ***unique combination of high brightness electron beams, from*** *the* ***SPARC*** *photo-injector,* ***with high intensity ultra-short laser pulses, from FLAME****. It will allow the* ***investigation of*** *all the* ***different configurations of plasma acceleration*** *and the* ***development of*** *a wide spectrum inter-disciplinary leading-edge research activity with* ***advanced radiation sources****, e.g. x-ray and THz radiation.*

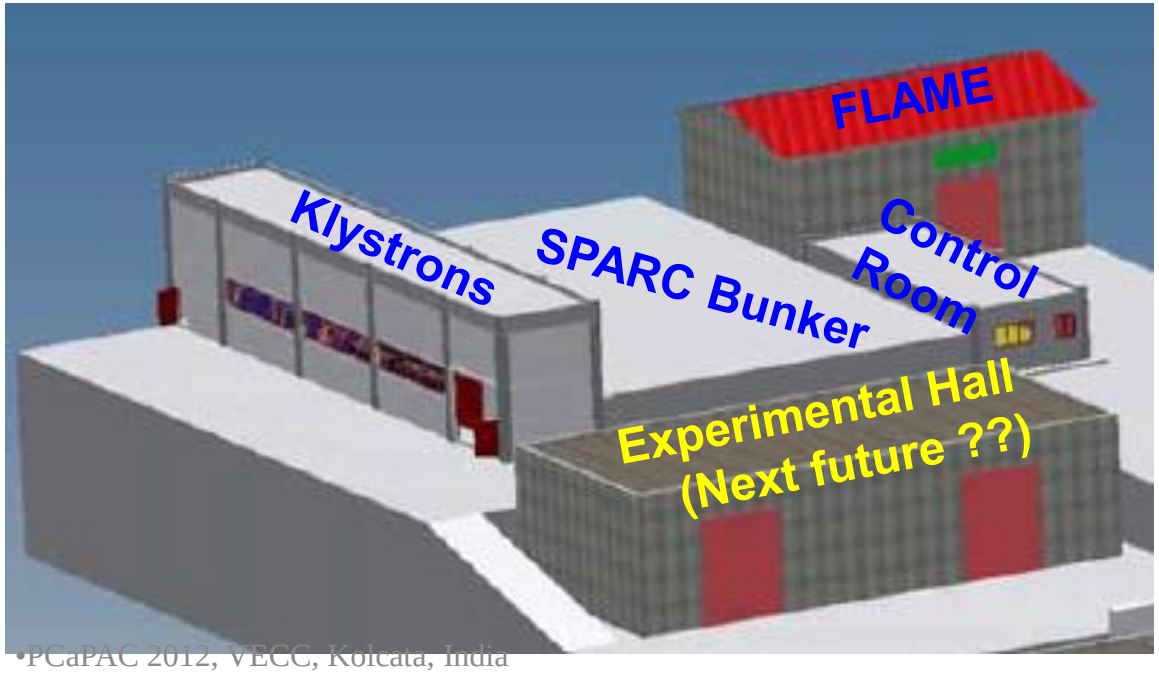
INFN Istituto Nazionale di Fisica Nucleare

ENEA Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile

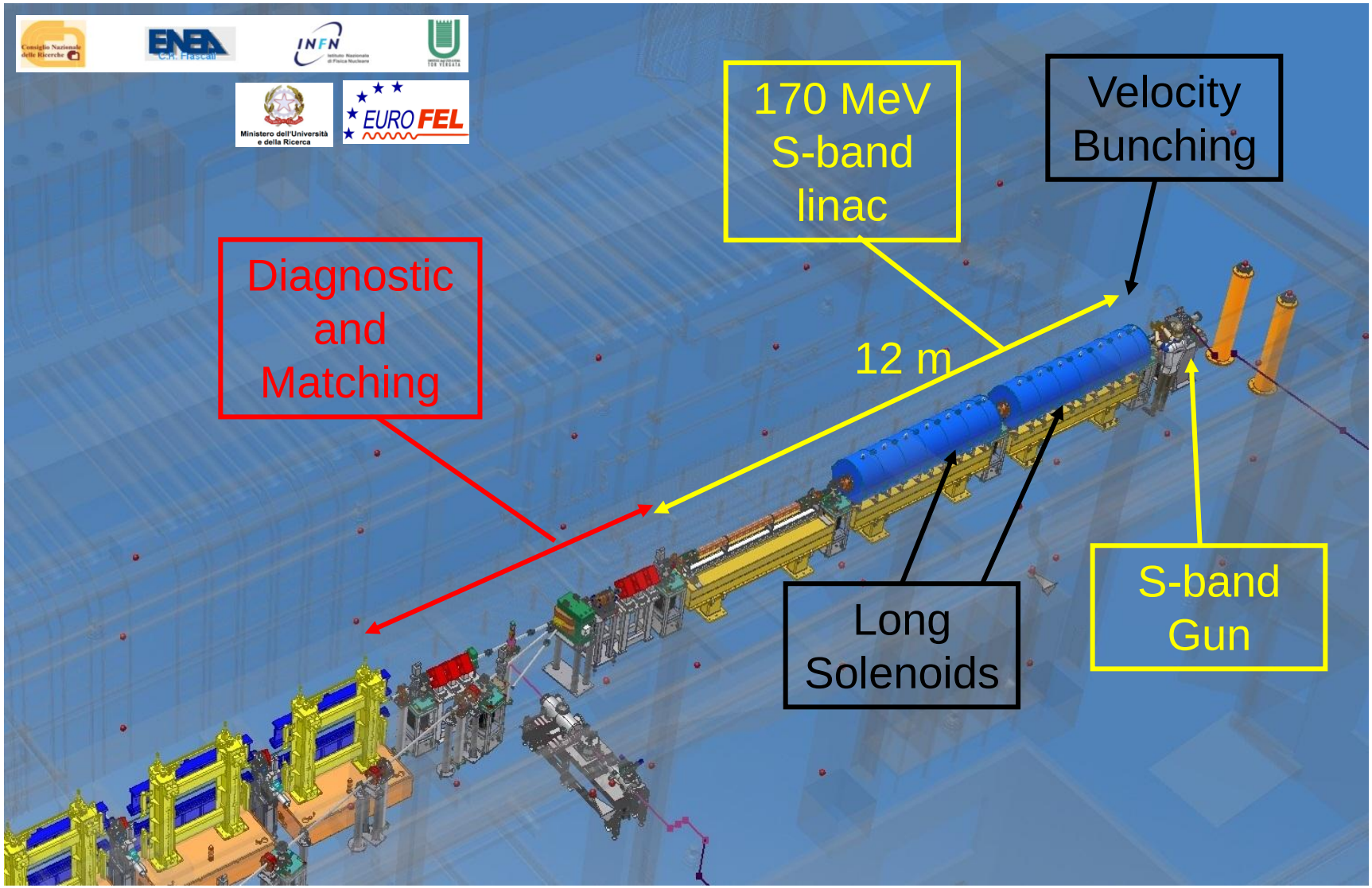
UNIVERSITA' degli STUDI di ROMA TOR VERGATA

Consiglio Nazionale delle Ricerche

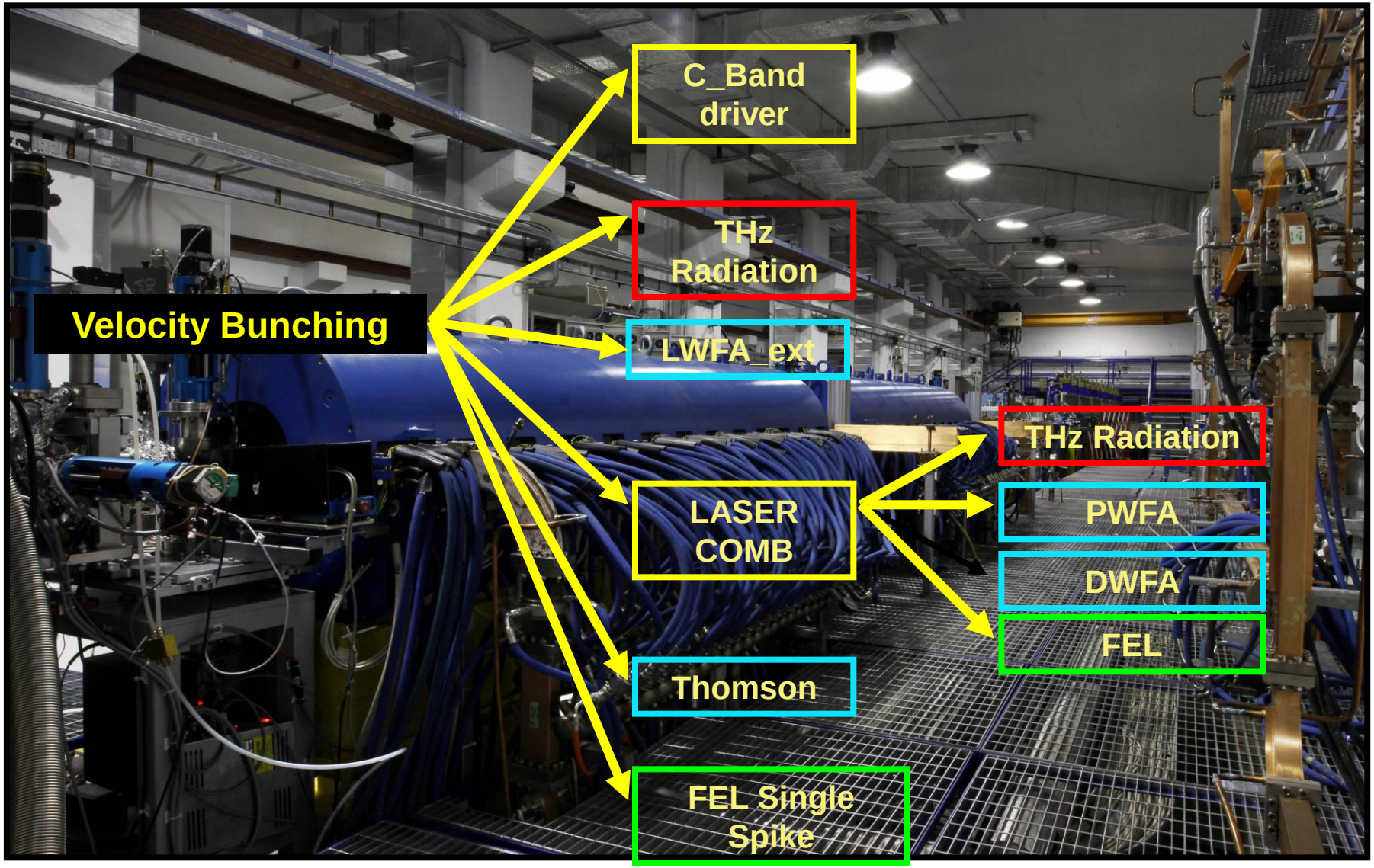
La Sapienza Università degli Studi di Roma



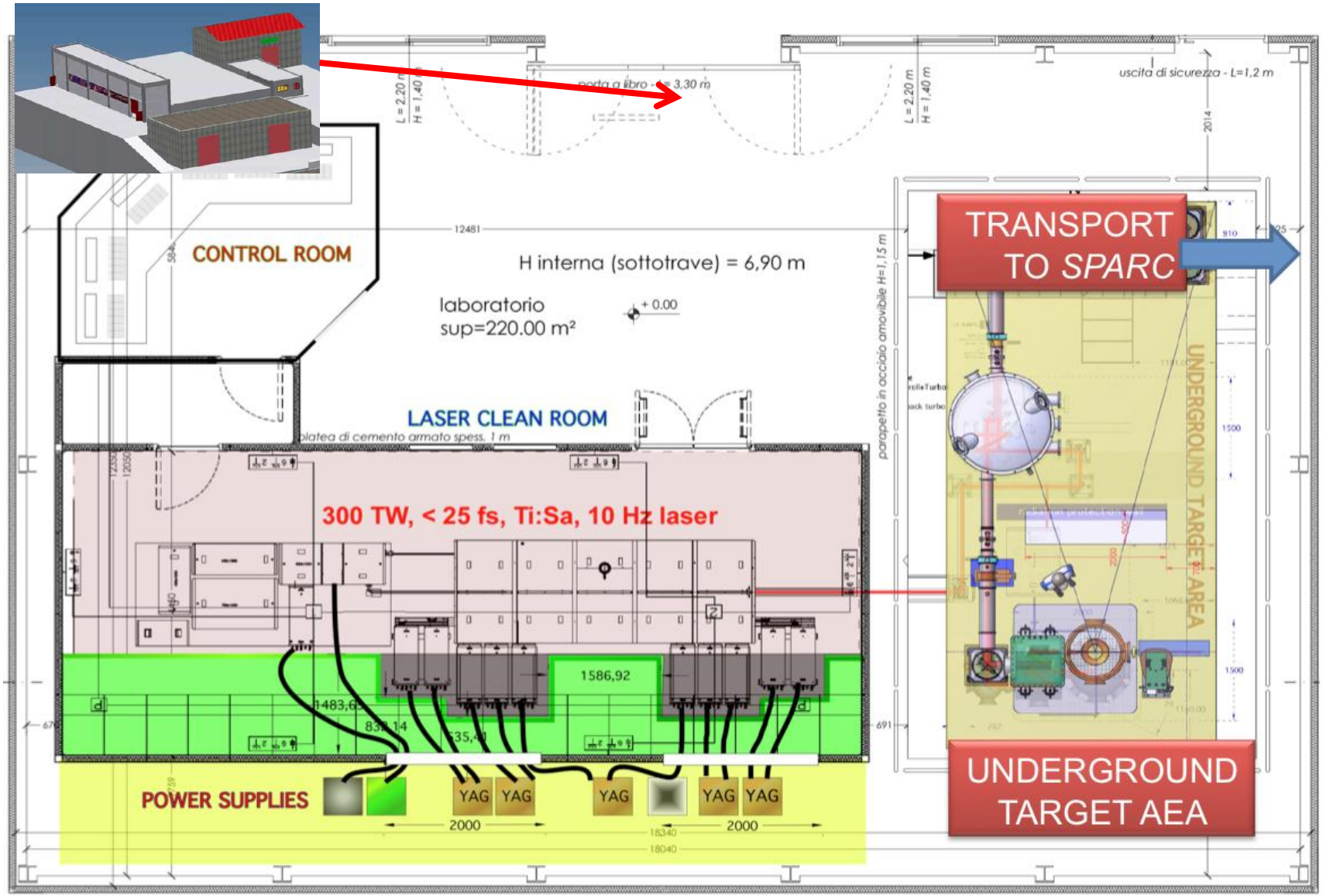
SPARC Photoinjector



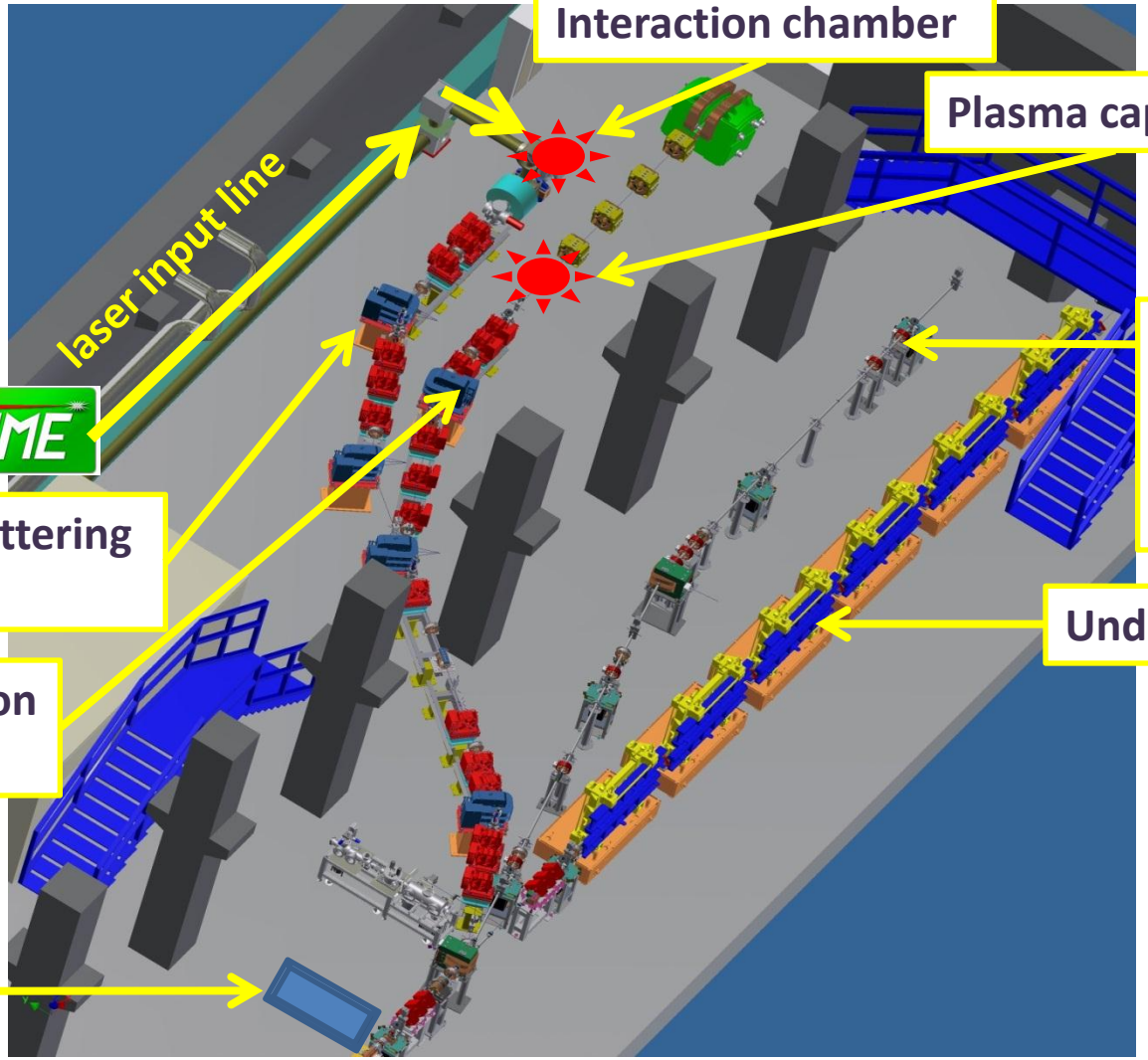
SPARC Photoinjector



FLAME: a 300 TW Ti:Sa Laser

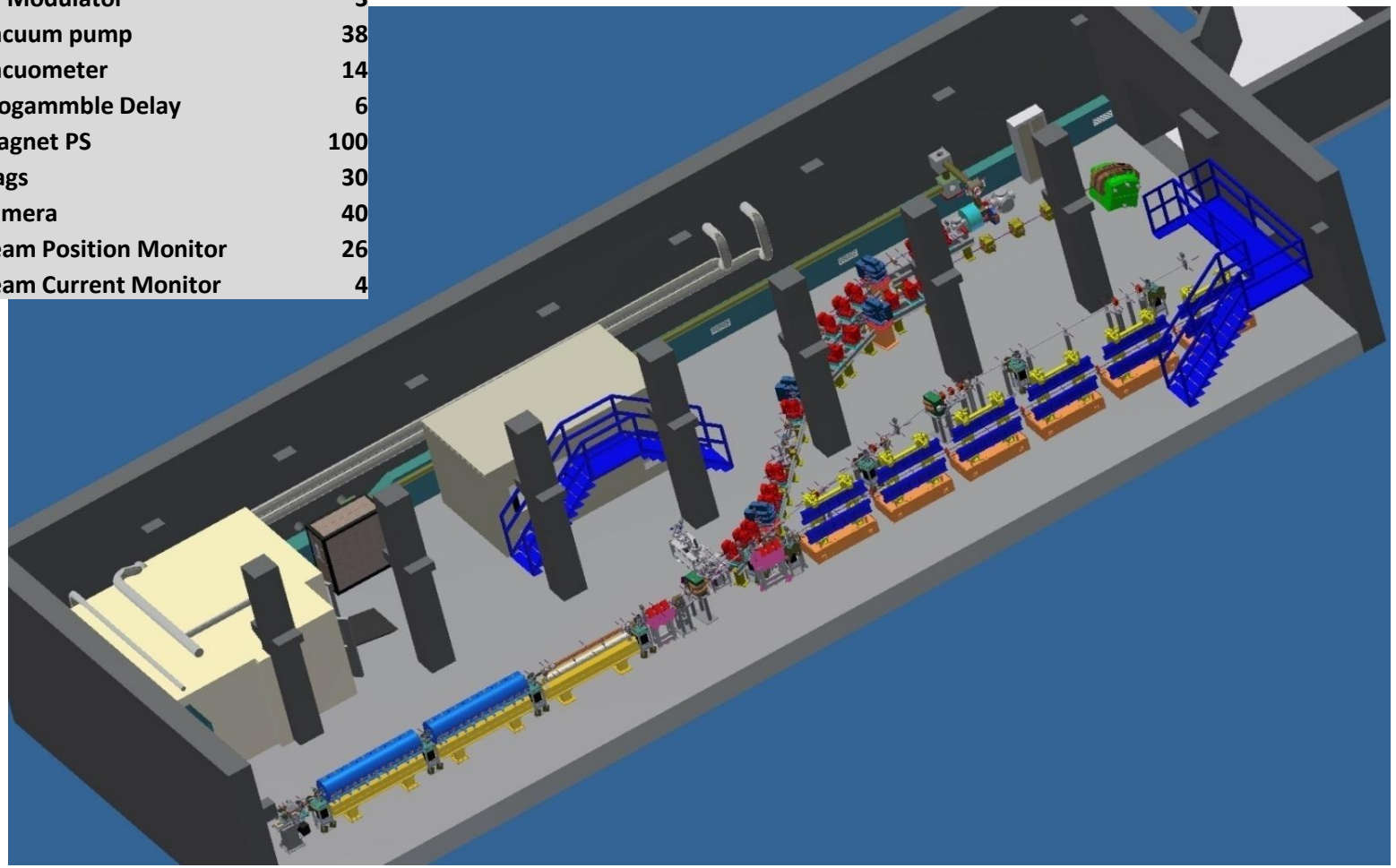


The SPARC_LAB Layout

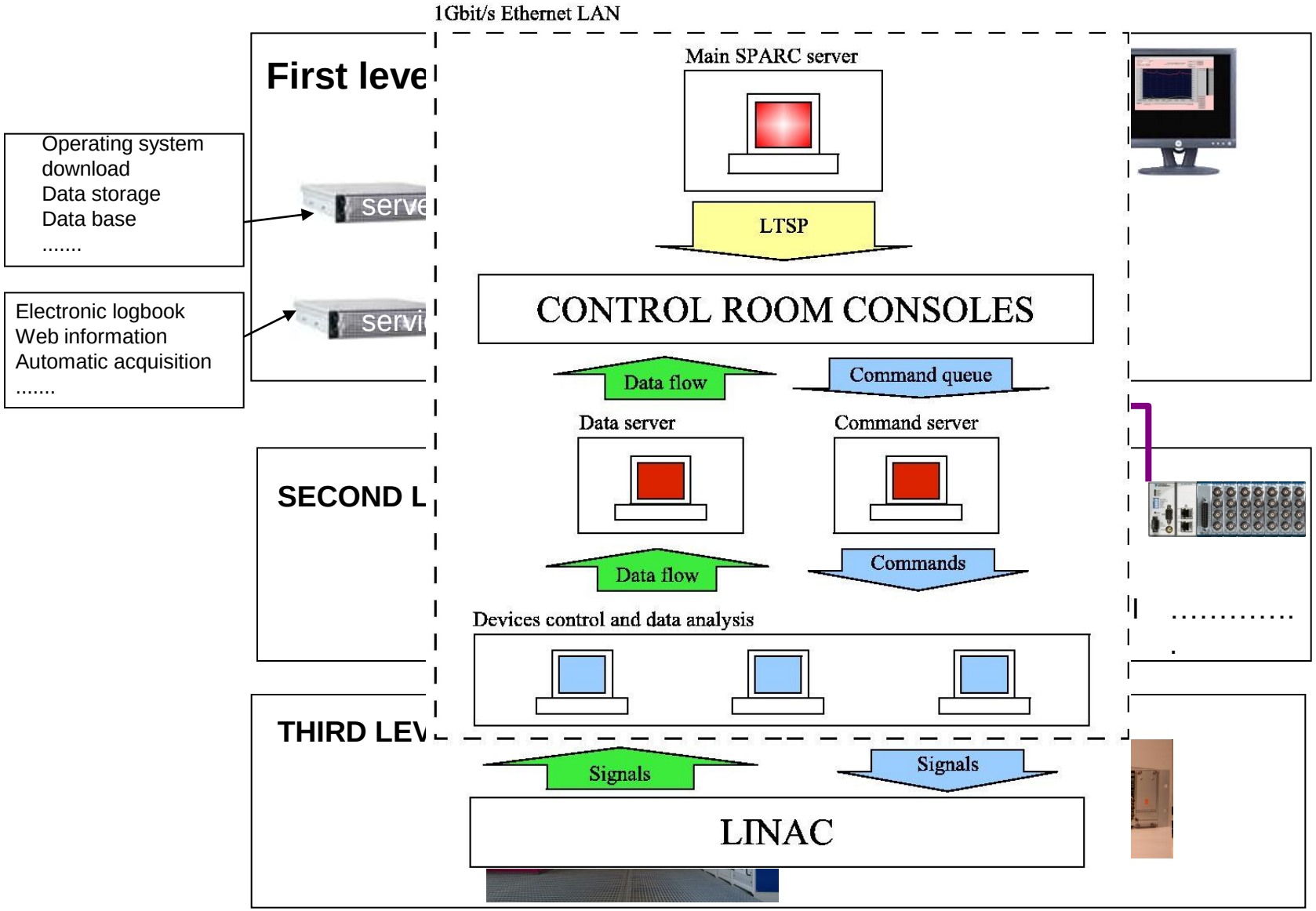


New SPARC Layout

RF Modulator	3
Vacuum pump	38
Vacuometer	14
Progammble Delay	6
Magnet PS	100
Flags	30
Camera	40
Beam Position Monitor	26
Beam Current Monitor	4

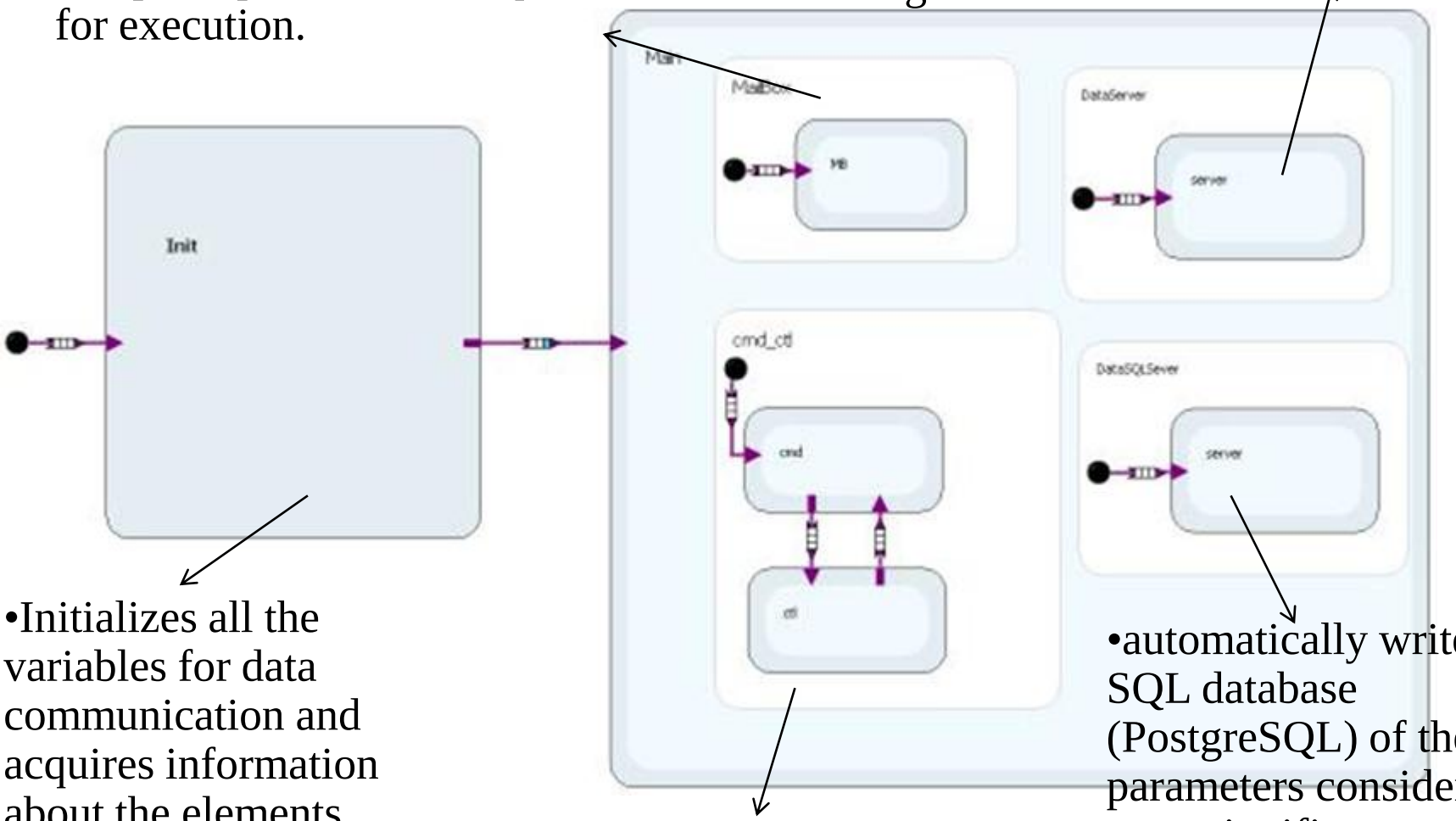


CONTROL SYSTEM



•acquires any commands from a console or from another computer puts them in a queue for execution.

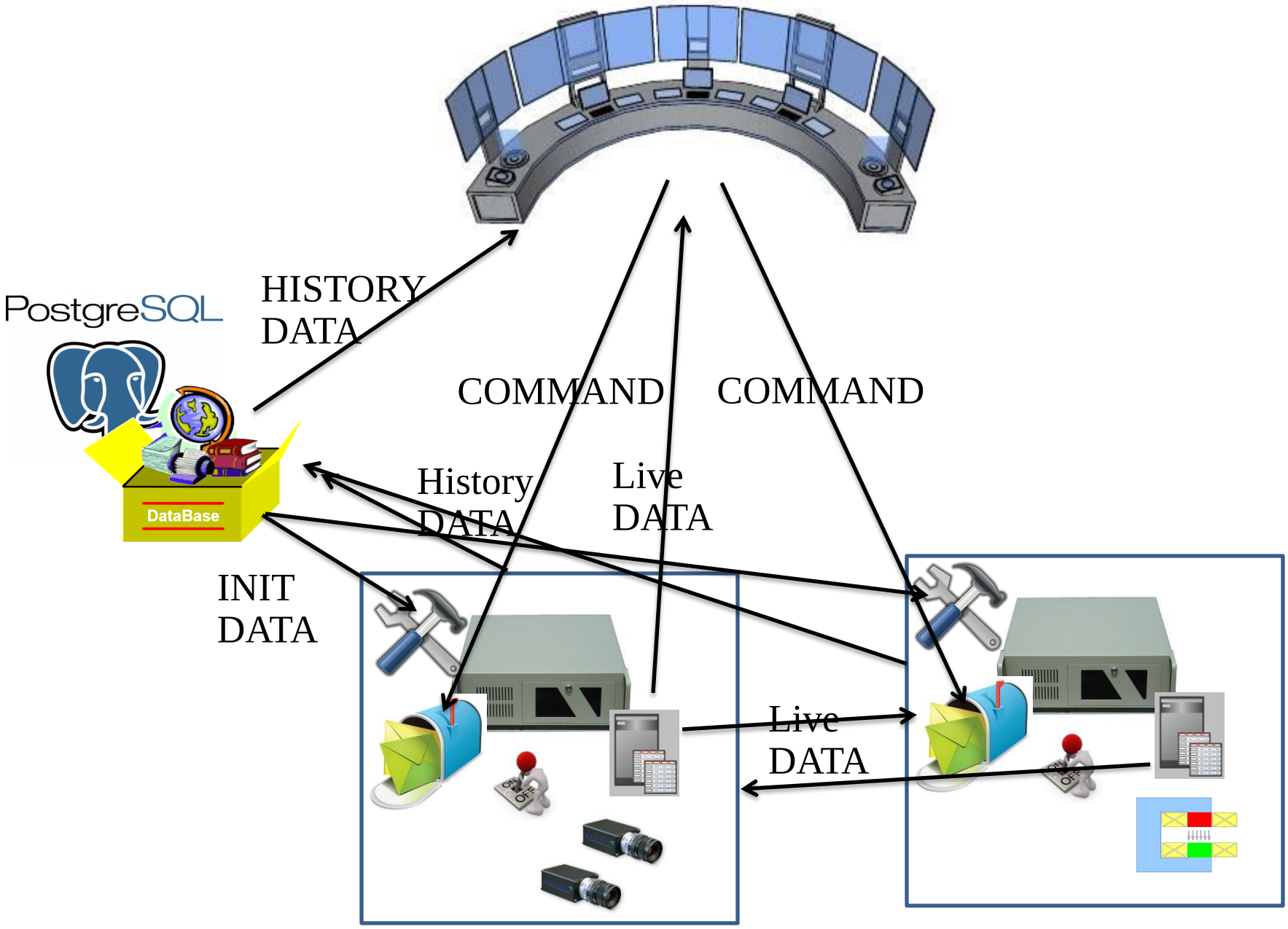
•Sends the clusters of the elements under control at the request of the higher level



•Initializes all the variables for data communication and acquires information about the elements under controlled and perform any hardware initialization

•execute the command if present in the command queue. The second operation is the control operation.

•automatically writes a SQL database (PostgreSQL) of the parameters considered most significant. Saving can be done either periodically or upon significant change


PostgreSQL
DataBase
HISTORY
DATA
COMMAND
COMMAND
History
DATA
Live
DATA
INIT
DATA
Live
DATA

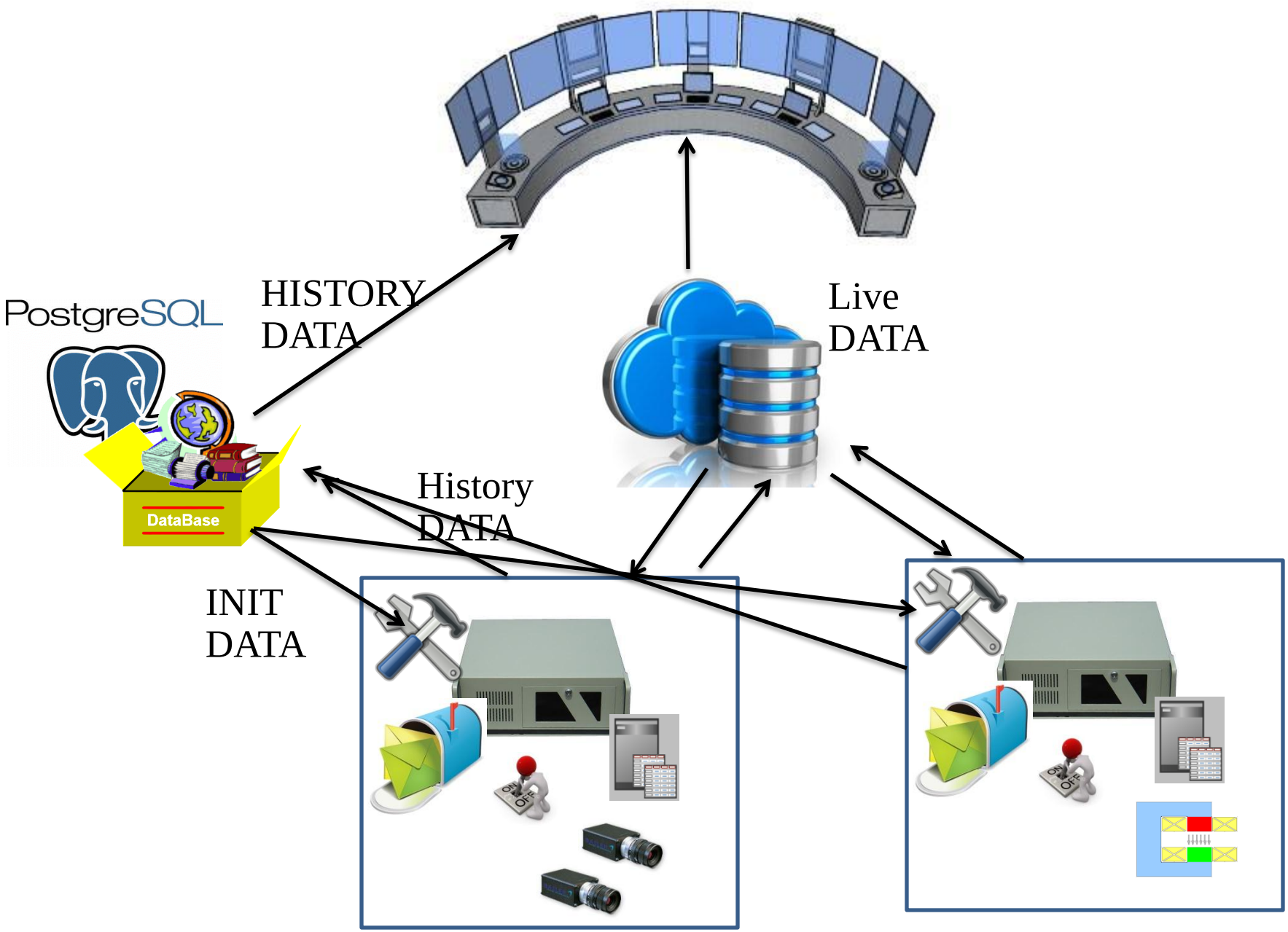
- The whole control system was developed using LabView
- The control system (hardware & software) is completely defined, tested and used.
- All element of the machine are under control

BUT

Bad performance on the console window when one front-end process is stopped

Slow performance when multiple interconnection

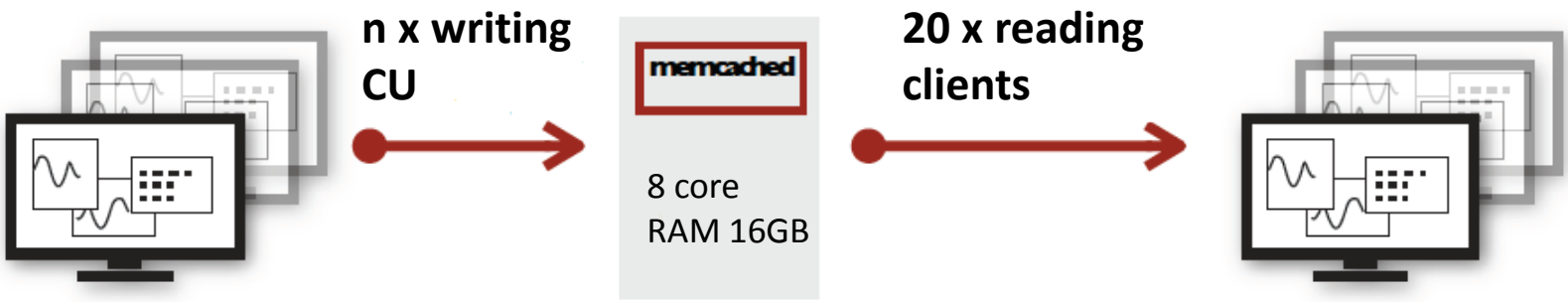
Cloud


PostgreSQL
DataBase
HISTORY
DATA
Live
DATA
History
DATA
INIT
DATA

!CHAOS

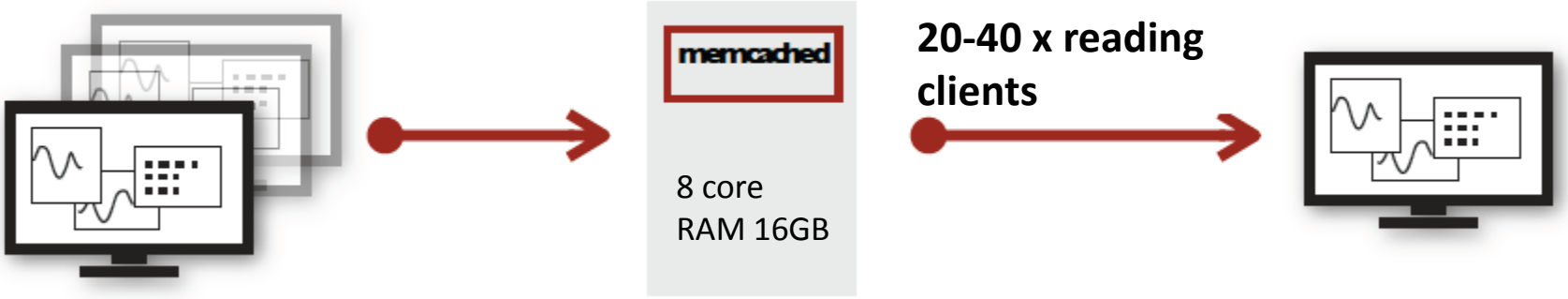
L. Catani
FRID01 Introducing the !CHAOS Control Systems Framework

Test #3.1



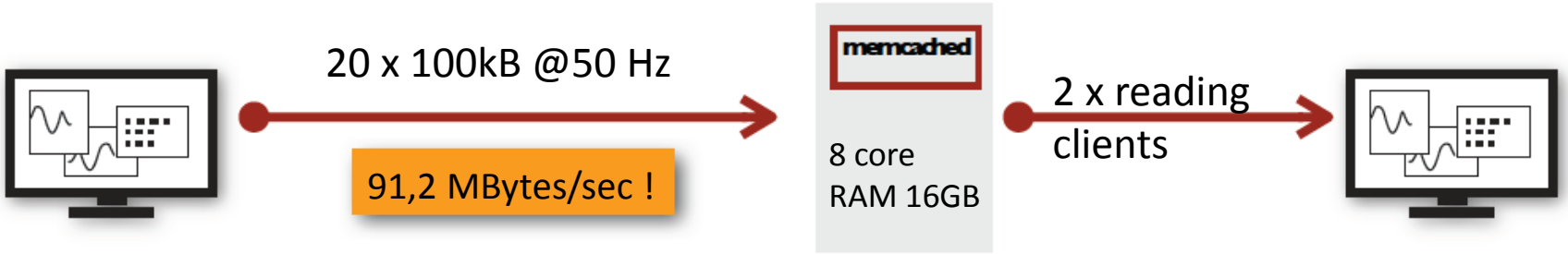
writing every... (msec)	#CU (Write)	#clients (Read)	#servers	#processes/ server	CPU load (%)
20	60	20	1	1	3-5
20	80	20	1	1	4-6
20	80	20	2	1	2-3
50	60	20	1	1	1-3
50	80	20	2	1	0-2
100	60	20	1	1	?
100	80	20	2	1	?

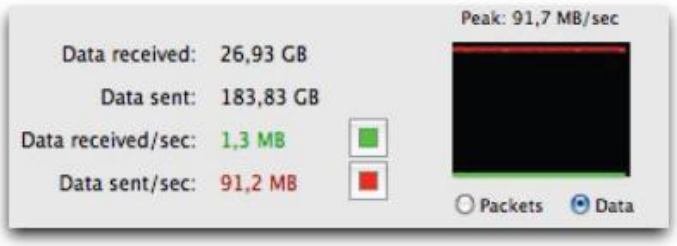
Test #3.2



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

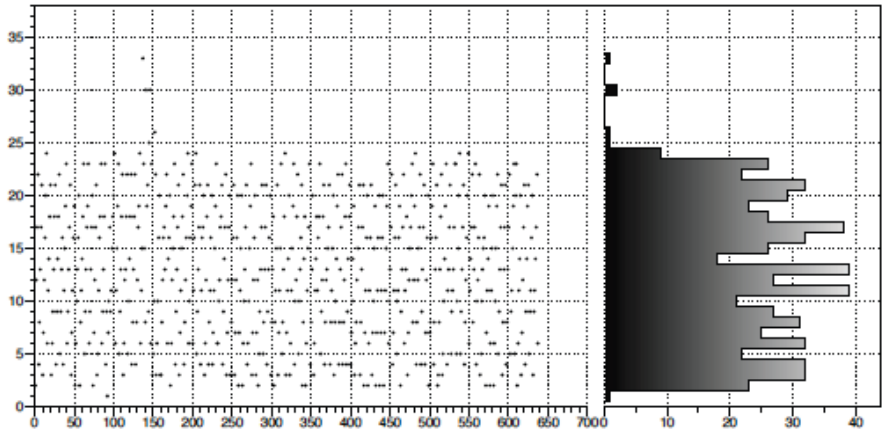
Test #4



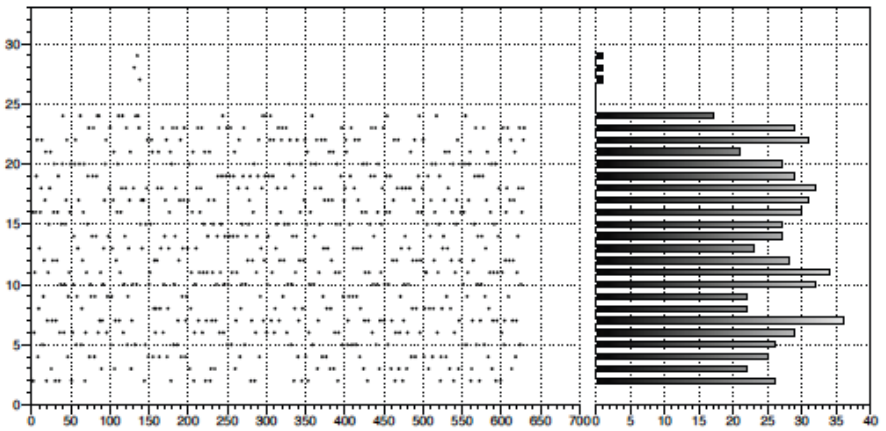


PID	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND
28059	dbuser	15	0	72236	10m	616	11.0	0.1	3:50.82	memcached
28066	dbuser	15	0	129m	5688	628	11.0	0.0	3:09.89	memcached
28052	dbuser	15	0	69812	8024	612	7.0	0.0	2:13.86	memcached
28074	dbuser	15	0	67568	5816	616	4.0	0.0	1:29.09	memcached

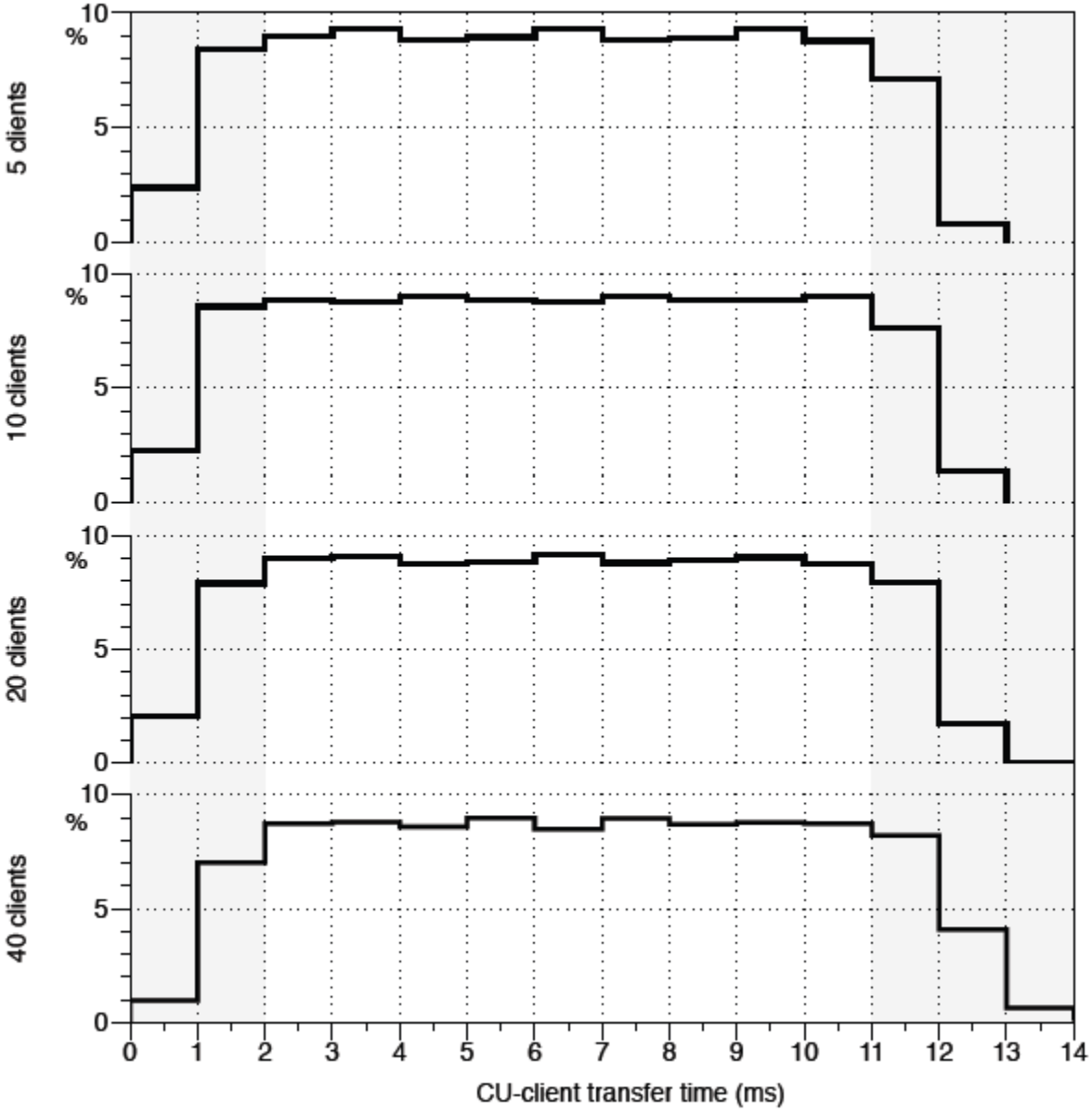
s4_hardware1_w20_m20_buff100000_rd10



s4_hardware1_w20_m20_buff100000_rd12



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.



Memcahed
Open source

LiVEJOURNAL

TypePad

YellowBot

WordPress.com

flickr

YouTube
Broadcast Yourself

digg

What is it

- Key/value store
- Store everything in RAM
- “Forgetting data is a feature”
- No persistent storage

What isn’t it

- Database
- HTTP accelerator

Architecture

- Divided into two parts:
- server
- client
- **Important:** "intelligence" split between client and server

Server

- Stores the key / value pairs

Clear stored data when:

- Is reached the deadline set by the user
- Runs out of memory available

Not 'aware of the other server:

- Do not talk to them
- There isn‘t synchronization
- There are no messages in broadcast
- Don’t know what is stored.

Client

- Available for the main programming languages
- Has a list of all servers
- Chooses each time the server on which to store data
- Know what to do when a server is not reach
- Takes care of serializing the values to be stored
- Know how to retrieve data

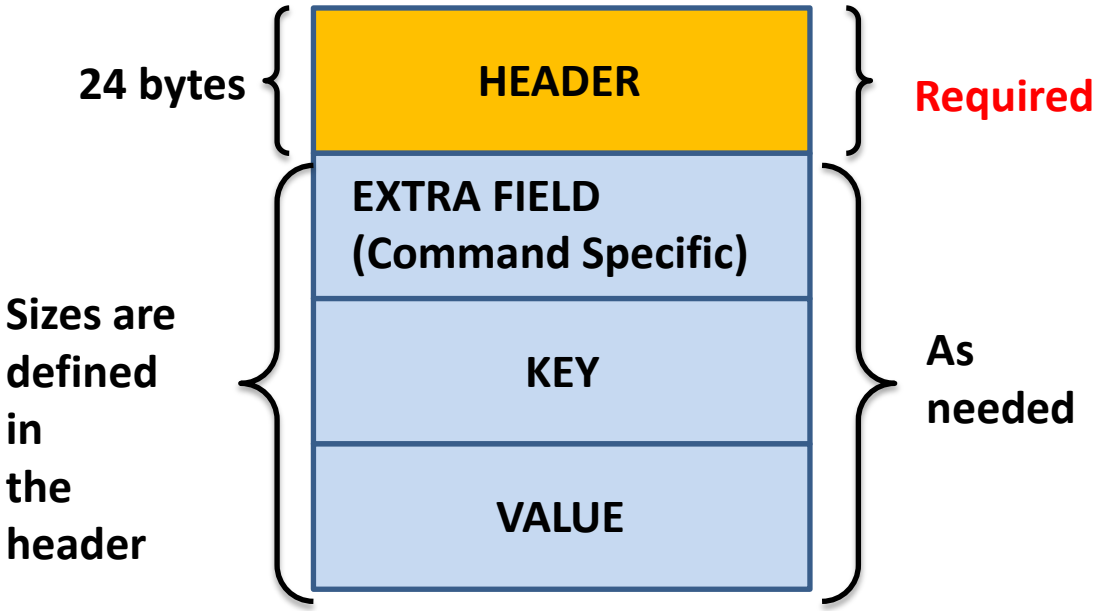
Labview implementation

- Exist C library

But

- Simple TCP/IP comunication

Packet Structure



Packet Structure

- Two types of headers:
- Request Header
- Response Header

The Request Header

MAGIC (1 byte)	Opcode (1 byte)	Key Length (2 bytes)
Extra Length (1 byte)	Data Type (1 byte)	Reserved (2 bytes)
Total Body Length (4 bytes)		
Opaque (4 bytes)		
CAS (Compare and Swap) (8 bytes)		

MAGIC

0x80 = Request Packet

The Response Header

MAGIC (1 byte)	Opcode (1 byte)	Key Length (2 bytes)
Extra Length (1 byte)	Data Type (1 byte)	Status (2 bytes)
Total Body Length (4 bytes)		
Opaque (4 bytes)		
CAS (Compare and Swap) (8 bytes)		

MAGIC

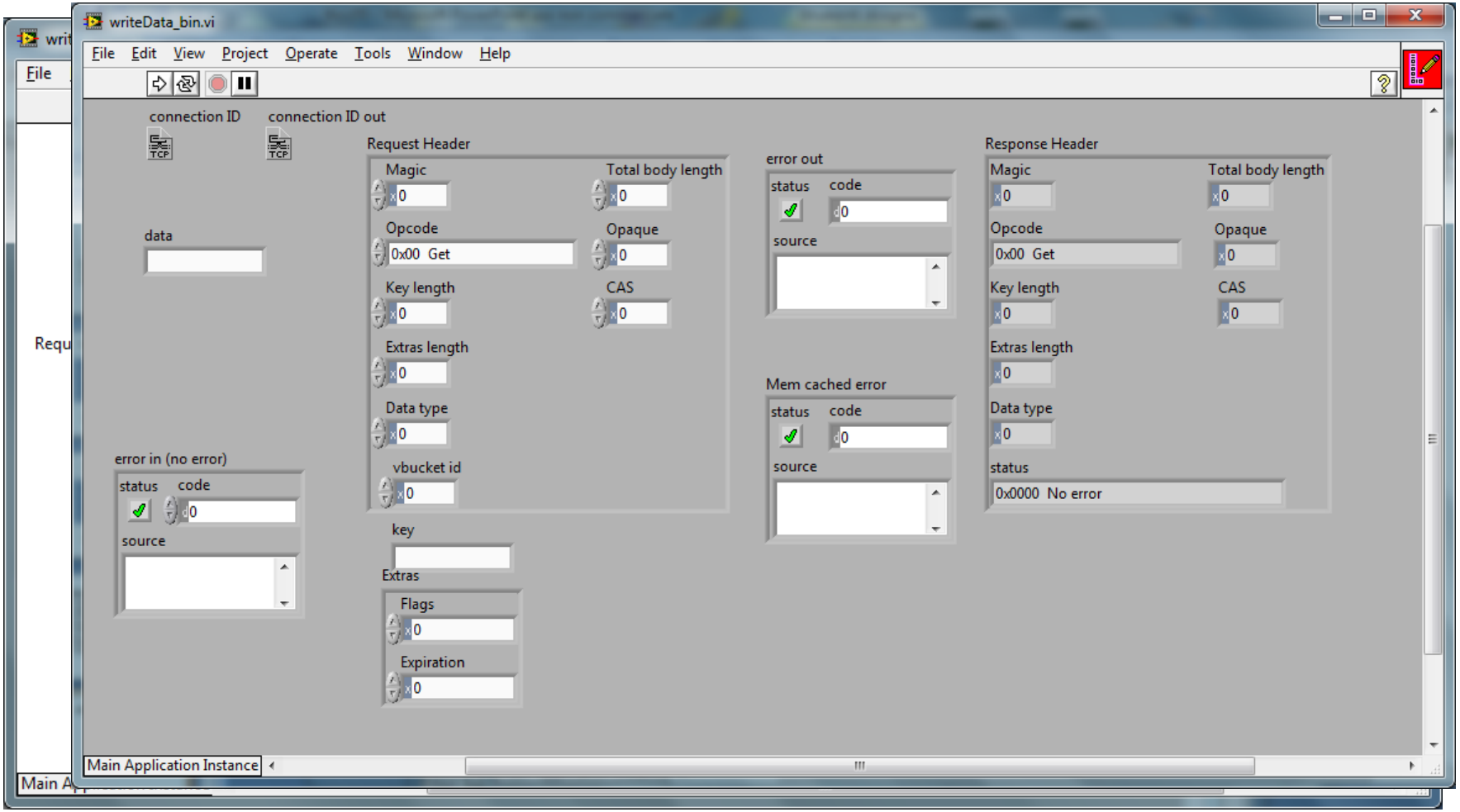
0x81 = Request Packet

OPCODE
Represents the Command Type

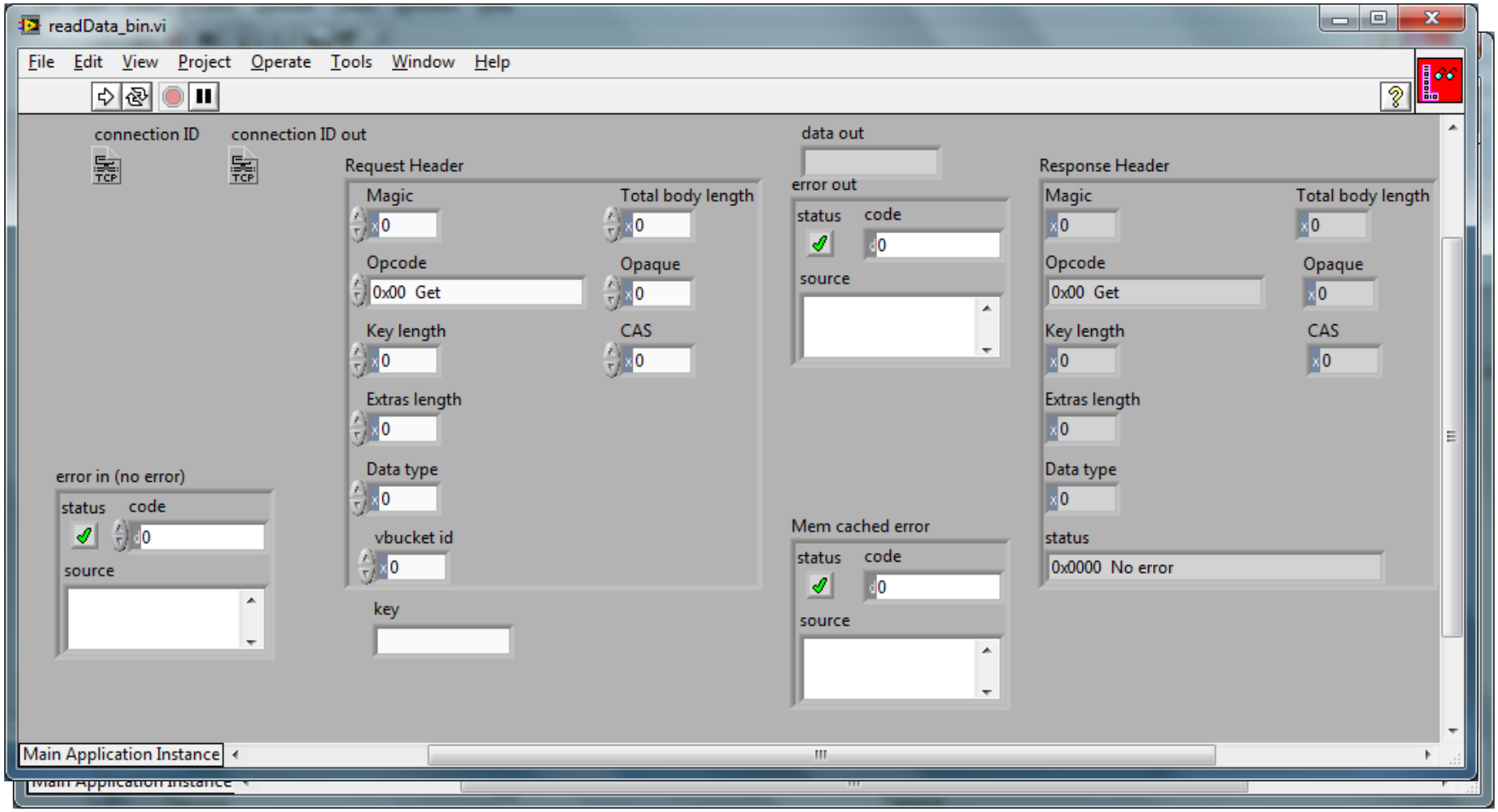
0x00 => Get
0x01 => Set
0x02 => Add
0x03 => Replace
0x04 => Delete
0x05 => Increment
0x06 => Decrement
0x07 => Quit

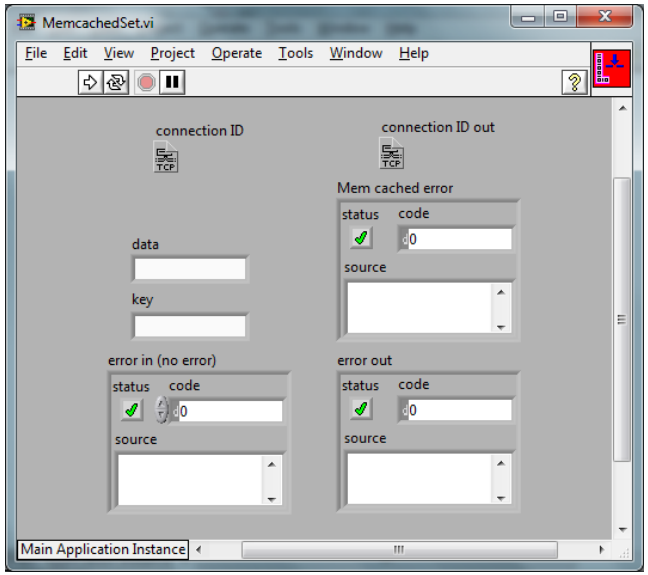
x08 => Flush
0x09 => GetQ
0x0A => No-op
0x0B => Version
0x0C => GetK
0x0D => GetKQ
0x0E => Append
0x0F => Prepend
0x11 => Stats

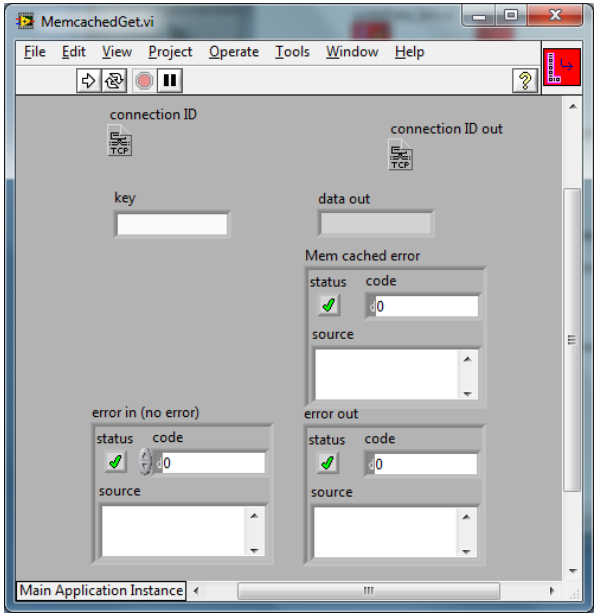
Write Data

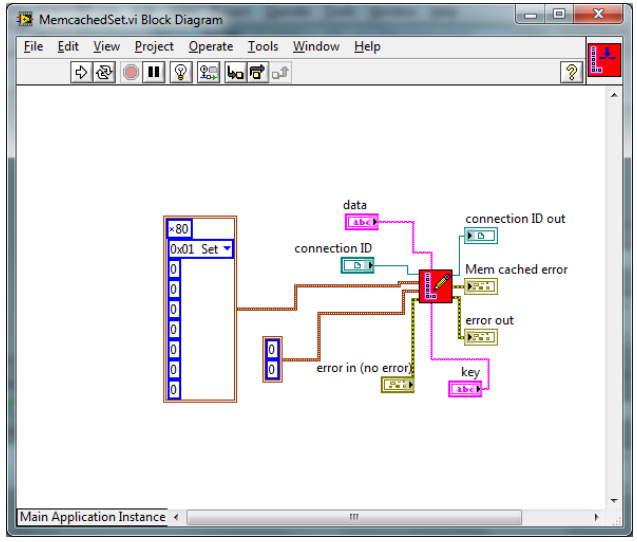

writeData_bin.vi
File Edit View Project Operate Tools Window Help
connection ID
connection ID out
data
Request Header
Magic
Total body length
Opcode
0x00 Get
Opaque
Key length
CAS
Extras length
Data type
vbucket id
key
Extras
Flags
Expiration
error out
status
code
source
Mem cached error
error in (no error)
Response Header
0x0000 No error
Main Application Instance

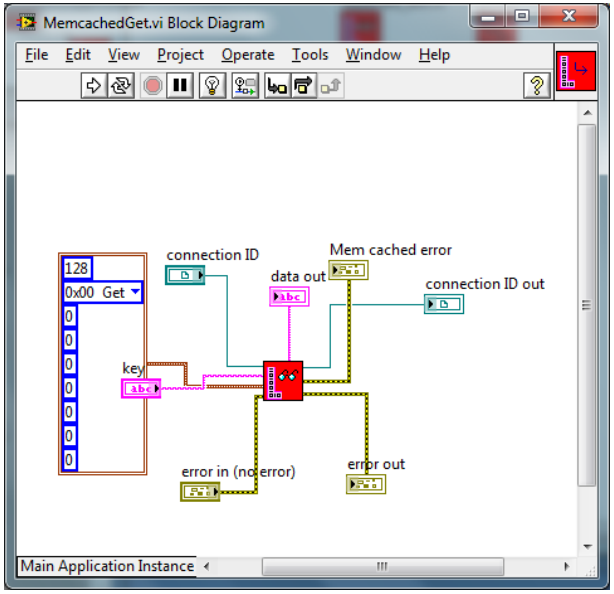
Read data

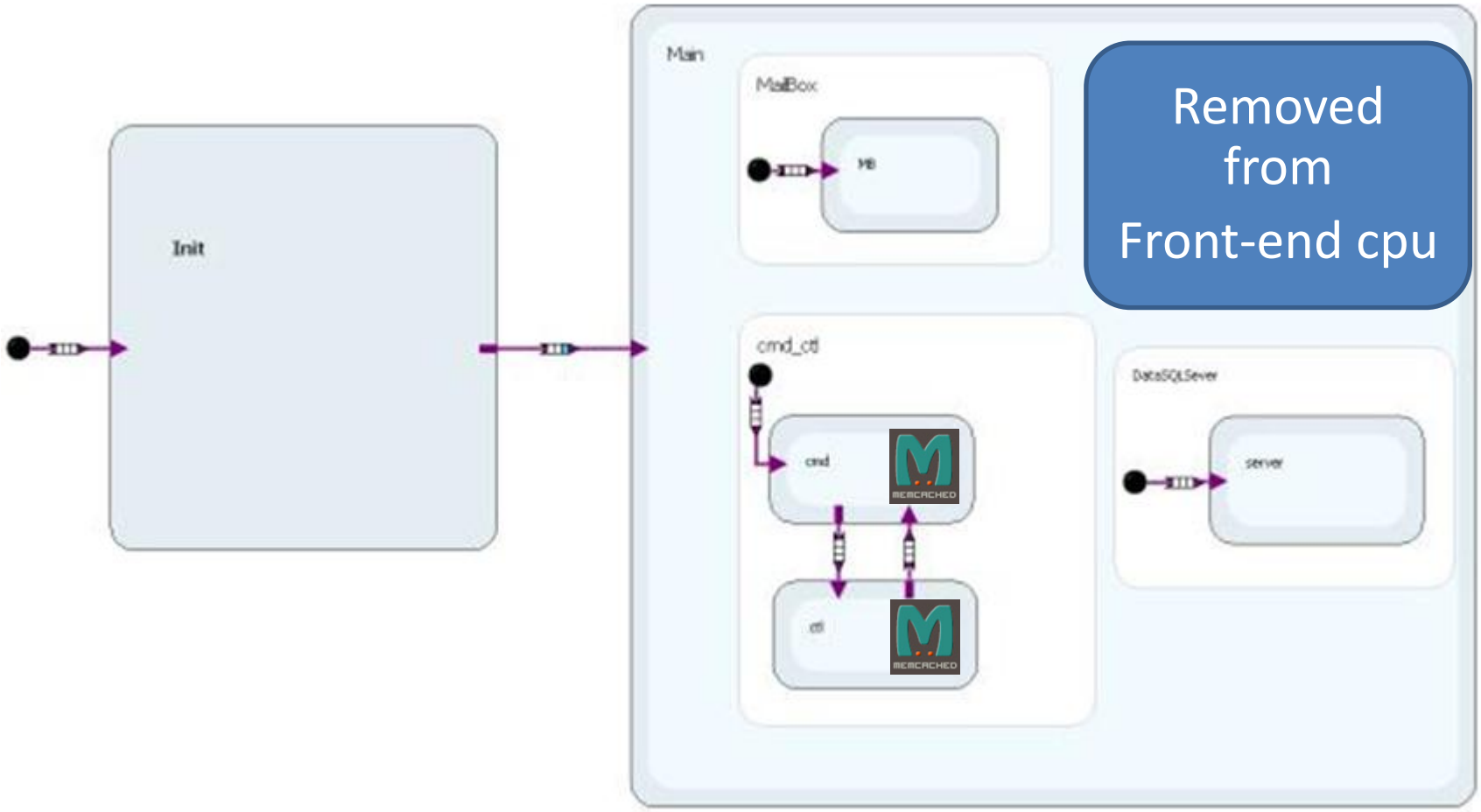

readData_bin.vi
File Edit View Project Operate Tools Window Help
connection ID
connection ID out
Request Header
Magic
Total body length
Opcode
0x00 Get
Opaque
Key length
CAS
Extras length
Data type
vbucket id
key
data out
error out
status
code
source
Response Header
Magic
Total body length
Opcode
0x00 Get
Opaque
Key length
CAS
Extras length
Data type
status
0x0000 No error
error in (no error)
status
code
source
Mem cached error
status
code
source
Main Application Instance


MemcachedSet.vi
File Edit View Project Operate Tools Window Help
connection ID
connection ID out
Mem cached error
status code
data
source
key
error in (no error)
error out
Main Application Instance

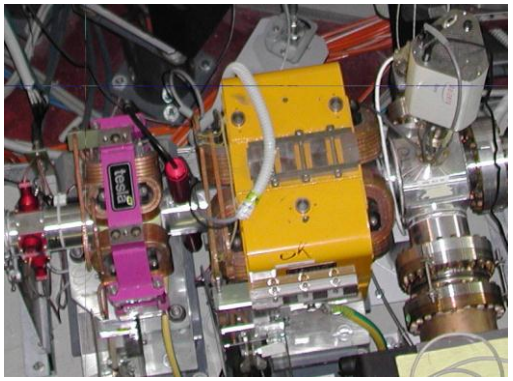

MemcachedGet.vi
File Edit View Project Operate Tools Window Help
connection ID
connection ID out
key
data out
Mem cached error
status code
source
error in (no error)
error out
Main Application Instance

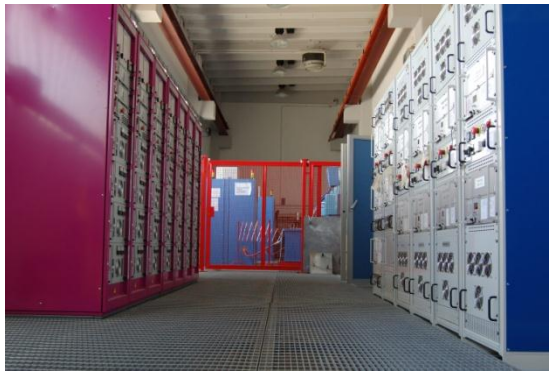

MemcachedSet.vi Block Diagram
File Edit View Project Operate Tools Window Help
data
connection ID out
connection ID
Mem cached error
error out
error in (no error)
key
0x01 Set
Main Application Instance


MemcachedGet.vi Block Diagram
File Edit View Project Operate Tools Window Help
connection ID
Mem cached error
data out
connection ID out
0x00 Get
key
error in (no error)
error out
Main Application Instance


Init
Main
MailBox
mb
cmd_ctl
cmd
ctl
MEMCACHED
MEMCACHED
DataSQLSever
server
Removed from Front-end cpu

Object abstraction

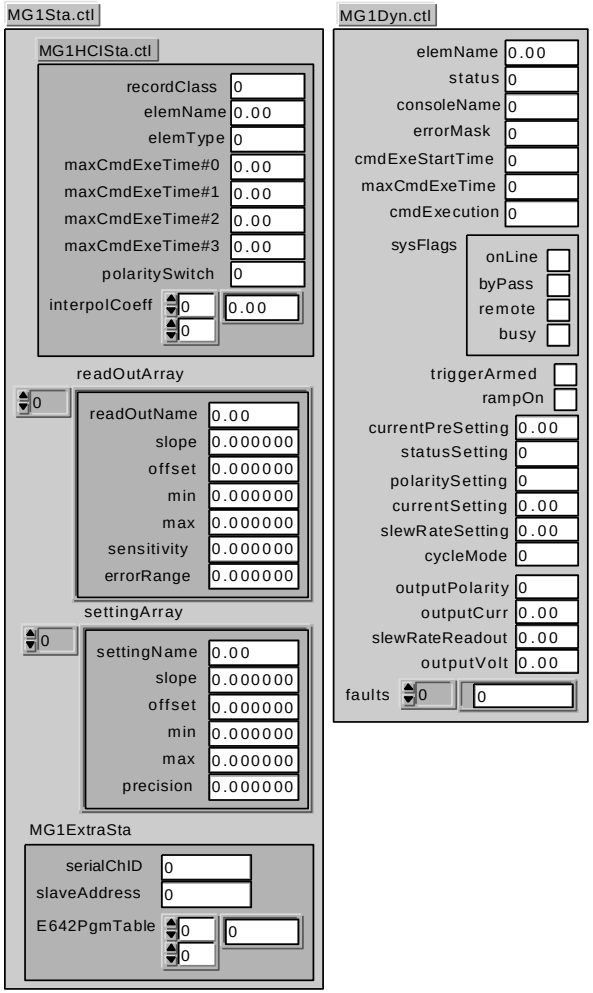


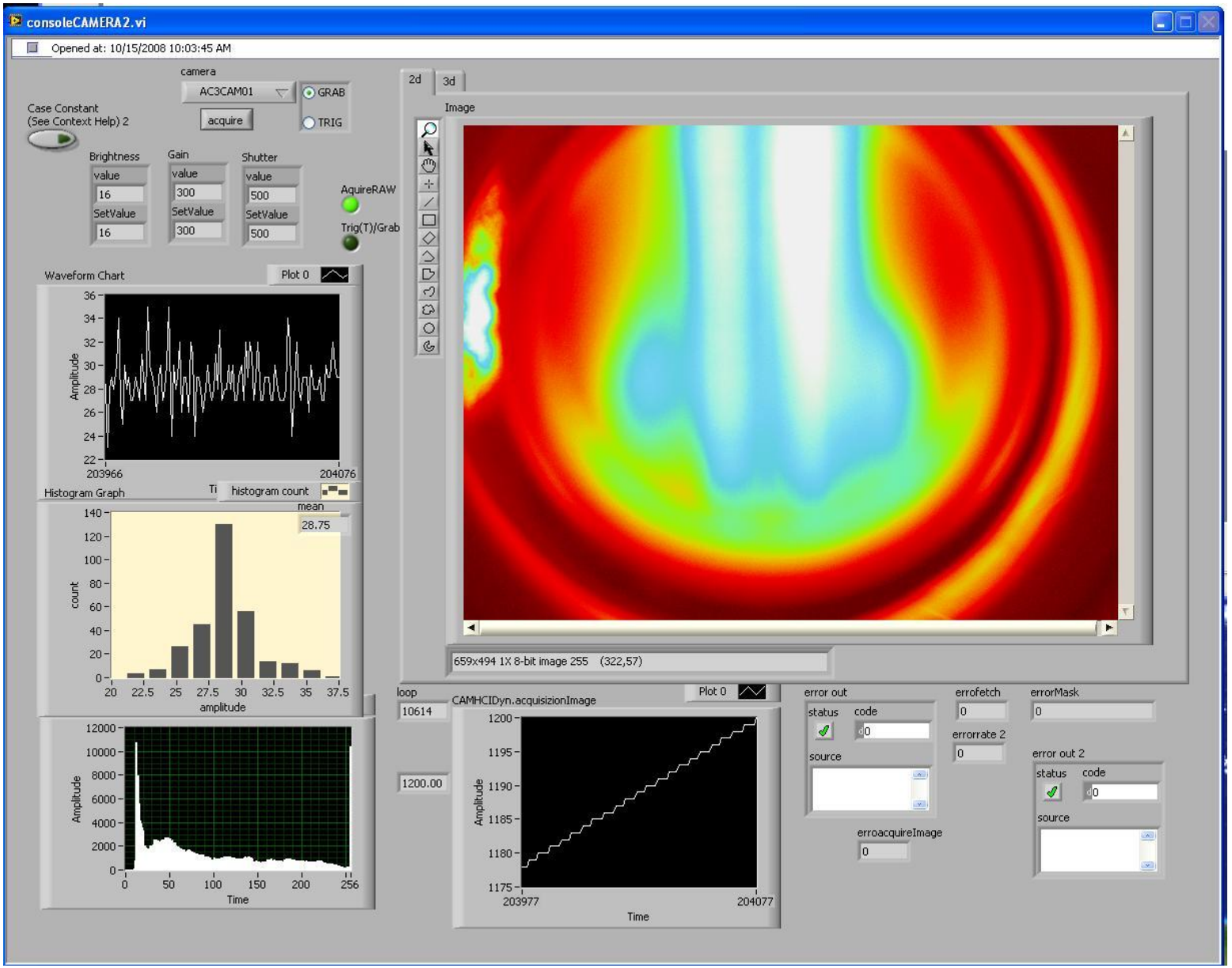


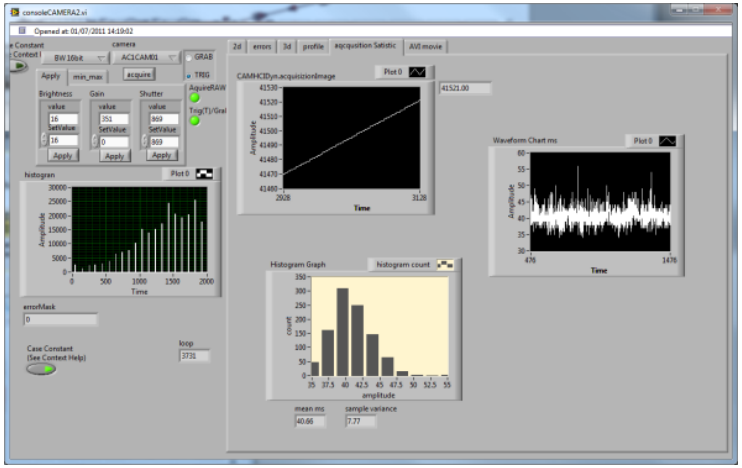
KEY definition
<element name>_DYN
<element name>_STA
<element name>_DAT

static variables (parameters)

dynamic variables




consoleCAMERA2.vi
Opened at: 10/15/2008 10:03:45 AM
camera
AC3CAM01
GRAB
TRIG
acquire
Case Constant
(See Context Help) 2
Brightness
value
16
SetValue
16
Gain
value
300
SetValue
300
Shutter
value
500
SetValue
500
AquireRAW
Trig(T)/Grab
Waveform Chart
Plot 0
Amplitude
203966
204076
Histogram Graph
histogram count
mean
28.75
count
amplitude
Amplitude
Time
2d
3d
Image
659x494 1X 8-bit image 255 (322,57)
loop
10614
1200.00
CAMHCIDyn.acquisizionImage
Plot 0
Amplitude
203977
204077
Time
error out
status
code
source
errofetch
errorMask
errorrate 2
error out 2
erroacquireImage



Hardware server

CPU	Intel Xeon 3 GHz
RAM	1 Gbyte
Ethernet	1 Gbit

Image statistic

Image 640x480 16 bit 40 ms sample variance 8 ms (614400 byte)

Image 640x480 8 bit 23 mssample variance 8 ms (307200 byte)

Don’t see variation from 1 until 4 consoles in acquisition

Conclusion

The use of cached data has demonstrated both in performance and in simplicity of realization.

Now the control system is completely past to use Memcached as RTDB.