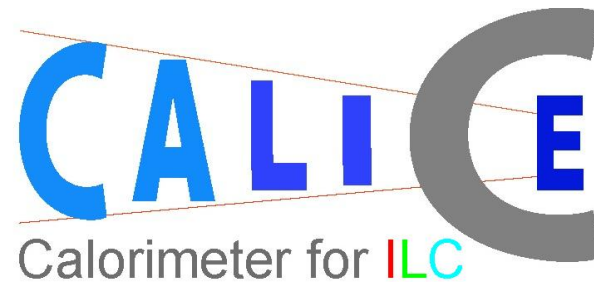




Status & development of the software for CALICE-DAQ



Tao Wu

On behalf of UK Collaboration

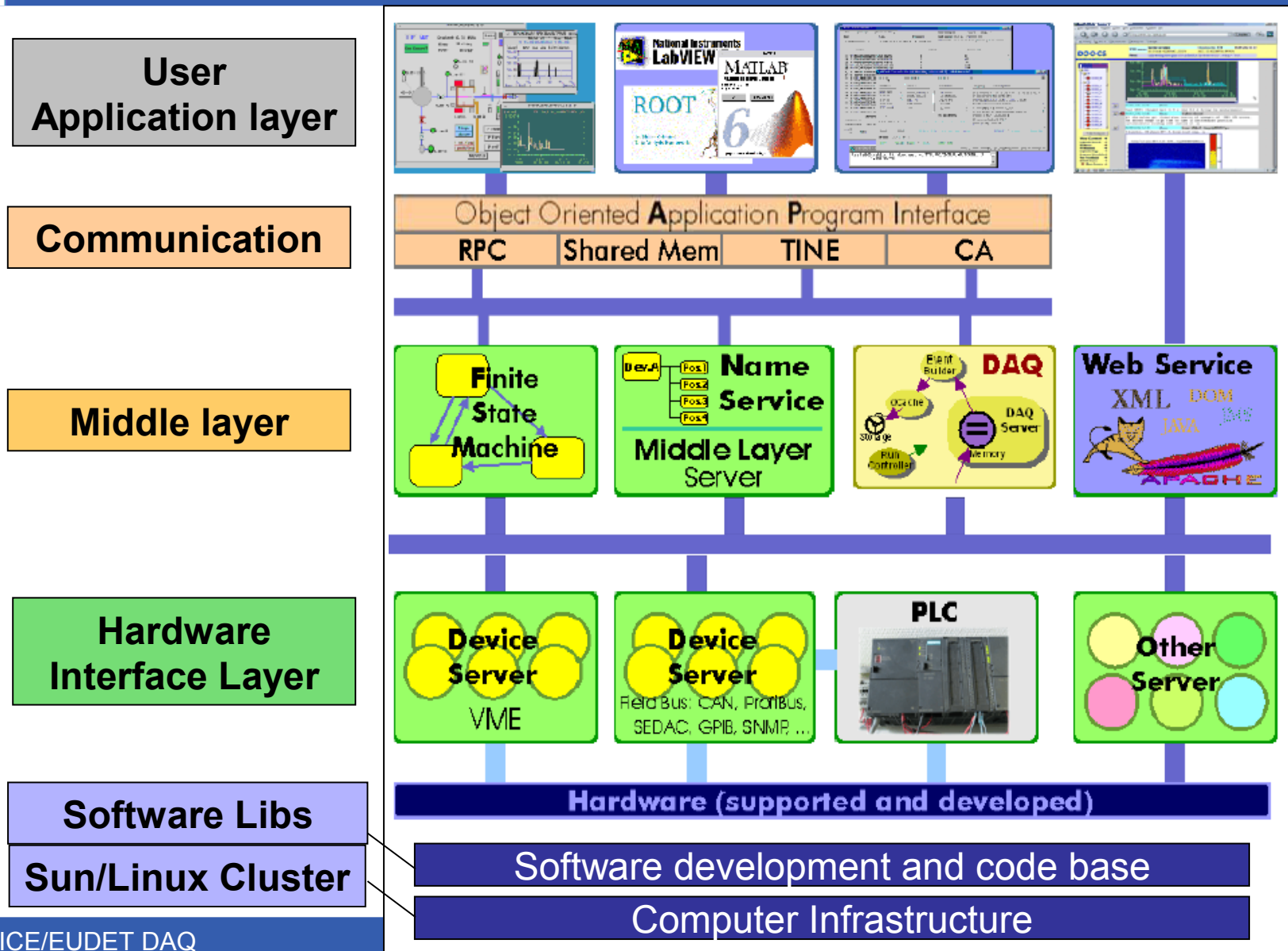


Outline

- Introduction of existing system;
- Development strategies;
- Task lists and its development status;
- Summary & outlook



Software Basis: DOOCs





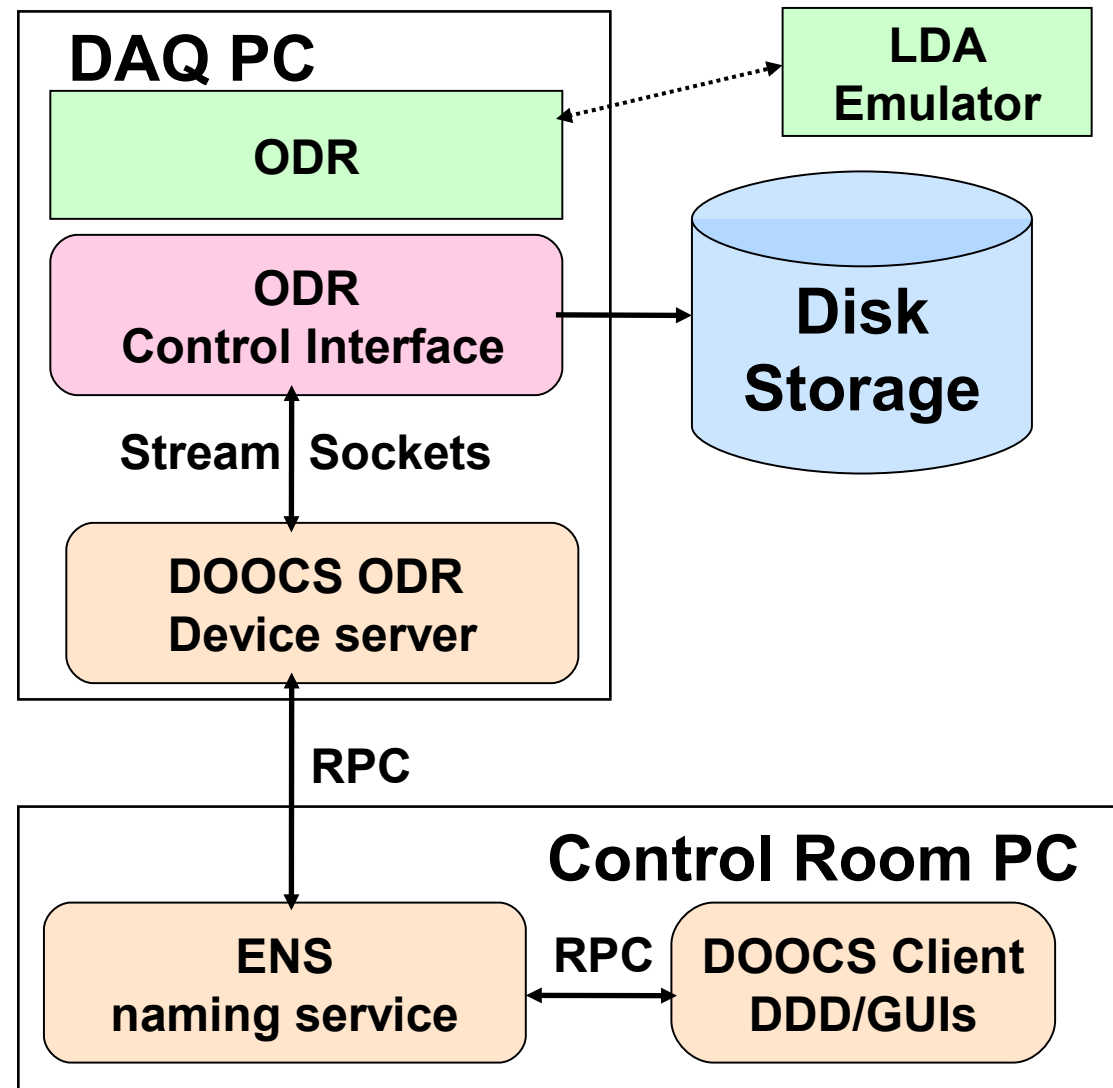
Device server & ENS

- Modeling hardware card via device server in OO
 - Existing: ODR & LDA device servers;
- **Equipment Name Server (ENS):**
 - Facility(**F**) / Device(**D**) / Location(**L**) / Property (**P**)
 - ▶ Detector type, components, instances & properties;
 - e.g. *CALICE.ECAL/ODR/ODR1/Status*
 - ▶ F: CALICE.ECAL, CALICE.AHCAL, CALICE.DHCAL
 - ▶ D: ODR, LDA, DIF, ASICs;
 - ▶ L: ODR1,ODR2,ODRX; LDA1,LDA2,LDAX; DIF1,DIF2,DIFX;
 - ▶ Property: X X X customize.
- To classify all properties and functionalities of each device for our DAQ system is ongoing!
- An interface talking to ODR has been built;



Existing test system

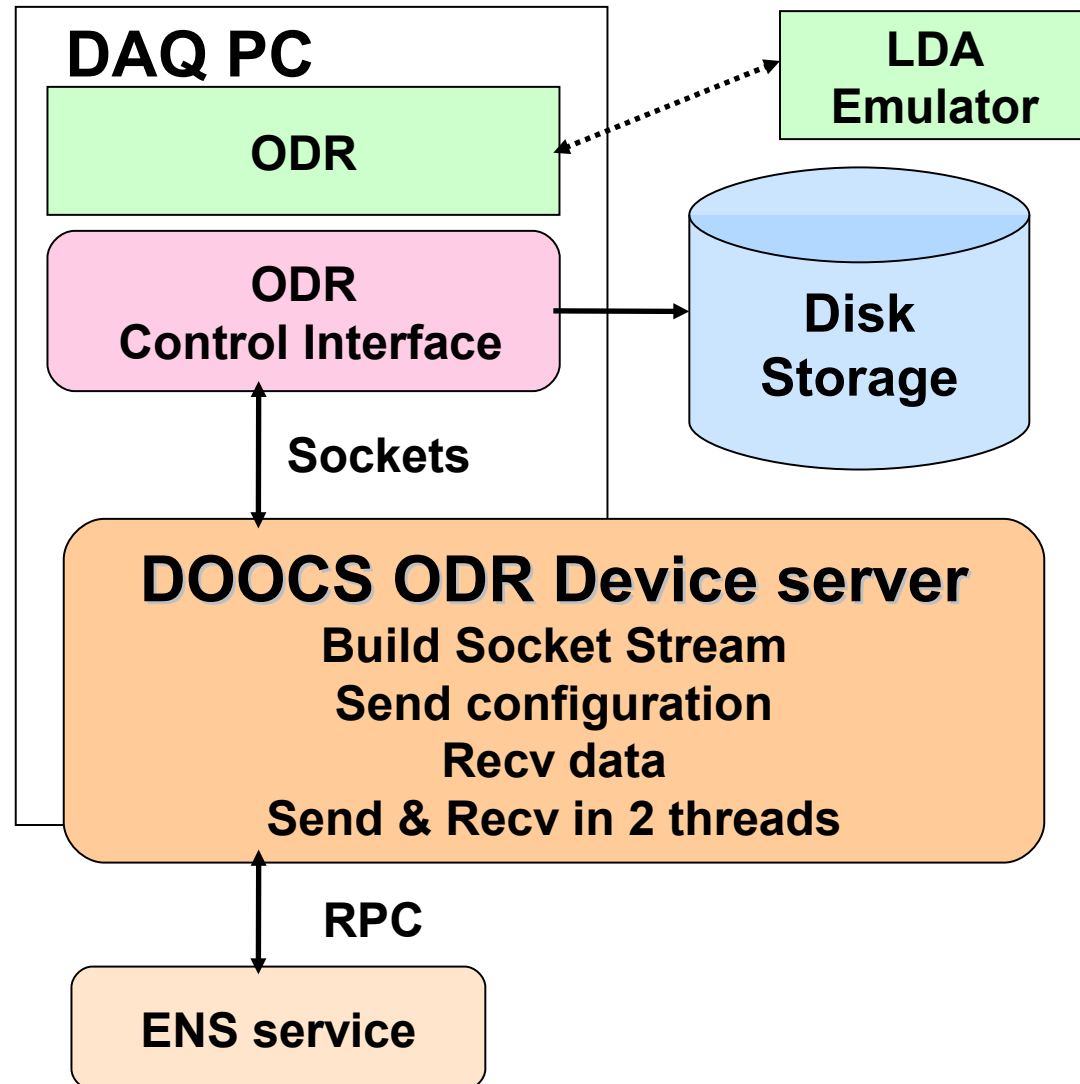
- Communication between different parts of DOOCS **Server/Client** by **RPCs**
- Configuration data & files are used when the system starts and runs:
 - ODR CTL s/w
 - ODR device server
 - ENS server





ODR device server

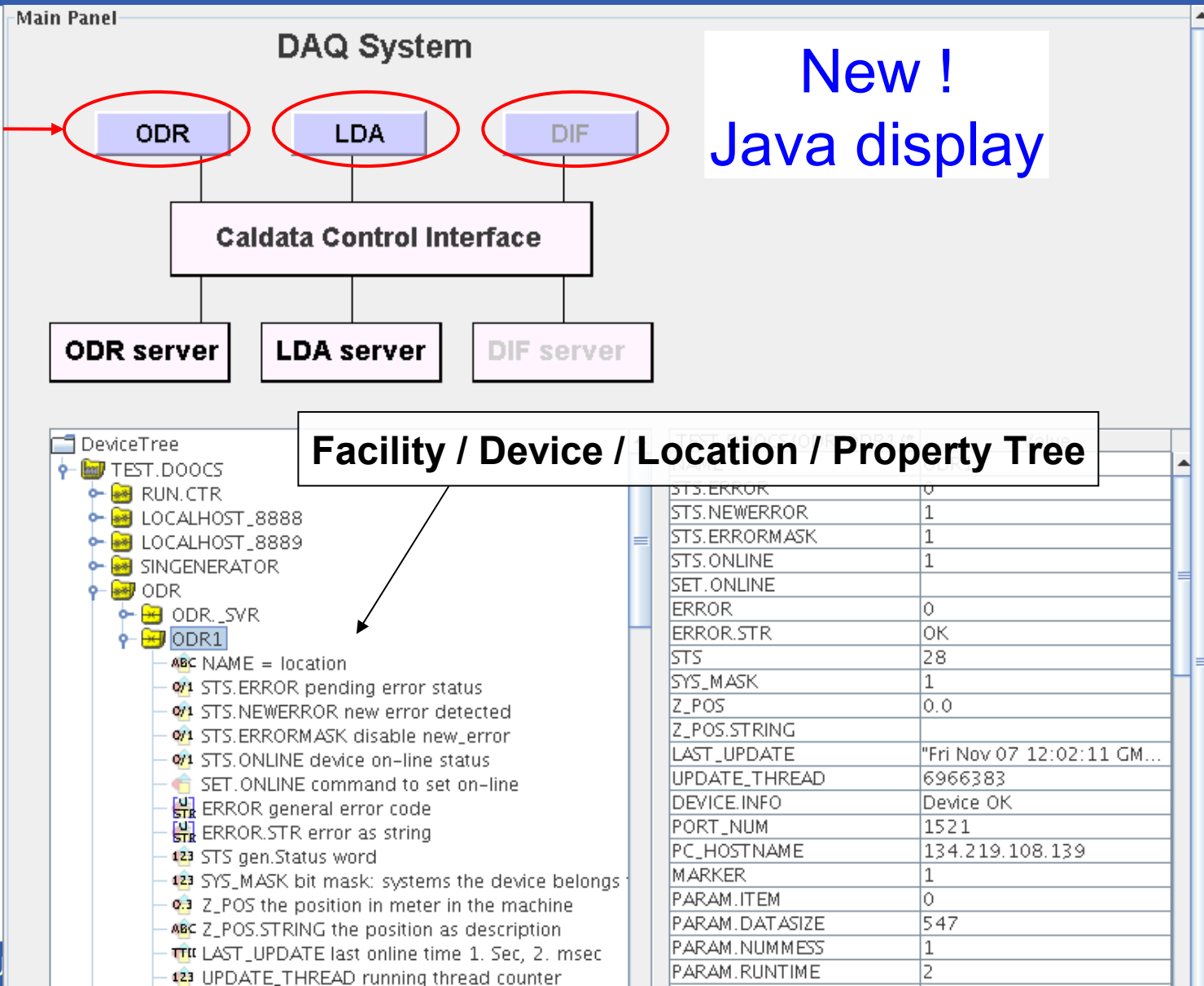
- **One device server** can serve **many instances** all connected via different ports and hostnames
- Using 2 **threads**: one for receiving, one for sending on the socket
- ODR control s/w
 - User-to-HD interface
 - Communication & I/O
- **Sockets format** is chosen to build an interface to the ODR and the LDA





Client GUI interface

Buttons



**New !
Java display**



ODR Client interface

ODR

ODR parameters

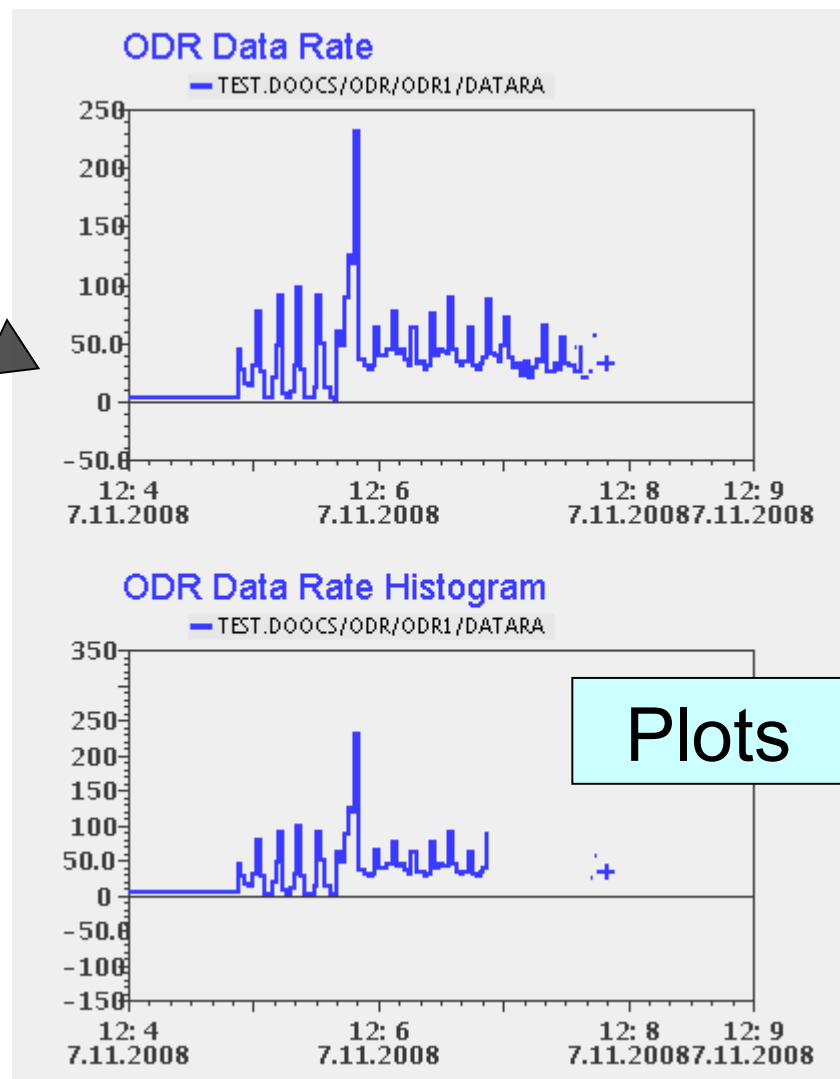
Parameter	Value	Action	Status
data size	547	send	●
no of messages	1	send	●
run time (s)	2	send	●
dump to screen	0	send	●
grouping	10000	send	●
no of IO Threads	1	send	●
dump data to disk 0/1	1	send	●
dump data size	10000000	send	●
word to extract	6	send	●
DG:0 Network:1	0	send	●
active channels 0/1	0	send	●
No of DMA	900	send	●
start data write	820	send	●
event grouping	0	send	●
DMA debug	0	send	●
text next message	0	send	●
statistics update freq	0	send	●
run	0	send	●
quit	0	send	●
empty	1	send	●

ODR Commands

- Start
- Get statistics
- Get parameters
- Stop
- ODR plots
- Quit All servers

TEST.DOOCS/ODR/0...

TEST.DOOCS/ODR/0...	Value
PARAM.ITEM	0
PARAM.DATASIZE	547
PARAM.NUMMESS	1
PARAM.RUNTIME	2
PARAM.DUMPSCR	0
PARAM.GROUP	10000
PARAM.IOTHREADS	1
PARAM.DUMPTOHD	1
PARAM.DUMPSIZE	10000000
PARAM.WORDEXTRACT	6
PARAM.DGNET	0
PARAM.ACTCHAN	0





LDA Client interface

LDA

LDA parameters

ODR MAC address

00:A0:CC:73:44:DF

emulator MAC address

00:E0:81:27:13:FD

LDA data size

512

LDA data type

3

LDA packet delay

11

TEST.DOOC5/ODR/OD...	Value
LDA.ODRMACDR	00:A0:CC:73:44:DF
LDA.EMUMACDR	00:E0:81:27:13:FD
LDA.DATASIZE	512
LDA.DATATYPE	3
LDA.PCKDELAY	11

LDA Commands

Start

Send parameters

Get parameters

Stop

Terminate

Commands are testing with new LDA emulator:

LDASTART
LDAConfig
LDASendConfig
LDASTOP
LDATERMINATE

New !



Development strategies

- Strategies before full chain available:
 - use an LDA emulator instead;
 - Continue to develop the rest parts of software for DAQ system in parallel;
 - Get each part ready as far as we can;
 - Detect new & fix pending problems and investigate better designs meantime.

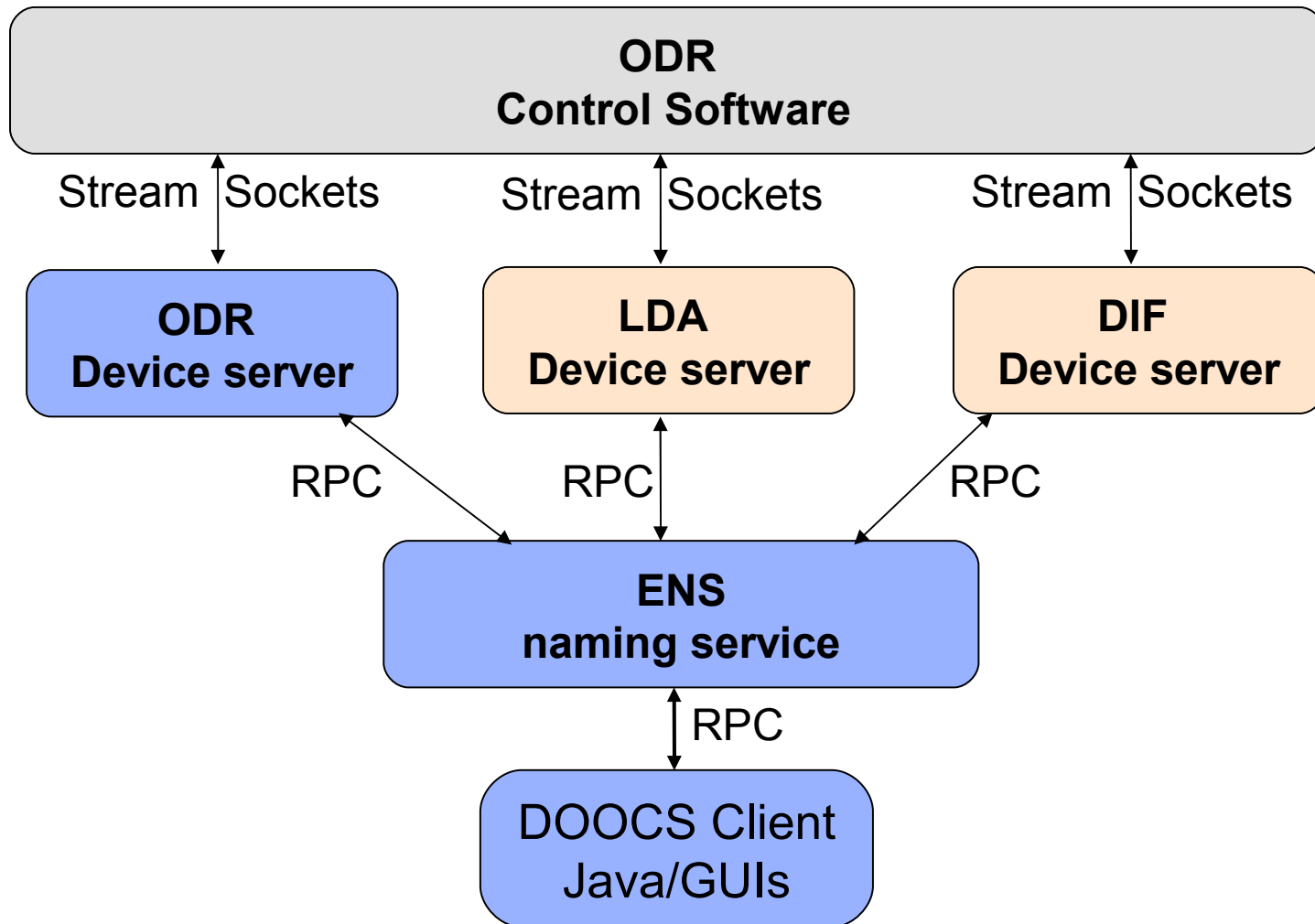


Task lists

Task	Status
ODR-LDA device server DIF, ASIC device server	Done Not yet
State Machine	Not yet
Data handling	Started
Error/Alarming handling	Started
Clock device server	investigating



Extendibility & integration of test system





Aim of the database

- Device database:
 - Connections between devices (ODR-LDA, LDA-DIF, DIF-DIF)
 - Starting parameters (e.g. port number, hostname)
- File database:
 - Host and path of file storage
 - Information about stored files
- Configuration database:
 - Static and volatile properties of all devices to be loaded
- All information related to make a RUN



Current status of database

- Choice of database: MySQL;
- Entity diagrams has been made;
- Use cases has been made and evaluated;
- database is populated with mockup data;
- C++ connector wrapped by MySQL++ to be built in DOOCS framework.

Optional solution !

Error handling



DOOCS Error Monitor (Version 5.1.39) wutao@larch.pp.rhul.ac.uk

List of Subsystems :

Facility	Object
TTF	WATCHDOG
TEST.DOOCS	ODR
TEST.DOOCS	SINGENERATO

Scan Time [sec] : 90 [Slider] Stop

New Errors Pending Errors Error Masks All Devices

Errors in Subsystem :

Location	New	Pending	Mask	Code	Time	Date	Error
TESLAHB2._SVR	0	0	0	0	17:10.43	24. Feb. 2004	OK
SIN	1	1	0	6	15:17.39	4. Dec. 2008	Device OFFLINE
COS	0	0	0	0	11:31.44	1. Oct. 2007	OK
TAN	0	0	0	0	19:23.45	13. Jun. 2003	OK

Functions on selected items :

Acknowledge Enable Disable Mask Choice: local mask

complete list: Log Files : Print History

Sad new error

Link to server is broken

Error: device offline

Exercising

- What type of error
- Its severity level

Some error types are defined.



ML based Alarm and Info Server Architecture

The architecture consists of three main components:

- GUI (Graphical User Interface):** Displays a table of device information with columns: Dev., Time, Sev., and Msg. It also shows a tree view of device locations (Fac., Dev., Loc., Prop.).
- Alarm and Info server:** Contains a **Central DOM Tree**, **XML backup**, **XML device history**, and a **Collect thread**. It also includes **Apache Tomcat Servlets** and a **JMS Server**, both using **JNI** for communication.
- DOOCS device server:** Manages device I/O and status reporting. It includes a **Queue** for **XML via RPC** and handles **Info**, **error**, and **status** messages.

Data Flow:

- XML Req / resp:** Transfers data between the GUI and the Alarm and Info server.
- JMS Event & session:** Transfers data between the GUI and the JMS Server.
- jDOOCS (opt.):** Transfers data between the GUI and the DOOCS device server.
- RPC call for device info:** Transfers data between the DOOCS device server and the Alarm and Info server.
- Collect thread:** Manages the flow of data from the DOOCS device server to the Central DOM Tree.

Central DOM Tree: A tree structure representing the device hierarchy, used for organizing and storing device information.

XML backup: A mechanism for saving device information in XML format.

XML device history: A mechanism for tracking device status changes over time.

Collect thread: A thread responsible for collecting data from the DOOCS device server and updating the Central DOM Tree.

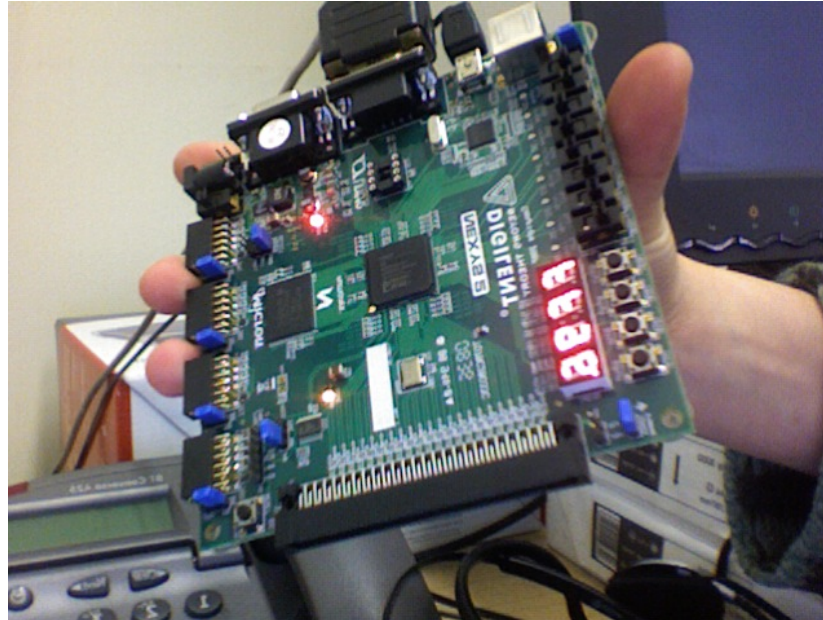
Queue: A data structure used for managing the flow of XML data via RPC.

Info, error, status: Messages exchanged between the DOOCS device server and the Alarm and Info server.

Device I/O: The physical interface for connecting the DOOCS device server to the devices.



Clock device server



- Simple hardware system to test with;
- No clock firmware to speak to yet, but
- Software sets registers
- Software handles device like a file
- Not yet integrate in a DOOCS device server

Exercising

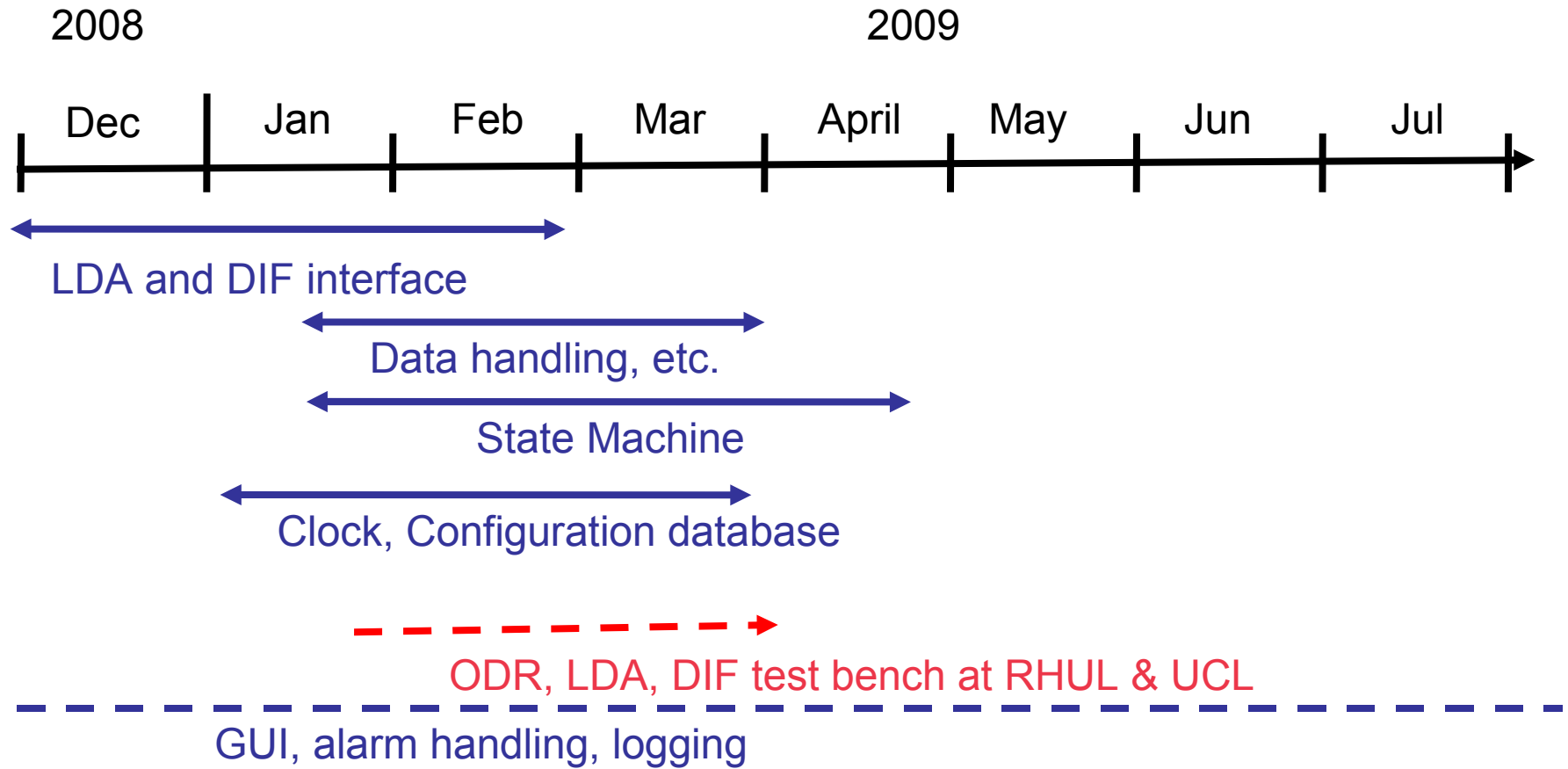


Some essentials

- The components are not integrated in software,
 - LDA / DIF / ASICs
 - Full chain will be available ~Jan/2009.
- No event building & LCIO converting
 - Data are saved to local disk in raw format;
 - Just provide interfaces in the framework;
 - More flexible way to leave them free for sub-detector groups to develop.



Timeline estimation





Summary & Outlook

- Software has integrated ODR into DAQ system;
- Software has successfully managed communications and control tests;
- Some further progress ongoing:
 - Device server for other DAQ components;
 - Data handling
 - Error/Alarm handling
 - Clock device server
 - Full chain of DAQ components will be available Jan/2009.



Backup



ODR update

- Tests are ongoing trying to emulate bunch-train data which is saved in single file to disk.

- A list of alarms is being defined:

- 1) ODR not reachable.
- 2) Internal ODR-receiver alarms:
 - 2a) ODR card not detected.
 - 2b) Cannot initialise ODR card.
 - 2c) Cannot initialise ODR network interface.
 - 2d) Cannot allocate memory.
 - 2e) Cannot start "receive" thread.
 - 2f) Cannot start IO thread.
 - 2g) Cannot save received data/error saving data.
 - 2h) Wrong virtual address of the data buffer.
 - 2i) Data error detected/DMA data corrupted.
 - 2j) Wrong data channel ID.
 - 2k) Unknown data page number - FIFO data corrupted.
 - 2l) Error stopping "receive" thread.
 - 2m) Error stopping IO thread.
- 3) DOOCS own, ODR related alarms:
 - 3a) No response from ODR.
 - 3b) Unknown response from ODR.
 - 3c) ODR response corrupted.
- 4) LDA related alarms:
 - 4a) No response from LDA.
 - 4b) Unknown response from LDA.
 - 4c) LDA response corrupted.

Andrzej Misiejuk RHUL