Minutes DAQ software discussion - 16/10/08

Priorities
- to be ready before testbench is ready-

- LDA - ODR - DIF device server disentanglement (Tao, Barry, Matt, Andrzej) => needs to be ready before starting state machine and error/alarm handling
- State machine (Tao, Andrzej, Barry)
- Data handling (Valeria)
- Error/alarm handling (Tao+Andrzej: implementing alarms and logs, Barry+Matt: input what needs to be handled, Valeria: DOOCS infrastructure)

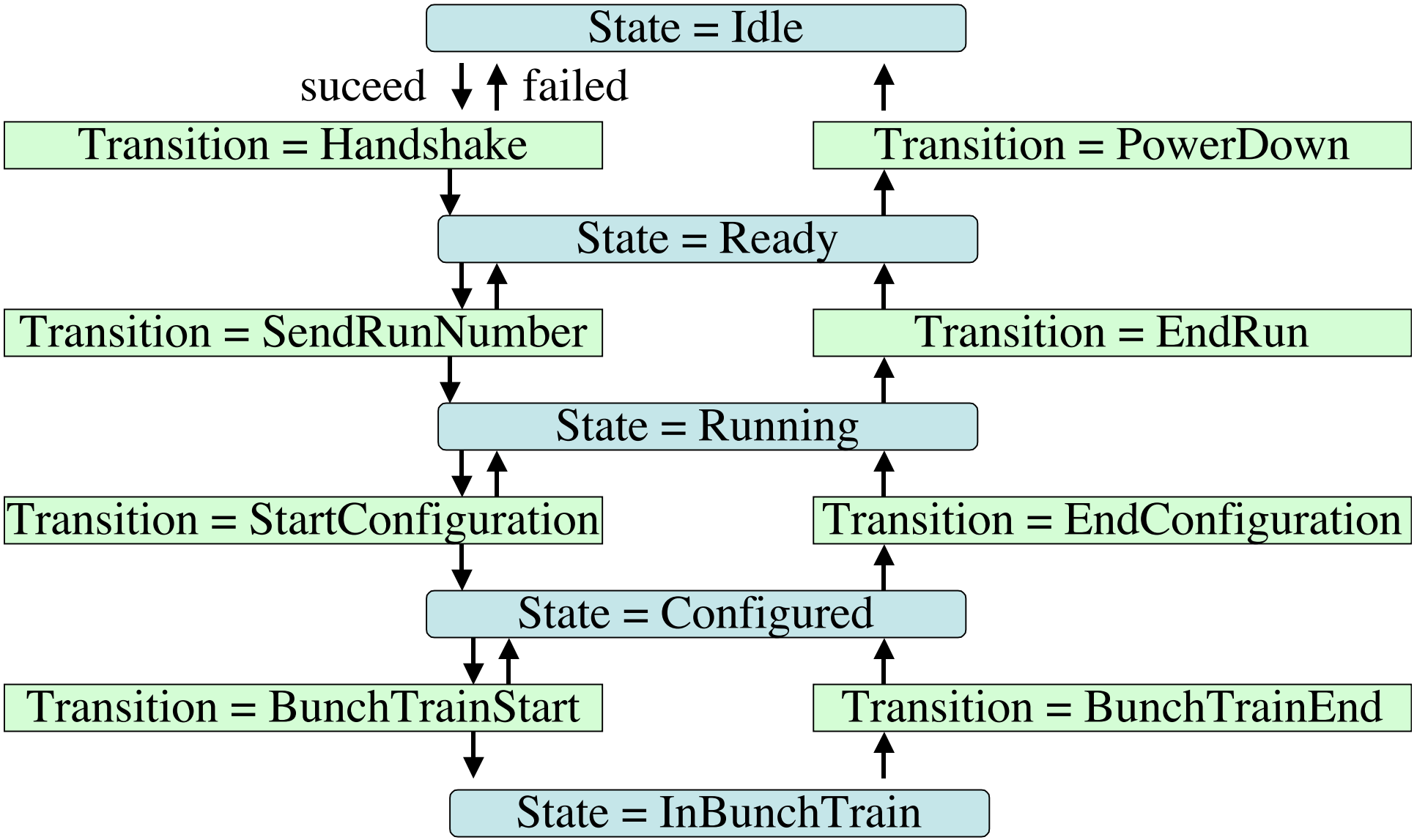
To investigate:

- Clock device server (Valeria, Matt)

State machine

- What we need to do to ramp up for data taking:
- Send hardware handshake to check connections (could also be done by getting conf.)
- Let file database know about run number
- Tell ODR which run number we have right now to put it into the file name
- Send conf.
- Receive automatic acknowledgement or send getConfiguration command

State Analysis



State machine

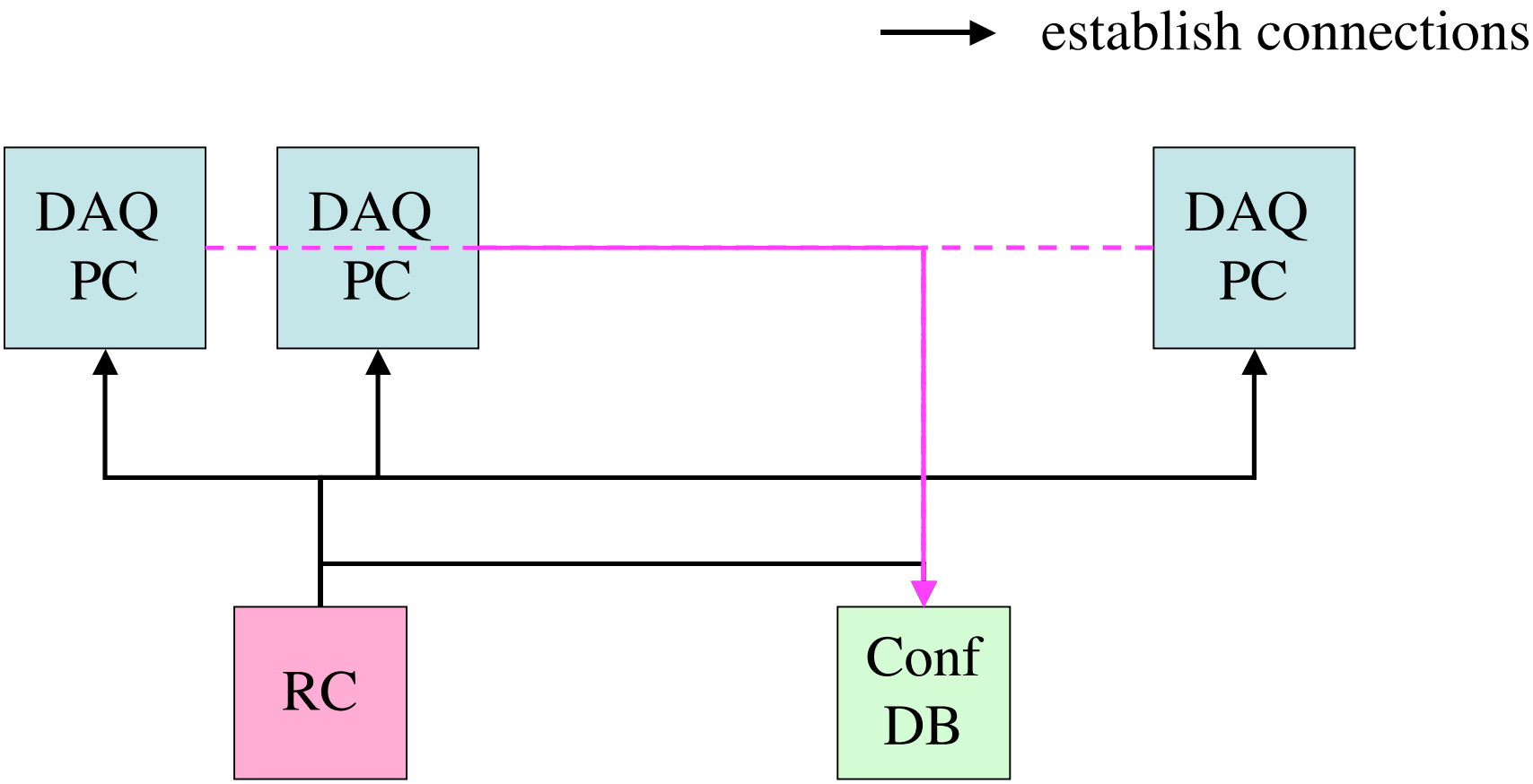
- Either take state machine on previous slide which was how we started out a year ago
- Or skip the first steps and put all functionality into the StartConfiguration step

⇒Comments from Barry: map functionality in steps of state machine

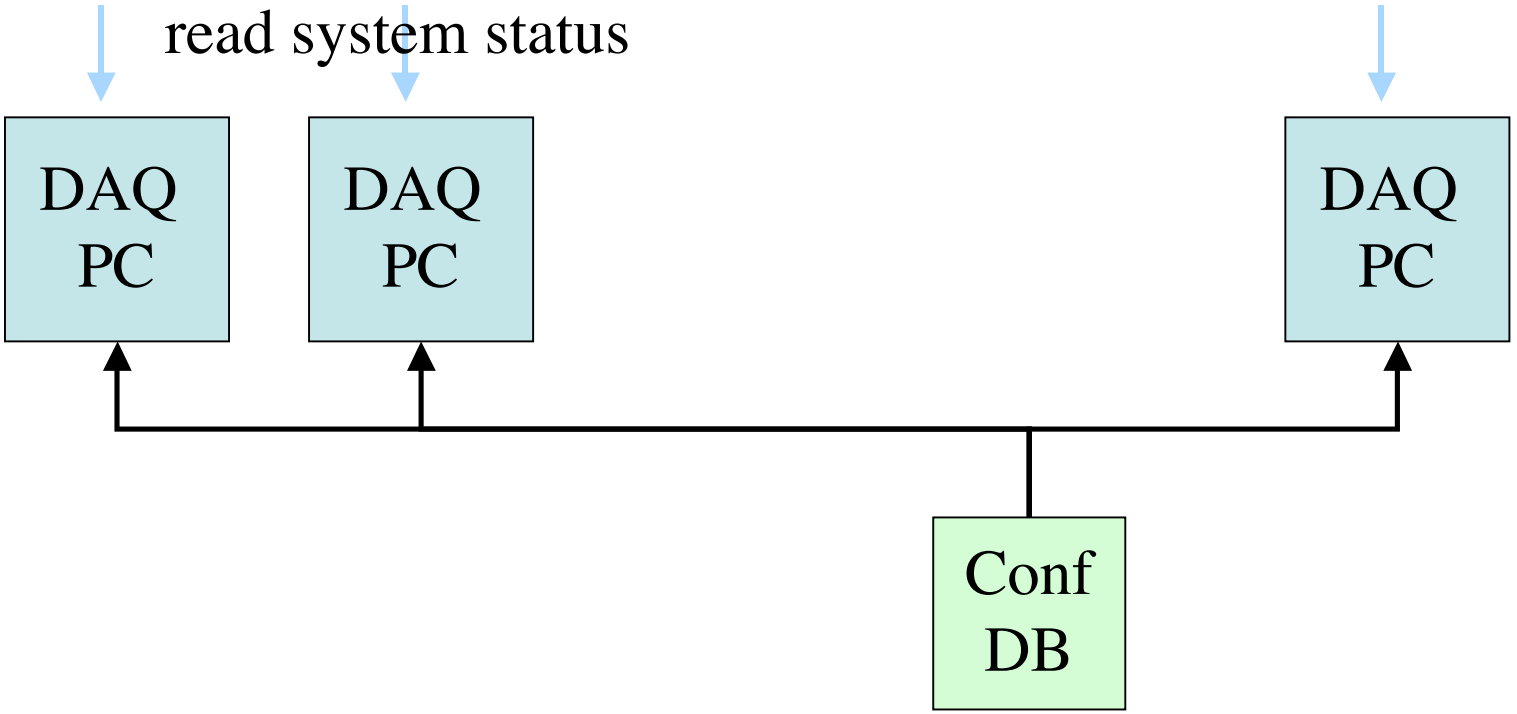
⇒ problem: if only one step, need to be careful to give any shifter more information in case something goes wrong

⇒Veronique: be careful with timeouts, the system has to start within 2 minutes

Transition: Handshake

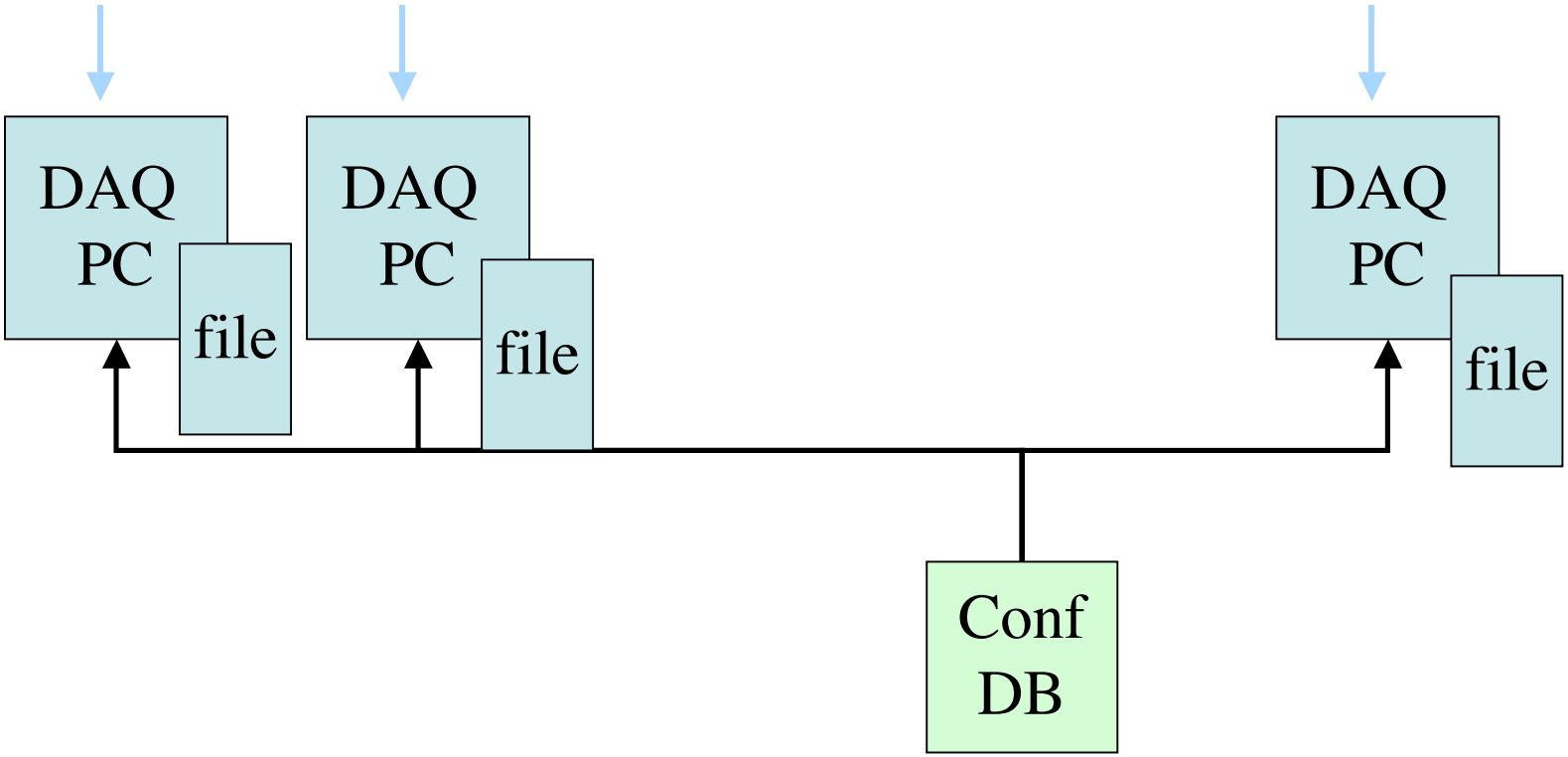


Transition: StartRun



Send run number to ODR software,
Make new run number plus unique in file database
(filename = [run_number + unique identifier])
and fill in configurations

Transition: StartConfiguration



Extract conf files for all device servers from db,
Recheck that configuration has been received

Alarm handling

Andrzej:

- will provide a list of possible problems of the ODR
- How to check them
- How to mitigate them

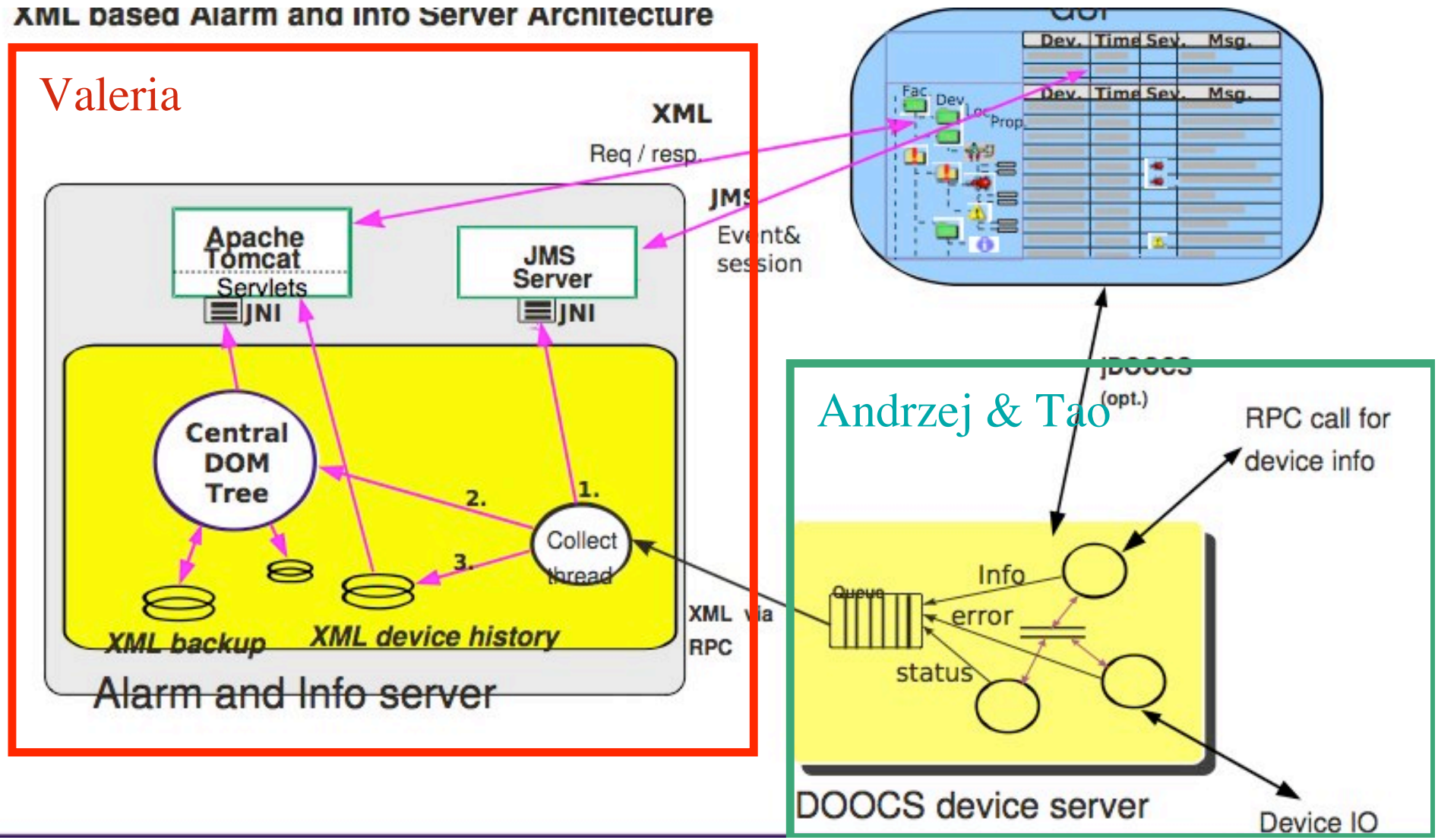
Tao:

- Probably best person to implement this list, because he can easiest communicate with Andrzej

Valeria:

- Can implement infrastructure for the Alarm handling

Alarm handling

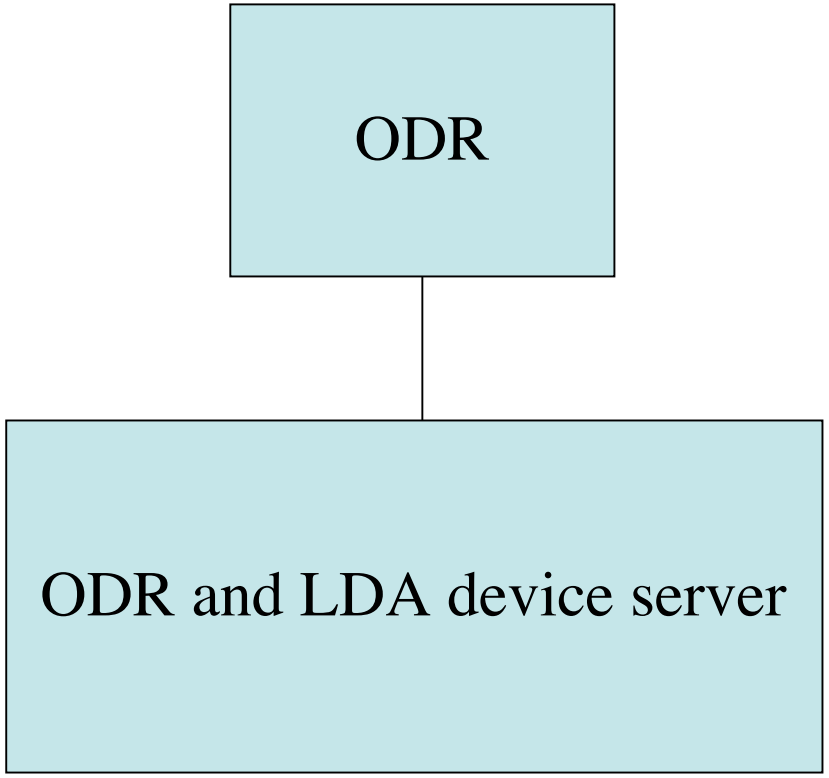


LDA emulator and possible DIF emulator

- Problem: how realistic is LDA emulator (Andrzej needs to check Mark Kelly's webpage for his definitions and probably communicate with Mark)
- DIF emulator: not much information at the moment, Veronique proposes to simply implement it like the LDA emulator, so that a whole system can be simulated (=> also build a DIF device server)
- Especially important, check that system with several LDAs and DIFs works

ODR, LDA and DIF device server - scenario I -

- Device servers need to be disentangled
- Right now the ODR and LDA device server are implement as one (see right hand side)



ODR, LDA and DIF device server - scenario II -

Easiest for DOOCS

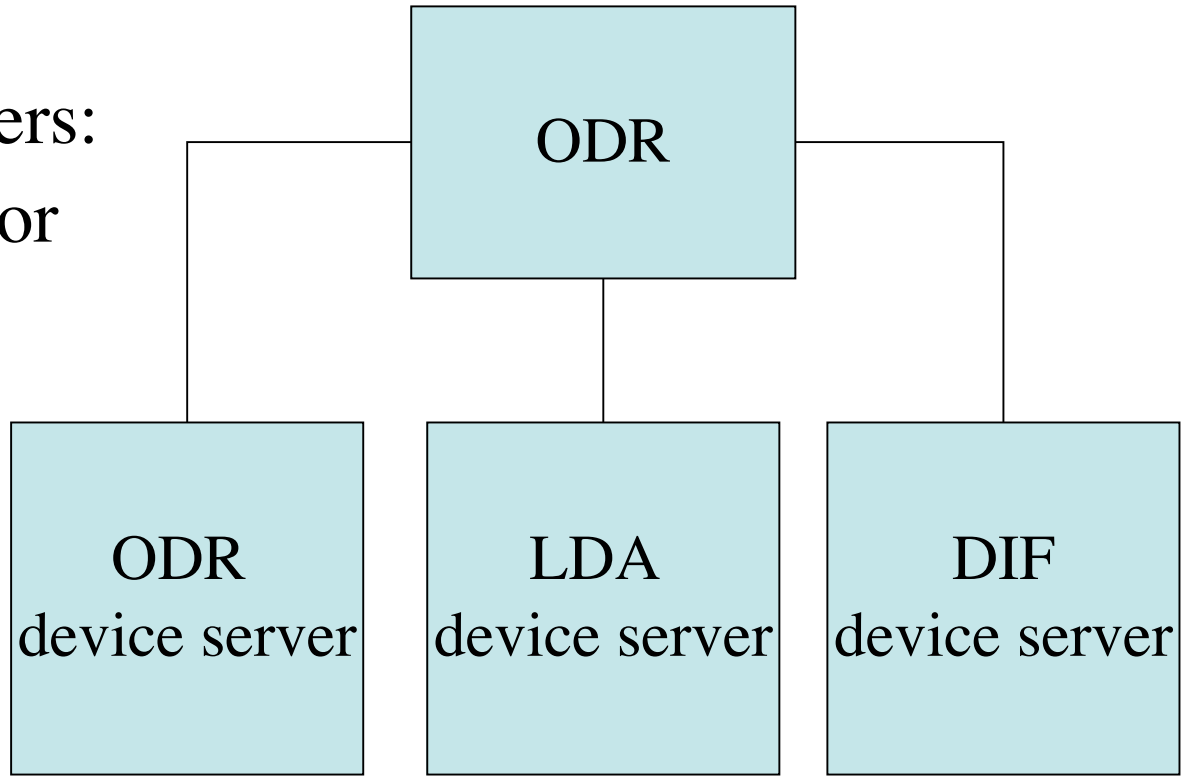
device server developers:

- A different socket for each device server instance
- 1 ODR socket,

4 LDA sockets,

32 DIF sockets

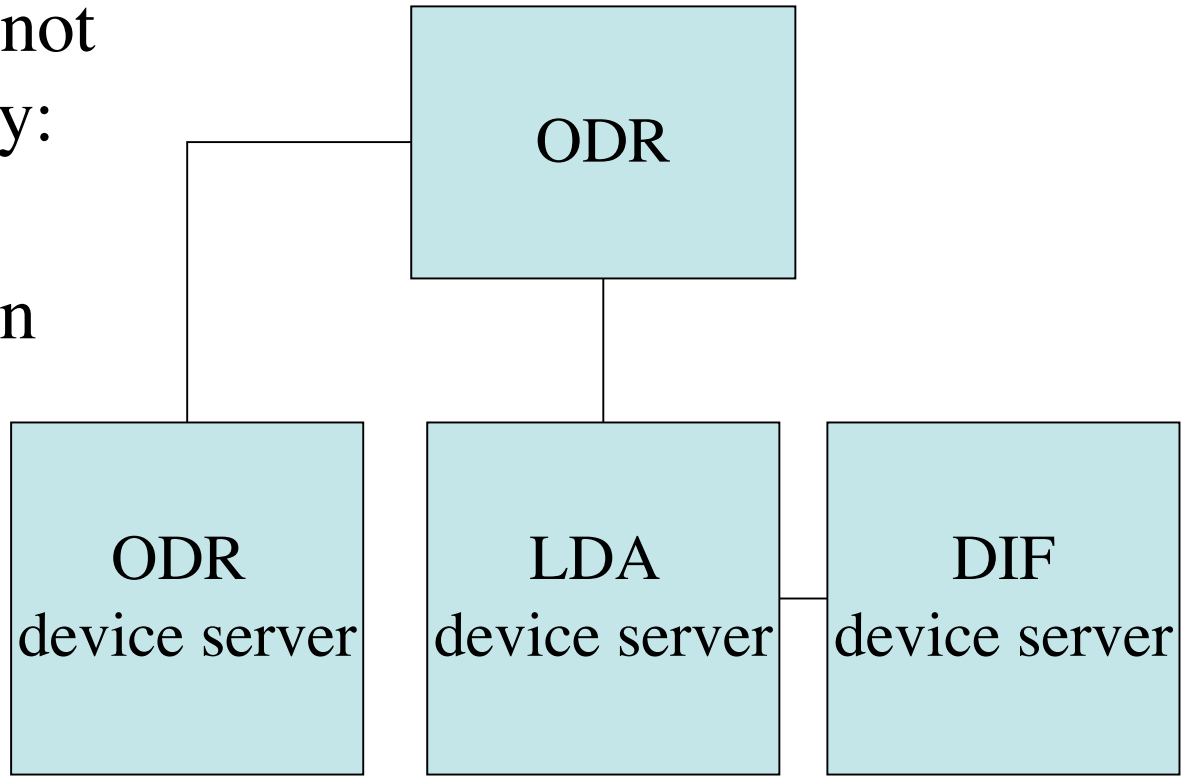
⇒Favoured scenario



ODR, LDA and DIF device server - scenario III -

Other (at the moment not favoured) possibility:

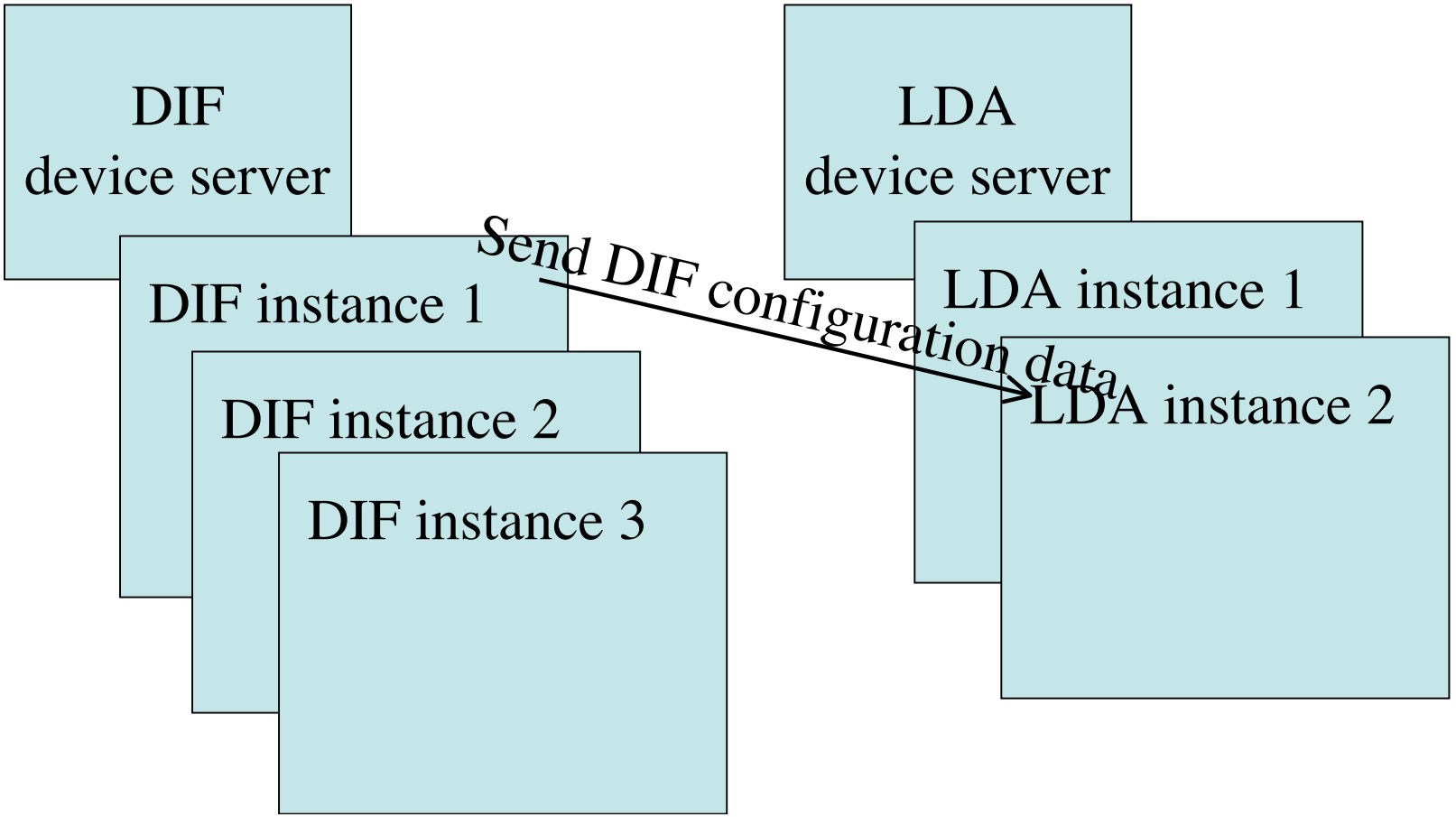
- Implement dependency between device servers



ODR, LDA and DIF device server - scenario III -

- needs communication between DOOCS device servers
- there are always several instances (objects) of the device server (equal to the number of real existing devices)
- Communication needs to be done with RPCs (like the ENS naming service)

ODR, LDA and DIF device server - scenario III -



ODR, LDA and DIF device server - hardware, firmware, driver solutions -

How to implement scenario II on the ODR driver, ODR firmware, hardware:

- Firmware: can easily distinguish between upstream (LDA/DIF data) and ODR data

⇒ Firmware needs to be tweaked a little for this

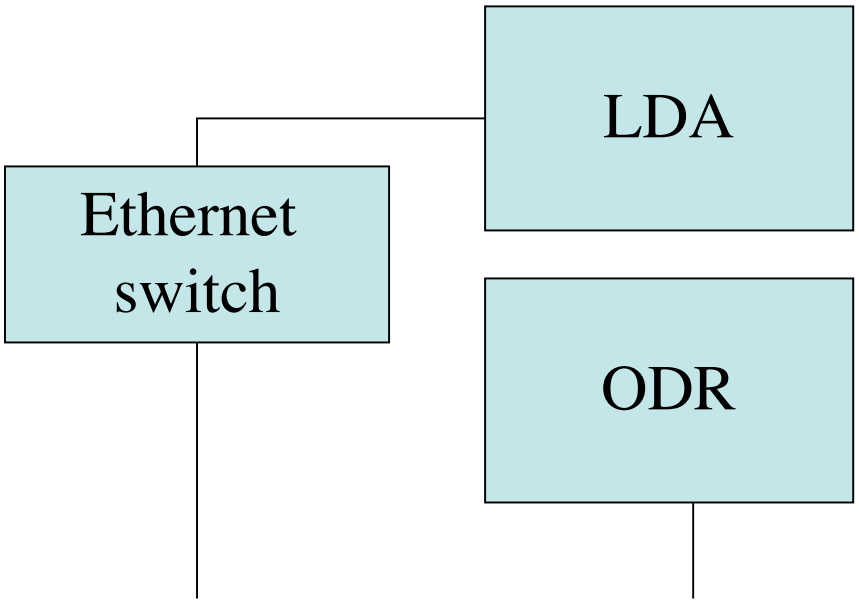
- Hardware: LDA configuration/control data can be routed over an ethernet switch

=>makes the difference between upstream and ODR data even more clear

- ODR driver: can look at upstream data

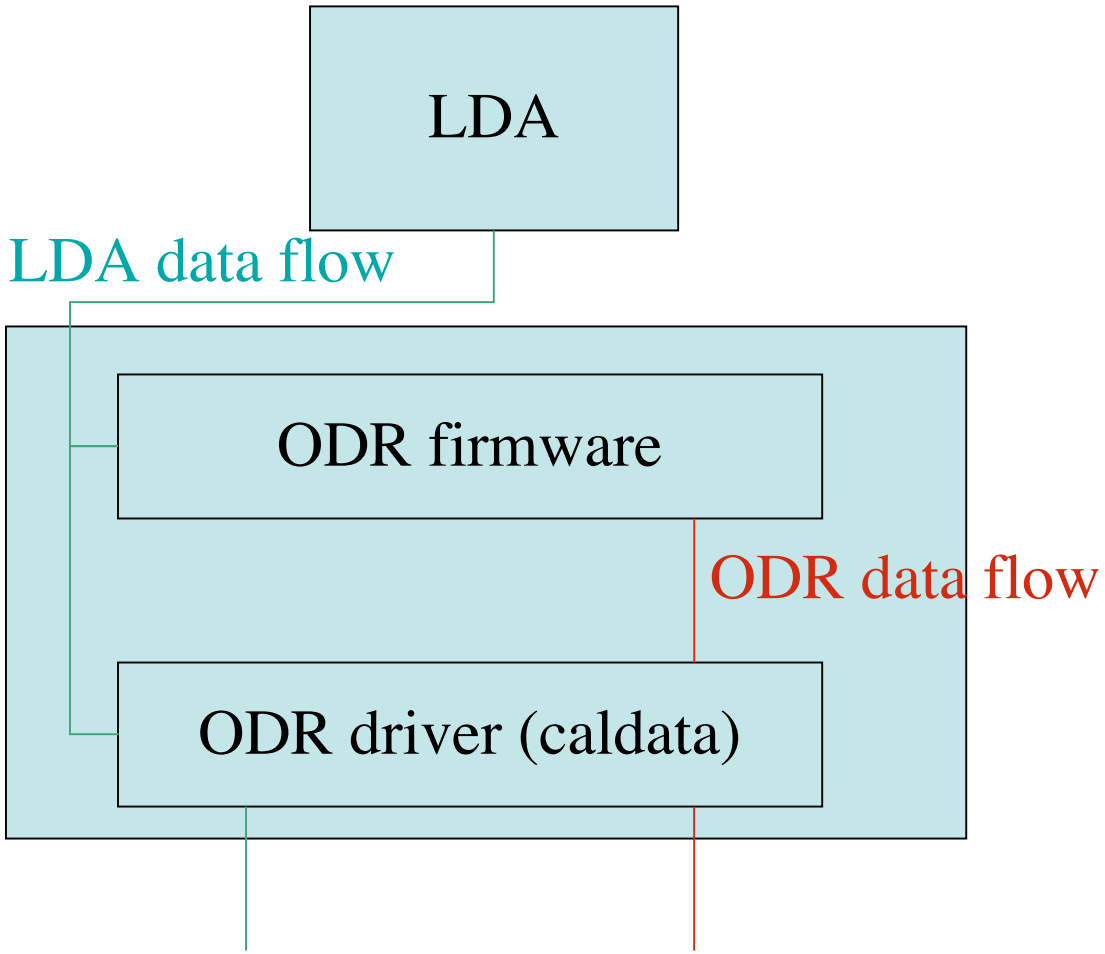
=>can distinguish between LDA and DIF data

ODR, LDA and DIF device server
- hardware, firmware, driver solutions -



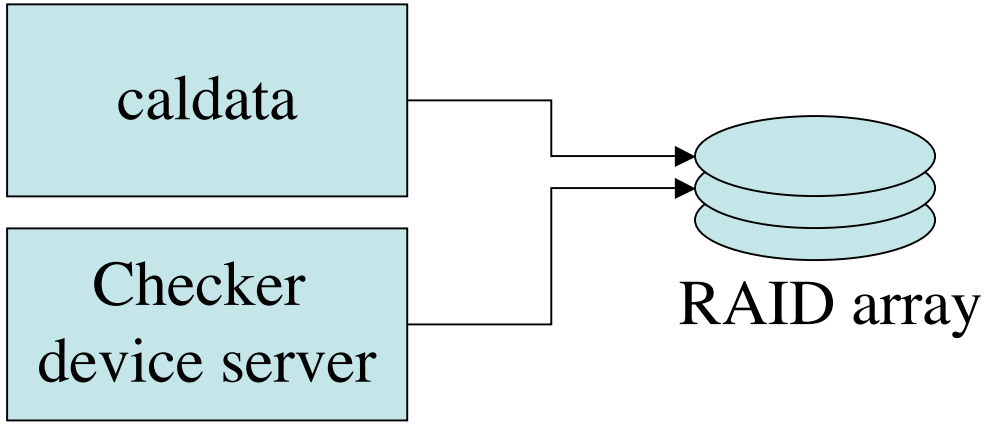
- different control/configuration data paths with the help of a switch

ODR, LDA and DIF device server
- hardware, firmware, driver solutions -



- different control/configuration data paths because ODR firmware can distinguish between data flow to/from ODR and upstream

DAQ - scenario I



- Crc checks to make sure that data are sane

(always check the latest data + skip all data received while performing checks), not clear what kind of check are done, might not even be crc checks

- Write file locations and configurations to database (mysql database)

First implementation step to be done

DAQ - scenario II

- Crc checks
- Write file locations and configurations to database
- Take care of offline event building and LCIO conversion, e.g. in the EUDAQ framework (make it in such a way, that the ASICS experts put in their expertise with the LCIO conversions), have a framework for the different detector types

For Valeria this is the desired scenario

DAQ - scenario III

- Crc checks
- Write file locations and configurations to database
- In addition online event building and LCIO conversion, can be a last step close to the testbeam (so that final data rates etc. are known)

=> depending on the data rates / our time this step might not be reached

Small to do's which occurred during the conversation

- Val: get CERN account, so that we can always update the odr.h, lda.h and other files between the hardware and software people
- Ganglia and nagios: both check how computing clusters work, could be used by us as additional check