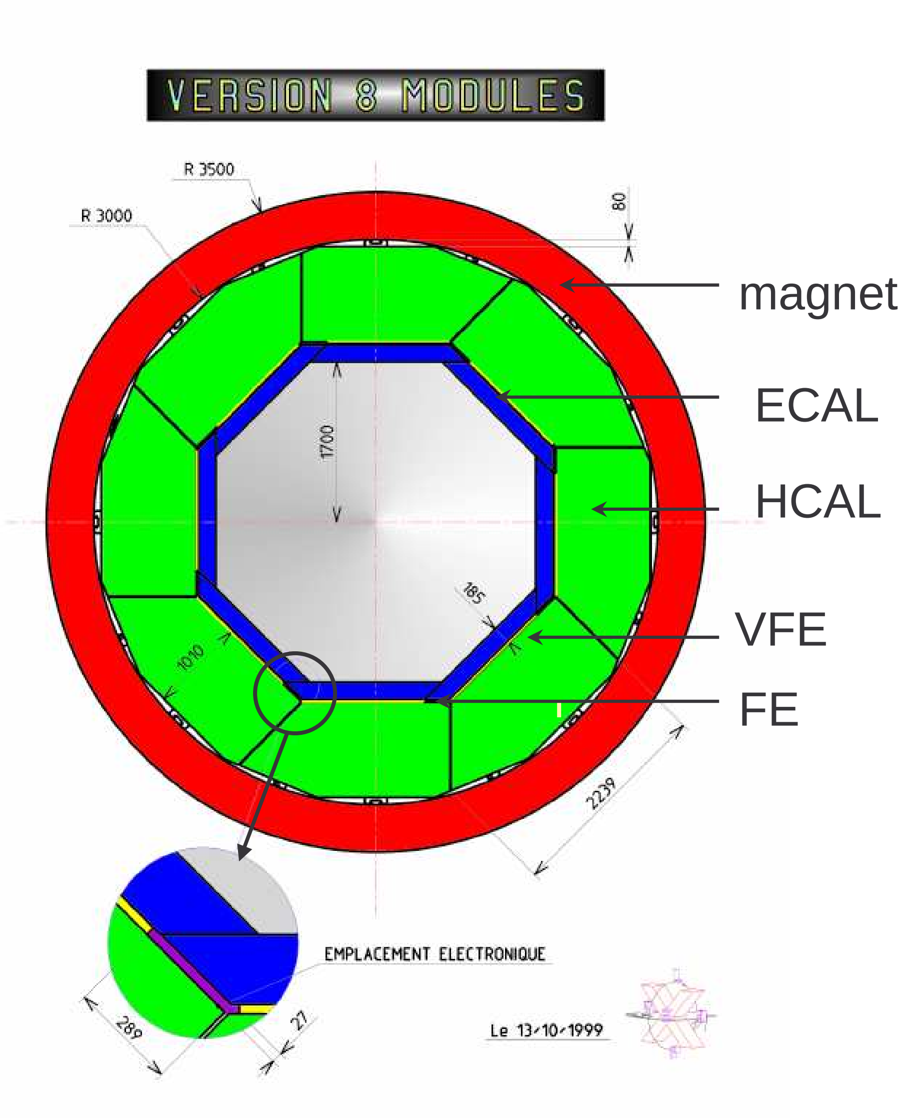
CALICE-DAQ software

Tao Wu

CALICE Collaboration Meeting

Prague, 11-13/Sep/2007

Detector & DAQ Hardware Layout



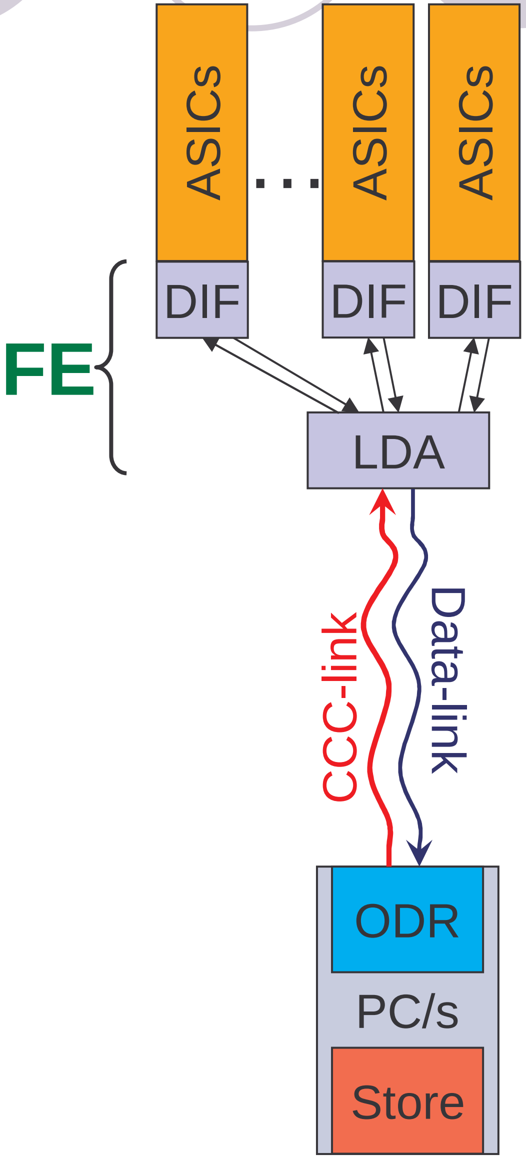
magnet

ECAL

HCAL

VFE

FE



ASICs … ASICs ASICs

DIF DIF DIF

FE

LDA

CCC-link

Data-link

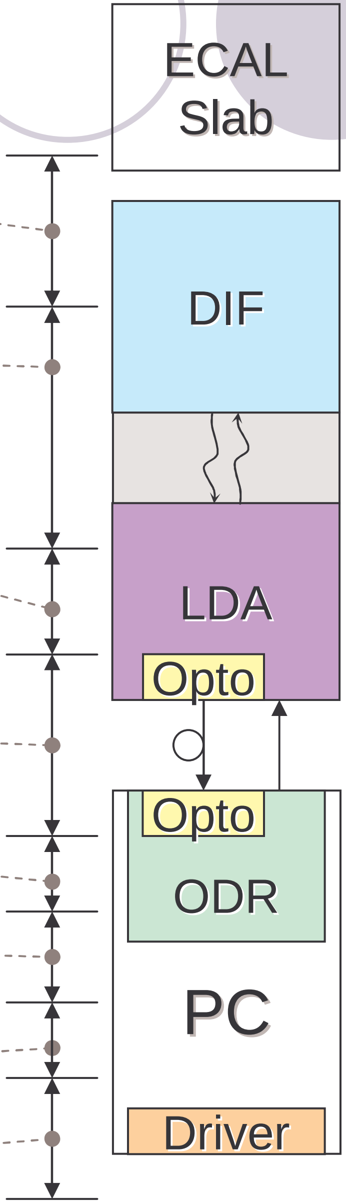
ODR

PC/s

Store

DAQ Overview

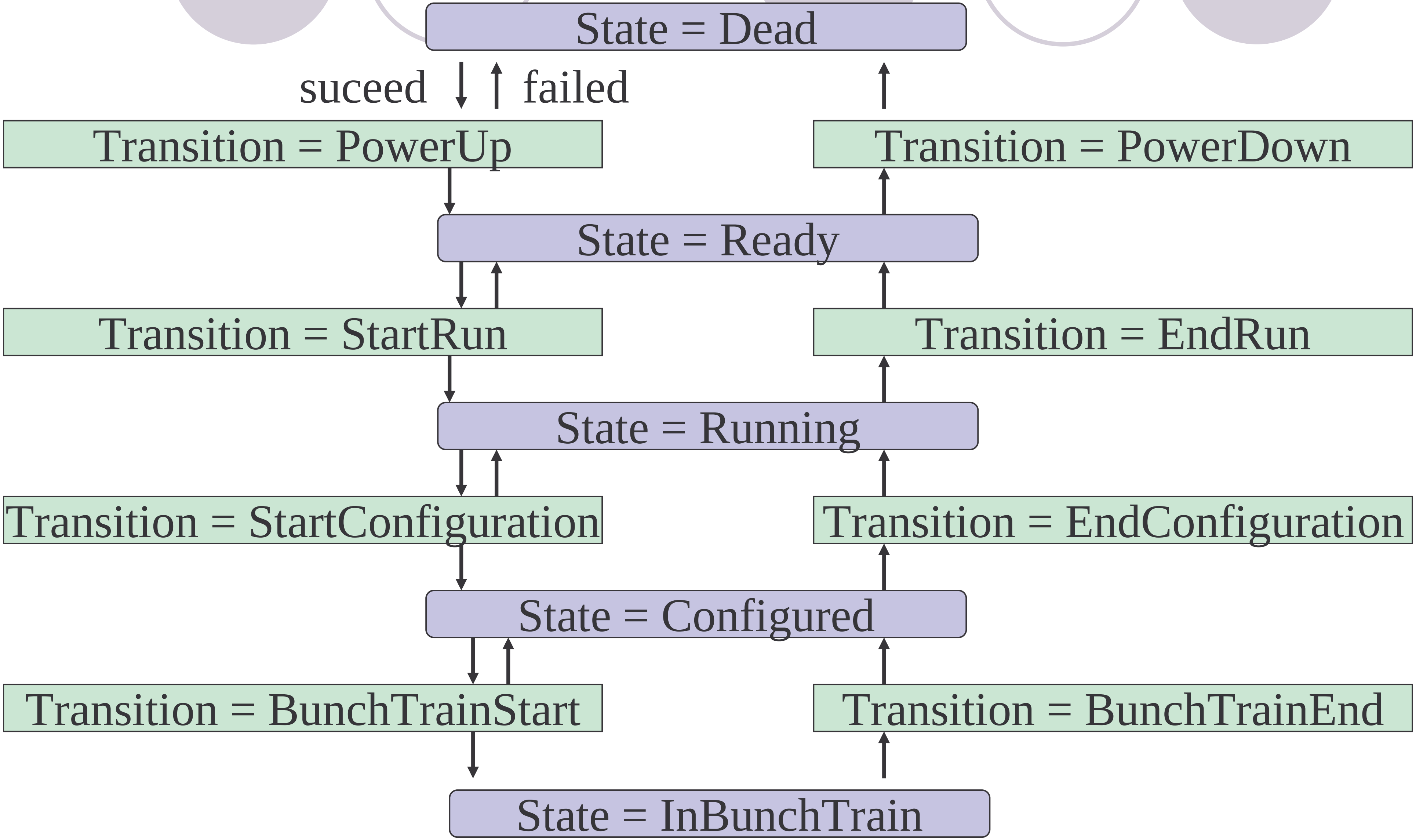
- Detector Interface (DIF)
  - Sub-detector specific, in conjunction with detector groups
- DIF to LDA
  - Generic, Copper links (25Mbit)
- Link/Data Aggregator (LDA)
  - Data format
  - Clock/Commands fan-out
- LDA to ODR opto-links
- Off-Detector Receiver (ODR)
- ODR to disk
  - PCI-Express driver software
- Local Software DAQ
- Full blown Software DAQ



DAQ software tasks

- Aim to develop a generic system
- Maximise use of off-the-shelf commercial components, cheap, scalable and maintainable
- Provide well defined interfaces between DAQ components to allow for simple upgrading or replacement in future without major re-design or cost
- Software control to integrate the rest of sub-systems of detectors
- Software to build event from bunch train data and disparate sources into single event data
- Manage Network and data storage

DAQ software for Eudet: State Analysis



State = Dead

suceed

failed

Transition = PowerUp

Transition = PowerDown

State = Ready

Transition = StartRun

Transition = EndRun

State = Running

Transition = StartConfiguration

Transition = EndConfiguration

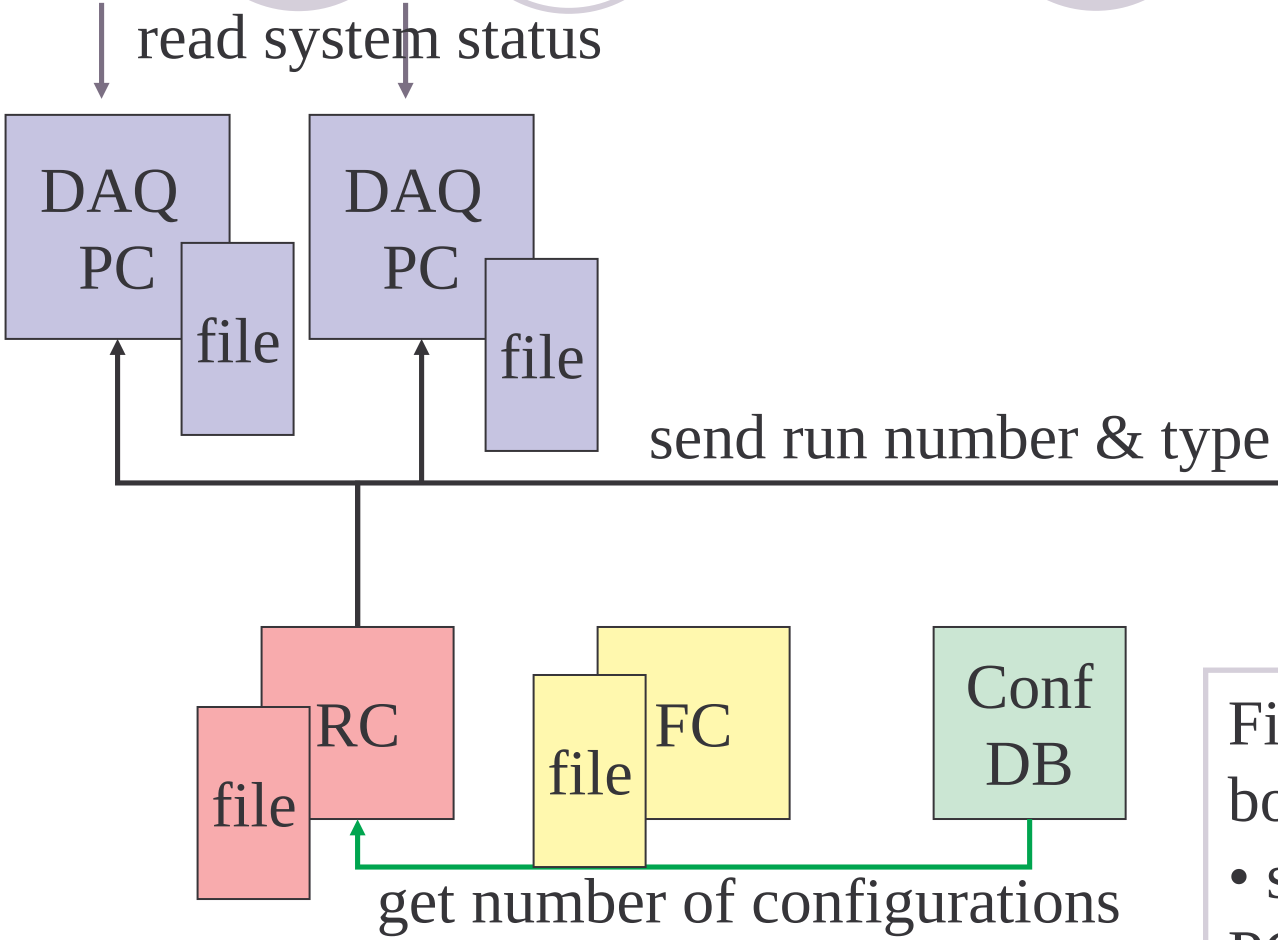
State = Configured

Transition = BunchTrainStart

Transition = BunchTrainEnd

State = InBunchTrain

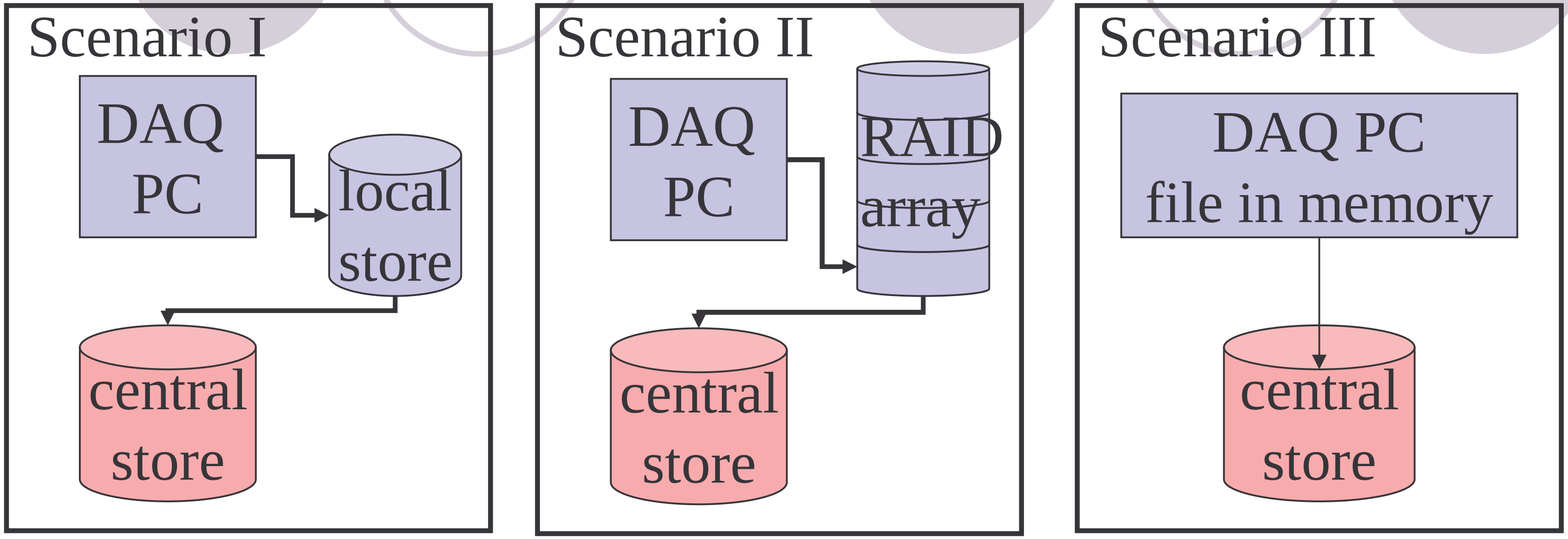
DAQ software for EUDET: Transition: StartRun



Files to be written for book keeping:

- system status by DAQ PC
- run info by RC PC
- system status by FC

DAQ software for EUDET: Data Storage



- which scenario to choose depending on the bandwidth with which the data gets produced: (I) up to 200Mbit/sec, (II) up to ~1600Mbit/sec, (III) from there on

- desirable to have files because transfer is easier and in case of timing problems error handling is easier, but keep system flexible for now

What DAQ software should be used?

- An effort and exploration is focused on EPICS;
- An alternative candidate is ACE;
- I am looking into ACE framework.

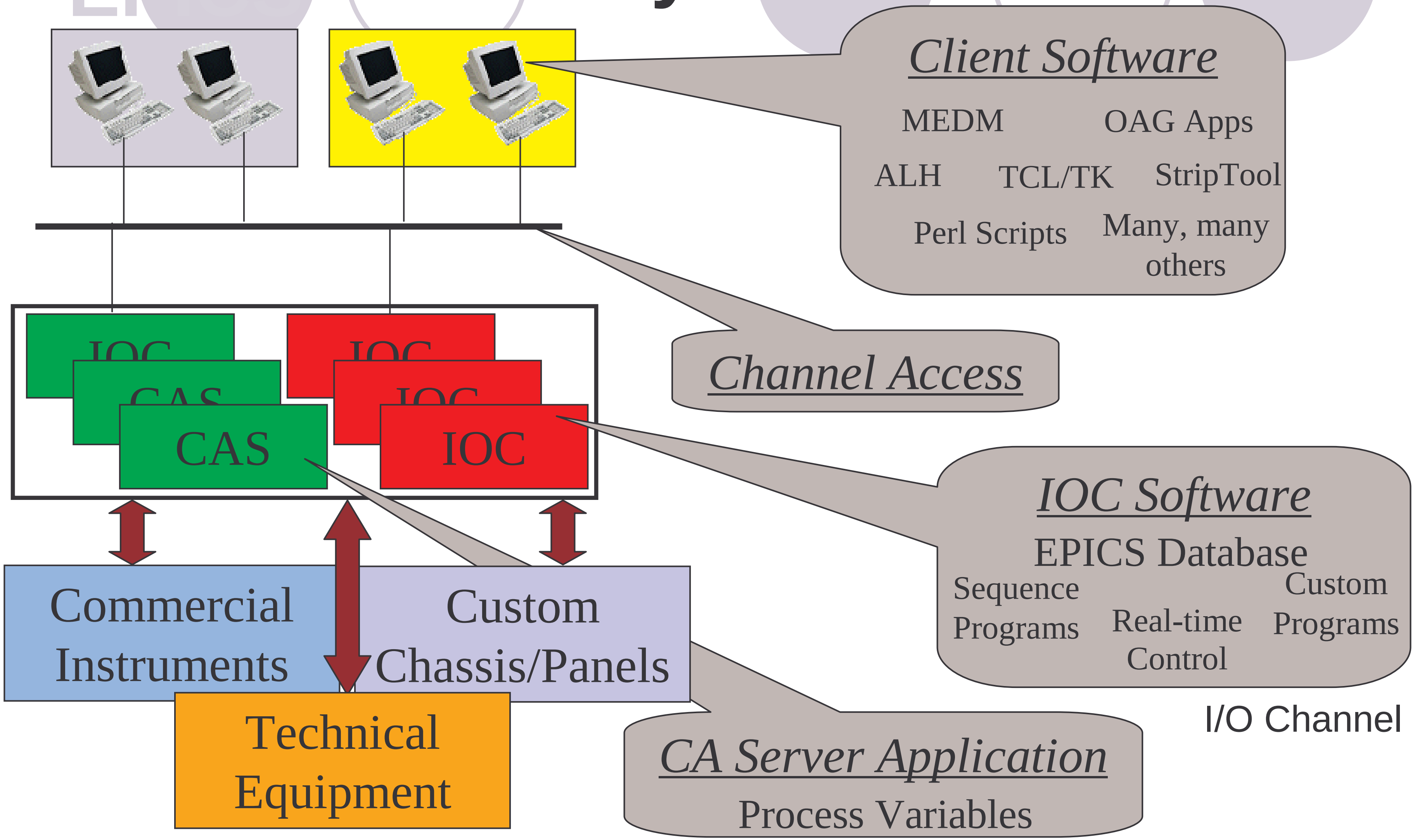
DAQ software candidate: EPICS

- What’s EPICS: Experimental Physics & Industrial Control System
- A World-wide Collaboration
- A Control System Architecture
  - Network-based “client/server” model with Channel Access Protocol for passing data
  - A distributed real-time database of machine values
- A Software Toolkit: A collection of software tools, comprehensive and scalable control system
- Successful cases: STAR/D0 …

So What Does it Do?

- EPICS tools are available to accomplish almost any typical Distributed Control System (DCS) functionality, such as:
  - Remote Control & Monitoring of Technical Equipment
  - Data Conversion/Filtering
  - Closed Loop Control
  - Access Security
  - Equipment Operation Constraints
  - Alarm Detection/Reporting/Logging
  - Data Trending/Archiving/Retrieval/Plotting
  - Automatic Sequencing
  - Mode & Facility Configuration Control (save/restore)
  - Modeling/Simulation
  - Data Acquisition
  - Data Analysis

Canonical Form of an EPICS Control System



Client Software

MEDM

OAG Apps

ALH

TCL/TK

StripTool

Perl Scripts

Many, many others

Channel Access

IOC

CAS

CAS

IOC

IOC

IOC

IOC Software

EPICS Database

Sequence Programs

Real-time Control

Custom Programs

I/O Channel

Commercial Instruments

Custom Chassis/Panels

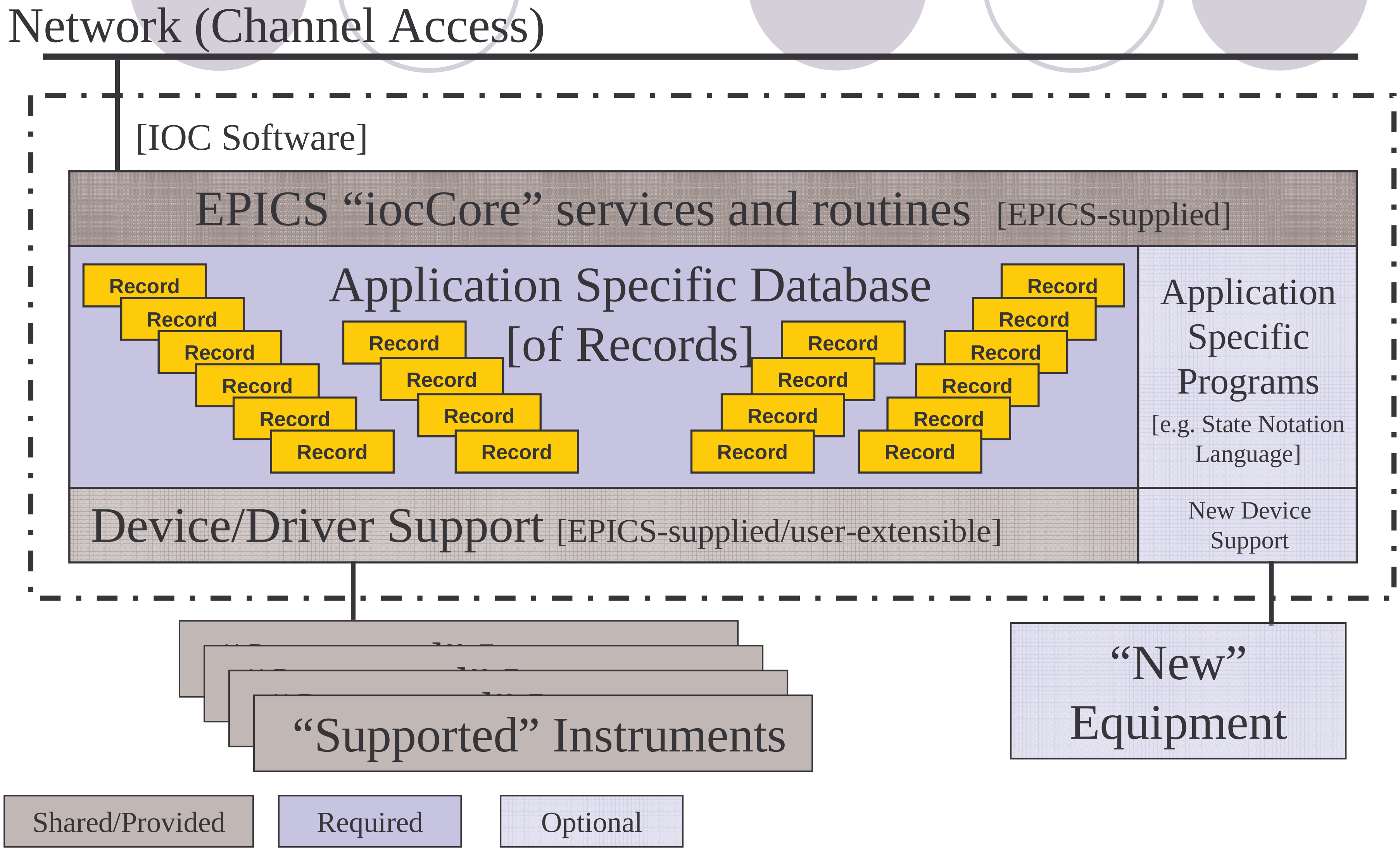
Technical Equipment

CA Server Application

Process Variables

Taken from the introduction course into EPICS

IOC Software in One Slide



Network (Channel Access)

[IOC Software]

EPICS “iocCore” services and routines [EPICS-supplied]

Application Specific Database [of Records]

Record Record

Application Specific Programs [e.g. State Notation Language]

Device/Driver Support [EPICS-supplied/user-extensible]

New Device Support

“Supported” Instruments

“New” Equipment

Shared/Provided

Required

Optional

Taken from the introduction course into EPICS

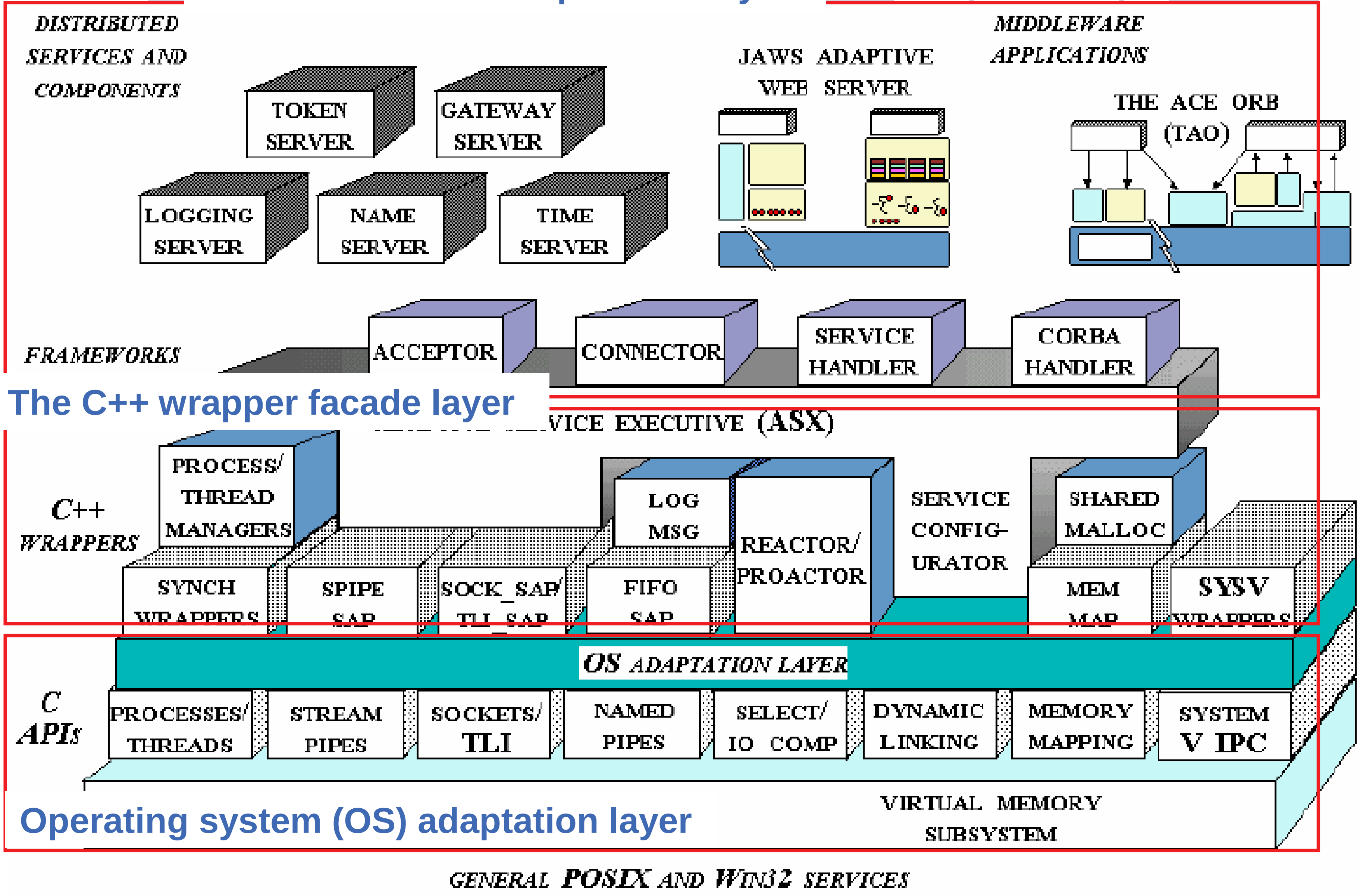
Main features linked to CALICE-DAQ

- Network-based “client/server” model with Channel Access Protocol
- Rich Client Software & Channel Access Server Application and I/O Channel software
- Toolkits: Commercial Instruments, Custom Chassis/Panels and Technical Equipment
- Common uses
  - Provide automated start-up sequences
  - Provide fault recovery or transition to a safe state
  - Provide automatic calibration of equipment
  - Benefit from Run Control and record management

ACE: alternative DAQ software candidate

- ADAPTIVE Communication Environment
- ACE is a free OO C++ toolkit, including reusable wrappers, classes and network programming frameworks, middlewares, which is portable & supportable in many Operation Systems.
- An off-the-shelf commercial components: Supported commercially by www.riverace.com

ACE Architecture



The frameworks and patterns layer

The C++ wrapper facade layer

Operating system (OS) adaptation layer

Main Functionalities of ACE

- ACE basics: Installation, Logging Facility, Containers
- Interprocess Communication: Sockets, Reactor, Proactor, Other IPC Types
- Process and Thread Management: Process, Signals, Thread, Thread Safety and Synchronization, Tasks and Active Object Pattern, Thread Pools
- Advanced ACE: Memory, Streams, Service Configurator, Acceptor & Connector, Naming Service, Message Queues
- Many topics uncovered ...

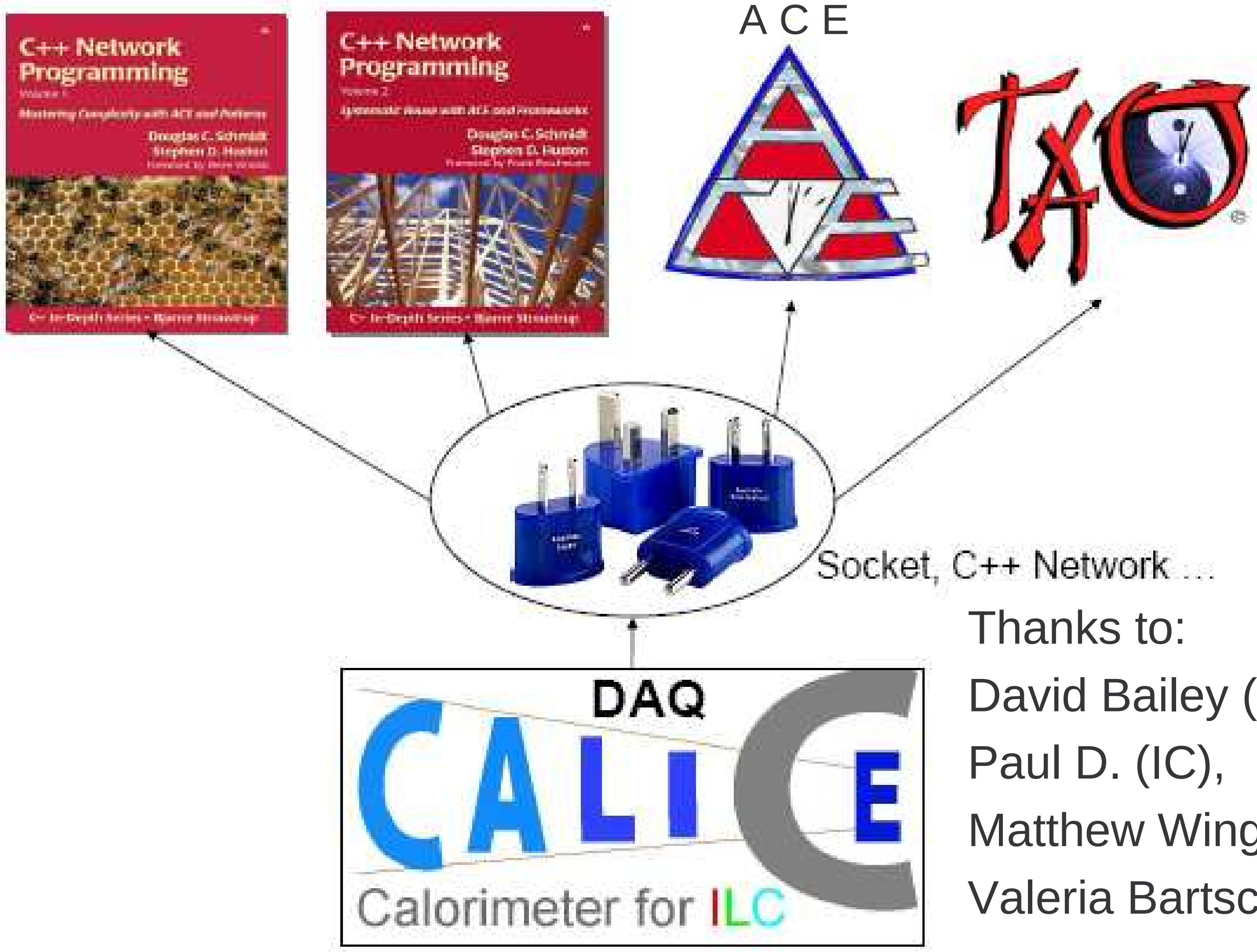
ACE functionality vs CALICE DAQ

DAQ software for EUDET	ACE
Transition state	Service configurator, message queues
Clock, control	Process, signal, timers
Book-keeping	Logging Facility
Data storage	Memory, stream
Network switch	Acceptor, connector
A/synchronous I/O capabilities	Reactor, proactor
Sub-detector talks	Unicast, broadcast & multi-cast

Summary

- DAQ software tasks are reviewed.
- Use cases of DAQ software for EUDET are discussed in some conceptions.
- An effort of DAQ software candidates is made: EPICS and ACE
- Some comparisons are made between ACE functionalities and DAQ software needs.
- Trigger open discussions of DAQ software framework? Optional: EPICS or ACE?

Thank you!



ACE

Socket, C++ Network …

Thanks to:

David Bailey (Manchester),

Paul D. (IC),

Matthew Wing, Matt Wing,

Valeria Bartsch (UCL)

ACE reference

- ACE main site: http://www.cs.wustl.edu/˜schmidt/ACE.html
- Obtaining ACE: http://download.dre.vanderbilt.edu/
- Linux Platform settings:

```
#! /usr/bin/tcsh -f
setenv ACE_ROOT
  /scratch0/wutao/ACE5.5/ACE_wrappers
setenv LD_LIBRARY_PATH
  ${ACE_ROOT}/ace:${ACE_ROOT}/lib:\
    ${LD_LIBRARY_PATH}
setenv PATH "${PATH}:${ACE_ROOT}/bin“
```

Then ``make” to compile

Backup

ACE Functionality (I)

- Logging Facility: good logging mechanism:
  - Use basic logging and tracing techniques
  - Enable and disable display of various logging message severities
  - Customize the logging mechanics …
- Interprocess Communication
  - Service access point wrappers:

    sockets, FIFO, stream pipe
  - Reactor & proactor: (a)synchronous I/O capabilities
  - Other IPC: unicast, broadcast & multi-cast, files, pipes, FIFOs, share-memory stream

ACE Functionality (II)

- Process & thread:
  - Start and terminate, (a) synchronize processes & signals
  - Thread management: creation, suspension, cancellation and deletion, locks, guards and conditions, sending, destroying, waiting, cooperation.
  - Priorities and scheduling classes in pools
  - Safety, synchronization and specific storage, and multi-thread programs
  - Active object and tasks managements

ACE Functionality (III)

- Rich array of memory management classes:
  - manage dynamic memory (memory allocated from the heap): more flexible, can be changed at runtime.
  - manage shared memory between processes: perform better, configured at compile time.
  - Map Interface: LIFO/FIFO, ACE MMAP Memory Pool,
  - ACE Shared Memory Pool, ACE Local Memory Pool
  - Memory Protection Interface & Synchronic Interface

ACE Functionality (IV)

- The streams
  - A one-way stream to record and process messages.
  - A Bidirectional Stream to implement a command stream

ACE Stream Class:

- open(), close(), wait();
- push(), pop(), top(), insert(), replace(), remove();
- get(), put();

ACE Functionality (V)

- ACE Acceptor:
  - Passive Connection Establishment
  - Handling of the connection after establishment
- ACE Connector:
  - Active Connection Establishment
  - Handling of the connection after establishment
- Uses TCP to establish the connection
- Uses UNIX domain sockets to establish the connection

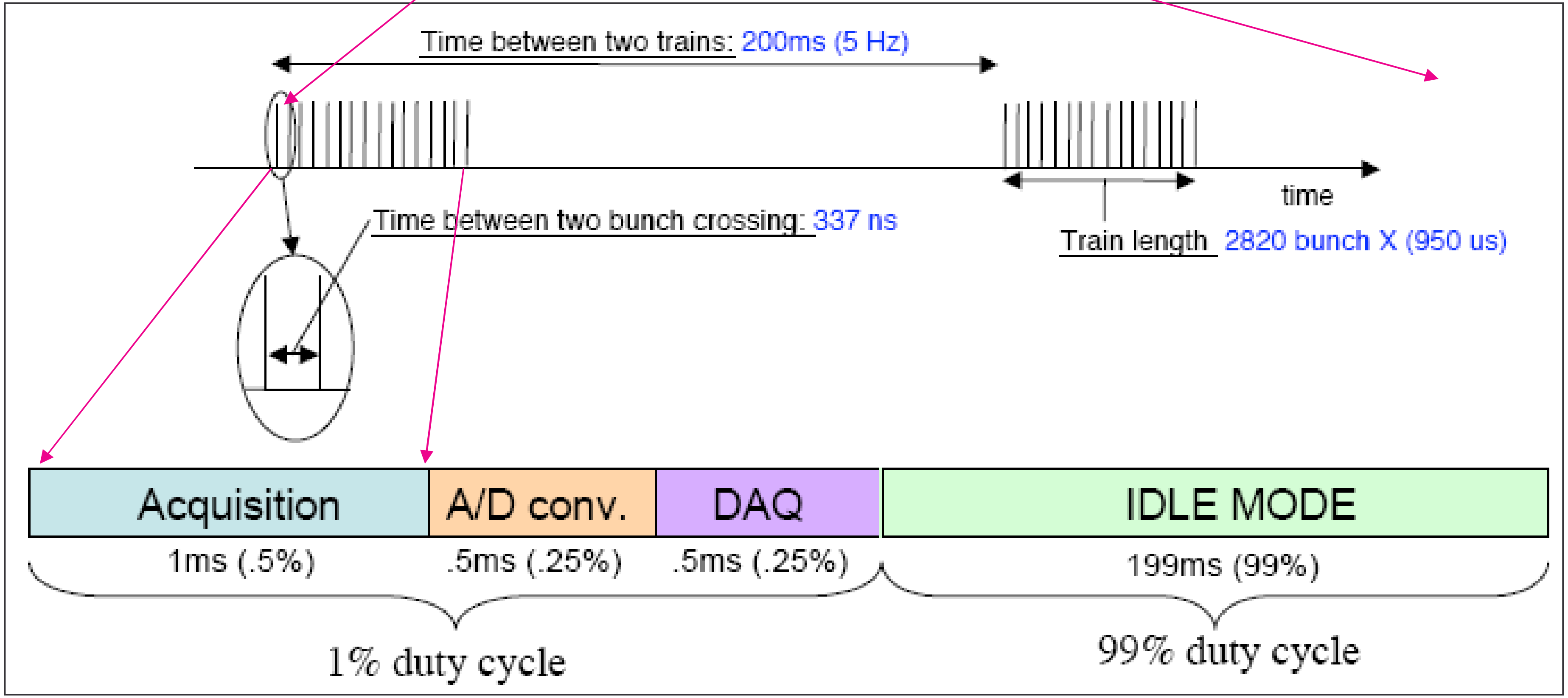
ACE Functionality (VI)

- Naming Services: Type of name space
  - A Single-Process Naming Context
  - Sharing a Naming Context on One Node
  - Sharing a Naming Context across the Network

Timing Consideration

1 run = several bunch trains
1 bunch train = many bunch crossings
1 bunch crossing = 1000 events

LCIO: event-by-event
Now no event is defined.


Time between two trains: 200ms (5 Hz)
Time between two bunch crossing: 337 ns
time
Train length 2820 bunch X (950 us)
Acquisition
A/D conv.
DAQ
IDLE MODE
1ms (.5%)
.5ms (.25%)
.5ms (.25%)
199ms (99%)
1% duty cycle
99% duty cycle

DAQ system general R&D work

- Make possibilities as to what can be done in the VFE/FE, Assume reading out higher data rate and can definitely do anything lower.
- Using commercial, off-the-shelf products, cheap, scalable and maintainable.
- Backplaneless readout
- Identify bottlenecks in this concept, effects on the calorimeter system.
- Perform data reformating, calibration, linearisation & digital filtering
- Should be applicable to the HCAL - other non-calorimeter components
- Test-bench work and demonstration of workability of concept.
- Be able to provide DAQ for prototype calorimeters being developed.