

New Hybrid Application Environment for Controls

Andrey Petrov, AD/Controls

ILC Controls Meeting, 2007-08-13

What is an application environment?

- A “glue” that turns a hundred of lose software components into a single orchestrated system
- Facilitates application development and administration through:
 - Reusable services
access to resources, security, etc.
 - Reusable code
 - Central management of application components
deployment, starting, and stopping
 - Uniform monitoring
 - Established development procedures
- Hides underlying complexity of the system

A users' perspective

- Valid questions from the users:

“How do I write a program...”

... to monitor machine parameters from home?

... to set a device from a D0 trailer?

... that constantly runs on a server collecting some data and provides reports in HTML format?

- Most users [perhaps] don't care which low-level control system is used, unless they have troubles over getting data

Why the **new** environment?

- Currently, the application environments—if exist—are offsprings of corresponding data acquisition systems
- Modern technologies are not supported; new applications emerge anyway, but live on their own
e.g., multi-tier architectures
- New challenges can't be fully addressed withing frameworks of the existing systems
e.g., security

Major issues

- Availability of data, applications
a legitimate user can't start a function program where needed
- Security
no real access control; firewalls rules the world
- Common application context
links between distributed software components, management & monitoring
- Development procedures
reasonably simple way to develop and deploy new stuff
- Integration of different control systems
can be done in a common middleware

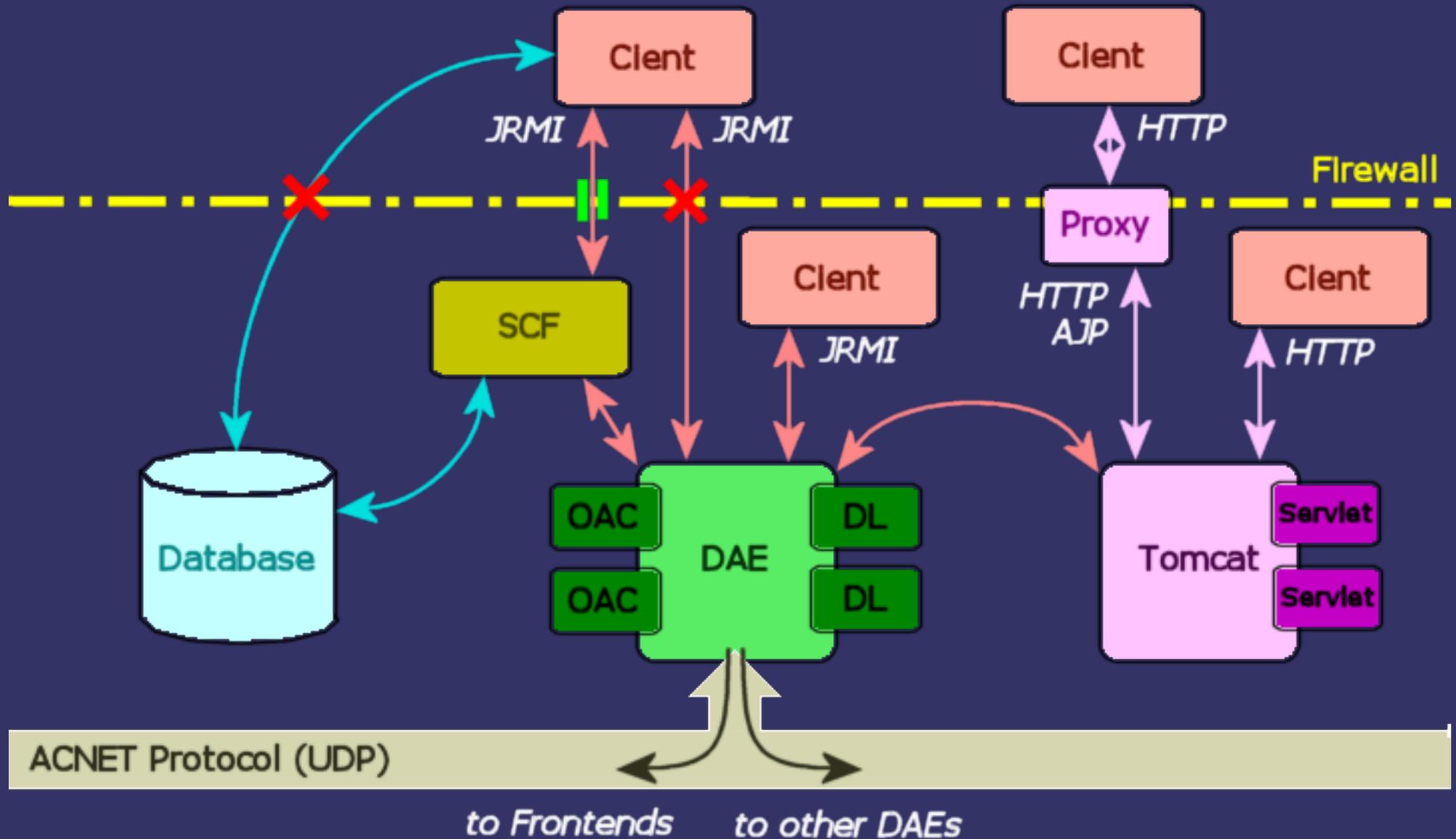
Concepts

- Separation of **low-level data acquisition systems (DAQ)** from the **top-level application environment (AE)**
- Development of a new hybrid AE independent of any particular DAQ, based on general technologies available at this time
- Keeping existing infrastructures unchanged

Objectives

- Short term:
 - to benefit existing control systems (ACNET, EPICS, DOOCS), by providing a flexible operating environment and a set of central services that are not currently available or need to be improved.
- Long term:
 - testing and demonstration of candidate technologies relevant to the ILC R&D program.

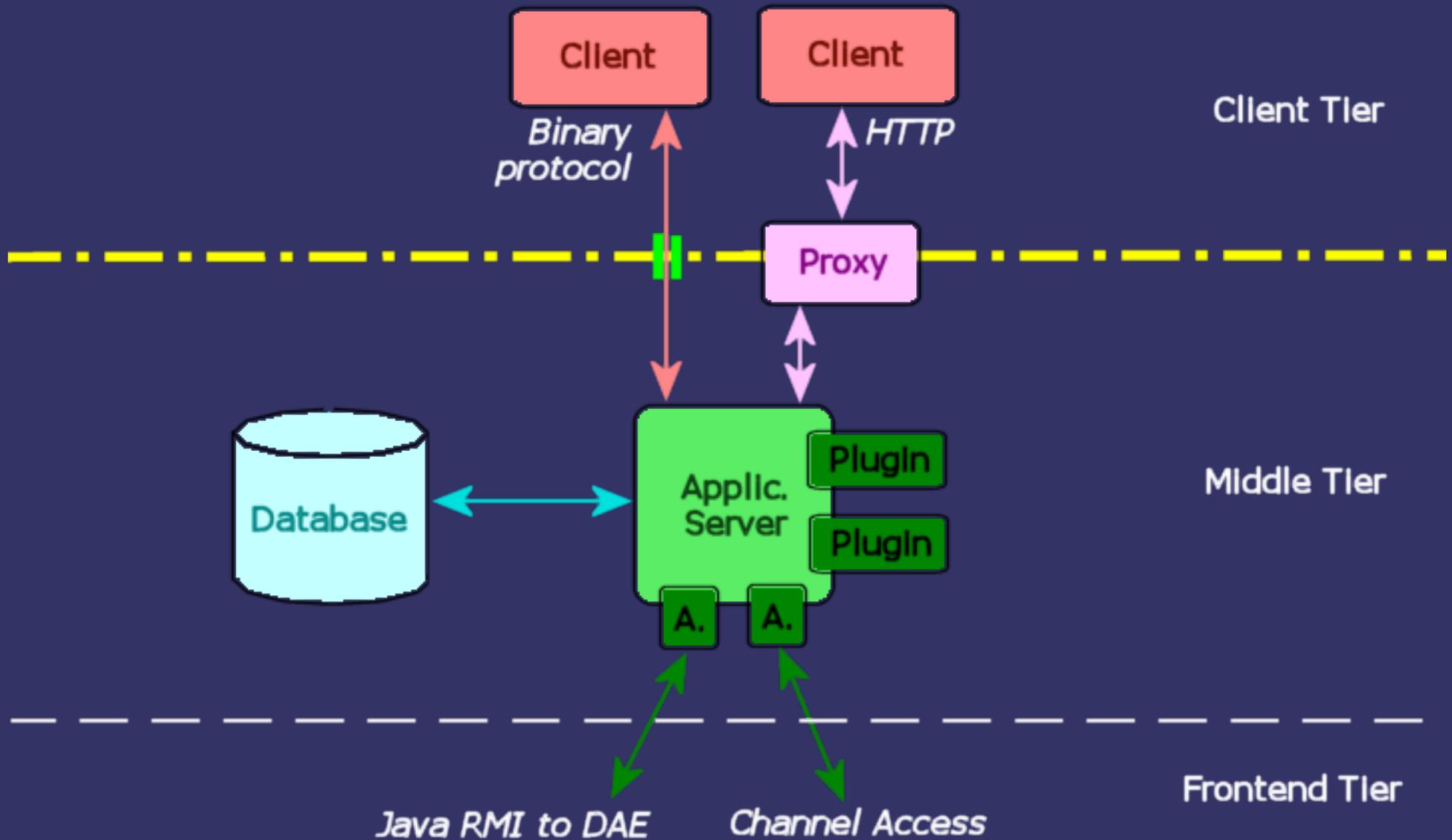
Existing AE in ACNET



Drawbacks in the current AE

- Three separate processes in the middle tier
- None of the containers supports general-purpose plugins
- Inefficient and proprietary binary protocol to the clients (Java RMI)
- Supports only one low-level data acquisition system
- Heavy dependencies within the code
- Configuration inherited from the legacy system

Proposed architecture



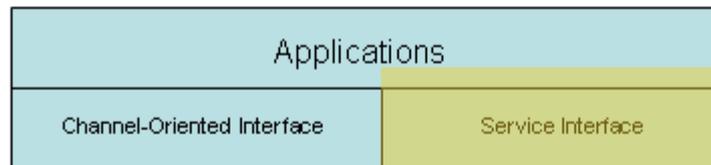
Principal components

- Application server
 - runs a set of loosely coupled components, which perform individual tasks*
- Data acquisition API, device and data abstractions
- Language independent client-server protocols
 - Binary (RPC or messaging), with asynchronous callbacks
 - Text-based (XML)
- Code management and release system

Services

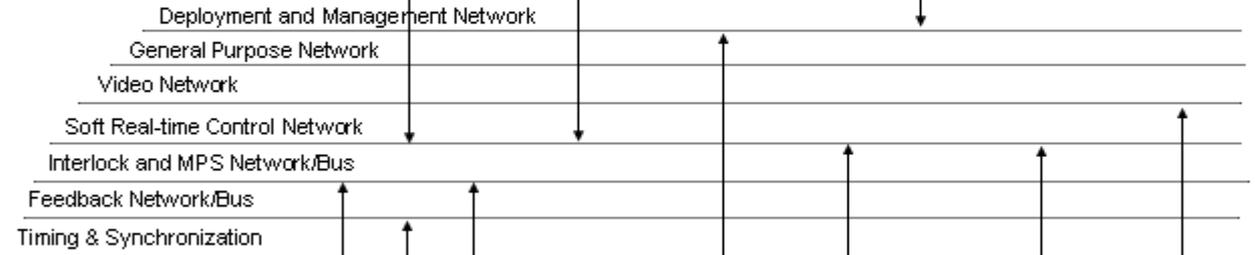
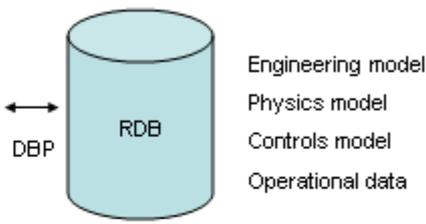
- Data acquisition
 - includes adapters to individual low-level DAQ
 - use to provide data to the clients, and to bridge data between different low-level DAQs
- Security
 - authentication & authorization, user database*
- Naming & directory service
 - database of property and devices, etc.*
- A simple service to run shell scripts
- Application index
 - distribution of user applications*

Client Tier

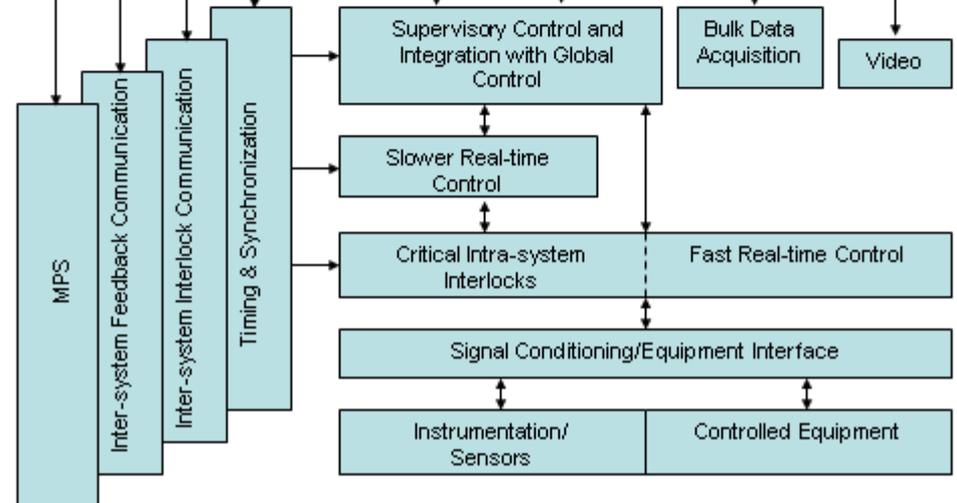


DOP – distributed object protocol
MQP – message queuing protocol
SRTP – soft real-time protocol
DBP – database protocol
DMP – deployment and management protocol

Services Tier



Front-end Tier



Candidate technologies

- Application server — ? + elements of Ice (see below)
- DAQ API, device and data abstraction — **DAL** [Data Acquisition Layer], developed by CosyLab for DESY
- Client-server protocols:
 - Binary — **ZeroC Ice** [Internet Communication Engine] (<http://zeroc.com>), an alternative to CORBA
 - Text-based — **Plain XML over HTTP**, probably **SOAP**
- Code management and release system — ?

Questions, suggestions, and participation
are welcome