Middleware Workshop

Project core team:

SL: Vito Baggiolini, Kris Kostro, Marc Vanden Eynden

PS : Franck di Maio, Alessandro Risso

Middleware Workshop

◆ Motivation	Kris
◆ Accelerator Device Model & Java API	Franck
◆ Object-Oriented Middlewares	
◆ CORBA	Kris
◆ DCOM	Alessandro
◆ Coffee break	
◆ Java Enterprise Beans and RMI	Vito
◆ Message-Oriented Middlewares	
◆ MSMQ, SmartSockets, IBUS	Marc, Vito
◆ OPC	Vito
◆ Selection criteria	Marc

Slides of the workshop

http://hpslweb.cern.ch/pssl/middleware/middleware.html

Motivation for the Middleware

Kris Kostro SL/CO

Why this workshop

- Middleware team asked for more information
- Postgraduate course on Middleware at EPFL Lausanne 1, 2, 3 of March
- We do not consider ourselves experts in the subject we present

Middleware Motivation

- PS/SL Convergence project early 1998
  Asked for a Vision of the new controls infrastructure
- Currently used technology is 15 years old - it's time to look for a new one
- Technology shift - Java, WWW, CORBA, DCOM, OO tools & techniques
- Data exchange with LHC subsystems and experiments
- Pushed by SPS 2001 (SPS as injector to LHC) project

What is Middleware

Middleware is the Software Bus for distributed applications

- Goes beyond the client-server model
- Additional layer which allows to interact with server (or service) abstraction.
  - Distributed objects
  - message queues
  - Unified DB access
- Added value such as location service, reliability, authentication, transaction semantics

Are SL-Equip and PS-Equip Middlewares ?

- Offer location service
- Based on standard RPC call
- In SL-Equip standard for plugging-in servers with SVMQ

Middleware project launched by PS/SL convergence

To provide a software communication architecture and services allowing inter-object communication, mainly for the Accelerator Device Model

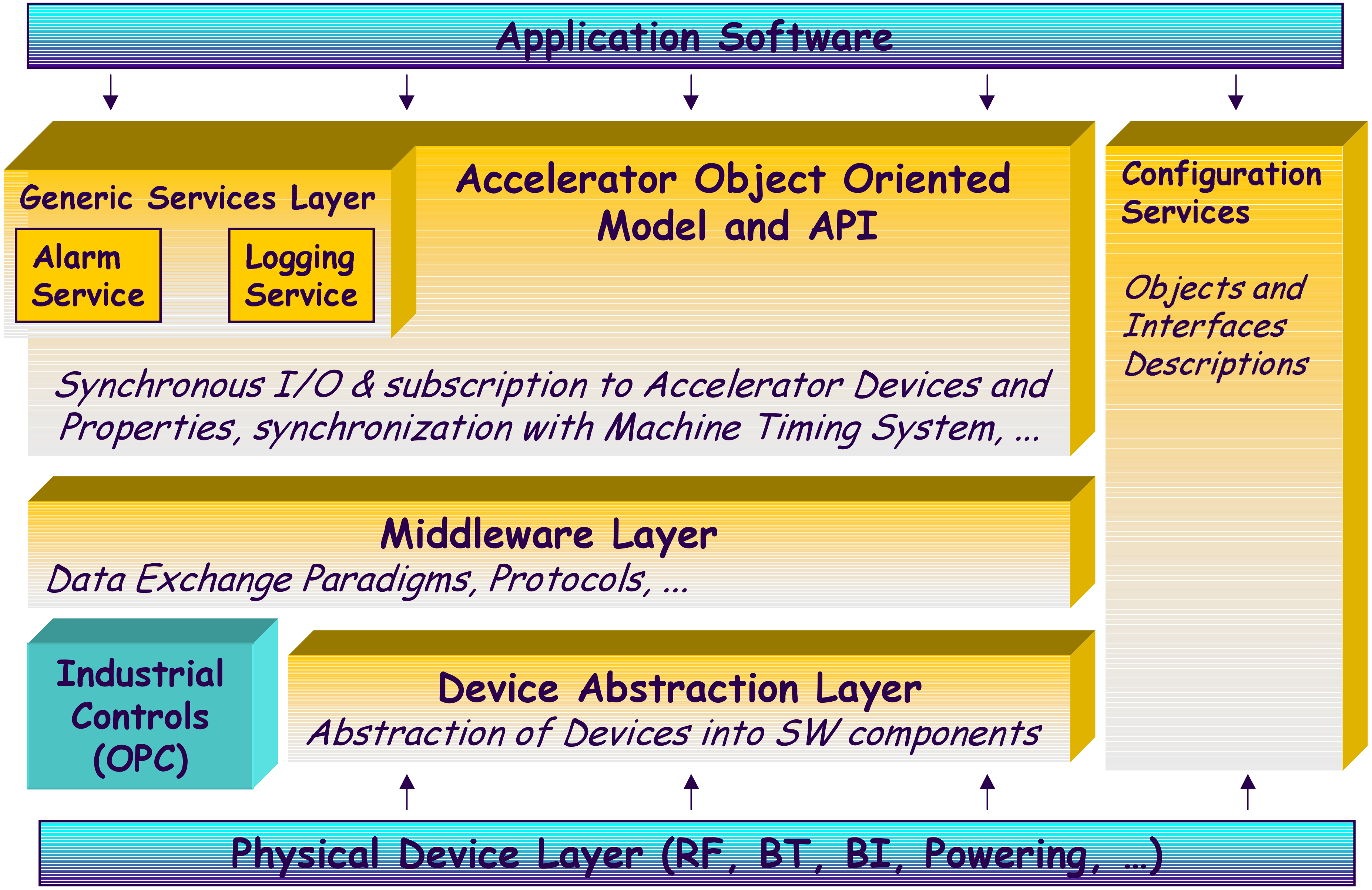
Scope

- **Synchronous device I/O**
- **Asynchronous distribution of device properties (publish-subscribe)**
- **“plugging-in” of physical & virtual devices**
- **Generic services (logging, alarms)**
- **Interoperability with industrial systems**

Project Phases

- **Phase I**
  - Requirement capture & analysis
  - Evaluation and choice of middleware technology
  - Definition and implementation of test cases.
- **Phase II**
  - Final architecture specification, Design and implementation.
- **Phase III**
  - Integration of existing servers, Connecting existing equipment.

Initial architecture


Application Software
Generic Services Layer
Alarm Service
Logging Service
Accelerator Object Oriented Model and API
Synchronous I/O & subscription to Accelerator Devices and Properties, synchronization with Machine Timing System, …
Configuration Services
Objects and Interfaces Descriptions
Middleware Layer
Data Exchange Paradigms, Protocols, …
Industrial Controls (OPC)
Device Abstraction Layer
Abstraction of Devices into SW components
Physical Device Layer (RF, BT, BI, Powering, …)

Middleware Introduction

Kris Kostro SL/CO

What is Middleware

Middleware is the Software Bus for distributed applications

- Goes beyond the client-server model
- Additional layer which allows to interact with server (or service) abstraction.
- Added value such as location service, reliability, authentication, transaction semantics

Middleware Families

- Object Request Brokers (ORB)
  - Extend the object model to distributed case - remote object encapsulates state, access via method calls
  - CORBA, Microsoft DCOM, Java RMI
- Message-Oriented Middlewares (MOM)
  - Mostly offer reliable messaging, some support multicasting
  - MSMQ, SmartSockets, IBUS

Middleware Facilities

- Basic facilities
  - How do you communicate
  - How do you describe the interfaces
  - How do you find servers and services
- Useful Services
  - Transactions
  - Security
  - Concurrency control

Middleware Facilities (2)

- Programming and deployment
  - Support for languages
  - Support for hardware platforms and interoperability
  - Support for Object-Oriented Model
  - Support for Components

  - Performance, scalability

Communication in ORB

- In OO middleware the basic method of communication is a synchronous method call
- Method call is like remote procedure call
- Fits well in the encapsulation principle
  - The remote object manages it's state
  - The state is only accessible through it's operations (methods)
- Subscription can be implemented with callbacks

Communication in MOM

- All communication is via messages
- Messages need guaranteed delivery
- There is normally a third party which can store messages
- MOM can be implemented in an ORB

Interface Description

- Describes allowed operations on objects
- Can include interface ID’s
- Normally in an independent descriptive language (IDL)
- Used for stub generation and for introspection

Naming and location

- Finding an object using a symbolic name
- Referencing an object so that the operation can be called
- Needs standards and databases

Other services

- Transactions
  - Group a number of interactions together
  - All-or-nothing semantics (commit or roll-back)
- Security
  - Authentication of users
- Distributed Locking
  - Concurrency control and synchronization
  - Exclusive access to shared resources

Components

Component is a piece of software small enough to create and maintain, big enough to deploy and support and with standard interfaces for interoperability

- For programming in large
- For non-programmers and visual programming
- Builder support