Progress with the Middleware

Alessandro RISSO

Outline

Project Overview
Middleware Capabilities
Adopted Technologies and Usage
Current & New Developments
Deployment
What Next?

Outline

Project Overview

Middleware Capabilities

Adopted Technologies and Usage

Current & New Developments

Deployment

What Next?

What is Middleware?

- Middleware is the **software** between the application programs and the operating system and base networking

- It is the ***SLASH*** in the term ***Client / Server***

What is the PS/SL **C**ontrols **M**iddle**W**are?

- **CMW** is the new communication mechanism **supported by PS/CO** and **SL/CO** groups

- It offers the infrastructure to **exchange data** and **commands** between different parts of a distributed control system

Project Members

- **PS/CO**: Steen Jensen, Bartek Paszkowski, Alessandro Risso, Nikolai Trofimov
- **SL/CO**: Vito Baggiolini, Francois Chevrier, Francesco Calderini, Kris Kostro, Marc Vanden Eynden

Project Overview

- **Summer 1999**
  - Requirements from PS/ SL control & equipment groups published
- **Winter 1999**
  - Technical choices published in the “Whitepaper”
- **Spring 2000**
  - Elaboration of Architecture and APIs
  - Evaluation of middleware products
- **Summer 2000**
  - Development of prototypes
- **Autumn 2000**
  - Selection of middleware products
- **Spring 2001**
  - Ready to deploy CMW v.1.0

Outline

Project Overview

Middleware Capabilities

Adopted Technologies and Usage

Current & New Developments

Deployment

What Next?

CMW Capabilities & APIs

Middleware provides:

- A comprehensive set of higher-level **distributed computing capabilities**
- A set of standards-based **interfaces** (API)
  - Technology independent
  - Stable
  - Available in Java, C++ (and C)
  - Documented at **cern.ch/controls-middleware**

CMW Capabilities (1/2)

DEVICE-PROPERTY Model

- HW and SW entities are represented as **Devices**
- Devices have **Properties**
  - Composed of elements of **Simple Data Type** like Integer, …, Double, String and **Arrays** of them
- Properties can be **Set**-**Get**
  - Blocking (synch) & non blocking (asynch)
- Properties can be **Monitored**
  - Publish/Subscribe on value change and on cycle event

CMW Capabilities (2/2)

TOPIC Model

- Based on the **Publish** & **Subscribe** communication paradigm
  - Used when multiple application need to receive the same message
  - Conceptually similar to the Newsgroup
- Communication is **Asynchronous**
- Well adapted to **Loosely Coupled** systems

Outline

Project Overview

Middleware Capabilities

Adopted Technologies and Usage

Current & New Developments

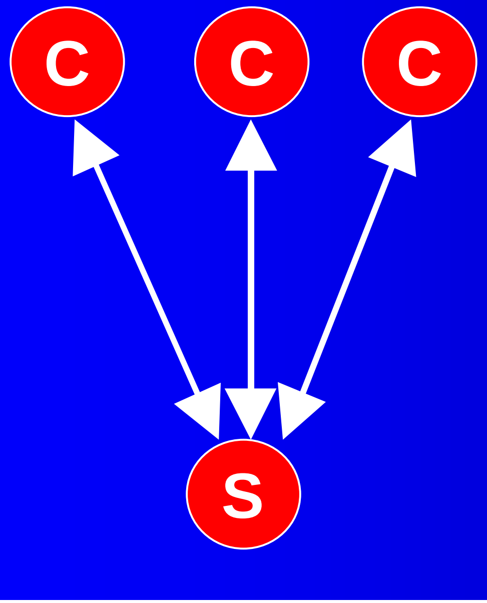
Deployment

What Next?

Chosen Technology

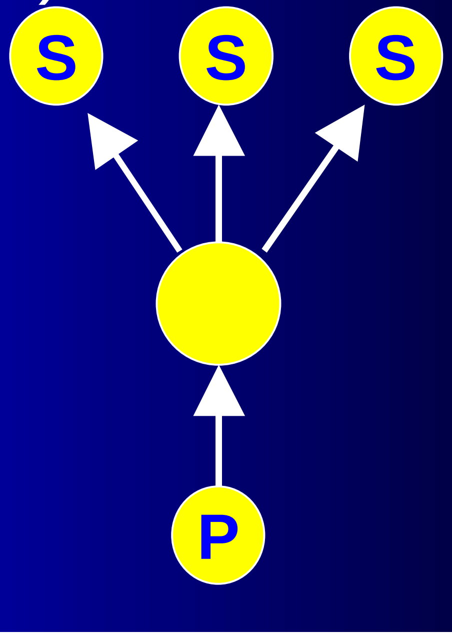
CORBA

- Establishes Client / Server relationship between objects
- OMG Standard



Message Oriented MW

- Implements the Publish / Subscribe communication paradigm
- Java Message Service (JMS) API



Why both **CORBA** & **MOM**?

- **CORBA** is the only fully interoperable middleware
- **CORBA** is available for
  - Any Common Programming Language
  - Any Operating System
  - Many Products

BUT

- **MOM** scales better
  - Message servers can be added as needed
- **MOM** is excellent for loosely coupled systems
  - Publisher only needs to know the topic
  - Subscriber only needs to know that a topic exists

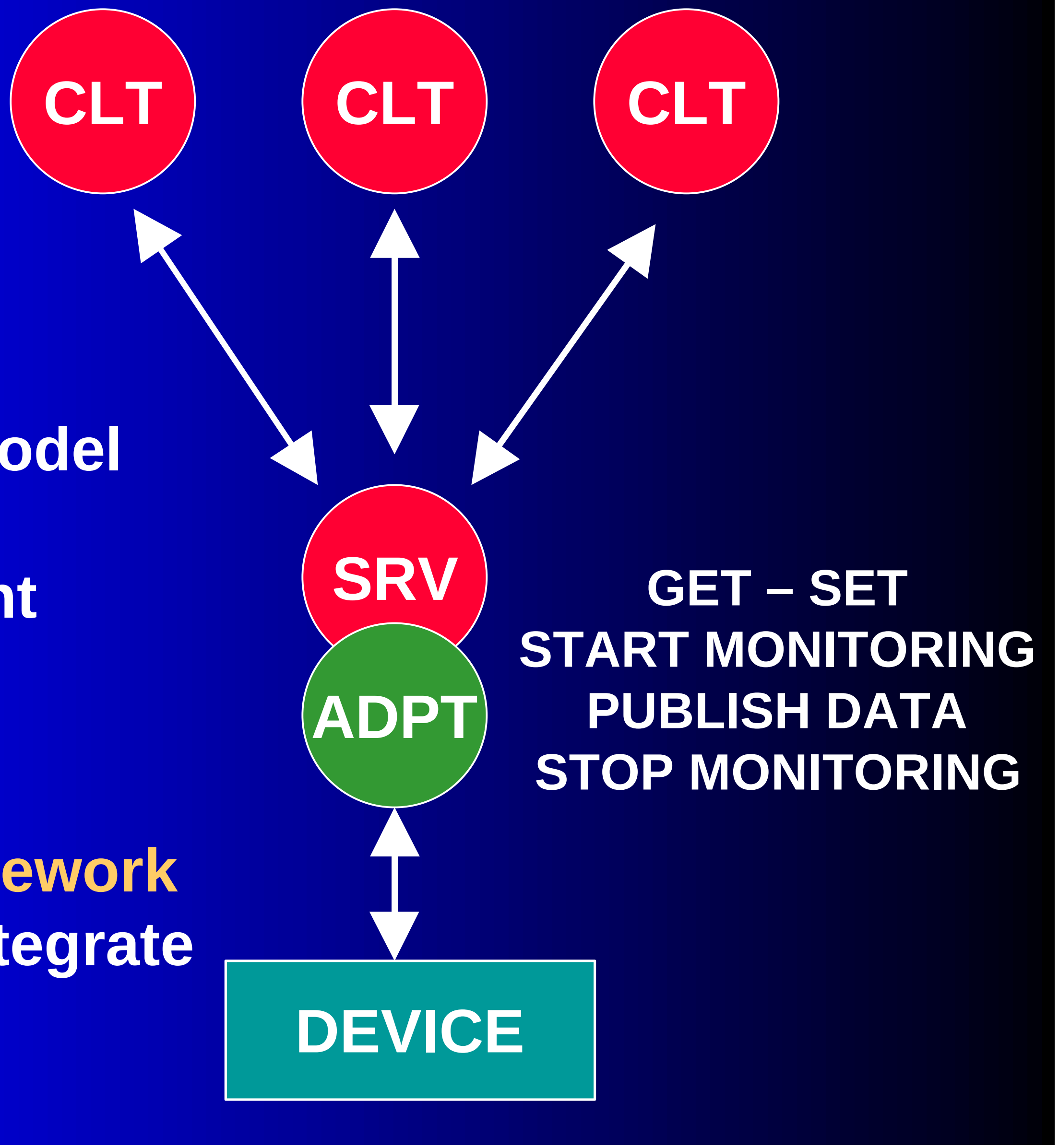
Support for the Dev/Prop Model

Client – Server
based on ORBacus
and ORBexpress

Remote Device Access Package
Support for Dev/Prop Model
Configuration Services
Connection Management

Generic Data Container

Integration Server Framework
Adapters are used to integrate new & legacy software



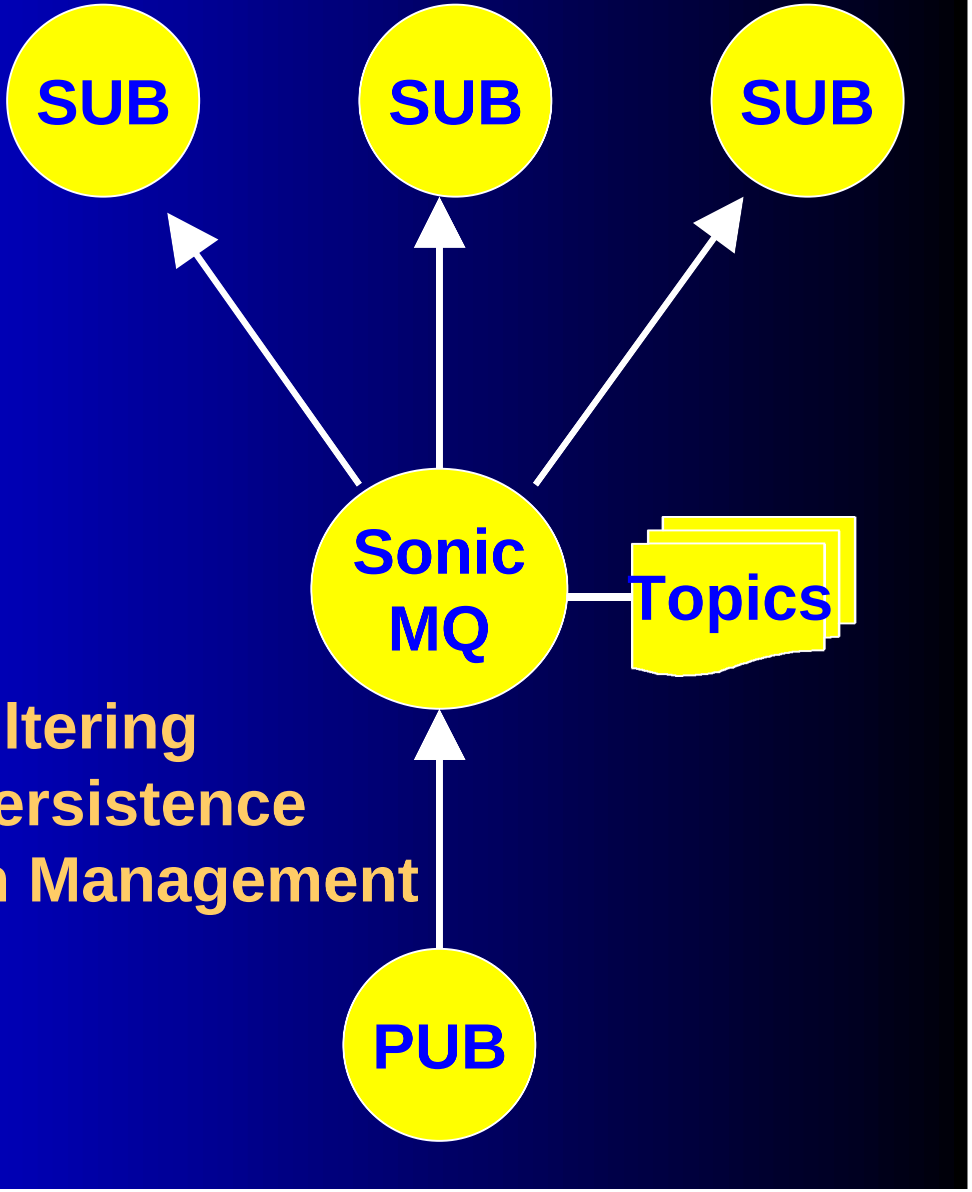
Support for the Topic Model

JMS API + Extensions

Hierarchy of Topics
partitioned in Domains
CMW.DEVICES
CMW.ALARM_SYSTEM
CMW.ADMIN

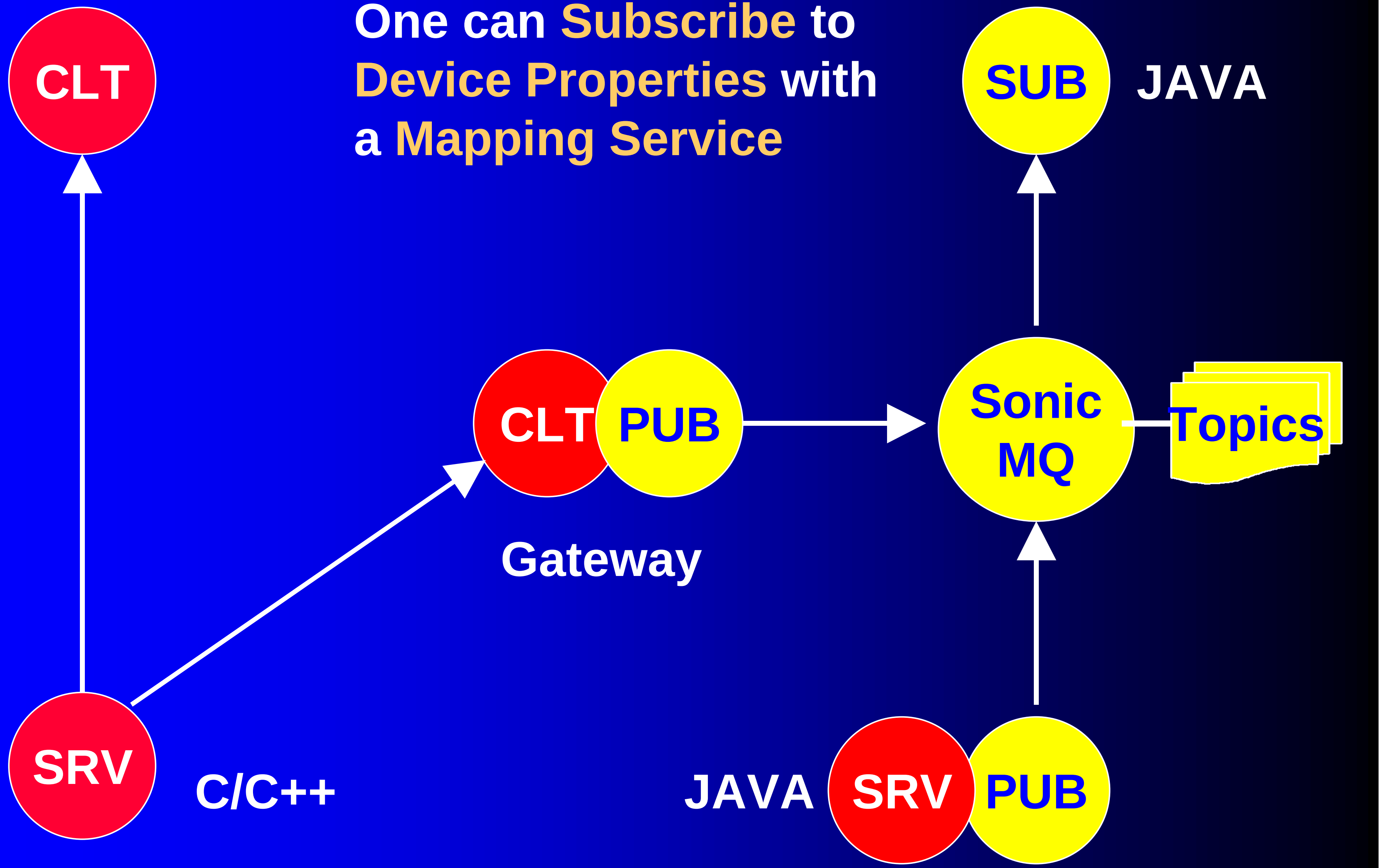
Message Types
Stream
Map
Text (XML)
Object
Bytes

Message Filtering
Message Persistence
Connection Management



Dev/Prop Subscription with MOM

One can Subscribe to Device Properties with a Mapping Service



Outline

Project Overview

Middleware Capabilities

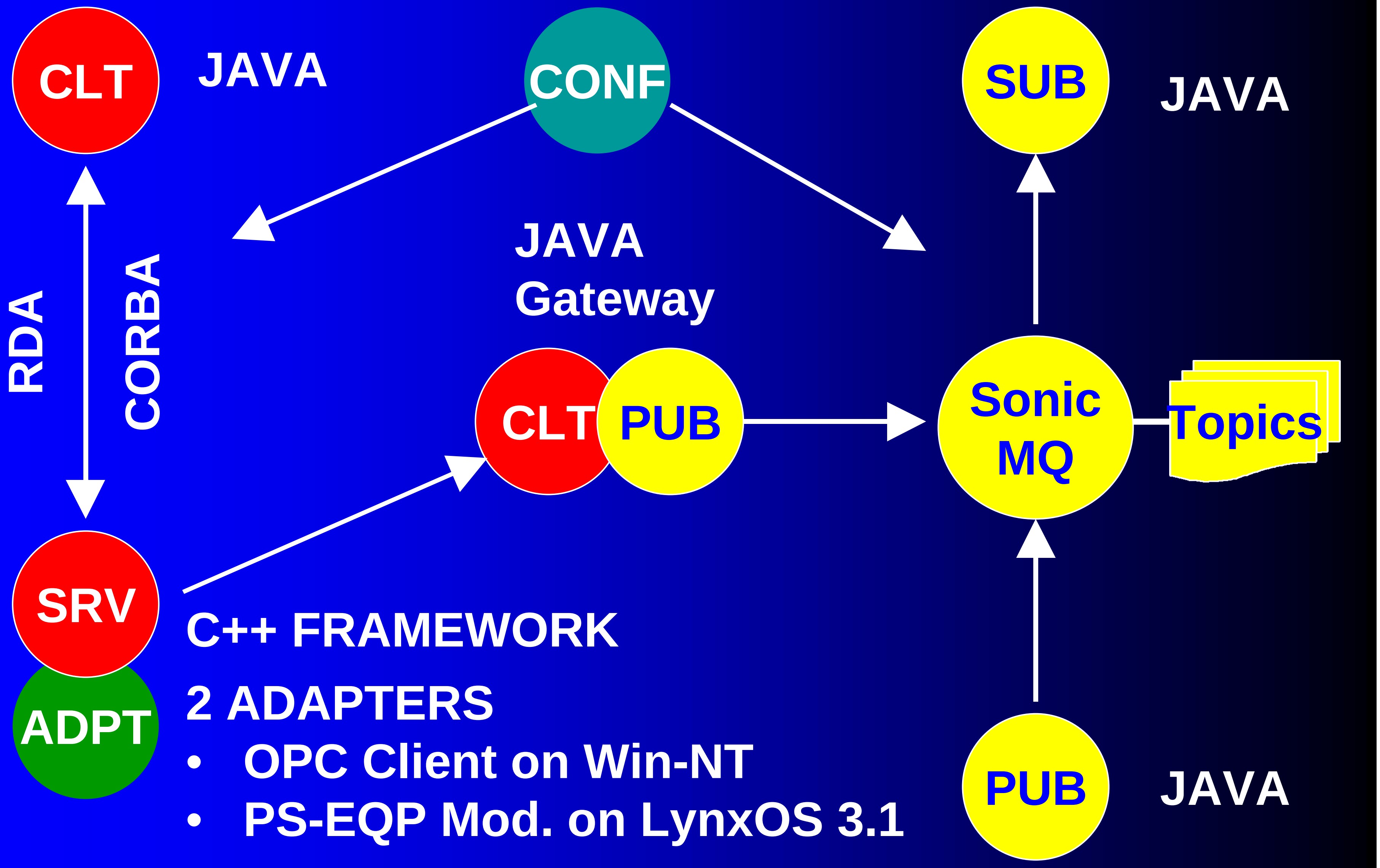
Adopted Technologies and Usage

Current & New Developments

Deployment

What Next?

What is available today?



C++ FRAMEWORK

2 ADAPTERS

- OPC Client on Win-NT
- PS-EQP Mod. on LynxOS 3.1

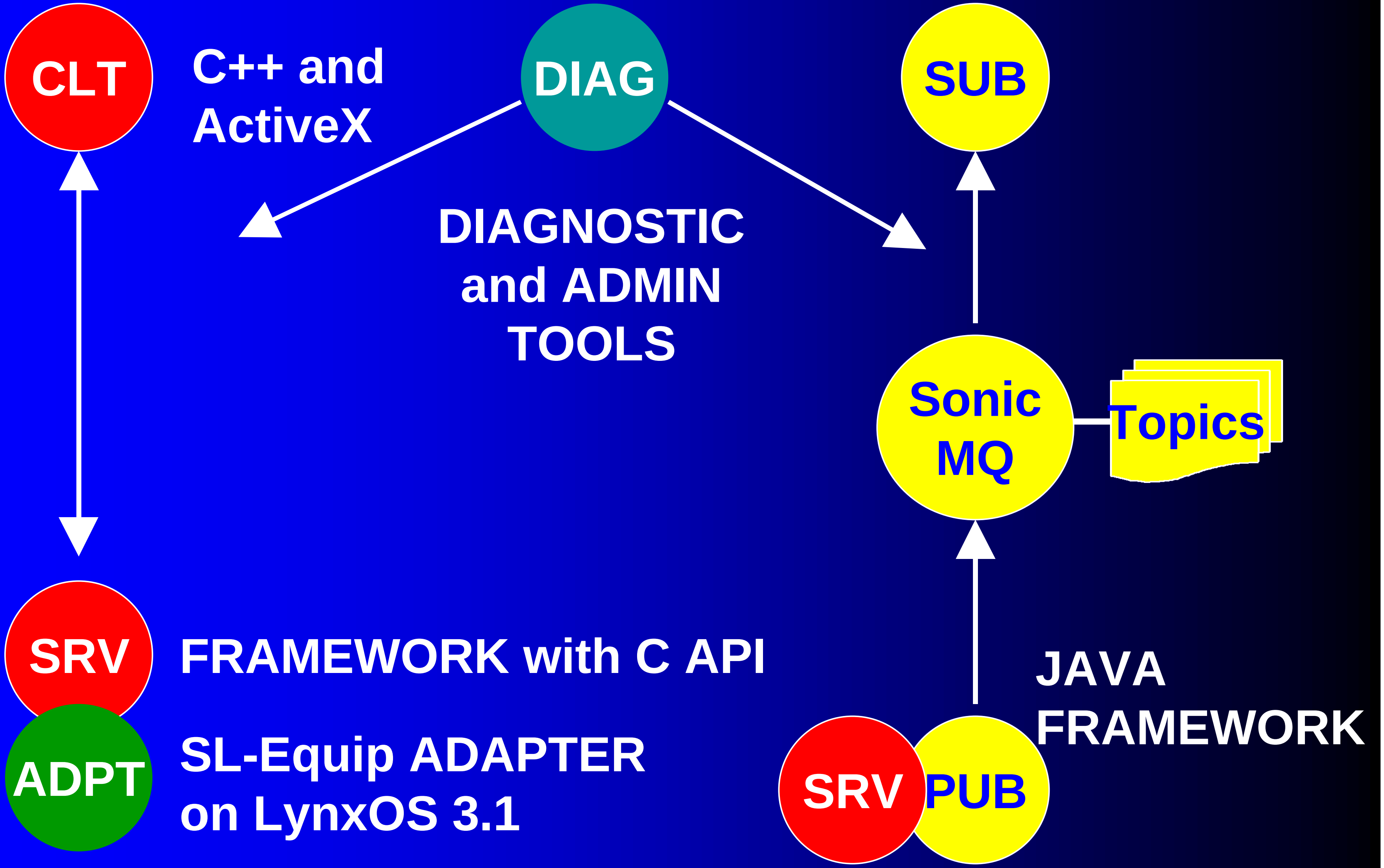
2001 CMW Developments

- SL-EQUIP adapter on LynxOS 3.1
- Integration Server Framework
  - C API
  - Java
- ActiveX Component on Windows platforms
  - Windows-UNIX Passerelle
- Diagnostic and Administration Tools

Further Activities

- Define a **Maintenance Plan**
- **Software User Manual**
- **Review of Requirements** for the Middleware
- **Add** More **Functionality**
  - Access Control
  - Reservation Facility
- **Review** of client & server **API** ‘s

2001 CMW Developments



CLT

C++ and ActiveX

DIAG

DIAGNOSTIC and ADMIN TOOLS

SUB

Sonic MQ

Topics

SRV

FRAMEWORK with C API

ADPT

SL-Equip ADAPTER on LynxOS 3.1

SRV PUB

JAVA FRAMEWORK

Outline

Project Overview

Middleware Capabilities

Adopted Technologies and Usage

Current & New Developments

Deployment

What Next?

CMW Deployment in PS

- **Deploy** the middleware **servers** on AD machine
  - Front-end computers running LynxOS 3.1
- **Connect JAVA** AD **programs** to the new middleware
- **Validate** the **infrastructure**
- **Deploy** on all PS accelerators

CMW Deployment in SL

- **CESAR Project**
  - Startup June 2001 with SL-Equip Adapter
  - Java Server Framework on Front-ends in Autumn 2001
- **LHC ALARM SERVICE**
  - Prototype of alarm service integration with Java Server Framework
  - Prototype of alarms gathering and distribution via TOPIC API
- **PASSERELLE Windows-UNIX**
  - ActiveX wrapper on CMW API

Outline

Project Overview

Middleware Capabilities

Adopted Technologies and Usage

Current & New Developments

Deployment

What Next?

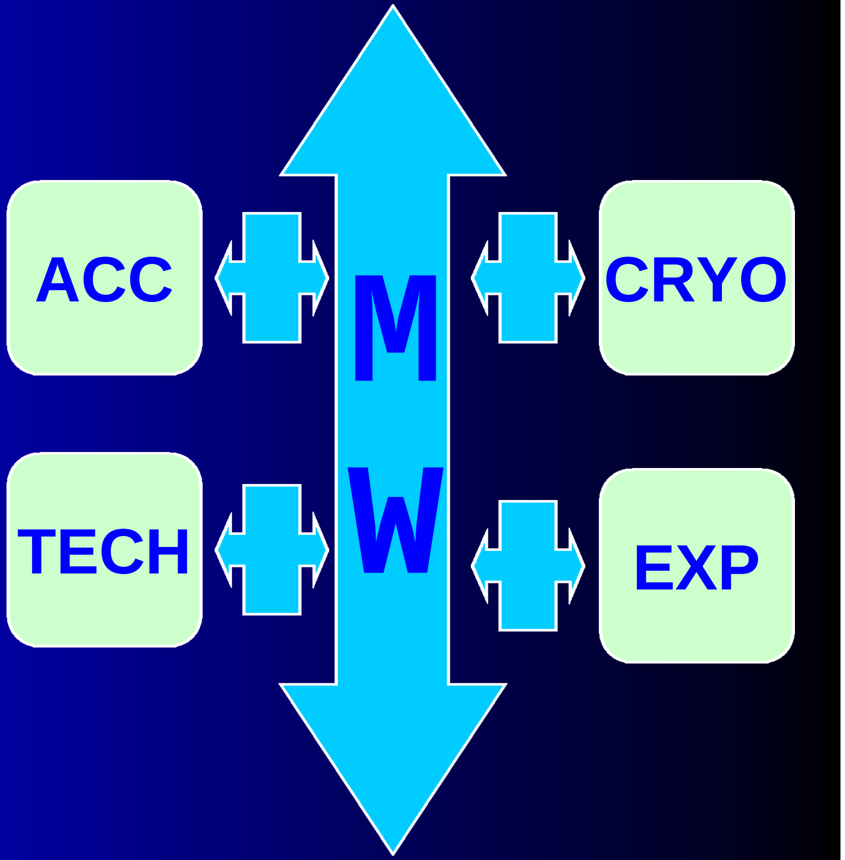
What does CMW offer ?

- Two communication models
- Strong support for Java and for standards
- Connection to both 'legacy' and industrial equipment (OPC)
- Integration framework for providing equipment servers

From MW Session on LHC-CP 2000

It outlined the need for:

- **Single Middleware** solution for each Domain
  - – Accelerator Domain
  - – Technical Infrastructure
  - – Experiments
  - – Cryogenics
- **Agreement on Interface** for Inter-Domain communication



CMW for inter-domain communication

CMW can be used for inter-domain communication

- The **Publish & Subscribe** communication paradigm, supported by the **TOPIC API**, it is ideal for **loosely coupled** domains

CMW for LHC Controls

CMW can play the role of Middleware for the Accelerator Domain

- The **decision** to adopt CMW in the whole Accelerator Domain can only be taken by the Accelerators **Control Groups** and **LHC-CP**
- **LHC-CP** should **evaluate** if **CMW** fulfills it's needs and if not request additions
- A new **organizational structure** should be put in place with the aim to review the CMW in the light of the LHC requirements

CMW Documentation

cern.ch/controls-middleware