EUROPEAN LABORATORY FOR PARTICLE PHYSICS
CERN – PS and SL DIVISIONS

CERN Note SL/99-16(CO)

CERN PS/SL Middleware Project

USER REQUIREMENTS DOCUMENT
PSSL-MW/URD/WORD/ISSUE 1/REVISION 3

Prepared by the PS/SL Middleware Project Team :

V.Bagiollini, F.Di Maio, K.Kostro, A.Risso, M.Vanden Eynden

Thank you to the following people for their contribution and comments :

G.Baribaud (SL/BI), A.Bland (SL/CO), E.Carlier (SL/BT), P.Charrue(SL/CO), E.Ciapala (SL/RF), N.DeMetz-Noblat (PS/CO), B.Denis (SL/CO), JJ.Gras (SL/BI), M.Jonker (SL/CO), R.Lauckner (SL/CO), J.Lewis (PS/CO), A. Marchand (SL/BT), F.Momal (LHC/IAS), P.Ribeiro (SL/CO), F.Weierud (SL/RF)

Abstract

The PS/SL Controls Convergence initiative launched in November 1998 *the "PS/SL Middleware project*" aimed at providing a modern common middleware communication infrastructure for the CERN Accelerator sector.
This paper introduces the context of the project and lists the functional and non-functional requirements of the middleware which were known at the date of publication of this document. This specification is a join submission from several members of the LHC, PS, SL and ST divisions.

This document will serve as a main reference, first in the process of choosing the middleware technology, and later on as a check list during the design and implementation phases.

DOCUMENT STATUS SHEET

DOCUMENT STATUS SHEET			
1. DOCUMENT TITLE: **CERN PS/SL Middleware Project** **User Requirements Document**			
2. DOCUMENT REFERENCE NUMBER: CERN **Note SL/99-**/PSSL-MW/URD			
3. ISSUE	4. REVISION	5. DATE	6. REASON FOR CHANGE
1	0	30/04/99	First issue
1	1	12/05/99	New and modified requirements
1	2	15/06/99	New SL/CO/FE requirements New SL/BI requirements New SL/BT requirements New LHC/IAE requirements New SL/RF requirements
1	3	02/08/99	Statements and qualifiers reviewed

DOCUMENT CHANGE RECORD

DOCUMENT CHANGE RECORD		DCR NO DATE ORIGINATOR APPROVED BY	1 6/08/99 M.VDE
1. DOCUMENT TITLE: CERN PS/SL Middleware Project User Requirements Document			
2. DOCUMENT REFERENCE NUMBER: CERN Note SL/99-116(CO)PSSL-			
3. DOCUMENT ISSUE/REVISION NUMBER: **1/3**			
4. PAGE	5. PARAGRAPH	6. REASON FOR CHANGE	
8	1.1	Formulation	
13	2.1.1	Formulation	
14	2.2	Formulation	
14	2.5	Formulation	
	Synchronous I/O service		
	SISO-001	New Statement	
	SIOS-003	New Statement and Qualifier	
	SIOS-004	New Qualifier	
	Asynchronous I/O service		
	AIOS-003	New Qualifier	
	AIOS-006	New Qualifier	
	Data Subscription service		
	DSS-001	New Statement	
	DSS-004	New Statement	
	DSS-005	Added	
	DSS-009	New Statement	
	DSS-012	New Statement	
	DSS-005 -> 015	New numbering !	
	Error Handling Service		
	EHS-001	New Qualifier	
	Directory Service		
	DS-001	New Statement	
	DS-002	New Statement	
	DS-003	New Statement	
	DS-004	Added	
	Access Control Service		
	ACS-001	New Statement and Qualifier	
	ACS-002	Removed	
	Reservation Service		
	RS-004	Added	
	InterOperability		

	IOP-001	New Statement
	IOP-002	Removed
	IOP-003	Is now IOP-002
	Scalability	
	SCA-001	New Statement
	SCA-002	New Statement
	Availability	
	AV-002	New Statement
	AV-004	New Statement
	AV-008	New Statement
	Performance	
	PERF-001	New Statement
	PERF-002	New Statement
	PERF-003	New Statement
	Administration	
	ADM-006	New Statement
	ADM-007	New Statement
	Maintenance	
	MAIN-002	New Statement
	MAIN-003	New Statement

TABLE OF CONTENTS

INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

This document is the reference requirements document for the PS/SL Accelerator Middleware Project. The contents of this document encompass the functional and constraints requirements coming from the following sources :

- The PS/SL Java API project [A3]
- The SPS-2001 software project
- The SPS EA Rejuvenation Project
- The PS and SL equipment and operations groups
- LHC and ST Controls entities requiring potential data exchange with the future Middleware

This baseline document will be used to choose the PS/SL Middleware technology and products, and to guide the design and integration according to the operational needs of both PS and SL divisions.
The intended readership of this document is the following :

- PS and SL equipment and operations groups. They can use this document to check if their requirements have been correctly captured
- Other CERN entities interested in the capabilities of the future PS/SL Middleware
- The PS/SL Middleware project team who will use this document as a baseline reference for the implementation of the Middleware

APPENDIX A provides a snapshot, as date of publication of this document, of the LHC global Real Time requirements. While communication protocols for LHC Real time (RT) closed-loop feedback systems fall out of the scope of the PS/SL Middleware project, all aspects of interfacing open-loop RT systems and providing mechanisms for application software to exchange data with those systems (e.g. RT knobs) will become an issue in a near future.

1.2 SCOPE OF THE SOFTWARE

The control of the CERN accelerator complex and the development of machine control application programs have shown the need for highly distributed communication software. Several significant but different investments were made in the PS and SL controls groups to address this issue.

The recent collaboration of the PS and SL controls groups in an Object Oriented Accelerator Device Model and Interface [A3] gives the opportunity to co-ordinate efforts to provide a common modern communication Middleware allowing the deployment of Object Oriented applications.

The purpose of the PS/SL Middleware project is [A2] :

"To provide a software communication architecture and services required to operate the PS/SL Accelerator complex, allowing, amongst others, inter-objects communications and supporting, as first priority, the CERN Java API technical specification, including the standard Accelerator Device Model."

The expected benefits of this software are :

- A common PS and SL OO distributed architecture based on modern technologies
- The full implementation of the PS/SL Java API technical specification, including all aspects related to data subscription
- A solid base for building PS and SL operational software projects in view of the LHC era
- A coherent software interface between the PS/SL Accelerator control system and other CERN systems, including industrial SCADA systems that will be used for the control of LHC sub-systems

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
DCR	Document Change Record
DSS	Document Status Sheet
ESA	European Space Agency
RID	Review Item Discrepancy
TBC	To be Confirmed
TBD	To be Defined
TOC	Table Of Contents
URD	User Requirements Document

1.4 REFERENCES

Two classes of references are detailed in this document:

- Applicable documents are considered to govern or form a part of this URD.
- Reference documents have a bearing on the content of the URD, but they do not govern or form a part of it.

1.4.1 Applicable documents

[A1] Software Engineering Standards, ESA Board for Software Standardisation and Control, 1994.

[A2] PS/SL Middleware Project – Launch Memorandum
CERN SL/CO Memorandum 98-41 November 25, 1998
PS/SL Controls Convergence Project Team

[A3] Java Controls API – Technical Specification
PS/CO Note 98-16 (Spec)
P.Charrue, J.Cuperus, I.Deloose, F.DiMaio, K.Kostro, M.Vanden Eynden, W.Watson

1.4.2 Reference documents

[R1] ESA PSS-05-02 Issue 1, Guide to the User Requirements Definition Phase, October 1991.

[R2] CERN PS/SL Controls Convergence Project – Project Definition Report
CERN PS Note 98-03 (CO), CERN SL Note 98-21 (CO)
P.Charrue, J.Cuperus, I.Deloose, F.DiMaio, K.Kostro, M.Vanden Eynden, W.Watson

1.5 OVERVIEW OF THE DOCUMENT

The document is structured as follows:

Chapter 2 describes the general factors that affect the product and its requirements. It does not state specific requirements; it only makes those requirements easier to understand.

Chapter 3 describes each of the software requirements following the PSS-05 guidelines[R1].

User Requirement Structure

Each explicit user requirement contains fields for the following information:

- Identifier
- Description
- Source
- Need
- Rank
- Stability
- Type

Identifier

Each identifier takes the basic form:

<ident>-<nr>

where **<ident>** is an abbreviation for one part or functionality of the system, and **<nr>** is a sequential number starting with 001. The list of part names is as follows:

AAA- *List your parts here*

UR insertions are made by appending a new level number (where necessary)_

<ident>-<nr>-**<new-nr>**

where **<new-nr>** is a new number sequence starting with 01, for each 'old number.

Description

This provides a natural language statement of the requirement. It is intended that all such statements are *clear* and *verifiable*.

Source

Identifies the origin of the requirement. Sources are made in respect of the applicable and reference documents list.

Need

This identifies whether a requirement is essential for the operation of the system or desirable (and consequently negotiable). The field therefore takes the values 'ESS' or 'DES'.

Rank

Identifies the priority of the requirement, in case incremental development is adopted. It takes a numeric value. One refers to the highest priority.

For the most part, ESS requirements are of either rank 1 or 2, whereas additional DES requirements are of rank 3.

Stability

Stability identifies the perceived status of the source information at the time of publication. It takes the values 'STABLE' or 'TBC'

Type

Refers to the requirement category. The values of this field are an abbreviation of the full name of the category:

Value	Category
FUNC	Functional
PERF	Performance
I/F	Interface
OPER	Operational
RES	Resource
VER	Verification
A/T	Acceptance testing
DOC	Documentation
SEC	Security
PORT	Portability
QUAL	Quality
REL	Reliability
MAINT	Maintenance
SAFETY	Safety

2. GENERAL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

As shown in Fig.1, the PS/SL Middleware software will be a major building block of the future common PS and SL Accelerator controls infrastructure currently under development in the scope of the PS/SL Controls Convergence Project [R2].

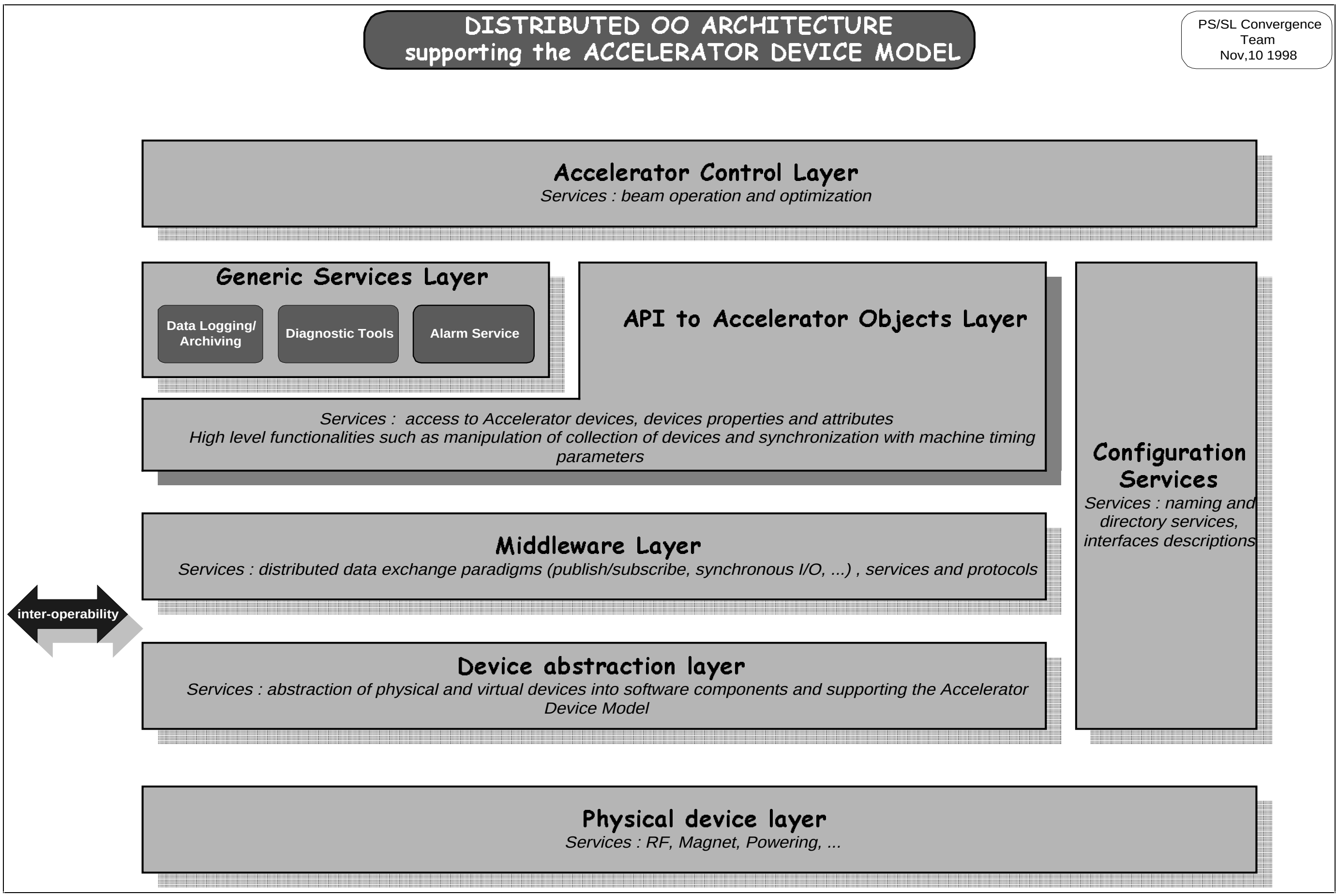

Fig.1 – PS/SL Distributed OO Architecture

Two important concepts will drive the choice of the Middleware software :

- The definition of a common PS/SL Accelerator Device Model which defines how Accelerator components will be viewed and accessed from the software application level.

- The technical specification of a common PS/SL Java Application Programming Interface (API). This specification, built on top of the Accelerator Device Model, defines a large set of functionalities like synchronous I/O operations, data subscription capabilities, introspection features, etc.

The PS/SL Middleware software will deliver a distributed controls architecture that will support this Accelerator Device Model and the technical specification of the Java API.

2.1.1 The CERN PS/SL Accelerator Device Model

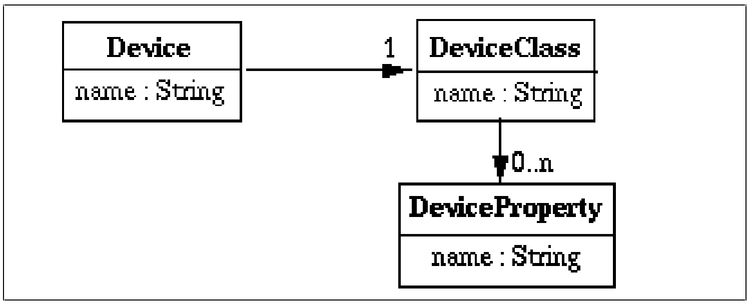

Fig.2 – The Accelerator Device Model (partial view)

As shown in Fig.2, this model provides a device-oriented view of the control system :

- The System consists of **named devices**
- A Device may represent a physical Device (e.g. a power converter or a vacuum valve), a composite device (e.g. a transfer line) or a more abstract entity related to controls (e.g. a beam process or a machine access request)
- The Devices are organised into **classes** that describe the Device Interface
- The Device Interface includes **get/set methods** that operate on named **properties**
- Properties that support the get/set operations also support the **subscribe/unsubscribe** operations
- Properties may have **characteristics** such as units, resolution, etc

2.2 GENERAL CAPABILITIES

The Middleware software architecture shall provide homogeneous functionality for implementing all communications functions between the equipment and the application programs. From a technical point of view, this project must provide a Middleware solution supporting the complete Accelerator Device I/O model of the CERN Java API technical specification, in particular:

- the synchronous and asynchronous I/O operations on Accelerator device data
- the distribution of Accelerator device data to client programs, based on the publish/subscribe paradigm

- the synchronisation of I/O interactions with Accelerator timing information
- API's, tools, run-time support and guidelines for integrating physical and virtual devices to the Middleware. This connection regards the server side (e.g. on the front-end computers) of the Middleware-based client/server architecture
- the access to general purpose services (e.g. Data logging, Alarms, persistence, etc)

2.3 GENERAL CONSTRAINTS

The following technical constraints shall be considered:

- the first priority of the software is to support the Accelerator Device Model and Device I/O services defined in the CERN Java API Technical Specification
- The specification and development of protocols must rely as much as possible on available standards

2.4 USER CHARACTERISTICS

As of date of publication of this document, the PS/SL Middleware will be used by the following entities :

- The PS and SL equipment groups who will use the Middleware at the server level to publish information about their equipment devices and at the client level for the development of specialist and operational programs
- The SPS-2001 project which will use the Middleware and the Java API layers to deploy the new generation of operational software needed to operate the SPS Accelerator as LHC injector. In particular, this project will define, on top of these layers, sets of standard interfaces for all SPS-2001 compliant equipment devices. Aspects like data subscription (publish/subscribe paradigm) are of first interest for this project
- The PS Controls Group in charge of the production of the operational software for the PS Accelerator complex
- Any CERN external entity interested in :
  - making proprietary data available to the PS/SL Accelerator control system
  - acquiring data from the PS/SL Accelerator control system
- Any commercial company in charge of connecting outsourced software systems to the PS/SL control system

2.5 OPERATIONAL ENVIRONMENT

Interoperability with the following systems is of major importance :

- Industrial control systems (e.g. OPC-based devices and SCADA systems)
- Real-Time systems developed for the LHC machine
- Existing data exchange and communication infrastructures at CERN (such as the ST Technical Data Server)

2.6 ASSUMPTIONS AND DEPENDENCIES

According to the criticality of the Middleware software, it is assumed that the existing PS and SL protocols and software components (e.g. SL-EQUIP, PS-EQP, …) will be maintained until the final Middleware has been deployed and declared operational in agreement with all concerned PS and SL equipment groups.

The Middleware has a strong dependence on the technical specification of the PS/SL Java API. As a result of this :

- Any change in the Java API specification shall be kept into account in the current document
- Some functional or non-functional requirements captured in the scope of Middleware project may propagate up to the Java API level and require modifications or extensions to the Java API Technical specification.

3. SPECIFIC REQUIREMENTS

3.1 CAPABILITY REQUIREMENTS

3.1.1 Synchronous I/O Service

UR number	Description	Source	Need	Rank	Stability	Type
SIOS-001	Client applications shall have the possibility, through a blocking synchronous call, to get and set the value of any Accelerator device property	Java API I/O general	ESS	1	STABLE	FUNC
SIOS-002	Client applications shall have the possibility to filter the information retrieved with a synchronous "get" command	SPS-2001 SL/BI (JJ.Gras) SL/BT (E.Carlier, A. Marchand)	ESS ESS ESS	1 1 1	STABLE STABLE STABLE	FUNC FUNC FUNC
SIOS-003	Client applications shall be able to specify a timeout (in milliseconds) after which a blocking synchronous device I/O operation is abandoned.	Java API IOTO-1	ESS	1	STABLE	FUNC

UR number	Description	Source	Need	Rank	Stability	Type
	This timeout information shall be passed to the server side.					
SIOS-004	Client applications shall have the possibility to specify the Accelerator Cycle Type or Timing Event for which a particular blocking synchronous I/O operation will apply	Java API IOTP-2	ESS	1	STABLE	FUNC
SIOS-005	Client applications shall have the possibility, through a synchronous call, to get and set the values of an array of device properties sharing the same interface (e.g. devices belonging or inheriting from the same class)	Java API IOAO-01 IOAO-04	ESS	1	STABLE	FUNC

3.1.2 Asynchronous I/O Service

UR number	Description	Source	Need	Rank	Stability	Type
AIOS-001	For each Device Property, client applications shall be able to initiate a get or set operation without waiting for completion (non-blocking mode)	Java API IOAS-1	ESS	1	STABLE	FUNC
AIOS-002	Client applications shall have the possibility to filter the information retrieved with an asynchronous “get” command	SPS-2001 SL/BI (JJ.Gras) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
AIOS-002	For any asynchronous get/set operation, a value or error information shall always be delivered to the client application	Java API IOAS-3	ESS	1	STABLE	FUNC
AIOS-003	The Middleware shall be able to accept and process (optimise) lists of get/set requests from client applications to server applications	Java API IOAS-2	DES	3	STABLE	FUNC
AIOS-004	Client applications shall be able to wait for the completion of an asynchronous get/set operation or to poll its status (e.g. ok, pending, error)	Java API IOAS-4	ESS	1	STABLE	FUNC
AIOS-005	Client applications shall be able to wait for the completion of a group of asynchronous get/set operations or to poll its status (e.g. ok, pending, error)	Java API IOAS-5	ESS	1	STABLE	FUNC
AIOS-006	Client applications shall have the possibility to specify the Accelerator Cycle Type or Timing Event for which a particular asynchronous get/set operation will apply (see also SIOS 004)	Java API IOAS-7	ESS	1	STABLE	FUNC

3.1.3 Data Subscription Service

UR number	Description	Source	Need	Rank	Stability	Type
DSS-001	Client applications shall have the possibility to "subscribe" to the value of each Device Property	Java API IODS-1	ESS	1	STABLE	FUNC
DSS-002	Client applications shall have the possibility to filter the information retrieved with a "subscribe" command	SPS-2001 SL/BI (JJ.Gras) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
DSS-003	Each value returned to a subscriber application shall be marked with a timestamp and optionally a beam stamp as specified in SDT-002 and SDT-004	Java API IODS-7	ESS	1	STABLE	FUNC
DSS-004	Client applications shall always be informed in case of problem in the Middleware subscription service (e.g. publisher failure, communication problem, etc)	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
DSS-005	Client applications shall be notified each time the value of a Device Property it has subscribed to is becoming unavailable E.g. virtual device not existing, MIL-1553 Bus problem, etc	SPS-2001 Java API IODS-8	ESS	1	TBC	FUNC
DSS-006	Client applications shall be notified each time the value of a Device Property it has subscribed to changing state from unavailable to available	SPS-2001	DES	3	STABLE	FUNC
DSS-007	Client applications shall be able to associate the rate of a subscription with an Accelerator Timing System Event	Java API IODS-2	ESS	1	STABLE	FUNC

UR number	Description	Source	Need	Rank	Stability	Type
DSS-008	Client applications shall be able to specify the maximum rate of a subscription with an absolute time interval	Java API IODS-2	ESS	1	STABLE	FUNC
DSS-009	Client applications shall be able to specify a subscription "on-change" only. In this case, a new Device Property value will only be delivered to the client if this value differs from the last value sent to the client by more than a determined deadband (expressed in % or absolute value)	Java API IODS-4 SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
DSS-010	Client applications shall be able to combine an "on-change" subscription with a guaranteed minimum rate at which data shall be returned	PS/SL Middleware SPS-2001 (MJ)	ESS	1	STABLE	FUNC
DSS-011	Any "on-change" subscription on a continuous Device Property shall use the resolution characteristic of this property	Java API IODS-5	ESS	1	STABLE	FUNC
DSS-012	A client application subscribing to a Property belonging to a Device not available at subscription time shall be notified	SPS-2001 SL/BT (E.Carlier, A. Marchand)	ESS	1	TBC	FUNC
DSS-013	A client application shall have the possibility to subscribe to a Device Property, even when it is temporarily unavailable	SPS-2001 SL/BT (E.Carlier, A. Marchand)	ESS	1	TBC	FUNC
DSS-014	Client applications shall be able to subscribe also to pre-defined "subjects" or "subscription groups"	SL/CO/AL (M.Tyrrell)	ESS	1	STABLE	FUNC
DSS-015	Client application shall be able to discover the list of pre-defined subscription subjects	SL/CO/AL (M.Tyrrell)	ESS	1	STABLE	FUNC
DSS-016	Client applications shall be able to discover the contents (list of devices and properties) of any subscription subject	SL/CO/AL (M.Tyrrell)	ESS	1	STABLE	FUNC

3.1.4 Error Handling Service

UR number	Description	Source	Need	Rank	Stability	Type
EHS-001	Client applications shall be informed about errors in any synchronous, asynchronous or subscription operation. This information shall at least be composed of : An error code identifying the kind of error An textual error message containing dynamic data related to the instance of this error	Java API IOCS-1	ESS	1	STABLE	FUNC
EHS-002	The Middleware should offer the possibility to organise errors into different categories	Java API IOCS-3	DES	3	STABLE	FUNC

3.1.5 Directory Service

UR number	Description	Source	Need	Rank	Stability	Type
DS-001	The Middleware shall provide introspection services to both client and server applications, allowing them to discover : - the properties (names, types, read/write access rights) of a given device class - the characteristics (units, resolution) of a given property	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
DS-002	Middleware applications shall be able to detect if additional parameters can be (or have to be) specified for a get/set request. Additional parameters comprise the following : - An Accelerator Cycle Type (String or Integer) - An Accelerator Timing Event Identifier (String or Integer) - A timeout value (in milliseconds)	Java API IOTP-1	ESS	1	TBC	FUNC
DS-003	The Middleware shall provide the possibility to dynamically create and destroy instances of virtual devices and to declare their existence/destruction to the Middleware directory service	SPS-2001	ESS	1	STABLE	FUNC
DS-004	Adding new Accelerator Devices to the MW must be possible without interrupting existing MW connections and on-going interactions	SL/CO/FE(A.Bland , P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
DS-005	Client applications shall be able to find device instances in the system. For this, clients shall be allowed to query device instances by specifying any of the following criteria and combinations thereof :	SL/CO/EA (V.Baggiollini) SPS-2001	ESS	1	STABLE	FUNC

UR number	Description	Source	Need	Rank	Stability	Type
	• symbolic device name (e.g. "MAGEA01") • substring of a device name (all devices containing "EA" in their symbolic name) • Device types (classes) (e.g. all devices of type "Magnet") • Geographical areas in which devices are located (e.g. all devices in a given tranfer line)					

3.1.6 Reservation Service

UR number	Description	Source	Need	Rank	Stability	Type
RS-001	The Middleware shall provide a Device Reservation mechanism by which client applications could reserve a Device in order to prevent access to this Device from other client applications (i.e. switching ON SL/RF equipment requires a sequencing of ~20 actions free of perturbation)	SL-RF (E.Ciapala, F.Weierud) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
RS-002	The Middleware shall provide information that allows owners of active Device reservations to be identified (client name and location) possibly with the help of other information sources (e.g. Users DB)	PS/CO SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
RS-003	The MW shall allow reservations of Devices to be unlocked at run-time	PS/CO	ESS	1	STABLE	SEC
RS-004	Reservations shall be based on a mechanism that prevents "forgotten reservations", i.e., reservations that remain active even if they are not needed anymore.	SL/CO/EA (V.Baggiollini)	ESS	1	STABLE	SEC

3.1.7 Access Control Service

	Description	Source	Need	Rank	Stability	Type
ACS-001	The Middleware shall provide an access control mechanism for granting and denying R/W access to Device Properties based on clients privileges	SL-RF (E.Ciapala, F.Weierud) Java API IOAC SL/CO/EA (V.Baggiollini) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	SEC
ACS-002	Middleware server applications shall be able to identify any incoming client application request by name and location (i.e. to allow servers to eventually develop their own access control mechanisms)	SL/RF (E.Ciapala, F.Weierud)	ESS	1	STABLE	SEC

3.1.8 Supported Data Types

UR number	Description	Source	Need	Rank	Stability	Type
SDT-001	The system shall allow the transport of the following basic data types : Byte, short, int, long (8,16,32,64 bit signed) Float, double (32, 64 IEEE floating point numbers) String (8 bits characters)	Java API IODT-1 IODT-2(a)	ESS	1	STABLE	FUNC
	Muti-dimentional matrices of these 7 basic data types	PS/CO	ESS	1	TBC	FUNC
SDT-002	The Middleware shall be able to transport data marked with a POSIX standard timestamp (number of seconds since Jan 1, 1970) having a resolution of at least milliseconds	Java API IOTS-1 IOTS-2	ESS	1	STABLE	FUNC
SDT-003	The Middleware shall be able to produce timestamps at the server side conforming to the SDT-002 format	PS/SL Middleware	ESS	1	STABLE	FUNC
SDT-004	The Middleware shall be able to transport data marked with a beam stamp composed of : - a cycle stamp unique for a minimum of 24h - an optional event identifier (32 bits int) - an optional cycle time (32 bits int) representing the time in milliseconds since the beginning of the cycle identified by the cycle stamp	Java API IOBS-2	ESS	1	TBC	FUNC

3.2 CONSTRAINT REQUIREMENTS

3.2.1 Platform, Network and Language

UR number	Description	Source	Need	Rank	Stability	Type
PNL-001	Client and Server interfaces to the Middleware shall be provided for the following platforms : - HP-UX 10.20 and higher (ESS), AIX 4.2 and higher (ESS), PowerPC LynxOS 2.5.1 and higher (ESS), Windows NT 4.0 and higher (ESS), PC LynxOS 2.5.1 and higher (DES), M68k LynxOS 2.5.1 and higher (DES), LINUX 5.0 and higher (DES)	PS/SL Convergence SL/BT (E.Carlier, A. Marchand) SL/CO/FE (A.Bland, P.Ribeiro)	ESS	1	STABLE	I/F
PNL-002	Both Client and Server interfaces to the Middleware shall be provided for the C, C++ and Java languages and for at least one scripting language	PS/SL Middleware SL/CO/FE (A.Bland, P.Ribeiro) SL/CO/EA (V.Baggiolini) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	I/F
PNL-003	Any process shall have the possibility to be at the same time client and server of the Middleware	SL/BT (E.Carlier, A. Marchand) SL/BI (JJ.Gras,	ESS	1	STABLE	I/F

UR number	Description	Source	Need	Rank	Stability	Type
		G.Baribaud)				
PNL-004	The Middleware shall run on the present PS/SL 10 BaseT Ethernet Network	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	I/F

3.2.2 Interoperability

UR number	Description	Source	Need	Rank	Stability	Type
IOP-001	The Middleware shall be able to be client of industrial SCADA systems offering an OPC server interface version 2.0 or higher	LHC/IAS (F.Momal) SL/BT (E.Carlier)	ESS	1	STABLE	FUNC
IOP-002	The Middleware shall offer a mapping between the Accelerator Device Model and the corresponding OPC data structures (groups and items)	LHC/IAS (F.Momal) SL/BT (E.Carlier)	ESS	1	STABLE	FUNC

3.2.3 Object Orientation Paradigm

UR number	Description	Source	Need	Rank	Stability	Type
OO-001	The Middleware shall allow invocation of methods of remote objects	PS/SL Convergence	ESS	1	STABLE	FUNC
OO-002	The Middleware shall support the OO paradigm and avoid as much as possible intermediate mappings and translations to other paradigms	PS/SL Middleware	ESS	1	STABLE	PERF
OO-003	The Middleware shall provide the same access capabilities to both local and remote objects	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC

3.2.4 Scalability

UR number	Description	Source	Need	Rank	Stability	Type
SCA- 001	The Middleware shall NOT impose any hard limits concerning the number of computers, operating system processes, or device instances connected to it. Anticipate figures for LHC era : 2000 servers and 500 clients processes Present figures @ PS : 150 Front-End computers (~120 in operation), 120 AIX workstations, 10 PCs used for controls Present figures @ SL : 210 LynxOS Front-End computers, 200 OS/9 Front-End computers, 50 HP-UX workstations	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	OPER

UR number	Description	Source	Need	Rank	Stability	Type
SCA-002	The MW infrastructure shall be able to manage at least 50000 active device instances PS figures : 15000 active device instances (PS-EQP DB) SL figures : 35000 active device instances (SL-EQUIP DB)	PS/CO (A.Risso) SL/CO (A.Bland) SL/BT (E.Carlier, A. Marchand)	ESS	1	TBC	FUNC

3.2.5 Availability (Hot backup, Redundancy, Server Relocation)

UR number	Description	Source	Need	Rank	Stability	Type
AV-001	The MW shall support redundant naming services	PS/CO (A.Risso) SL/BT (E.Carlier, A. Marchand)	ESS	1	TBC	REL
AV-002	The MW shall offer mechanisms allowing a server process to archive an image of devices information (i.e. save it to persistent storage). It shall be possible to select the information to be archived	PS/CO (A.Risso) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	REL
AV-003	The MW shall offer mechanisms allowing a server process to restore an archived image of devices information it is responsible for (i.e. to check and re-synchronise HW with required settings after a server crash/restart)	PS/CO (A.Risso) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	REL
AV-004	A subscriber crash/restart shall be transparent to the publishers.	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	REL
AV-005	A publisher crash/restart shall be transparent to the subscribers. In case of publisher crash/restart, the Middleware is responsible for re-establishing the subscription between the publisher and its associated	PS/SL Middleware SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	REL

UR number	Description	Source	Need	Rank	Stability	Type
	subscribers. The subscribers shall not be involved in this recovery process					
AV-006	It shall be possible to shutdown and restart the MW without stopping or crashing server-side users processes (delaying them is acceptable)	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
AV-007	Restarting the MW system processes on a given machine shall not take longer than 30 seconds	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	DES	2	TBC	OPER
AV-008	Adding new Accelerator Devices Instances to the MW directory service shall be effective within a maximum time of 1 minute	SL/BI (JJ.Gras)	DES	2	STABLE	

3.2.6 Reliability

UR number	Description	Source	Need	Rank	Stability	Type
REL-001	User processes shall not be in a position to crash or deadlock a MW system process)	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	REL
REL-002	Any MW client/server transaction shall not be blocked or affected by other MW client/server transactions	SL/CO/FE SL/BT (E.Carlier, A. Marchand)	ESS	2	STABLE	REL
REL-003	The MW shall provide timeout mechanisms preventing client or server applications to enter deadlock situations when communicating via the MW	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	REL

3.2.7 Integration in Existing Infrastructure

UR number	Description	Source	Need	Rank	Stability	Type
INT-001	The Middleware shall be able to transport groups of get/set commands towards the same server application	Java API IOAO-03	ESS	1	STABLE	I/F PERF
INT-002	The Middleware shall detect and exploit existing PS server capabilities for handling arrays of get/set operations on device properties	Java API IOAO-03	ESS	1	STABLE	I/F
INT-003	It shall be possible to install the MW in the server front-ends in parallel with the old communication infrastructure (SL-EQUIP mhm @ SL)	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	OPER
INT-004	The Middleware has to fit into the available memory on the front-ends : @ SL : total front-end Memory : 32Mbytes MW available memory : 3 Mbytes (worst case)	SL/CO/FE (A.Band, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	RES
INT-005	Communications mechanism shall be provided between the SL OS/9 front-end platforms and the MW	SL/CO/FE (A.Bland, P.Ribeiro)	DES	2	TBC	I/F

3.2.8 Performance

UR number	Description	Source	Need	Rank	Stability	Type
PERF-001	When a single client carries out an I/O request to an array of device properties, the MW shall group the requests, so that only a single I/O interaction is carried out with each device server	PS/CO (A.Risso)	ESS	1	STABLE	PERF
PERF-002	In normal operational conditions (normal load on the network and on the involved computers), the MW subscription service shall guarantee a delivery time of less than 10 msec, measured from the publisher application to the subscriber application. (required to transport machine timing information).	PS/CO (J.Lewis) SL/CO (M.VDE) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	PERF
PERF-003	The Middleware shall support the transmission over the network of 6000 property values belonging to 1500 devices every 3~4 seconds (Alarm program behaviour @ PS)	PS/CO (A.Risso)	ESS	1	TBC	PERF

3.2.9 Administration (tools, diagnostics, reboots, …)

UR number	Description	Source	Need	Rank	Stability	Type
ADM-001	The MW shall allow to activate/deactivate at run-time a human readable logging (ASCII) of the information regarding the client applications accessing a server (all accesses or errors only) : - identity of the program initiating an interaction (host + program id like process name and/or pid and uid) - identity of the program being contacted - interaction start time and duration - success or failure of the interaction	PS/CO (A.Risso) SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
ADM-002	The MW shall offer administration mechanisms for grouping and administrating servers in categories (i.e. finding all servers for the AD machine, sending commands to all servers of a given category)	PS/CO (N.DeMetz Noblat)	ESS	1	TBC	FUNC
ADM-003	The MW shall offer a command line interface for exercising and debugging access to device properties in synchronous, asynchronous and subscription modes	PS/SL Middleware SL/BT (E.Carlier, A. Marchand) SL/CO/FE (A.Bland)	ESS	1	STABLE	FUNC
ADM-004	Tools shall be provided for editing the contents of the Middleware directory service (Add/remove/Modify Device Definitions)	SPS-2001	ESS	1	STABLE	FUNC
ADM-005	The MW shall offer tools for creating and administrating subscription subjects in a hierarchical way	SL/CO/AL	ESS	1	STABLE	FUNC

UR number	Description	Source	Need	Rank	Stability	Type
		(M.Tyrrell)				
ADM-006	The following figures shall be provided regarding the MW system processes in order to integrate the surveillance of the MW into the current SL monitoring tools (clic) : - max size of each process - max number of instances of each process - max file descriptors used - etc	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC
ADM-007	The MW shall provide counters (with 200 days lifetime) that describe the system behaviour in a statistical way : number of (un)successful requests, maximum rate of requests, number of active clients, total number of clients	SL/CO/FE (A.Bland, P.Ribeiro)	ESS	1	TBC	FUNC
ADM-008	It shall be possible to include, in an orderly manner, MW start and shutdown commands in any machine reboot procedure	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	FUNC

3.2.10 Maintainability

UR number	Description	Source	Need	Rank	Stability	Type
MAIN-001	CERN shall have access to the Middleware Source Code. TBD in the future commercial contract.	PS/CO (N.Demetz Noblat)	DES	2	TBC	MAIN
MAIN-002	Diagnostics and corrective intervention support must be provided during the whole life time of the MW, both for off-the-shelf software and house-made software	PS/SL Middleware	ESS	1	STABLE	MAIN
MAIN-003	On-site CERN competence must be available for 1st level problem solving	SL/CO/FE (A.Bland, P.Ribeiro) SL/BT (E.Carlier, A. Marchand)	ESS	1	STABLE	MAIN
MAIN-004	On-site competence must be backed-up by a corrective maintenance contract that can react within 8 working hours	SL/CO/FE (A.Bland, P.Ribeiro)	ESS	1	STABLE	MAIN

3.2.11 Ease of Use

UR number	Description	Source	Need	Rank	Stability	Type
EOU-001	Server-side skeletons must be provided as a starting point for users to integrate their server software with the Middleware.	SL/CO/AL (M.Tyrrell) SL/CO/FE (A.Bland, P.Ribeiro)	DES	2	STABLE	DOC

4. APPENDIX A : LHC GLOBAL REAL TIME REQUIREMENTS

Communications Performance to meet LHC Global Real Time Requirements.

The purpose of this note is to establish a baseline for the required performance of the LHC communication system in order to fulfil the global real time communication requirements for beam control. These facilities are intended to supplement the non time critical communications, which would rely on similar mechanisms used for accelerator control at CERN today. The estimates are based on the user requirements which have been expressed, often from discussions in the Dynamic Effects Working Group. They are not aimed at the performance of any particular communication technology and are based on the end-users needs rather than the performance of the control system and it's components. This baseline does not include local links, only information to be carried on the site wide communication system.

The following processes must run concurrently:

1. Orbit Correction

Purpose:
Provide closed loop orbit correction during injection and ramping
Description:
For each beam collect data from 500 pickups, compute correction, and send results to 500 power converters.
Data:
Each pickup: 1 ID, 3 properties, 2 positions, and 1 intensity, tag
Each corrector: 1 ID, 1 property, and 1 current

Rate: 1 Hz
Jitter: 100 ms
To Control Room: 2 rings * 500 monitors * 8 variables = 8000 variables
From Control Room: 2 rings * 500 correctors * 3 variables = 3000 variables
Notes: There are about 500 orbit monitors but these may be grouped before the data arrives on the site wide link. There are also about 500 orbit correctors but depending on the algorithm

chosen to correct the orbit the number of results to be loaded could be reduced to about 10. The rate of 1 Hz was taken from DEWG discussions on the evolution of the orbit errors during the start of the ramp.

2. Orbit Display

Purpose:
Provide permanent real time orbit display in the control room
Description:
For each beam collect data from 500 pickups, display
Data:
Each pickup: 1 ID, 3 properties, 2 positions, and 1 intensity tag
Rate: 1 Hz
Jitter: 100 ms
To Control Room: Uses correction data
From Control Room: None
Notes: There are about 500 orbit monitors but these may be grouped before the data arrives on the site wide link. The rate is chosen to meet the operator’s ability to watch the orbit evolve and because it fits with the orbit correction requirement. The jitter is a guess at the operator’s perception threshold and is the same as orbit correction requirement.

3. Beam Loss Display

Purpose:
Provide whole ring beam loss display in the control room
Description:
For each beam collect data from 500 pickups, display in control room
Data:
Each pickup: ID, 2 properties, dose, intensity, tag
Rate: 1 HZ
Jitter: 100 ms
To Control Room: 2 rings * 500 monitors * 6 variables = 6000 variables
From Control Room: None
Notes: There may be more than 500 loss monitors so the volume may be more. Display rate and jitter are chosen to be the same as the orbit display.

4. Beam Property Display

Purpose:
Provide tuning information displays in the control room such as intensity, luminosity, beam profile, bunch distributions.
Description:
For 20 diverse sources collect information from a single source, display in the control room
Data:
Each source: ID, 1 property, 250 data values
Rate: 10 Hz
Jitter: 10 ms
To Control Room: 20 sources * 250 variables = 5000 variables
From Control Room: None

Notes: This will provide the operator with signals for open loop tuning and also some channels for fixed observation. It is not intended for 3D so 250 points would provide similar detail to a scope trace. Rate and jitter are based on ergonomically guesses.

5. RT Knobs

Purpose:
Provide open loop feedback for beam control of parameters such as orbit bump, chromaticity, accelerating voltage. 5 systems may operate concurrently
Description:
For 5 different parameters, take a single value from operator, convert to hardware values, and send to 8 devices
Data:
Each device: ID, 2 properties, and 2 data values
Rate: 10 Hz
Jitter: 50 ms
To Control Room: None
From Control Room: 5 systems * 8 devices * 5 variables = 200 variables
Notes: this is the counterpart of the beam property display, the 8 devices could be a tune quadrupole in each octant, 5 systems seems already beyond the chaos threshold, 10 Hz is chosen
considering hardware bandwidth and operator perception. 50 ms guess at the operator's perception.

6. Beam Control Feedback

Purpose:
Provide closed loop feedback for beam parameters such as tune, chromaticity on the ramp.
Description:

For 2 parameters of each beam collect data from 2 pickups, compute correction, and send results to 8 converters
Data:
Each pickup: ID, 2 properties, 2 values, tag
Each corrector: ID, property, current
Rate: 10 Hz
Jitter: 10 ms
To Control Room: 2 beams * 2 parameters * 2 pickups * 6 variables = 48 variables
From Control Room: 2 beams * 2 parameters * 8 converters * 3 variables = 96 variables
Notes: these are the feedback channels to compensate unpredictable and irreproducible field errors in the superconducting magnets. 2 pickups is adequate except for whole ring data i.e.orbit. 8 converters chosen with tune quads in mind. Rate and jitter seem generous compared with evolution of LHC beams.

7. Orbit Control at the Collimators and IPs

Purpose:
Provide closed loop feedback on the closed orbit at the locations of the beam cleaning collimators and the IPs.
Description:
For each beam collect orbit data from 10 pickups per plane in each of the 2 cleaning insertions and the 4 IPs i.e. 6 insertions. Compute corrections, either centrally or locally depending on
strategy to decouple this system from the global feedback - 1 above. Send corrections to 8 correctors in each insertion.
Data:
Each pickup: 1 ID, 3 properties, 2 positions, 1 intensity, tag
Each corrector: 1 ID, 1 property, 1 current
Rate: 10 Hz
Jitter: 10 ms
To Control Room: 2 beams * 6 insertions * 2 planes * 10 pickups * 8 variables = 1920 variables
From Control Room: 2 beams * 6 insertions * 2 planes * 8 correctors * 3 variables = 576 variables

Notes: The rate must be high enough to deal with rapid orbit changes provoked by such events as a corrector tripping. The design of the cleaning system should consider a trip of a corrector involved in the local orbit feedback.

8. Feedback from reference magnets

Purpose:
Apply compensation of multiple field errors as measured on the reference magnets.
Description:
No work has been done. As a start here it is proposed that 5 multipoles are sent from a reference magnet representing each octant to the centre. A correction is sent from the centre to a
power converter in each octant for each multipole.
Data:
For each converter: 1 ID, 1 property, 1 current
From each multipole: 1 ID's, 1 properties, 1 strength
Rate: 1 Hz
Jitter: 100 ms
To Control Room: 2 beams *8 octants * 5 multipoles * 3 variables = 240 variables
From Control room: 2 beams * 8 octants * 5 converters * 3 variables = 240 variables