



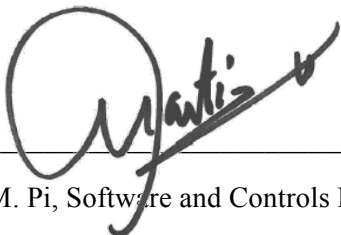
GMT Software and Controls Standards

GMT Requirements Document

Document ID: GMT-REF-00029	Revision: Rev. A
Date: 05/08/2020	Status: Released
Author: J. Filgueira	

Signatures

Owner

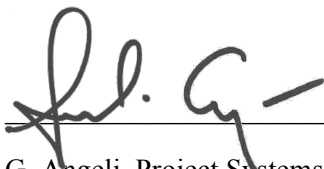


M. Pi, Software and Controls Manager

02/20/2020

Date:

Approvers



G. Angeli, Project Systems Engineer

04/19/2020

Date:



D. Ashby, Project Engineer

04/19/2020

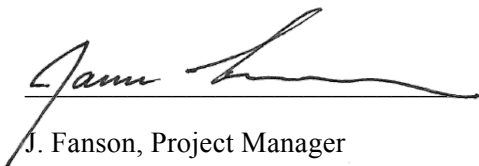
Date:



W. Burgett, Deputy Project Manager

04/23/2020

Date:



J. Fanson, Project Manager

04/23/2020

Date:

Revision Log

Revision	Date	Affected Sections	Change Request #	Comments	Change Author
A	05/08/2020	All	GMT-CR-04264	Authorized Release	J.M.Filgueira

For detailed revision history in DOORs, click [here](#).

Table of Contents

1 Introduction.....	7
1.1 Purpose	7
1.2 Scope	7
2 Definitions, Acronyms, and Reference Documents.....	8
2.1 Definitions	8
2.2 Acronyms.....	9
2.3 Applicable Documents	10
2.4 Referenced Documents.....	10
2.4.1 Industry	10
2.4.2 Project	12
2.5 Auxiliary Documents.....	13
3 Device Control System Reference Architecture	14
3.1 Introduction	14
3.2 Terminology	14
3.3 DCS Context.....	15
3.4 DCS Definition.....	17
3.5 Component Architecture.....	19
3.5.1 Component State.....	20
3.5.2 Component Connections.....	23
3.5.3 Component Behaviors.....	23
3.5.4 DCS Component Containers.....	25
3.5.5 DCS Applications	26
3.6 DCS Interfaces.....	26
3.7 Instrument Software and Controls Integration	26
4 Device Control System Specification.....	26
4.1 Functional Requirements.....	26
4.1.1 Functional Breakdown	26
4.1.2 DCS Product Breakdown Structure	30
4.1.3 Process Data.....	31
4.1.4 Process Control	32
4.1.5 Process Supervision	32
4.1.6 Process Monitoring	34
4.1.7 State Variables	36
4.1.8 Fault Management.....	40
4.1.9 Alarm Management.....	42
4.1.10 Error and Status Logging.....	45
4.1.11 Process Configurable Properties Management	46
4.1.12 Operation Support.....	47
4.2 Non-Functional Requirements.....	51
4.2.1 Security	51
4.2.2 Performance	51
4.2.3 Availability.....	52
4.2.4 Diagnostics.....	52
4.2.5 Safety	53
4.2.6 Naming Convention	55



4.3 Hardware Requirements	57
4.3.1 Device Control Computer	57
4.3.2 PLC-based DCC	59
4.3.3 PC-based DCC	60
4.3.4 Local ISS Safety PLC	60
4.3.5 Deployment Strategies	61
4.3.6 Computing Resources Sizing	68
4.3.7 Field Devices Interface	69
4.3.8 Motion Control Deployment	70
4.3.9 DCS Software Platform	71
4.3.10 Technical Cameras Interface	71
4.3.11 Science CCD Detectors Interface	71
4.3.12 Control Cabinets	72
4.3.13 Control Signal Cabling Rules	72
4.3.14 Networking	73
4.4 Software Requirements	73
4.4.1 Operating Systems	74
4.4.2 Programming Languages and Tools	74
4.4.3 Modeling Requirements	76
4.4.4 Development Requirements	78
4.5 Lifecycle Requirements	80
4.6 Interface Specification	83
4.6.1 Control System Interfaces	83
4.6.2 Interlock and Safety System Interfaces	90
5 Non-Conformances	93
APPENDIX: Accepted EtherCAT Devices	94

List of Figures

Figure 3-1 [ID 91671]: General Control Structure	15
Figure 3-2 [ID 91672]: DCS in the Context of the GMT Observatory Control System	16
Figure 3-3 [ID 91673]: DCS Detailed Context	17
Figure 3-4 [ID 91674]: DCS Functional Decomposition	19
Figure 3-5 [ID 91675]: Structure of a Component in the Context of a DCS	22
Figure 3-6 [ID 91676]: Reactive Closed Control Loop	24
Figure 3-7 [ID 91677]: SDK Core Framework Built-in State Machines	25
Figure 4-1 [ID 91678]: Component, Port and Connector	27
Figure 4-2 [ID 91679]: DCS Package Decomposition	30
Figure 4-3 [ID 91680]: Component op_state State Machine	37
Figure 4-4 [ID 91681]: Sim Mode State Machine	39
Figure 4-5 [ID 91682]: Control Mode State Machine	40

Figure 4-6 [ID 91683]: Fault State Machine.....	41
Figure 4-7 [ID 91684]: Alarm State Machine.....	43
Figure 4-8 [ID 91685]: ISS Architecture – Logical View	54
Figure 4-9 [ID 91686]: ISS Architecture – Physical View.....	55
Figure 4-10 [ID 91687]: Single-Tier Deployment Strategy Logical View.....	62
Figure 4-11 [ID 91688]: Single-Tier DCS Deployment Strategy Physical View	63
Figure 4-12 [ID 91689]: Two-Tier DCS with Hybrid PLC/PC Logical View	64
Figure 4-13 [ID 91690]: Two-Tier Controller with Hybrid PLC/Supervisor Physical View.....	65
Figure 4-14 [ID 91691]: Two-Tier Deployment (SDK) Logical View	66
Figure 4-15 [ID 91692]: Two-Tier Deployment (SDK) – Physical View.....	67
Figure 4-16 [ID 91693]: DCS Implementation Strategy Overview.....	77
Figure 4-17 [ID 91694]: Controlled Subsystem Physical Interface.....	84
Figure 4-18 [ID 91695]: High-Level – Low-Level Interface (Logical View).....	89
Figure 4-19 [ID 91696]: Global ISS – Controlled Subsystem Physical Interface.....	91
Figure 4-20 [ID 91697]: Global ISS – Controlled Subsystem Logical Interface	92
Figure 4-21 [ID 91698]: OCS – Controlled Subsystem Physical Interface.....	93

List of Tables

Table 2-1 [ID 91710]: Definitions	8
Table 2-2 [ID 91712]: Acronyms	9
Table 2-3 [ID 91714]: Applicable Documents	10
Table 2-4 [ID 91717]: Industry Referenced Documents	10
Table 2-5 [ID 91719]: Project Referenced Document.....	12
Table 4-1 [ID 91785]: DCS Functional Packages	27
Table 4-2 [ID 91846]: Component op_state Specification.....	37
Table 4-3 [ID 91862]: Kind Feature Values Definition.....	41
Table 4-4 [ID 91872]: Alarm States	43
Table 4-5 [ID 91874]: Alarm Kind Values.....	44
Table 4-6 [ID 91884]: Logging Levels.....	45
Table 4-7 [ID 92015]: Motion Control Deployment Mode.....	70

1 Introduction

1.1 Purpose

The GMT Software and Controls encompasses the software and hardware components necessary to control and monitor the GMT optical and electromechanical subsystems and to safely and efficiently operate the GMT observatory. Its design is driven by a set of general guidelines: to use industry components and standards that improve the cost-effectiveness of the system; to design an architecture around well-established practices and design patterns; to validate the technical platform and architecture by prototyping and incremental delivery and integration; to use a model-based development approach integrated with an Agile and lean based management process; and to efficiently support and collaborate with the parties involved in the development of the different GMT subsystems.

The GMT Observatory has a large number of electro and optomechanical Devices that required automated control—the Telescope and Instruments account for almost thirty Controlled Subsystems. The control function of each subsystem is implemented in a Device Control System (DCS). Some of them are developed in house, while others are procured from the partner institutions or commercial vendors located around the world. In order to facilitate the specification, development, integration and standardization of common design aspects, DCSs follow a reference component architecture that ensures the internal consistency of subsystems, minimizes the interfaces between them and facilitates the composition of individually tested subsystems.

1.2 Scope

This document defines rules for the design of each DCS in order to guarantee the integration, interoperability, maintainability and cost efficiency of the GMT Software and Controls.

This document is organized as follows: Chapter 2 describes applicable definitions and references. Chapter 3 introduces the DCS reference architecture. Chapter 4 defines the DCS development process and life cycle. Chapter 5 defines the DCS specifications. Chapter 6 specifies the non-conformance policy.

2 Definitions, Acronyms, and Reference Documents

2.1 Definitions

Table 2-1 [ID 91710]: Definitions

Term	Definition
Actuator	Device or technical system which converts commands from the controller into physical effects on the controlled plant
AIT	Assembly, Integration and Testing
Control objective	Goal that the controlled system is supposed to achieve
Control performance	Quantified capabilities of a controlled system
Control System	Part of a controlled system which is designed to give the controlled plant the specified control objectives
Controlled Plant	Physical system, or one of its parts, which is the target of the control problem. Includes sensors and actuators
Controlled Subsystem	Controls relevant part of a system to achieve the specified control objectives. This includes the control system and the controlled plant
Controller	Control component designed to give the controlled plant a specified control performance. The controller interacts with the controlled plant through sensors and actuators. In its most general form, a controller can include hardware, software, and human operations. Its implementation can be distributed over the telescope and facilities.
Device	The basic active elements of a controlled plant including sensors and actuators
Sensor	Device that measures states of the controlled plant and provides them as feedback inputs to the controller
State	Set of variables or parameters describing the dynamic behavior of the controlled system at a given time

2.2 Acronyms

Table 2-2 [ID 91712]: Acronyms

Acronym	Description
COTS	Commercial of the Shelf
CSS	Controlled Subsystem
DCS	Device Control Subsystem
DCP	Device Control Package
ETG	EtherCAT Technology Group
GMF	GMT Modelling Framework
GMT	Giant Magellan Telescope
GMTO	Giant Magellan Telescope Organization
FAT	Factory Acceptance Testing
FTA	Fault Tree Analysis
ICMP	Information and Configuration Management Plan
I/O	Input / Output
ISS	Interlock and Safety System
OCS	Observatory Control System
PO	Project Office
RAM	Reliability, Availability and Maintainability
SAT	Site Acceptance Testing
SOW	Statement of Work
SWC	Software and Controls
TBD	To Be Determined

2.3 Applicable Documents

Table 2-3 [ID 91714]: Applicable Documents

Document Number	Title	DocuShare Link
GMT-REF-00191	GMT Electronics Standards	https://docushare.gmto.org/docushare/dsweb/Services/Document-10312
GMT-DOC-03205	GMT Concept of Operations Document	https://docushare.gmto.org/docushare/dsweb/Services/Document-58009
GMT-REQ-00027	GMT System Level Requirements	https://docushare.gmto.org/docushare/dsweb/Services/Document-6396
GMT-REQ-03214	GMT Observatory Requirements Document	https://docushare.gmto.org/docushare/dsweb/Services/Document-57998
GMT-OCS-xxx	GMT OCS Data Products	TBD
GMT-DOC-00003	GMT Information and Configuration Management Plan	https://docushare.gmto.org/docushare/dsweb/Services/Document-1872
GMT-DOC-00002	GMT Systems Engineering Management Plan	https://docushare.gmto.org/docushare/dsweb/Services/Document-60
GMT-REF-00362	GMT Project Acronyms and Glossary	https://docushare.gmto.org/docushare/dsweb/Services/Document-14403/

2.4 Referenced Documents

2.4.1 Industry

Table 2-4 [ID 91717]: Industry Referenced Documents

Document Number	Title
ISO/IEC 12207:2008	Systems and software engineering – Software life cycle



	processes
ECSS-E-60A	Space engineering - Control Engineering
IEC 61800-7-201	Adjustable speed electrical power drive systems – Part 7-201: Generic interface and use of profiles for power drive systems – Profile type 1 specification
ECSS-E-HB-40A	ESA Software Engineering Handbook http://www.esa.int/TEC/Software_engineering_and_standardisation/TECP5EUXBQE_0.html
https://en.wikipedia.org/wiki/Component-based_software_engineering	Component-based software engineering
http://www.plcopen.org/pages/tc2_motion_control/index.htm	PLC Open MCL
http://www.beckhoff.com/english.asp?twincat/twincat-3.htm	TwinCAT 3
https://opcfoundation.org/about/opc-technologies/opc-ua/	OPC-UA
https://centos.org	Linux CentOS
http://www.plcopen.org/pages/tc1_standards/iec_61131_3/	IEC 61131-3
http://www.etherlab.org/en/ethercat/	Igh EtherCAT master
http://www.ethercat.org	Ethercat standard
https://www.python.org	Python
https://isocpp.org/std/the-standard	ANSI C++17
http://nanomsg.org	Nanomsg
http://msgpack.org	MessagePack
https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO	RT_PREEMPT
http://www.ecma-international.org/publications/standards/Ecma-262.htm	JavaScript

https://nodejs.org	node.js
http://coffeescript.org	CoffeeScript
IEC 61800-7-301	Adjustable speed electrical power drive systems – Part 7-301: Generic interface and use of profiles for power drive systems – Mapping of profile type 1 to network technologies
https://www.sebokwiki.org/wiki/Reliability,_Availability,_and_Maintainability	Reliability, Availability and Maintainability

2.4.2 Project

Table 2-5 [ID 91719]: Project Referenced Document

Document Number	Title	DocuShare Link
GMT-REQ-00612	GMT Observatory Control System Requirements Document	https://docushare.gmto.org/docushare/dsweb/Services/Document-78936
GMT-REQ-01280	GMT SEC Requirements Document	TBD
GMT-REQ-01454	SEC User Design Requirements Document	TBD
GMT-ICD-01455	SEC to Observatory Interface Control Document	TBD
GMT-DOC-01337	GMT Standard Electronics Cabinet Design Concept	TBD
GMT-OCS-REF-00963	GMT WBS Dictionary	https://docushare.gmto.org/docushare/dsweb/Services/Document-30311
GMT-OCS-REF-00023	GMT Product Tree	https://docushare.gmto.org/docushare/dsweb/Services/Document-2393
GMT-RVW-00410, Vol. 1	System Level PDR Report	https://docushare.gmto.org/docushare/dsweb/Services/Document-13491
GMT-RVW-	System Level PDR	https://docushare.gmto.org/docushare/dsweb/Services/Document-

00410, Vol. 2	Report	13492
GMT-RVW-00410, Vol. 3	System Level PDR Report	https://docushare.gmto.org/docushare/dsweb/Services/Document-13493
GMT-RVW-00410, Vol. 4	System Level PDR Report	https://docushare.gmto.org/docushare/dsweb/Services/Document-13490

2.5 Auxiliary Documents

This document defines the requirements that apply during the development of Device Control Systems. Additional processes, workflows and guidelines will be made available in auxiliary documents.

3 Device Control System Reference Architecture

3.1 Introduction

This chapter gives a description of the GMT Device Control Subsystems (DCS) terminology and architecture, its design philosophy and its main functions.

The adoption of a set of standards based on a reference architecture offers the following benefits: Enables the inter-operation between different Software and Controls subsystems. Promotes commonality. Enables reuse of components. Simplifies operation and maintenance thanks to the use of common features. Simplifies the definition of interfaces. Reduces the number of spare parts needed to guarantee the downtime budget. Fosters the use of same solutions to similar problems. Provides a guideline for the design of the DCS subsystems. Reduces the support effort. Provides system level testing strategies.

Due to the expected duration of the project, these standards should not be considered frozen at present time. It is advisable for the project to make use of the most recent concepts and tools as they become available and they demonstrate an advantage over the existing ones (e.g. improved reliability and lower cost). Any change to this document will follow the formal review process as specified in the ICMP.

3.2 Terminology

The diagram in Figure [ID 91671] describes the control engineering terminology used in this document. The control engineering terminology is based on the ECSS-E-60-A - Control Engineering standard. Other terms used in the following sections are based on the theory of dynamical systems. As shown the Controlled Plant or system under control, includes also the Sensors and Actuators of the system.

A Controlled Subsystem is the control relevant part of a system to achieve the specified control objectives. This includes the control system and the controlled plant. (e.g. M1). A Controlled Subsystem has two parts:

- *Device Control System* that includes all the hardware and software required to control a Controlled Subsystem Plant.
- *Controlled Subsystem Plant* that consists of the physical system, or one of its parts, which is the target of the control problem. Includes sensing and actuating Devices. (e.g. M1 support system).

Note: In the State Analysis terminology, the plant is known as system under control.

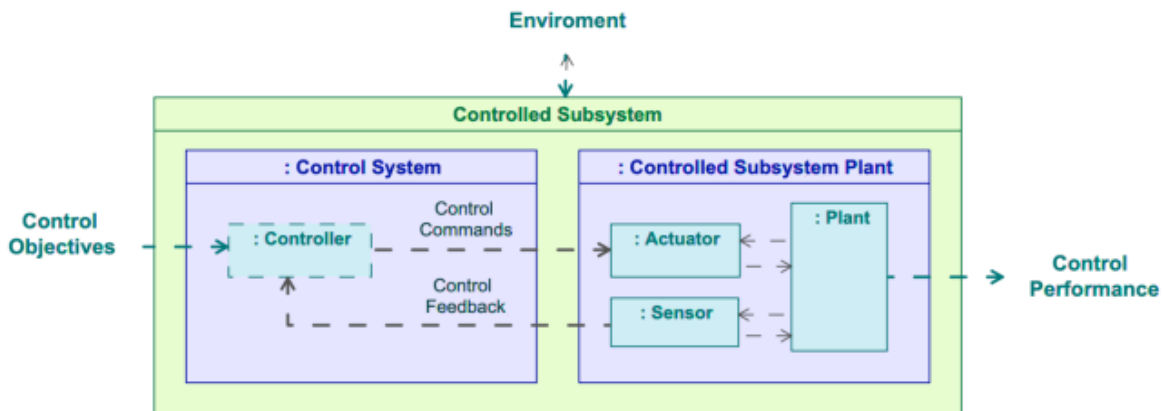


Figure 3-1 [ID 91671]: General Control Structure

3.3 DCS Context

The DCS Context includes the parts of the GMT Software and Control that are relevant to the definition of a DCS. Figure [ID 91672] shows the decomposition of the GMT Software and Controls in two parts, the GMT Control System, which is responsible of the integrated operation and control functions of the Observatory and the Interlock and Safety System that implements the functional safety of the Observatory.

The GMT Control System is subsequently divided in three parts: (a) the Observatory Control System (OCS) that controls the Observatory as an integrated entity and allows the scientific and operation users to interact with the Observatory, (b) the Telescope Device Control Systems (TDCS) that includes the DCSs of enclosure, telescope and adaptive optics and (c) the Instrument Device Control Systems (IDCS) that includes the DCSs of the GMT scientific instruments.

The Interlock and Safety System (ISS) implements the functional safety of the Observatory as determined by the Observatory Hazard Analysis. The ISS is divided in two parts: (a) the Global Interlock and Safety System that implement the system level functional safety that involves more than one subsystem, and (b) the Local Interlock and Safety Subsystems that implement the local functional safety functions if required by the specific hazard analysis of the corresponding Controlled Subsystem.

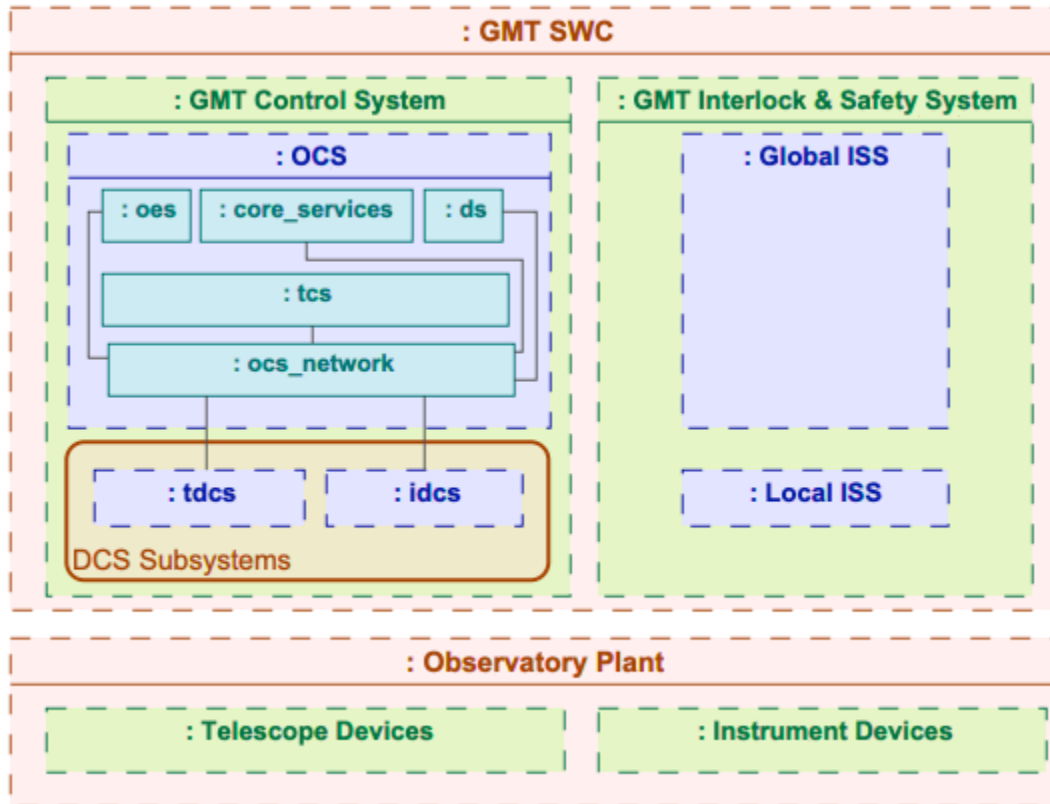


Figure 3-2 [ID 91672]: DCS in the Context of the GMT Observatory Control System

The *Observatory Control System* includes all the hardware and software required to operate the observatory and telescope as a System and to coordinate all the DCSs. It comprises:

- The *Observing Execution System* (oes) which includes all the hardware and software required for the high-level integrated operation of the observatory (e.g. scientific observations scheduling and execution).
- The *Core Services* (core_services) which includes essential observatory services that addressed cross cutting concerns (e.g. creation, propagation, processing and storages of logging or telemetry data).
- The *Telescope Control System* (tcs) which includes the system level telescope control functions (e.g. wavefront control, telescope pointing, tracking and guiding).
- The *Observatory Data Systems* (ds) which support data processing and archiving functions.
- The *OCS Network* (ocs_network) which provides all the communication infrastructure of the OCS.

The following diagram shows the DCS in the context of the overall GMT Control System:

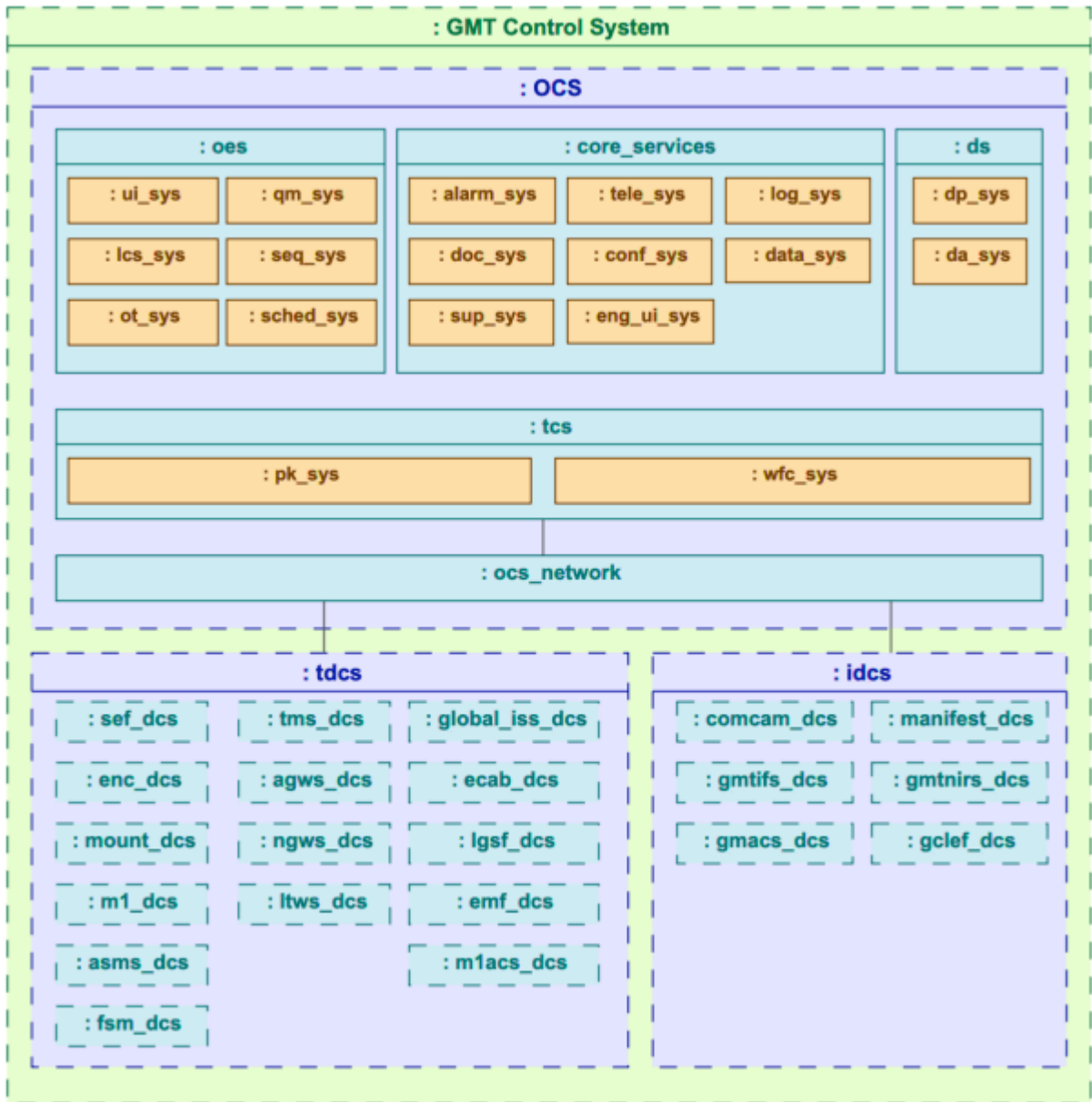


Figure 3-3 [ID 91673]: DCS Detailed Context

3.4 DCS Definition

The main goal of a DCS is to implement the control function of a dynamical system. To that purpose each DCS includes all the hardware (e.g. control computers) and software (e.g. control algorithms) necessary to control and operate the corresponding telescope or instrument opto/electro/mechanical Controlled Subsystem (e.g. primary mirror control system, adaptive secondary control system, instrument control system, etc.). For example, a DCS commands opto-mechanical degrees of freedom or captures data from optical and other sensors. Degrees of freedom are represented by a set of state variables that define the



observed and desired state of the system and evolve following a control law which often consists of a set of nested closed control loops. Although the primary function of a Device Control Subsystem is the control of hardware devices, an efficient and robust operation requires also the development of:

- Calibration software that is required to produce error maps or models that allow the control system to achieve the required performance;
- Diagnosis software that provides the capability to observe and verify the behavior of hardware and software subsystems, especially during the commissioning phase;
- Safety functions that detect the possible occurrence of an unsafe state; preventing, if possible, the engagement of the Interlock and Safety System, which is ultimately responsible for functional safety and acts usually in a more ‘dramatic way’; and
- Supervision software that coordinates several components or subsystems to guarantee their correct states and that implements fault management strategies that enable the system to continue its normal operation or the graceful degradation of performance in non-nominal scenarios.

A DCS is partitioned in modular functional packages classified in two main categories:

- *Device Control Packages* that include all the software necessary to control a Controlled Subsystem Plant. The basic building block of a Device Control Package is a **Device Controller** which includes all the software necessary to operate a Controlled Subsystem Device.
- *Operation Support Packages* include all the software necessary to operate a Controlled Subsystem integrated with the rest of the observatory. Operation Support Packages are further defined in the following sections.

Section 4.1.1 provides a more detailed description of the DCS package structure.

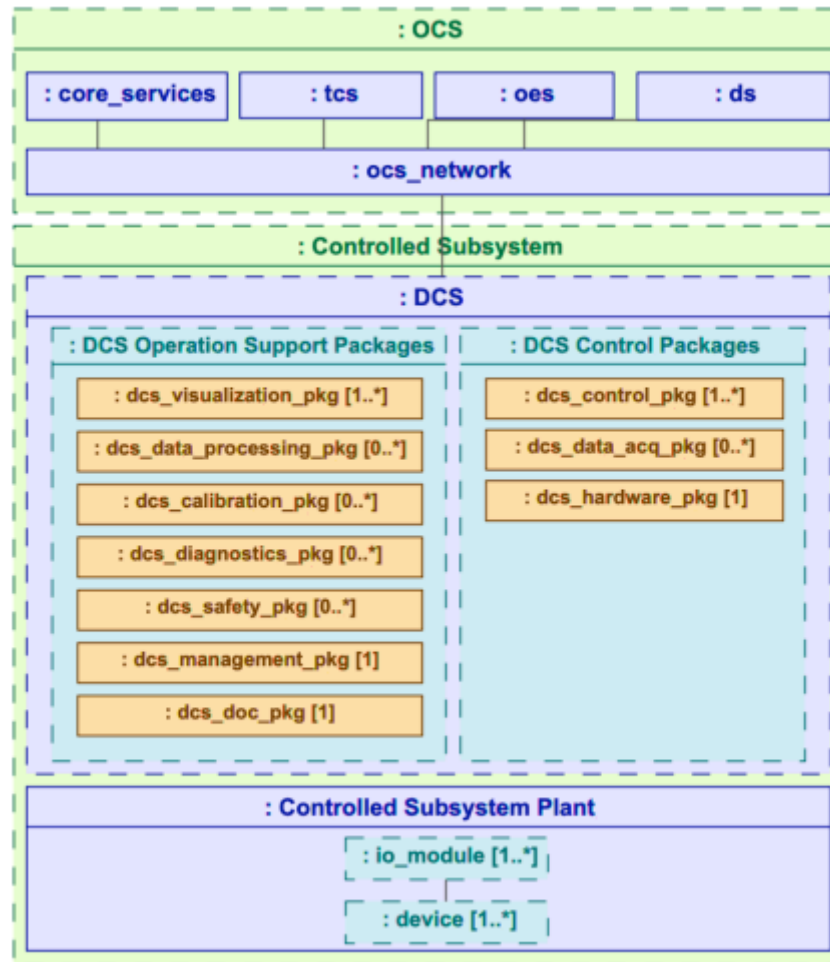


Figure 3-4 [ID 91674]: DCS Functional Decomposition

3.5 Component Architecture

A Component Architecture is based on independent, modular components that help to manage complexity and encourage re-use. A software Component is a module that encapsulates a set of related functions or data. From a modeling point of view *Components* are a software representation of a significant entity in the domain of the problem. In the case of a Device Control System, where the problem domain is prominently control and data acquisition systems, some examples of components would be: Controllers, Data Processing Pipelines or Hardware Adapters.

The design of the GMT Software and Controls System is based on a distributed component architecture, where *Components* like *Controllers* and *Pipelines* interact with the GMT opto-mechanical devices (system under control) by means of *Hardware Adapters*. As the GMT has a large number of such devices distributed across the telescope and observatory, the DCS Components are deployed in specialized computers, each one connected directly to a hardware Devices (e.g. the telescope mount, the primary mirror).



Components can be grouped to create composite modules that implement elaborated functions, like the high-level functions of the Observatory. For that purpose, *Components* can exchange information relative to each one state. For example, to manage the complexity derived from the large number and heterogeneous nature of the observatory hardware devices, *Components* are arranged in a hierarchical partition of responsibilities. In this hierarchy, low-level *Components* control directly the elementary degrees of freedom (e.g. mechanical, thermal) of the observatory, while high-level *Components* control higher-level degrees of freedom, often obtained as a computation over the basic ones (e.g. optical).

Supervisory functions are carried out by a specialized *Supervisor* component that implements the top-level interfaces of the DCS; maintains the integrity and robustness of the control system and the system under control (e.g.: DCS top-level fault management); manages the life cycle and configuration of other DCS *Components*; or coordinates the Controlled Subsystem modal transitions amongst other functions.

Hardware Adapter *Components* provide an implementation independent interface with the Observatory hardware Devices often using and standardized fieldbus as described in the Hardware standards in section 4.3.7.

3.5.1 Component State

Components are the building blocks that allows us to construct the control and data acquisition functions of a dynamical system. From this point of view, *Components* embody the *state* and *behavior* of such functions.

As part of a *dynamical system*, the state of a *Component* at a given moment in time comprises all the information necessary to compute the future state of the *Component* as determined by the *Component* dynamic laws (*behavior*). In order to harmonized how the state of a *Component* is represented, the following *Component* features are defined:

- *Inputs* represent the process data that can flow into the *Component*. Input process data may come from the Plant through a Hardware Adapter or from another *Component*, when they are arranged as a collaborating group.
- *Outputs* represent the process data that can flow from the *Component*. *Outputs depend on the current value of the State Variables and the Inputs*. Output process data may be directed to affect the Plant through a Hardware Adapter or to another *Component* when they are arranged as a collaborating group (e.g. axis group with multi-axis kinematics).
- *State variables* are variables whose values evolve through time in a way that depends of the values they have at any given time and also depends on the externally imposed values of *input* variables. In Controller components the evolution of state variables is governed by the control law.
- *Properties* represent configuration data that affects the behavior of a component but that doesn't change dynamically. It has a predefined default value which is independent of the Plant state. Properties are mostly changed by operators when they want to alter the behavior of a *Component*.



- *Faults* are specialized state variables that represent the occurrence of a predefined fault condition that the Component is able to detect as a function of its current state representation. Some Faults may have a predefined action to recover from them.
- *Alarms* are specialized state variables that represent the status of a predefined alarm condition that the Component is able to detect as a function of its current state and that requires awareness of the observatory operators. The Alarm status depends on the alarm detection function and the interaction with the operators.

In the case of a Controller component, state variables allow steering the state of the system under control by imposing constraints on the value that the controllable state variables need to achieve at a given moment in time. The diagram in the Figure [ID 91676] shows the activities involved in the estimation and updating of state variables.

The diagram in Figure [ID 91675] shows the structure of a Component as implemented in the SDK Core Framework:

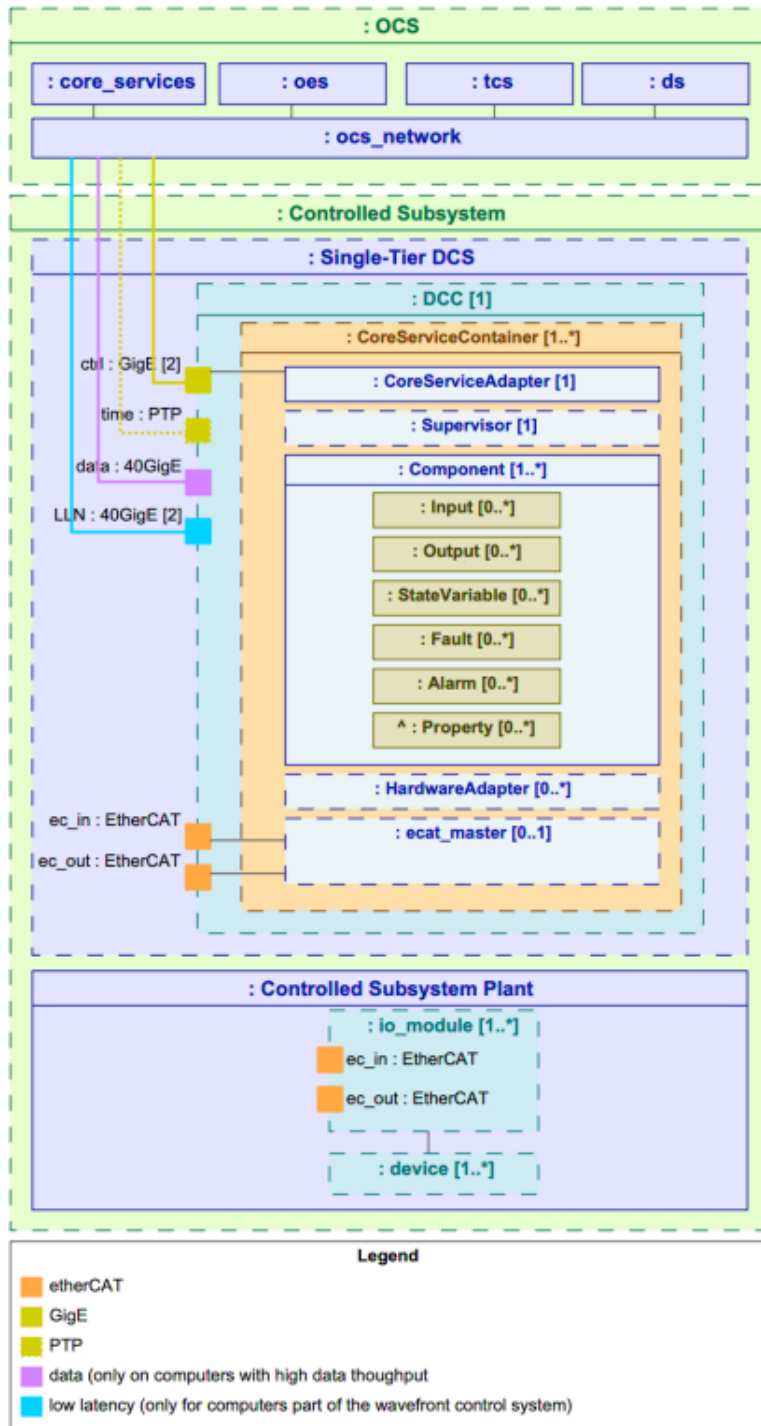


Figure 3-5 [ID 91675]: Structure of a Component in the Context of a DCS

3.5.2 Component Connections

The values of the features of a Component can be accessed externally and represent the interface of the Component while the details of each Component behavior are encapsulated. In some applications Components need to communicate information to each other, this is accomplished by defining Connectors that allow the values of features flow between components. For example, Connectors are used to (a) integrate standardized reusable control Components with a given field bus configuration; (b) connect Component responses with user interface components; or (c) connect Components with the Observatory Core Services.

3.5.3 Component Behaviors

In our problem domain, control and monitoring systems, the state of a Component changes with time. For each Component we define a *step* function which calculates the future state of the Component based in the current state. When Components are Controllers, whether discrete, continuous or hybrid, the step function implements the control law. The *step* function defined as: $f_{step}: (I, S) \rightarrow O$, takes the current Inputs and current State (as represented by State Variables) of a Component and generates a new set of Outputs that produce an effect on the context of a Component (e.g. other connected components or the system under control via the Hardware Adapters).

The diagram in Figure [ID 91676] represents an overview of the control, supervision and monitoring functions performed by a DCS Component during the execution of its step function and their relation to state variables. Each component must implement a step function that is invoked by the SDK Core Framework at a predefined evaluation rate. In each iteration of the loop the component must process its inputs and outputs and health status. The estimation of state variables is defined by the state variable sampling rate and it may use an estimation model.

The control of the state variable is defined by the state variable control rate. It must calculate the control output, if necessary, with a control model. The evaluation rate shall be always higher that the maximum of the sampling or control rate of any of its states variables

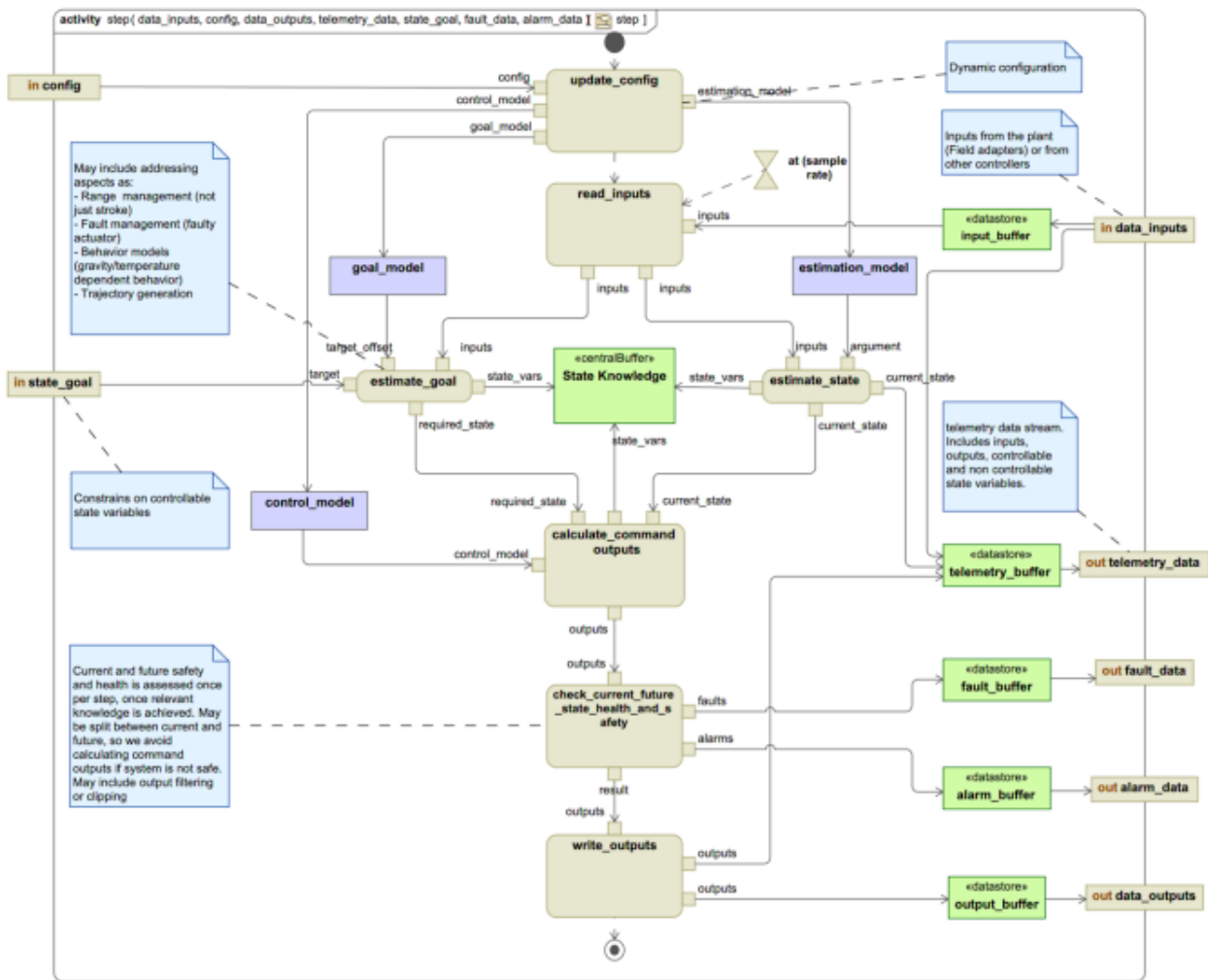


Figure 3-6 [ID 91676]: Reactive Closed Control Loop

In some cases, the *step* function may be enough to implement all the functions allocated to a Component. However, in other cases, if the functions are complex, it may be useful to organize them using more elementary function blocks named Behaviors. Behaviors must specialize an *apply* function which is invoked by the Core Framework at the evaluation rate of the Component (in the SDK the evaluation rate is defined by the *scan_rate* property). In case the Behavior requires to be evaluated at a different rate from the Component evaluation rate a specific rate attribute can be defined.

It is possible to define input and output parameters as part of a Behavior specification. This allows to organize them following the structure that suits the decomposition of the problem by connecting the inputs and outputs of a Behavior to another. For example, to create a control hierarchy inside a Component.

3.5.3.1 State Machines

A common way of implementing discrete functions is by using State Machines. The SDK Core Frameworks support the use of Moore and Mealy machines. State machines are specialized behaviors

which are evaluated in a reactive way. State Machines can be composed to create arbitrary large hierarchical and parallel machines. They can also be used to compose hybrid control architectures that include continuous and discrete behavior inside a Component. The State Machines are defined as a tuple $S = \{S, S_o, I, O, f_t, f_o\}$ where:

- S is the set of valid states
- S_o is the value of the initial state
- I is the set of valid inputs
- O is the set of valid outputs
- $f_t: (I, S) \rightarrow S$ is a transition function that takes as arguments the inputs and current state and determines the next state.
- $f_o: (I, S) \rightarrow O$ is an output function that takes as arguments the inputs and current state and updated the output of the StateMachine.

Additionally, for each state it is possible to define: an *entry* function that is executed when the state is entered, an *exit* function that is executed when the state is exited, and an *on* function that is executed every time the State Machine is evaluated and is in the state.

The following diagram shows as example some of the SDK Core Framework built-in state machines.

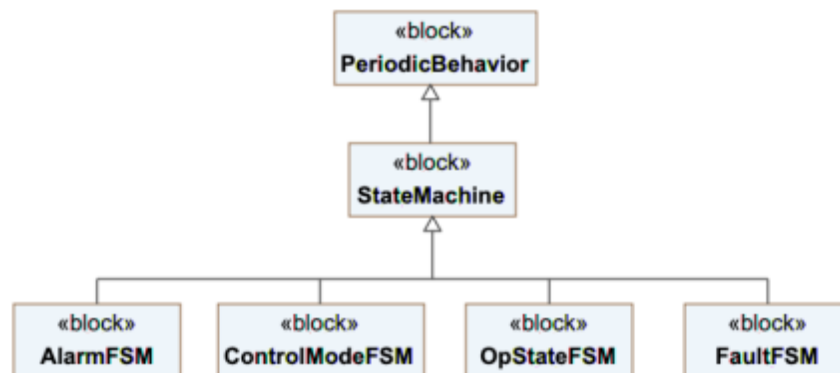


Figure 3-7 [ID 91677]: SDK Core Framework Built-in State Machines

3.5.4 DCS Component Containers

The OCS Core Services implement cross-cutting aspects of the GMT Control System like the Telemetry or the Logging Services. As part of the integration of each DCS with the rest of the Observatory, Components are required to use the OCS Core Services. Components are deployed in Component Containers that provide a simplified Application Programming Interface (API) that allows them to access the OCS Core Services.

3.5.5 DCS Applications

Components and Containers need to be deployed as processes in a host computer during operations. Applications are transformed in an executable that instantiates the Components inside a Container that allows them to use the OCS Core Services. Additionally, applications allow the management of command line options.

3.6 DCS Interfaces

As mentioned above, the features of a Component represent their interface. The Interface of a DCS is defined by the set of features that are connected to Components that are external to the DCS. In other words, the interface of a DCS is defined as the list of Connectors that have any feature of the DCS Components as an endpoint. The endpoints can be inbound or outbound. The section 4.6 of this document defines the specification of DCS interfaces.

3.7 Instrument Software and Controls Integration

The GMT instruments software and controls follow the same architectural design as other Device Control Subsystems. An Instrument Device Control Subsystem includes most of the packages described in Figure [ID 91674] and is developed following the standards and rules described in this document. The operation of instrument, telescope and adaptive optics DCSs will be coordinated by the OCS (e.g. as operation sequences executed by the Sequencer) in order to execute automated observation or calibration operations. The IDCS data pipeline packages are integrated in the OCS (i.e. in the Data Processing System) so Data Products management is operated in a uniform way across the Observatory. Analogously, instrument specific user interface panels and observing tools are integrated in the Observing Execution System to ensure consistent and seamless implementation of observatory workflow.

4 Device Control System Specification

This chapter defines common requirements applicable to each GMT DCS. It comprises DCS architecture, software and hardware specifications.

4.1 Functional Requirements

4.1.1 Functional Breakdown

Each DCS is made up of Components organized into Packages according to their functional affinity or relationships. The following diagram shows the formal specification of a DCS structure:

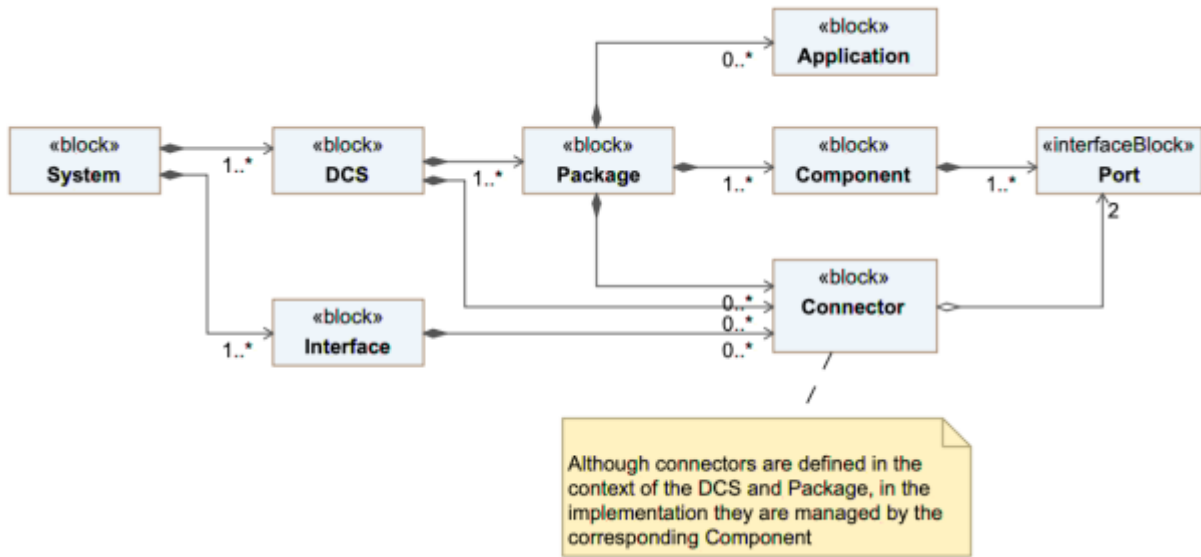


Figure 4-1 [ID 91678]: Component, Port and Connector

Which Packages exist in which DCS depends on their specific functionality (e.g., some subsystems do not require special calibration components, or do not interface with hardware devices). Table [ID 91785] DCS Packages are split in two categories:

- Device Control Packages – These packages are included in Controlled Subsystems that involve the control of optical and electromechanical hardware Devices.
- Operation Support Packages – These Packages include software components necessary to support health monitoring, automation and proper operation of a Controlled Subsystem. Diagnosis and calibration packages are emphasized early on in the design. This is an area that is often overlooked despite the fact that they may take a significant amount of development effort, especially in the case of complex adaptive optics control subsystems.

The next table defines different categories of Packages.

Table 4-1 [ID 91785]: DCS Functional Packages

Device Control Packages		
Package Name	Description	Typical Components
Control Package	Contains software Components that implement the control and supervisory functions of a Device Control Subsystem (e.g., Mount Device Control System Control Package).	Supervisor, Controller, Hardware Adapter



Data Acquisition Package	Contains software Components that implement the supervisory and data acquisition functions of a Detector Control Subsystem (e.g., AGWS Slope Processor Package). Only Controlled Subsystems that contain detectors (e.g wavefront sensor, acquisition/guide camera or a science detector) need to provide a Data Acquisition Package	Supervisor, Controller, Pipeline
Hardware Package	Contains hardware elements in which to deploy the Device Control or Data Acquisition Package software Components and the hardware to interface with the electromechanical Devices.	Device Control Computer, I/O Module
Operation Support Packages		
Package Name	Description	Typical Components
Sequencing Package	Contains Sequences necessary for the operation of the Controlled Subsystem.	Sequence
Diagnosis Package	Contains software Components necessary to implement diagnosis functions when required. This may involve the development of special control or operation modes.	Supervisor, Controller, Pipeline, Sequence
Calibration Package	Contains software Components necessary for the calibration and characterization of hardware Devices. This may include the development of special control or operation modes.	Supervisor, Controller, Pipeline, Sequence
Data Processing Package	Contains software Components necessary for the calibration and processing of science and WFS detectors.	Supervisor, Pipeline
Visualization Package	Contains Panels and Widgets that provide custom visualizations necessary for the efficient operation of a given Controlled Subsystem (e.g., M1 global status Panel). Note that default engineering Panels are available as part of the Engineering UI.	Panel, Widget
Observing Tool Plugin Package	Observing Tool (OT) plugins provide instrument specific editors that integrate with the Observatory Observing Tools to facilitate the specification of instrument specific observation parameters.	Panel, Widget, Pipeline
Quality Monitoring Package	Quality Monitoring components —usually in the form of data processing Pipelines provide analysis of the data products of an observation with the goal of	Pipeline



	assessing the correctness of the observing process	
Safety Package	Contains software/hardware that implement Controlled Subsystem specific safety functions. These Components often interface with the corresponding Local ISS, but could be independent (e.g., M1 safety controller).	Supervisor, Controller
Operation Workflows Package	Contains Workflows that allow the automation of high-level operations that involve the Controlled Subsystem (e.g., unit test workflow, or calibration workflow in case that several sequences and human operations are involved).	Workflow
Management Package	Contains Components that capture the development backlog and the Assembly Integration and Testing plans.	Plan, Workflow
Documentation Package	Contains Documents that describe different aspects of the DCS (e.g. DCS requirements document, DCS architecture)	Document
Common Package	A Common Package may be defined to contain common elements reused in the other packages of the DCS	Various

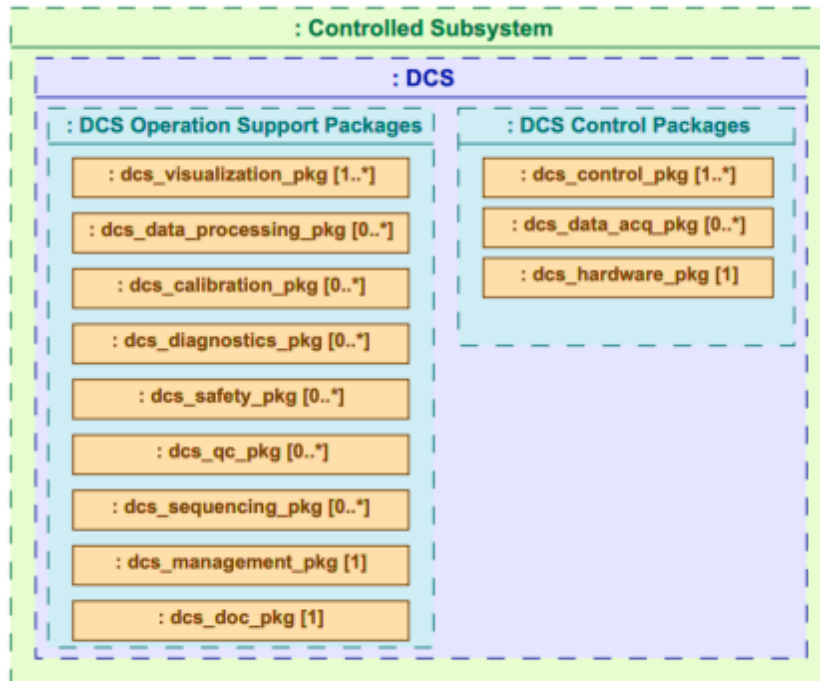


Figure 4-2 [ID 91679]: DCS Package Decomposition

4.1.2 DCS Product Breakdown Structure

The Product Breakdown Structure defines the overall organization of a DCS.

REQ-L3-SCS-91788: DCS Product Decomposition

The DCS shall be partitioned in functional packages as defined in Table [ID 91785].

REQ-L3-SCS-91790: DCS Package Decomposition

The DCS packages shall be partitioned in Components.

REQ-L3-SCS-91791: DCS Control Package

The DCS shall have at least one Control Package.

REQ-L3-SCS-91792: DCS Hardware Package

The DCS shall have at least one Hardware Package.

REQ-L3-SCS-91793: DCS Visualization Package

The DCS shall have at least one Visualization Package.

REQ-L3-SCS-91794: DCS Management Package

The DCS shall have one Management Package.

REQ-L3-SCS-91795: DCS Documentation Package

The DCS shall have one Documentation Package.

4.1.3 Process Data

Process Data refers to the data needed to represent the dynamical processes of the Controlled Subsystem Plant.

REQ-L3-SCS-91798: Process Data Identification

The DCS shall identify all the process data relative to the operation of the Controlled Subsystem.

REQ-L3-SCS-91799: Process Data Representation

The DCS shall represent all the process data relative to the operation of the Controlled Subsystem as one of the following categories: process inputs, process outputs, state variables, properties, faults and alarms.

REQ-L3-SCS-91800: Process Data Value Definition

The DCS shall define the sets (continuous or discrete) that contain the values that can be assigned to Process Inputs, Process Outputs or State Variables.

REQ-L3-SCS-91801: Process Data Types

The DCS shall define the data types of the Process Outputs, Process Inputs and State Variables.

REQ-L3-SCS-91802: Process Data Initialization

The DCS shall ensure that the Process Outputs, Process Inputs and State Variables have an initial value that is contained in the corresponding valid set.

REQ-L3-SCS-91803: Process Data Units

The DCS shall define the units of Process Outputs, Process Inputs and State Variables that represent a physical magnitude.

Notes: As a general rule, SI units shall be used.

REQ-L3-SCS-91805: Continuous Process Data Value Ranges

The DCS shall define a nominal operating range for Process Outputs, Process Inputs and State Variables that represent a continuous value.

REQ-L3-SCS-91806: Local Storage of Data

The DCS shall not store locally permanently any data.

Notes: All data generated by any DCS is sent to the OCS, where is stored for analysis and archival.

Legacy ID: REQ-L3-SCS-0017

4.1.4 Process Control

REQ-L3-SCS-91808: DCS Control Functions

The DCS shall implement all the control functions identified by the Controlled Subsystem functional architecture.

Legacy ID: REQ-L3-SCS-0012

REQ-L3-SCS-91809: Control Function Integrity

The DCS shall ensure that the Process Outputs values produced by the control function are defined for any valid combination of Process Inputs and State Variables values.

REQ-L3-SCS-91810: Control Loop Optimization

The DCS shall optimize the control loops in order to reduce the frequency of actuation and the wear of the Controlled Subsystem Devices.

Legacy ID: REQ-L3-SCS-0014

4.1.5 Process Supervision

REQ-L3-SCS-91812: DCS Supervisory Functions

The DCS shall implement all the supervisory functions identified by the Controlled Subsystem functional architecture.

Legacy ID: REQ-L3-SCS-0015

REQ-L3-SCS-91813: Non-Responsive Component Detection

The DCS shall detect when a Component is not able to respond, in which case it shall assert the fault `<component>_not_responding`.

REQ-L3-SCS-91814: Non-Operational Component Detection

The DCS shall detect when a Component operation state fails to enter its RUNNING state when requested, in which case it shall assert the fault `<component>_not_operational`

REQ-L3-SCS-91815: Controlled Subsystem Integrity

The DCS shall ensure that the Controlled Subsystem is operated in a way that ensures its integrity (e.g., collision avoidance)

REQ-L3-SCS-91816: Supervised Components Configuration

The DCS shall ensure that the DCS Components are configured in the correct sequence and with the predefined configuration properties.

REQ-L3-SCS-91817: Controlled Subsystem Robustness

The DCS should perform its functions in presence of non-nominal conditions, (e.g. fault management and tolerance) to the extent that this is possible.

REQ-L3-SCS-91818: Controlled Subsystem Life-Cycle

The DCS shall implement the Controlled Subsystem life-cycle functions (e.g. startup and shutdown).

Legacy ID: REQ-L3-SCS-0033

REQ-L3-SCS-91819: Controlled Subsystem Operation Modal Transition

In cases in which the Controlled Subsystem has different modes of operation, the DCS shall implement the transition between them.

REQ-L3-SCS-91820: Controlled Subsystem Input/Output Health

The DCS shall monitor the health of the input and outputs that connect the DCS to the Controlled Subsystem hardware plant (e.g. sensors or actuator devices)

REQ-L3-SCS-91821: Status Information

The DCS shall provide status information required to operate and debug the Controlled Subsystem.

Notes: The status information includes Controlled Subsystem operating states, state variable transitions, alarm conditions, log messages and configuration change events.

Legacy ID: REQ-L3-SCS-0016

REQ-L3-SCS-91822: Central Operation

The DCS shall be able to be operated remotely from the observatory control room(s).

Legacy ID: REQ-L3-SCS-0018

4.1.6 Process Monitoring

REQ-L3-SCS-91824: Process Input

The DCS shall be able to sample each process input at a rate that allows constructing a temporal data representation of the Controlled Plant signals consistent with the associated control function.

REQ-L3-SCS-91825: Process Input Sampling

The DCS shall be able to sample each process input with a unique identifier, a time stamp and an error identification in case of error.

REQ-L3-SCS-91826: Process Output Sampling

The DCS shall be able to sample each process output with an identifier, a time stamp and an error identification in case of error.

REQ-L3-SCS-91827: State Variable Sampling

The DCS shall be able to sample each process state variable with an identifier, a time stamp and an error identification in case of error.

Notes: Units, name, Quality of Service parameters and description of the state variable are not required in the sampled data as they are defined in the DCS System Definition Files.

Legacy ID: REQ-L3-SCS-0019

REQ-L3-SCS-91828: Maximum Sampling Rate

The DCS shall be able to sample each process input, process output and state variable up to a maximum rate defined for each one of them.

REQ-L3-SCS-91829: Sample Rate Configuration

The DCS shall be able to modify the sampling rate of any process input, process output or state variable in runtime.

REQ-L3-SCS-91830: Raw Data Conversion

The DCS should apply the conversion from raw data to engineering data (scaling) as near as possible to the source of the data.

Legacy ID: REQ-L3-SCS-0020

REQ-L3-SCS-91831: Time Stamping Latency

The DCS shall time stamp state variables as close as possible to the source of data.

Legacy ID: REQ-L3-SCS-0021

REQ-L3-SCS-91832: Calibration Factor

The DCS shall provide the capability to configure the calibration factor and conversion formula applied to each state variable.

Legacy ID: REQ-L3-SCS-0022

REQ-L3-SCS-91833: Process Data Transmission

The DCS shall provide the capability to transmit both raw data and engineering data to the OCS.

Legacy ID: REQ-L3-SCS-0023

REQ-L3-SCS-91834: Process Data Buffering

The DCS shall store temporally data only in the case that a circular buffer is required to manage high throughput telemetry.

Notes: All the GMT telemetry data is stored and archived by the OCS Telemetry Service outside the DCS.

Legacy ID: REQ-L3-SCS-0024

REQ-L3-SCS-91835: Process Data Integrity

The DCS shall detect and notify any case in which the process data doesn't meet its nominal conditions.

4.1.7 State Variables

REQ-L3-SCS-91837: State Variable Definition

The DCS shall define the minimum set of state variables to represent the dynamic state of the Controlled Subsystem.

REQ-L3-SCS-91838: State Variable Control Rate

The DCS shall define the control the rate at which the associated control law has to be updated.

REQ-L3-SCS-91839: Controllable State Variable Sampling Dead Band

The DCS shall define the sampling and control dead bands of state variables whose possible values are continuous.

REQ-L3-SCS-91840: Controllable State Variables

The DCS shall define which state variables are affected by the control law (e.g. position of a linear stage).

REQ-L3-SCS-91841: State Variable Goals

The DCS shall be able to achieve the goals set on its controllable state variables provided that the goals are valid values in the set that defines the state variable.

REQ-L3-SCS-91842: Non-Controllable State Variables

The DCS shall define which state variables are needed to estimate the state, but are not controllable (e.g. external wind speed).

4.1.7.1 Op_state Variable

The op_state state variable represents the operational state of a Component. A set of states addresses the distributed nature of the Component and its life cycle management. The diagram in the Figure [ID 91680] shows the op_state state machine. Only the description of each state is shown. Details about entry actions, transitions and activities are omitted in this diagram.

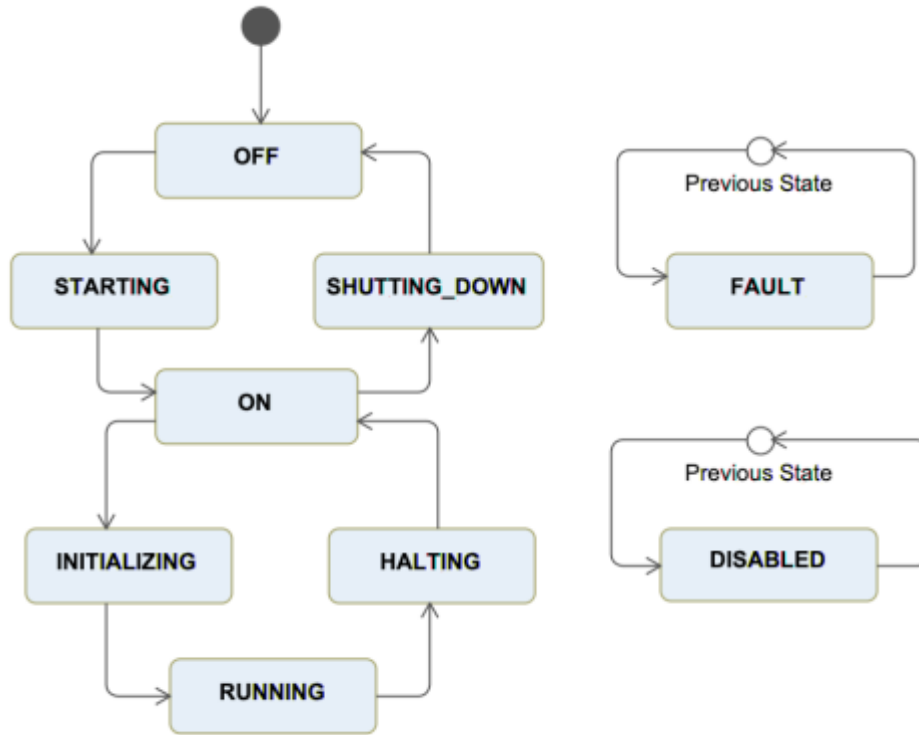


Figure 4-3 [ID 91680]: Component op_state State Machine

Controllers are specialized Components that interact with Devices. Although the op_state state machine is the same, the semantics of the Controller states must consider the plant equipment. The table Table [ID 91846] provides the specification of the Controller state machine.

Table 4-2 [ID 91846]: Component op_state Specification

State	Description
OFF	Initial pseudo state. The Controller is not operational because it has not been created yet. The Controller cannot inform this state as it is not processing its inputs and outputs. In this state, the software is not running, and controlled equipment is not accessible.
STARTING	Auto transition to ON if auto_start is true
ON	Enables the processing of inputs and outputs. The Controller can access the connected Devices which can be properly initialized and configured. When connecting, the Controller



	external equipment outputs shall be in safe state (e.g., brake engaged, motion drives disabled, locking pins inserted). This state can be the final state after a reset or after a power failure.
INITIALIZING	Enables the processing of property, state variables, alarms and fault. While the Controller is in this state the necessary procedures required to make the controller ready to receive operation requests (e.g., find fiducial marks) are executed.
RUNNING	The state variables following mode is set to 'FOLLOWING'. The Controller is running and can be idle or achieving the requested set points for its state variables.
HALTING	The state variables following mode is set to 'NOT_FOLLOWING'. The inputs, outputs, alarms and faults are processed, but the state variables will not follow any new goal.
SHUTTING_DOWN	Disables the processing of inputs and outputs. The Controller stops having access to its connected Devices.
FAULT	State variables are set to 'NOT_FOLLOWING'. The Controller has detected a severe fault condition and is waiting for an event to occur (e.g., operator input) to correct such situation.
RESETTING	Return to a safe and known state. For example, when the Controller has entered into a FAULT state, due to the ISS triggering an interlock condition (which can disable drives, remove power, etc.), a reset command must be sent to the Controller.
DISABLED	In this state, the Controller rejects attempts to perform any control action. This is especially important with Controllers connected to Devices. In this state the Controller does not send demands to the equipment requesting motion or a change (a message is sent to the client indicating that the Controller is disabled). Note that the Controller is ready and it will answer requests that ask for some status, but it will not execute any commands that lead to actions on connected Devices. This state can be reached from any state, and when enabled, will return to the previous state.

REQ-L3-SCS-91847: DCS Operational State State Machine

Each DCS component shall implement the [op_state](#) state machine.

Legacy ID: REQ-L3-SCS-0079

REQ-L3-SCS-91848: State Machine Monitoring

Each DCS component shall send a status message for each state transition to the OCS.

Legacy ID: REQ-L3-SCS-0080

4.1.7.2 Sim_Mode State Variable

Controllers that interface with hardware support specialized operation modes, on-line and simulated:

- In *on-line mode*, controllers try to detect and setup the hardware elements connected to them during startup. If some of the required hardware devices are not available and prevent the controller to perform its functions, the Controller will transition to fault mode. This is the default mode when the system is deployed for operation at the observatory.
- In *simulation mode*, controllers will setup the I/O framework in simulation mode. Communication messages with the hardware will be logged, but will not be sent to the hardware devices. Hardware devices will not be powered up during the startup sequence. This mode is intended to be used during development when the hardware is not yet available or is only partially available. It also enables controller debugging once the hardware is integrated.

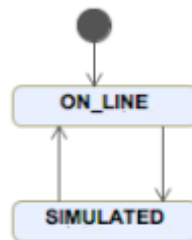


Figure 4-4 [ID 91681]: Sim Mode State Machine

REQ-L3-SCS-91851: On-Line Operation Mode

The DCS shall support the on-line operation mode.

Legacy ID: REQ-L3-SCS-0081

REQ-L3-SCS-91852: Simulation Operation Mode

The DCS shall support the simulation operation mode.

Legacy ID: REQ-L3-SCS-0082

4.1.7.3 Control_Mode State Variable

The DCS distributed Components shall support two operation modes, standalone and integrated:

- In *integrated mode*, components will try to connect with the OCS Observatory Services. If the services are not available, the component will stop its startup sequence. This is the default operation mode when components are integrated and deployed in the Observatory or during development when used with the OCS Development Runtime.
- In *standalone mode*, components do not try to connect to the observatory services (e.g. log and alarms send their messages to the console or a file). This operation mode is intended to be used during initial component development or when network services are not available.

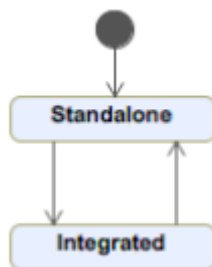


Figure 4-5 [ID 91682]: Control Mode State Machine

REQ-L3-SCS-91855: Normal Operation

The DCS shall always be in integrated mode during normal operation.

Notes: Use of local standalone mode should be minimized as much as possible.

Legacy ID: REQ-L3-SCS-0083

4.1.8 Fault Management

The previous section described some strategies to implement the control, monitoring and supervisory functions of a Component. In addition to these functions control systems that are required to exhibit robustness and reliability must address the detection and management of non-nominal operating conditions. Often the strategy for addressing fault management does not necessarily follow the structure of the control function, in which case it is convenient to have an independent implementation.

Faults can be organized in a tree structure similar to the ones used in Fault Tree Analysis (FTA) to model more complex fault states. The main purpose of a Fault is to detect, and if possible handle, non-nominal operating conditions.

The GMT SDK Core Framework allows the definition of the faults that a Component may handle. For each Fault an evaluation function can be defined to detect if the fault condition is active. Additionally, in cases in which a strategy can be established to handle the fault condition a recover function can be defined. Each Fault is associated by default with a fault state machine that governs the transitions between fault states as described in the following diagram.

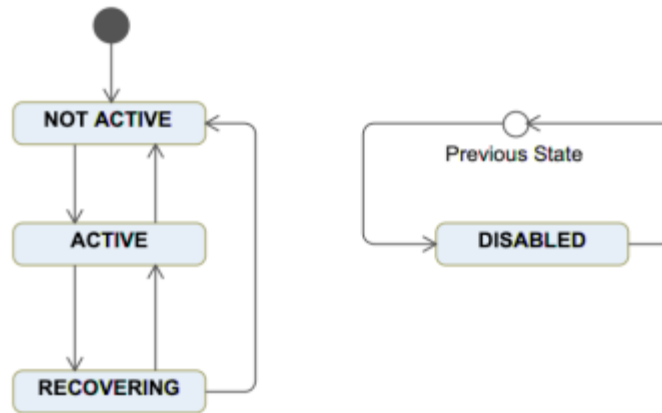


Figure 4-6 [ID 91683]: Fault State Machine

The definition of a Fault shall include the following features:

- Kind: The kind feature can have the following possible values:

Table 4-3 [ID 91862]: Kind Feature Values Definition

Fault Kind	Description
primary	Primary faults detect the occurrence of a fault condition
secondary	Represent a transfer from another fault tree
or	OR gate. The fault occurs if any of the children faults occurs
and	AND gate. The fault occurs if all of the children faults occur
xor	The fault occurs if only one of the children faults occurs
count	The fault occurs if at least count number of the children faults occurs

- Parent: The parent feature is an attribute that contains the name of the parent fault in a fault tree. If the fault is the root of the fault tree the value shall be the empty string. Root nodes can be used to connect with other fault trees secondary (transfer in) nodes.
- Level: The level of severity of the fault. Severity levels are TBD.
- Rate: The rate feature is an attribute that defines the frequency at which the fault condition is evaluated.



- Threshold: The threshold feature is an attribute that defines the number of cycles in which the fault condition occurs before the fault becomes active.
- Count: The count feature is an attribute that defines the number of children faults when the fault is of kind count.

REQ-L3-SCS-91864: Fault Definition

The DCS shall define the Faults that are detectable by the Controlled Subsystem.

Notes: The specification of the faults of a DCS is captured in the DCS System Definition Files (SDF).

REQ-L3-SCS-91866: Fault Detection

The DCS shall be able to detect the fault conditions allocated to the DCS by the fault management design of the Controlled Subsystem.

REQ-L3-SCS-91867: Fault Recovery

The DCS shall be able to recover from a Fault condition if a fault recovery function has been defined.

4.1.9 Alarm Management

Alarms are used to notify operators of operating conditions that require their attention. The GMT SDK Core Framework allows to define which Alarms are associated with a Component. For each Alarm an evaluation function has to be defined to determine if an alarm condition is active. Alarms can be arranged, grouped and connected in a tree structure.

Although the implementation mechanism is similar, Faults and Alarms have different functions and their logic are independent, although in some cases the occurrence of a Fault condition or the inability to recover from a Fault condition may require the activation of an Alarm to notify the operators. Once an Alarm condition occurs the alarm follows a life-cycle similar to the one defined in the [IEC62682 Management of alarm system for the process industries standard](#).

The life-cycle of an alarm is modeled as a state machine as described in the following diagram.

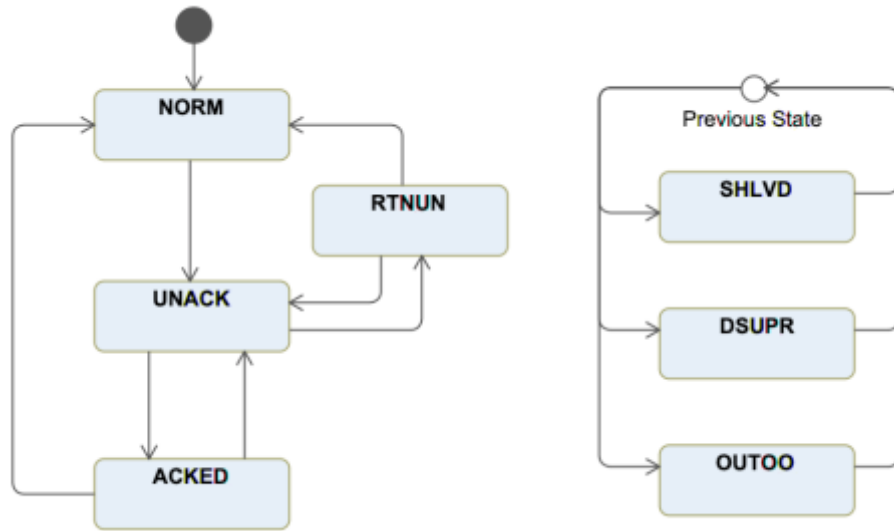


Figure 4-7 [ID 91684]: Alarm State Machine

Table 4-4 [ID 91872]: Alarm States

Alarm State	Description
NORM	Normal
UNACK	Unacknowledged
ACKED	Acknowledged
RTNUN	Returned to Normal Unacknowledged
SHLVD	Shelved
DSUPR	Design suppressed
OOSRV	Out of Service

The definition of an Alarm shall include the following features:

- Level: The level feature is an attribute that defines the severity of the alarm. Severity levels are TBD.
- Rate: The rate feature is an attribute that defines the frequency at which the alarm condition is evaluated.



- Threshold: The threshold feature is an attribute that defines the number of cycles in which the fault condition occurs before the alarm becomes active.
- Kind: The kind feature is a String attribute with the following possible values:

Table 4-5 [ID 91874]: Alarm Kind Values

Alarm Kind	Description
primary	Primary alarms detect the occurrence of an alarm condition
secondary	Represent a transfer from another alarm tree
or	OR gate. The alarm occurs if any of the children alarm occurs
and	AND gate. The alarm occurs if all of the children alarms occur
xor	The alarm occurs if only one of the children alarms occur
count	The alarm occurs if at least count number of the children alarm occurs

REQ-L3-SCS-91875: Alarm Definition

The DCS shall identify and monitor the Controlled Subsystem alarm conditions and generate an alarm event when these conditions take place.

Notes: The specification of the alarms of a DCS is captured in the DCS System Definition Files (SDF).

Legacy ID: REQ-L3-SCS-0040

REQ-L3-SCS-91876: Active Alarm Status

The DCS shall transmit any changes in the status of alarm conditions.

Legacy ID: REQ-L3-SCS-0026

REQ-L3-SCS-91877: Alarms and Operating State Consistency

The DCS shall take into account the operating states of the Controlled Subsystem when monitoring alarm conditions to avoid sending alarms when they are not significant for a given operation state.

Legacy ID: REQ-L3-SCS-0027

REQ-L3-SCS-91878: Alarm Event Information

The DCS shall notify an alarm with an alarm event that contains:

- A timestamp

- The source of the alarm condition
- A severity
- A value
- An alarm description
- Alarm state

Legacy ID: REQ-L3-SCS-0028

4.1.10 Error and Status Logging

The logging function enables to record the history of events, whether normal or abnormal, surrounding the GMT operations. Log events are intended for view and access on an operation console and stored in a persistent database.

REQ-L3-SCS-91882: Log Event Information

The DCS shall use the following log message structure:

- A timestamp
- The source of the log message
- A text explaining the event
- A message level (info, debug, trace, metric, warning, error, fatal).

Legacy ID: REQ-L3-SCS-0029

REQ-L3-SCS-91883: Logging Levels

The DCS shall use the following log message levels:

Table 4-6 [ID 91884]: Logging Levels

Logging level	Abbr	Description
fatal	FTL	fatal - errors which make the application unusable
error	ERR	error - errors that preclude to archive a specific request
warning	WRN	warning - problems that may cause unexpected results
info	INF	info - information about the general execution of the application
debug	DBG	debug - information to provide an understanding of the internal of the application

trace	TRC	trace - information that may server to identify a potential problem
metric	MET	metric - information to record performance metrics relative to the execution of the application

REQ-L3-SCS-91885: Log Events

The DCS shall record and transmit the following messages to the OCS Logging Service:

- Each DCC, PLC or embedded system events or state changes.
- Each change of configuration properties.
- Each transition in operating states.
- Each goal sent by the OCS to the DCS.
- Each state variable validity change.
- Each action done locally by operators.
- Any error shall be detected, and an error message shall be generated and communicated to the OCS.

Legacy ID: REQ-L3-SCS-0030

4.1.11 Process Configurable Properties Management

REQ-L3-SCS-91887: Configurable Properties

The DCS shall be designed to be configurable by means of a set of properties.

Notes: The specification of the configurable properties of a DCS is captured in the DCS System Definition Files (SDF).

Legacy ID: REQ-L3-SCS-0031

REQ-L3-SCS-91888: Configuration Parameters

The DCS shall provide the capability to modify any configuration property with no disturbance to the correct operation of the Controlled Subsystem.

Legacy ID: REQ-L3-SCS-0032

REQ-L3-SCS-91889: Properties Configuration



The settings which are expected to be changed, however rarely, in course of the Controlled Subsystem lifetime, should be made configurable without additional program recompilation and, preferably, without program restart.

Legacy ID: REQ-L3-SCS-0033

REQ-L3-SCS-91890: Properties Value Definition

The DCS shall define the sets (continuous or discrete) that contain the values that can be assigned to Properties.

REQ-L3-SCS-91891: Properties Data Types

The DCS shall define the data types of the Properties.

REQ-L3-SCS-91892: Properties Data Initialization

The DCS shall ensure that the Properties have an initial value that is contained in the corresponding valid set.

REQ-L3-SCS-91893: Properties Data Units

The DCS shall define the units of Properties that represent a physical magnitude.

REQ-L3-SCS-91895: Continuous Process Data Value Ranges

The DCS shall define a nominal operating range for Process Outputs, Process Inputs and State Variables that represent a continuous value.

4.1.12 Operation Support

4.1.12.1 Visualization

REQ-L3-SCS-91898: DCS Specific User Interface Elements

The DCS shall provide specialized user interface elements when the basic set contained in the User Interface Framework is not adequate for efficient operation of the system under control.

Notes: The GMT “User Interface Framework” (ui_fw) provides visualization components that target general use cases. They may be appropriate for general engineering and basic operation. However, with the DCS controlling a complex subsystem, specialized user interface components may need to be deployed.

Legacy ID: REQ-L3-SCS-0025

4.1.12.2 Data Processing

REQ-L3-SCS-91900: Data Processing Function

The DCS shall implement the data processing functions required to operate the system under control.

Legacy ID: REQ-L3-SCS-0026

4.1.12.3 High-Level Operations

REQ-L3-SCS-91902: Sequencing

The DCS shall implement sequencing functions so it can be operated from the OCS sequencing tools.

Legacy ID: REQ-L3-SCS-0027

REQ-L3-SCS-91903: Operation Workflows

The DCS shall implement operation workflows for those functions that required a collaborative interaction between the Controlled Subsystem and the operators.

Legacy ID: REQ-L3-SCS-0028

REQ-L3-SCS-91904: Operation Commands

The DCS shall implement the behaviors that implement the required changes to its State Variables.

Legacy ID: REQ-L3-SCS-0029

4.1.12.4 Diagnostics

REQ-L3-SCS-91906: Diagnosis Function

The DCS shall implement diagnosis functions to characterize non-nominal behavior of the system under control or other DCS components.

Notes: Diagnosis functions are necessary when the operational complexity of the Controlled Subsystem makes hard to understand its behavior under nominal and non-nominal conditions.

Legacy ID: REQ-L3-SCS-0030

4.1.12.5 Calibration

REQ-L3-SCS-91908: Calibration Function

The DCS shall provide calibration functions when the operation of the Controlled Subsystem requires parameters that have to be obtained after the execution of measurements.

Legacy ID: REQ-L3-SCS-0031

4.1.12.6 Quality Assessment

REQ-L3-SCS-91910: Quality Assessment Requirement

The DCS shall ensure the quality of the data being produced during the operation of the Controlled Subsystem.

Legacy ID: REQ-L3-SCS-0034

4.1.12.7 DCS Operation Parameters Definition

REQ-L3-SCS-91912: Operation tool plugins

The DCS shall implement the tools necessary to estimate the required values of its state variables needed during operation (e.g. time exposure calculator for scientific observations).

Legacy ID: REQ-L3-SCS-0035

REQ-L3-SCS-91913: Operation Tool Input Parameters

The DCS shall define the input parameters needed by the Operation tool plugins.

Legacy ID: REQ-L3-SCS-0036

4.1.12.8 DCS Data Products

REQ-L3-SCS-91915: Controlled Subsystem Specific Data Products

DCS data products shall conform with the GMT OCS Data Product Specification Document.

Legacy ID: REQ-L3-SCS-0037

4.1.12.9 DCS Controlled Devices

REQ-L3-SCS-91917: DCS Controlled Devices

The DCS shall provide descriptions of the Controlled Subsystem Plant Devices under its control. These descriptions should capture the information relevant to perform the control functions and to operate the Devices.

Notes: The SDK metamodel specifies the features (e.g. vendor, model, location) necessary to model a Device.

Legacy ID: REQ-L3-SCS-0038

REQ-L3-SCS-91918: Device Calibration Data Provenance

The DCS shall include information about the serial number and location of Controlled Subsystem Plant Devices that can be exchanged so the provenance for the calibration data can be ensured.

Legacy ID: REQ-L3-SCS-0039

4.1.12.9.1 Computing Resources Management

REQ-L3-SCS-91920: Remote Control Functions

The DCS shall provide remote control functions for its computing resources (e.g. reboot, configure, start, stop, switch to standalone/integrated control mode).

Notes: Remote control functions shall comply with the safety rules of the GMT site.

Legacy ID: REQ-L3-SCS-0034

REQ-L3-SCS-91921: Monitoring Function

The DCS shall provide the capability to monitor DCC functions and equipment.

Legacy ID: REQ-L3-SCS-0035

REQ-L3-SCS-91922: Equipment to be Monitored

The DCS shall monitor at least:

- DCS hardware (DCC, PLC) and software
- Device Controllers
- Fieldbus networks
- Interface with OCS

Legacy ID: REQ-L3-SCS-0036

REQ-L3-SCS-91923: Monitored Equipment Status

The DCS shall provide the operational status (operational, partially operational or not operational) of any monitored equipment.

Legacy ID: REQ-L3-SCS-0037

REQ-L3-SCS-91924: Equipment Performance Monitoring

The DCS shall provide the capability to monitor the performance of the DCS equipment.

Notes: Performance information such as field bus status, CPU load and memory usage or network bandwidth utilization shall be recorded.

Legacy ID: REQ-L3-SCS-0038

REQ-L3-SCS-91925: Monitoring Function Health

The DCS monitoring function shall include self-tests and live tests.

Legacy ID: REQ-L3-SCS-0039

4.2 Non-Functional Requirements

4.2.1 Security

REQ-L3-SCS-91928: DCS Access Control

DCS shall restrict access to authorized systems/people.

Notes: The Core Framework defines an Access Control List (ACL) for each Component that defines which DCS/user profiles can set goals on each Component state variables.

Legacy ID: REQ-L3-SCS-0040

4.2.2 Performance

REQ-L3-SCS-91930: Availability

The DCS shall be compliant with the availability allocations of the RAMS requirements of the Controlled Subsystem and GMT.

Legacy ID: REQ-L3-SCS-0041

REQ-L3-SCS-91931: Sensor Data Transport QoS

The DCS shall ensure that the duration for update of information from sensors to the OCS Control Network shall meet the Quality of Service requirements as defined in each corresponding SDF.

Legacy ID: REQ-L3-SCS-0048

REQ-L3-SCS-91932: Command Transport QoS

The DCS shall ensure that the duration for request of change to the DCS state from OCS Control Network to actuators shall meet the QoS requirements as defined in each corresponding SDF.

Legacy ID: REQ-L3-SCS-0049

REQ-L3-SCS-91933: Wavefront Control Transport QoS

The DCS shall meet the QoS requirements as defined in each corresponding SDF when participating in the GMT wavefront control system.

Legacy ID: REQ-L3-SCS-0050

REQ-L3-SCS-91934: Time Synchronization

The DCS shall be synchronized with the GMT central time reference.

Legacy ID: REQ-L3-SCS-0051

4.2.3 Availability

REQ-L3-SCS-91936: Hot Swapping

The DCS shall use hot swapping whenever it is required by the RAM analysis of the Controlled Subsystem.

Legacy ID: REQ-L3-SCS-0052

REQ-L3-SCS-91937: Redundancy

The DCS shall use redundancy whenever it is required by the RAM analysis of the Controlled Subsystem.

Legacy ID: REQ-L3-SCS-0053

4.2.4 Diagnostics

REQ-L3-SCS-91939: DCC and PLC Diagnosis

The DCS computers, PC and PLC based DCCs, and equipment shall have provisions for self-diagnostics.

Notes: DCCs and equipment should repeat self-checks at scheduled times.

Legacy ID: REQ-L3-SCS-0054

4.2.5 Safety

REQ-L3-SCS-91941: Safe Operation

The DCS shall be able to autonomously maintain safe operation of the Controlled Subsystem Devices in case of loss of communication with the OCS.

Legacy ID: REQ-L3-SCS-0055

REQ-L3-SCS-91942: Software Safety

The DCS shall not have ultimate responsibility for the safety of the Controlled Subsystem or persons.

Legacy ID: REQ-L3-SCS-0056

REQ-L3-SCS-91943: Limit Protection

The DCS shall have built-in absolute-limit protection to prevent errors.

Legacy ID: REQ-L3-SCS-0057

REQ-L3-SCS-91944: Limit Protection

The DCS shall have timeouts to ensure correct operation in case of OCS failure.

Legacy ID: REQ-L3-SCS-0058

REQ-L3-SCS-91945: Plant State Non-Assumption Strategy

The DCS shall avoid any assumption about the status of the Controlled Subsystem equipment or the plant during the start-up process or normal operation.

Notes: Although the DCS records the status of the equipment when it was powered off, human intervention may have change the configuration of the Controlled Subsystem equipment.

Legacy ID: REQ-L3-SCS-0059

REQ-L3-SCS-91946: Fieldbuses Independence

The DCS fieldbus shall be separated from the ISS fieldbus in such a way that a hardware failure to the DCS components will not affect the Local ISS for that Controlled Subsystem, which should continue operating autonomously.

Legacy ID: REQ-L3-SCS-0060

Figure [ID 91685] and Figure [ID 91686] represents the logical and physical architecture of the ISS. The ISS Safety Networks (global and local) are independent of the OCS Control Network.

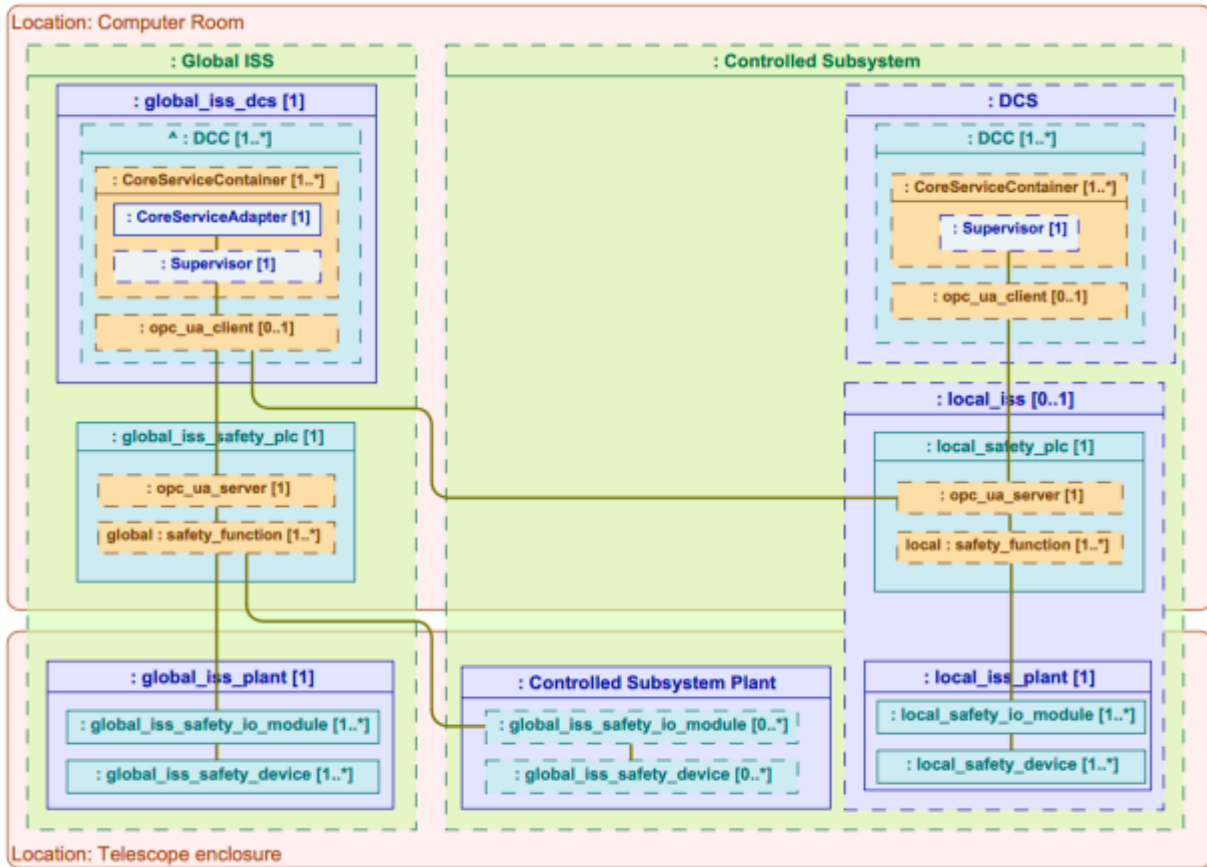


Figure 4-8 [ID 91685]: ISS Architecture – Logical View

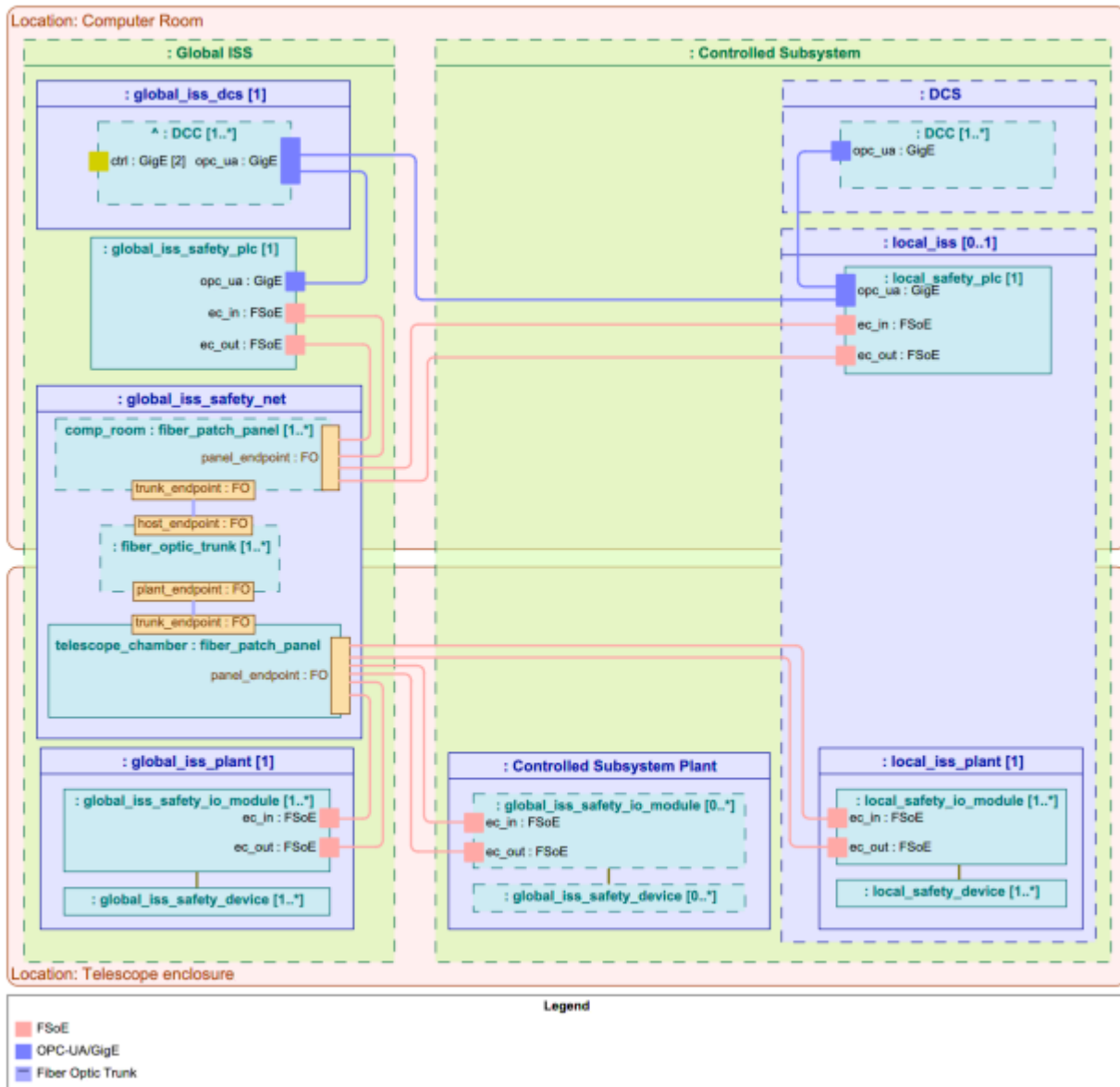


Figure 4-9 [ID 91686]: ISS Architecture – Physical View

4.2.6 Naming Convention

REQ-L3-SCS-91949: Unique Naming

Each DCS, DCS package and DCS component shall have a unique name.

Legacy ID: REQ-L3-SCS-0011



REQ-L3-SCS-91950: Use of Abbreviations

Abbreviations used in the composition of names shall be contained in the glossary n-grams.

Notes: See OCS glossary for reference.

Legacy ID: REQ-L3-SCS-0150

REQ-L3-SCS-91951: DCS Naming

Each DCS shall be named according to the following format:

```
<SUBS>_dcs      # where
<SUB>: Abbreviation of the DCS
Example: m1_dcs
new RegExp ///^#{SUBS}_dcs$///.test dcs.name #formal naming test
```

Legacy ID: REQ-L3-SCS-0151

REQ-L3-SCS-91952: Package Naming

Each DCS Package shall be named according to the following format:

```
<SUBS>_<ID>_<CAT>_<MCLASS>      # where
<SUB>: Abbreviation of the DCS
<CAT>: Abbreviation of the functional category of the package
as defined in the glossary n-grams
<ID>: Optional. Abbreviation of the specific package in case
more than one of the same class exists
<MCLASS>: Metamodel class abbreviation (in this case Packages -
> pkg)
Example: m1_ctrl_pkg, inst_vacuum_ctrl_pkg, inst_cryo_ctrl_pkg
new RegExp ///#{SUBS}_({CATS.join "|"})_pkg$///.test
pkg.name #formal naming test
```

Legacy ID: REQ-L3-SCS-0152

REQ-L3-SCS-91953: Component Naming

Each DCS Component shall be named according to the following format:

```
<SUBS>_<CMP>_<MCLASS>      # where
<SUB>: Abbreviation of the DCS
<CMP>: Component abbreviation
<MCLASS>: Metamodel class abbreviation of the
component class (e.g. Controller -> ctrl) as defined in the glossary
n-grams
```



```
Example: m1_support_actuator_ctrl  
  
new RegExp ///#{SUBS}_[a-z]_+_(#{MCLASS.join "|"})$///.test  
cmp.name      #formal naming test
```

Legacy ID: REQ-L3-SCS-0153

4.3 Hardware Requirements

The hardware standards described here apply to all the electronic equipment and electronic components that will be used in a DCS. This includes the control of the instruments and other equipment that may be developed outside of the project office. DCSs developed externally shall be compliant with these standards.

4.3.1 Device Control Computer

The DCC (Device Control Computer) is a standardized computer specified by GMTO to be used in the deployment of the DCS. The DCC is connected to the OCS Networks. Each DCC hosts the software components that implement the functional and physical part of the control and data acquisition functions of the Controlled Subsystem devices, i.e. the Controlled Subsystem Plant in the Figure [ID 91674]. Each DCC includes a processor and an interface to the I/O modules required to connect to hardware Devices. I/O interfaces are either I/O adapters within the DCC (e.g. detector controller interface cards), or use a field bus to connect with remote I/O modules (i.e. Motion drives or discrete I/O signal interfaces).

Two types of DCCs are supported:

- PC-based DCCs Intel-based Industrial Computers running Linux with real time extensions (RT_PREEMPT) are used in the following use cases:
 - o High bandwidth telemetry is required
 - o The DCS has to be integrated with the low latency network.
 - o High computing performance is required.
 - o Implementing complex operation scenarios is required
 - o Control loops or data acquisition rates are faster than 100 Hz.
- PLC-based DCCs Programmable Logic Controllers (PLCs) are used in the following use cases:
 - o The bandwidth necessary for telemetry is low.
 - o Simple supervision, coordination or operation scenarios.
 - o The implementation of Safety Functions as specified in the corresponding Local ISS.

When PLC-based DCCs are used, a Master Supervisor and a set of Proxies is deployed in a PC-based DCC that communicates with the PLC(s) using OPC UA. The purpose of the Proxies is to map the data exported by the PCL-based DCCs into SDK compliant Components that can be integrated with the OCS Core Services.

The development of this DCC Master Supervisor depends on the procurement agreement and may be provided by GMTO if the DCS provider is not familiar with the technologies used for DCS / OCS integration.

REQ-L3-SCS-91960: DCS Application Software

The DCS DCC shall host the supervisory and control DCS software Components.

REQ-L3-SCS-91961: DCS OCS Fieldbus Software

The DCS DCC shall host the software that enables the communication with the fieldbus (e.g. EtherCAT master).

REQ-L3-SCS-91963: DCS OCS Network Interface

The DCS DCC shall host the physical interfaces that connects the DCC with the OCS Network(s).

REQ-L3-SCS-91964: DCS Master Supervisor Deployment

The DCS shall have one and only one DCS Master Supervisor deployed in one of the DCS DCCs.

Legacy ID: REQ-L3-SCS-0086

REQ-L3-SCS-91965: DCC Physical Interface

The DCS DCC shall provide at least two ports to connect to the OCS Control Network.

Legacy ID: REQ-L3-SCS-0087

REQ-L3-SCS-91966: DCC – DCS Integration

The DCS DCC shall be integrated into the DCS.

Legacy ID: REQ-L3-SCS-0088

REQ-L3-SCS-91967: DCC Location

The DCS DCC shall be installed in the GMT Computer Room.

Legacy ID: REQ-L3-SCS-0089

REQ-L3-SCS-91968: Standard DCC Power Consumption

The DCS DCC shall not exceed 200W (TBC) of power consumption.

Legacy ID: REQ-L3-SCS-0090

REQ-L3-SCS-91969: DCC with Hardware Accelerator Power Consumption

The DCS DCC shall not exceed 2000W (TBC) in case that hardware accelerators (e.g. GPUs) are required.

REQ-L3-SCS-91971: DCC Physical Envelope

The DCS DCCs shall not use more than 2U (TBC) in a 19” rack.

Legacy ID: REQ-L3-SCS-0092

REQ-L3-SCS-91972: DCC Processor

The DCS DCCs shall use an Intel compatible processor.

Legacy ID: REQ-L3-SCS-0093

REQ-L3-SCS-91973: DCC Physical Environment

The DCS DCC shall comply with the Environment requirements (seismic, etc). TBD

Legacy ID: REQ-L3-SCS-0092

4.3.2 PLC-based DCC

In addition to the DCC requirements, PLC-based DCCs must comply with the following requirements:

REQ-L3-SCS-91976: Physical interface

The DCS physical interface between PLC-based DCCs and the DCS Master Supervisor DCC or any other PLC(s) shall be Gigabit Ethernet Internet.

Legacy ID: REQ-L3-SCS-0103

REQ-L3-SCS-91977: Data Interface

The DCS data interface protocol between PLC-based DCCs and the DCS Master Supervisor DCC or any other PLC(s) shall be OPC UA.

Legacy ID: REQ-L3-SCS-0107

REQ-L3-SCS-91978: PLC

The DCS PLC-based DCC shall be Beckhoff series TBD or equivalent.

Legacy ID: REQ-L3-SCS-0108

REQ-L3-SCS-91979: Standard Products

The DCS PLC-based DCC, I/O Modules and field buses shall be selected from SDK supported products list.

Legacy ID: REQ-L3-SCS-0109

4.3.3 PC-based DCC

In addition to the DCC requirements, PC-based DCCs must comply with the following requirements:

REQ-L3-SCS-91982: PC-based Hardware Architecture

The DCS PC-based DCC shall use Device Control Computers (DCC) based on Intel industrial computers with PCI Express bus system and adequate interface to fieldbus and OCS Control Network.

Legacy ID: REQ-L3-SCS-0111

REQ-L3-SCS-91983: DCS OCS Communication Software

The DCS DCC shall host the SDK framework that enable communication between the DCS and the rest of the OCS.

REQ-L3-SCS-91984: DCC Standard Products

The DCS PC-based DCC, I/O Modules, terminal blocks and field buses shall be selected from the SDK supported products list.

Legacy ID: REQ-L3-SCS-0112

4.3.4 Local ISS Safety PLC

In addition to the DCC requirements, the Local ISS Safety PLC must comply with the following requirements:

REQ-L3-SCS-91987: Safety PLC

The Local ISS Safety PLC shall be a TwinSafe enabled Beckhoff series TBD or equivalent.

REQ-L3-SCS-91988: Physical Interface

The physical interface between the DCS Master Supervisor DCC and the Local ISS Safety PLC(s) shall be Gigabit Ethernet.

REQ-L3-SCS-91989: Data Interface

The data interface protocol between the DCS Master Supervisor DCC and the Local ISS Safety PLC(s) controllers shall be OPC UA.

4.3.5 Deployment Strategies

The DCS reference architecture defines a generic design that shall be refined and adapted to each specific Controlled Subsystem. A specific DCS may need more than one DCC or may use multiple DCC types (PLC-based, PC-based). The next sections describe which deployment strategies can be used to decide how to deploy the DCS software Components into the DCS DCCs.

4.3.5.1 Single-Tier Deployment

In a Single-Tier deployment, all the Controllers of the DCS are deployed in a single PC-based DCC. The DCC is connected with the Controlled Subsystem Plant input/output modules using the EtherCAT fieldbus. The DCC host the interfaces needed to communicate with the OCS Network.

The Device Control Package is fully developed using the OCS SDK with allows the DCS Controllers to communicate with the OCS.

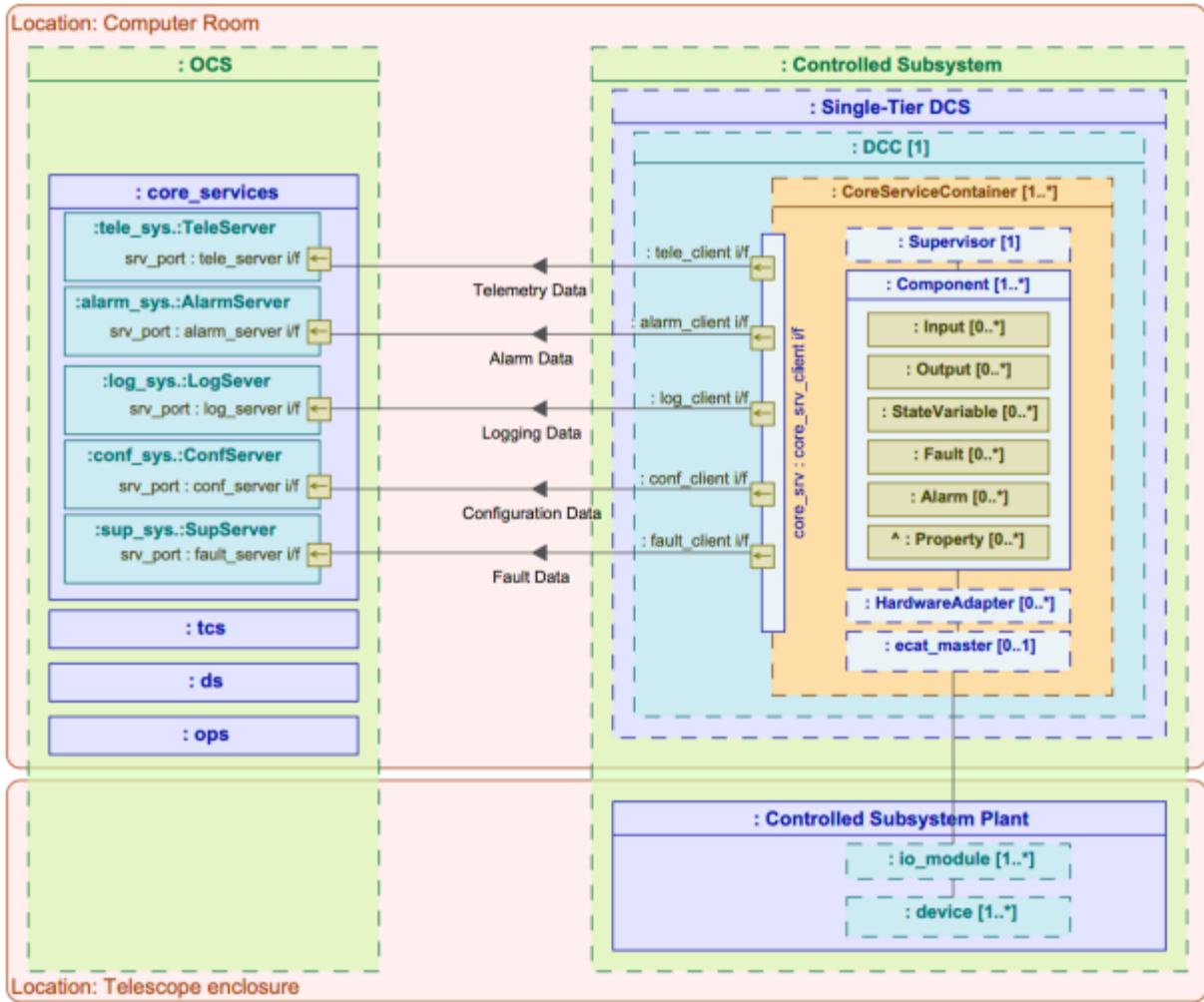


Figure 4-10 [ID 91687]: Single-Tier Deployment Strategy Logical View

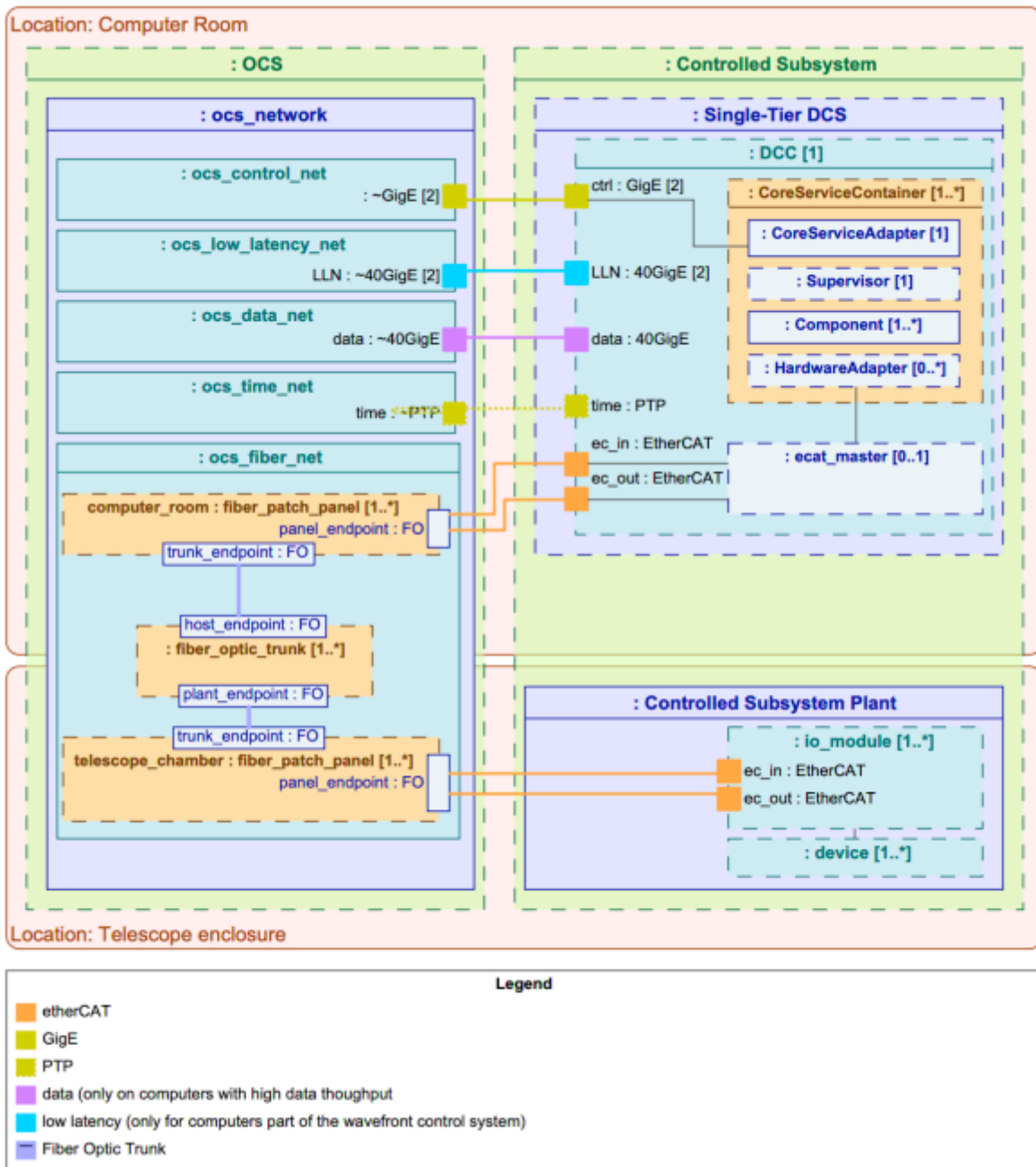


Figure 4-11 [ID 91688]: Single-Tier DCS Deployment Strategy Physical View

4.3.5.2 Two-Tier Deployment (Hybrid Platform)

In a Two-Tier Deployment (Hybrid platform) the Control function is divided in two parts, (1) the High-Level Control that is built with the OCS SDK, provides supervision functions and interfaces with the OCS using the SDK Core Frameworks and (2) the Low-Level Control that is built the PLC programming environment, provides control functions and interfaces with the Controlled Subsystem Plant via a

fieldbus. The High-Level and Low-Level communicate with each other using the OPC-UA protocol, where the Low-Level Controller includes an OPC-UA server and the High-Level Controller includes an OPC-UA client.

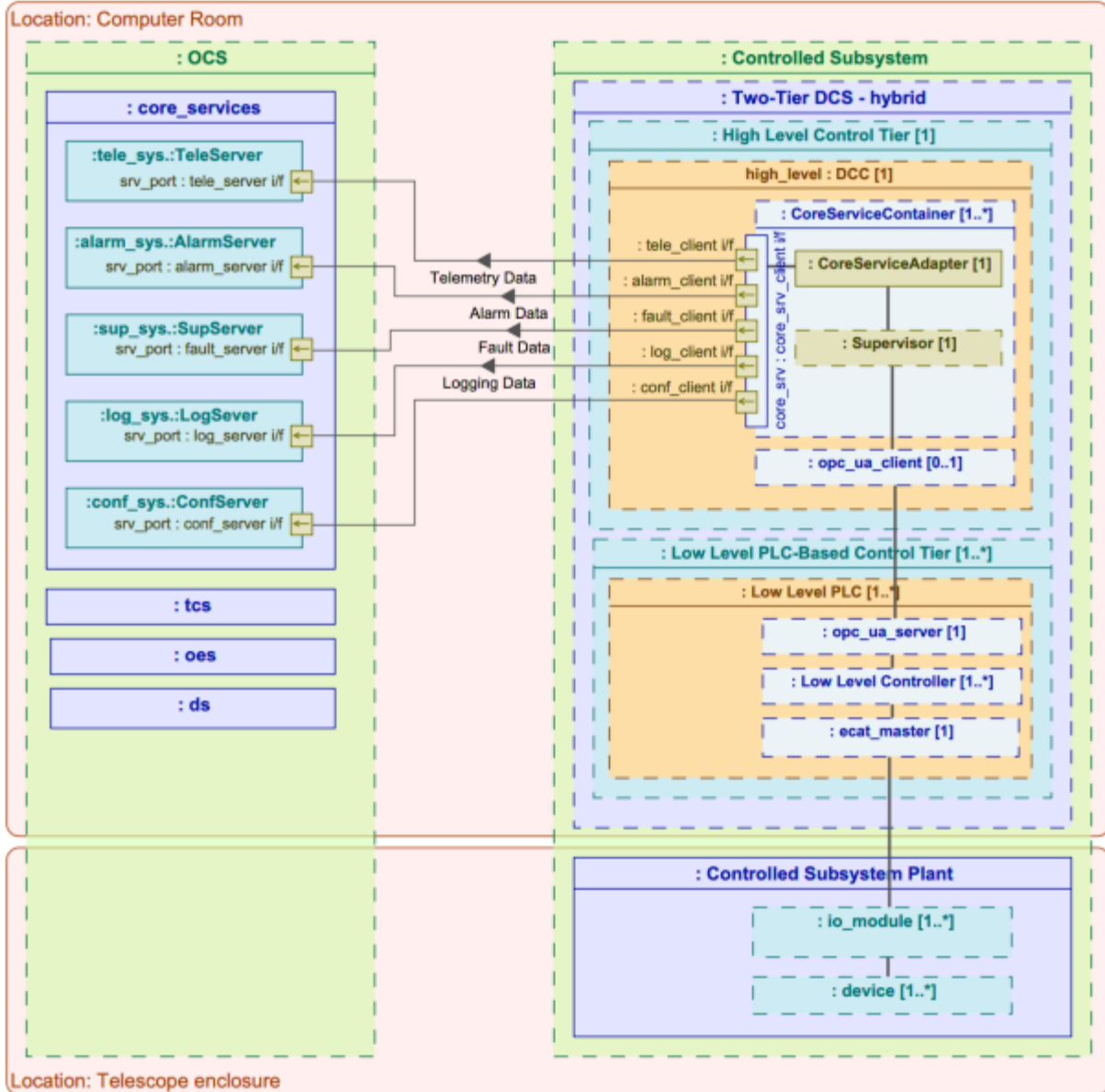


Figure 4-12 [ID 91689]: Two-Tier DCS with Hybrid PLC/PC Logical View

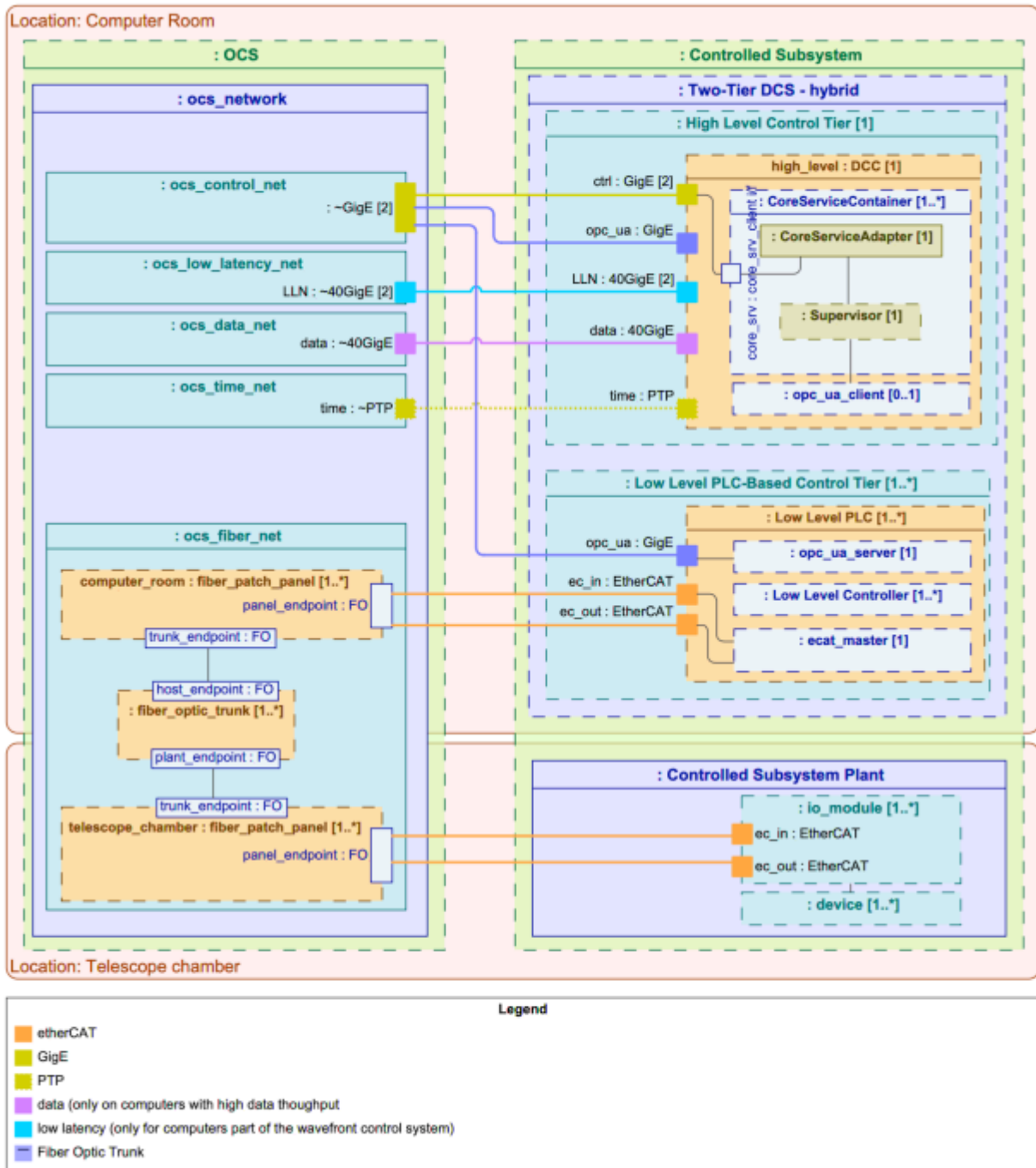


Figure 4-13 [ID 91690]: Two-Tier Controller with Hybrid PLC/Supervisor Physical View

4.3.5.3 Two-Tier Deployment (SDK Platform)

In a Two-Tier Deployment (SDK platform) the Control function is divided in two parts, (1) the High-Level Control that is built with the OCS SDK, provides supervision functions and interfaces with the OCS using the SDK Core Frameworks and (2) the Low-Level Control that is also built with the OCS SDK, provides control functions and interfaces with the Controlled Subsystem Plant via a fieldbus. The High-Level and Low-Level communicate with each other using the standard SDK Connectors.

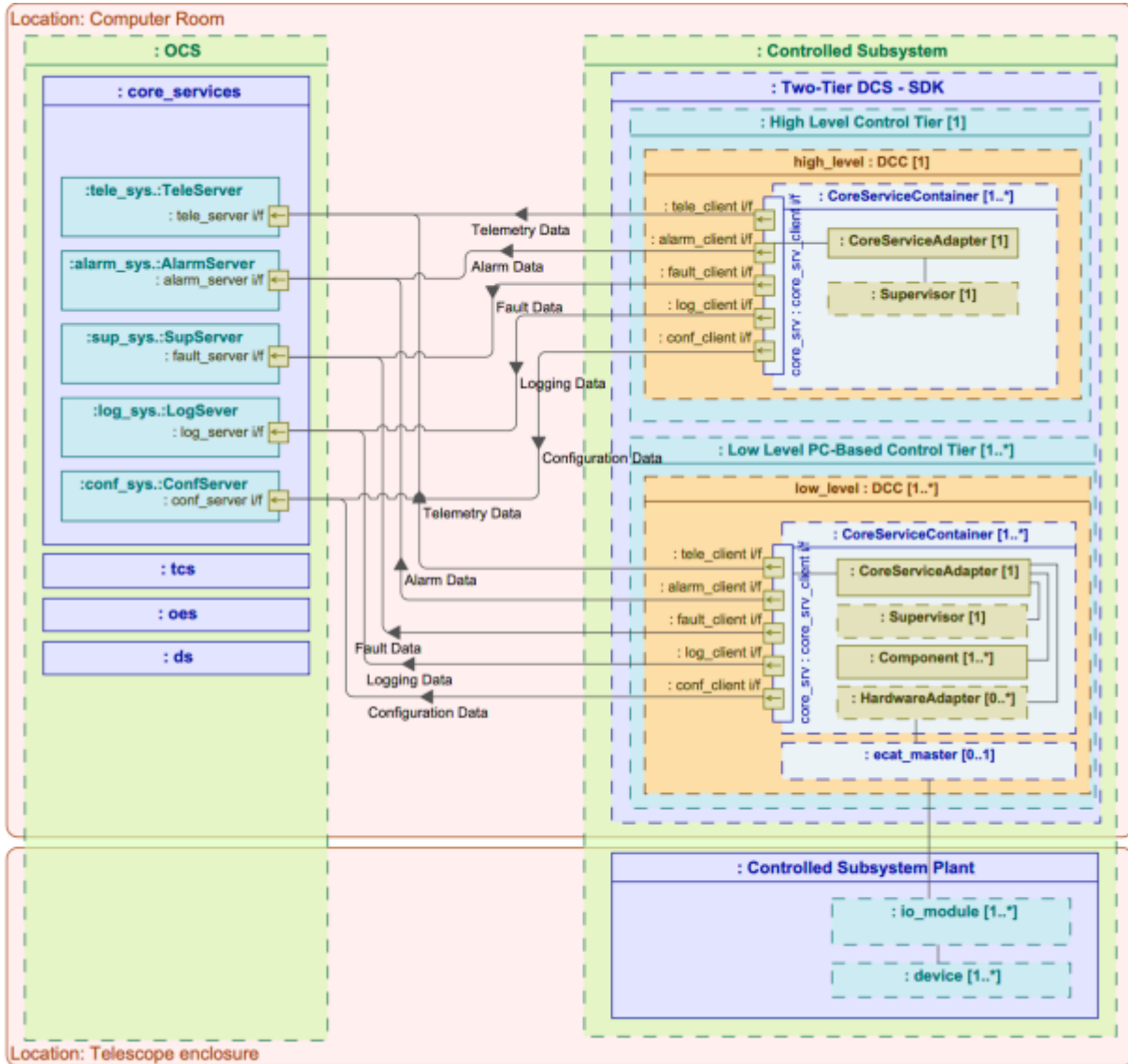


Figure 4-14 [ID 91691]: Two-Tier Deployment (SDK) Logical View

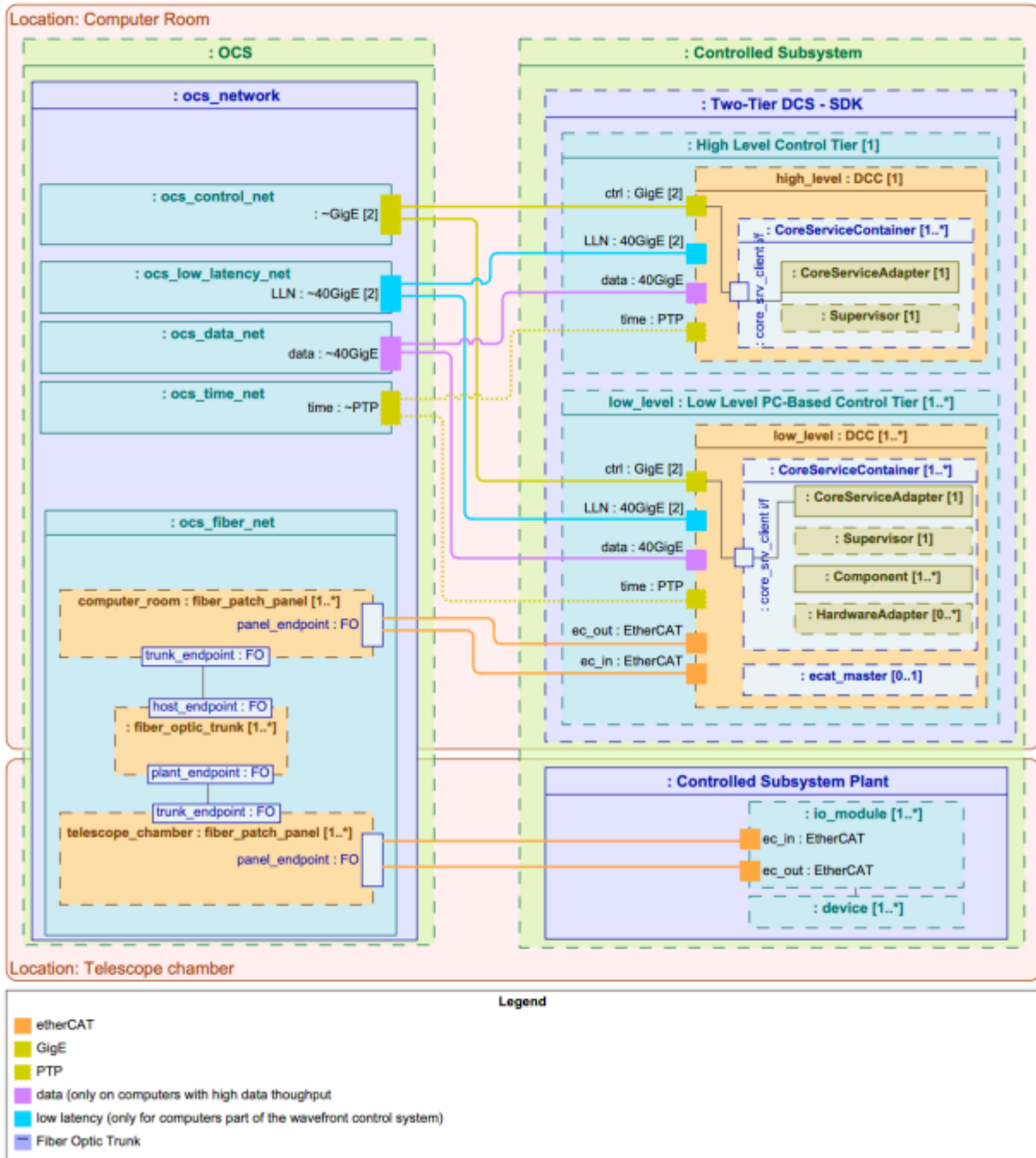


Figure 4-15 [ID 91692]: Two-Tier Deployment (SDK) – Physical View

4.3.6 Computing Resources Sizing

REQ-L3-SCS-92000: Computing Resources

The DCS computing resources shall be adequate to support the correct operation of the DCS.

Legacy ID: REQ-L3-SCS-0042

REQ-L3-SCS-92001: Computing Resource Sizing

The DCS computing resources shall provide sufficient margins to allow adding additional functions.

Legacy ID: REQ-L3-SCS-0043

REQ-L3-SCS-92002: Load Factor Characterization During Acceptance Tests

The acceptance tests shall provide measurements of the load of the DCS computing and network resources during operation.

Legacy ID: REQ-L3-SCS-0044

REQ-L3-SCS-92003: IO Module Sizing

The recommended additional reserve space for IO Modules per DIN rail in the allocated *Standard Electronics Cabinet* (SEC) should be more than 20% (TBC).

Notes: This is a design guideline. Depending on the complexity of the Controlled Subsystem this might not be realizable because the space in the SEC is limited.

Legacy ID: REQ-L3-SCS-0045

REQ-L3-SCS-92004: Not equipped I/O Channel Additional Reserve

The recommended additional reserve I/O channels (not equipped) per type should be more than 20% (TBC).

Notes: This is a design guideline. Depending on the complexity of the Controlled Subsystem this might not be realizable because the space in the SEC is limited.

Legacy ID: REQ-L3-SCS-0046

REQ-L3-SCS-92005: Equipped I/O Channel Additional Reserve

The recommended additional reserve I/O channels (equipped) per type should be more than 5% (TBC).

Notes: This is a design guideline. Depending on the complexity of the Controlled Subsystem this might not be realizable because the space in the SEC is limited.

Legacy ID: REQ-L3-SCS-0047

4.3.7 Field Devices Interface

REQ-L3-SCS-92007: DCS General Interface

The interface between the DCC (both PLC and PC based) and the general field Devices shall be the EtherCAT fieldbus.

Notes: General field Devices refers to the Devices usually found in industrial automation systems (e.g. motion control, process control). The interface for specialized Devices like Technical Cameras or Building Automation Devices are defined in specific requirements in this document.

Legacy ID: REQ-L3-SCS-0093

REQ-L3-SCS-92008: DCS Building Automation Field Devices Interface

The interface between PLC based DCCs and field Devices that are part of a Building Automation System shall be BACnet.

Notes: Building automation Devices refers to the field devices used in the centralized control of a building (e.g. heating, ventilation and air conditioning (HVAC), lighting).

REQ-L3-SCS-92009: DCS Safety Field Devices Interface

The interface between the safety PLCs and the field Devices shall be the EtherCAT FailSafe over EtherCAT (FSoE).

REQ-L3-SCS-92010: DCS Fieldbus Layout

The fieldbus IO Modules shall be installed in the Observatory plant SECs or smaller boxes when the performance requirements for the electrical signal path requires it.

Legacy ID: REQ-L3-SCS-0094

REQ-L3-SCS-92011: DCS Fieldbus Cabling

The cabling between the first IO Module of the DCS and the rest shall be an optical fiber. GMTO provides fiber links between the Electronics Room and the enclosure building as part of the OCS Field Network. The cabling between the field IO Modules shall be Ethernet UTP CAT 6 (TBC).

Legacy ID: REQ-L3-SCS-0095

REQ-L3-SCS-92012: DCS Cable rRdundancy

The fieldbus layout shall be designed to support cable redundancy as stated in EtherCAT protocol.

Legacy ID: REQ-L3-SCS-0096

4.3.8 Motion Control Deployment

REQ-L3-SCS-92014: Motion Control Deployment Mode

The DCS shall use one of the recommended control deployment modes defined in Table [ID 92015] for each individual degree of freedom.

Notes: Motion drives are assumed to comply the IEC 61800-7-201 and IEC61800-7-301 mapping to EtherCAT.

Legacy ID: REQ-L3-SCS-0085

Table 4-7 [ID 92015]: Motion Control Deployment Mode

Control Deployment mode	Description
PVT	Position, Velocity and Torque control loops are implemented on a motion drive. This mode is recommended for single degree of freedom controllers with single or dual encoder feedback. In this mode encoder feedback is connected to the drive
P-VT	The Position control loop is implemented in the DCC, while the Velocity and Torque loops are implemented in the drive. This mode is recommended when the position feedback cannot be connected directly to the drive.
PV-T	The Position and Velocity control loops are implemented in the DCC, while the Torque control loops is implemented in the drive. This mode is necessary when several actuators act on the same axis.

REQ-L3-SCS-92017: Motion Control

Motion control functions shall be implemented using motion drives that implement the CiA 402 profiles as defined in IEC 61800-7-201 / 301.

Notes: This is especially recommended in the case of single axis with single or dual feedback and single actuator.

Legacy ID: REQ-L3-SCS-0097

REQ-L3-SCS-92018: Motion Control Signals

Motion control related signals shall be connected to the drive auxiliary inputs/outputs.

Notes: Example of motion signals are encoder feedback or home switches.

Legacy ID: REQ-L3-SCS-0098

REQ-L3-SCS-92019: Motion Control Loops

Depending on the Motion Deployment mode (see REQ-L3-SCS-0085), motion control loops shall be closed in the drives using one of the standard profiles defined in IEC 61800-7-201/301.

Legacy ID: REQ-L3-SCS-0099

4.3.9 DCS Software Platform

REQ-L3-SCS-92021: DCS Software Platform

Each DCC shall be configured by the DCS developer using the GMT SDK provided by GMTO.

Legacy ID: REQ-L3-SCS-0100

4.3.10 Technical Cameras Interface

REQ-L3-SCS-92024: Technical Camera Data Interface

The interface between the DCC and a Technical Camera shall be GenICam (TBC).

Notes: GigE Vision, USB3 Vision and CameraLink cameras with GenICam profile may be also acceptable (TBC).

Legacy ID: REQ-L3-SCS-0113

4.3.11 Science CCD Detectors Interface

REQ-L3-SCS-92026: Science CCD Detector Data Interface

The interface between the DCC and a science CCD detector shall be TBD.

Legacy ID: REQ-L3-SCS-0113

4.3.12 Control Cabinets

REQ-L3-SCS-92028: Standard Electronics Cabinets

DCS field hardware shall be installed in the assigned *Standard Electronics Cabinet (SEC)* as defined in the *GMT Electronics Standards*, in the SEC DRDs

Legacy ID: REQ-L3-SCS-0113

REQ-L3-SCS-92029: DCC and PLC Location

PC-based and PLC-based DCCs shall be installed in the Computer Room electronics cabinets.

Legacy ID: REQ-L3-SCS-0114

REQ-L3-SCS-92030: DCC and I/O Module Interface

DCCs and field I/O Modules are connected using the OCS Field Network fiber trunk lines.

Notes: The GMT control network is a combination of fiber distribution units, fiber transceivers and CAT6 electrical cable to field elements.

Legacy ID: REQ-L3-SCS-0115

4.3.13 Control Signal Cabling Rules

REQ-L3-SCS-92032: Cabling Rules

DCS cabling shall be compliant with the *GMT Electronics Standards* and *SEC User DRD* for signal cabling and routing.

Legacy ID: REQ-L3-SCS-0116

REQ-L3-SCS-92033: Controlled Subsystem Plant Single Connection Point

A particular Controlled Subsystem Plant signal shall not be connected to different DCSs.

Notes: In the case in which more than one DCS required access to a signal, the corresponding data shall be transmitted through the OCS Control Network.

Legacy ID: REQ-L3-SCS-0117

REQ-L3-SCS-92034: Long Distance Signal Connection

If the Controlled Subsystem Plant equipment and the DCS I/O Modules connected to it are far away from each other, then an optical fiber connection shall be used.

Legacy ID: REQ-L3-SCS-0119

REQ-L3-SCS-92035: Types of Cables

The cables used to connect the DCS with the Controlled Subsystem Plant shall be compliant with the *GMT Electronics Standards* and *SEC User DRD*.

Legacy ID: REQ-L3-SCS-0120

REQ-L3-SCS-92036: Analog Signal Cabling

Analog signal cabling shall be compliant with the *GMT Electronics Standards* and *SEC User DRD*.

Legacy ID: REQ-L3-SCS-0121

REQ-L3-SCS-92037: Signal Standards

Signals ranges and types shall be compliant with the *GMT Electronics Standards* and *SEC User DRD*.

Legacy ID: REQ-L3-SCS-0121

4.3.14 Networking

REQ-L3-SCS-92039: External Network Connection Prevention

The DCS shall not be connected to any communication network external to the OCS.

REQ-L3-SCS-92040: Internal Network Prevention

The DCS shall have no internal networking equipment.

4.4 Software Requirements

A set of software packages, named GMT Software Development Kit (SDK), is distributed by GMTO for the development, test and operation of the DCS. This package includes the required Core Frameworks and Observatory Core Services distribution.

REQ-L3-SCS-92043: DCS Communication with the OCS

The DCS shall use the GMT Core Framework for the communication to/from DCS Controllers and Supervisors.

Legacy ID: REQ-L3-SCS-0062

4.4.1 Operating Systems

REQ-L3-SCS-92045: DCC Operating System

The DCS shall use Linux, CentOS 8 or later (TBC) as its Operating System for the DCC.

Legacy ID: REQ-L3-SCS-0063

4.4.2 Programming Languages and Tools

REQ-L3-SCS-92047: Software Version Control Tool

The DCS shall use the software version control tool git.

Legacy ID: REQ-L3-SCS-0064

REQ-L3-SCS-92048: Software Repository

The DCS software repository shall be located in GMT GitHub private account.

REQ-L3-SCS-92049: Software Collaboration

The DCS shall use GitHub to submit new versions for acceptance and to manage issues related to the DCS software.

REQ-L3-SCS-92050: PLC-based DCC Programming Language

The DCS PLCs shall be programmed using IEC 61131-3.

Legacy ID: REQ-L3-SCS-0065

REQ-L3-SCS-92051: PLC-based DCC Motion Control

The DCS PLC motion functions shall be implemented using the PLCOpen Motion Control standard.

Legacy ID: REQ-L3-SCS-0066

REQ-L3-SCS-92052: PLC-based DCC Communications

The DCS PLCs shall implement and OPC UA server to enable communication from/to the DCC High-Level Master Supervisor.

Legacy ID: REQ-L3-SCS-0067

REQ-L3-SCS-92053: Inter PLC-based DCC Communications

The DCS PLCs shall use OPC UA for data exchange between them.

REQ-L3-SCS-92054: PLC-based DCC Software

The DCS PLC software shall be programmed using TwinCAT v3.1.

Legacy ID: REQ-L3-SCS-0068

REQ-L3-SCS-92055: PLC-based DCC Safety Software

The DCS PLC Safety software shall be programmed using TwinSafe/TwinCAT v3.1

REQ-L3-SCS-92056: PC-based DCC Software

The DCS shall use the GMT SDK software and environment to develop and test the PC based DCC software.

Legacy ID: REQ-L3-SCS-0069

REQ-L3-SCS-92057: PC-based DCC Fieldbus Master

The DCS shall use the igH etherCAT master in order to acquire the process image of the field devices connected to the fieldbus.

Notes: The I/O Framework provided by GMTO provides a simplified way of accessing the Fieldbus process image.

Legacy ID: REQ-L3-SCS-0070

REQ-L3-SCS-92058: Middleware Agnostic

The DCS Components shall be independent of the communication middleware used.

Notes: The SDK provides mechanisms that allow the DCS to communicate with the OCS independently of the middleware.

Legacy ID: REQ-L3-SCS-0072

REQ-L3-SCS-92059: Distributed Middleware Transport

The DCS shall use Nanomsg (TBC) for the communication between distributed Components.

Notes: The GMT Core Framework provides independence of the middleware and is the recommended way of implementing distributed communications.

Legacy ID: REQ-L3-SCS-0073

REQ-L3-SCS-92060: Distributed Middleware Serialization

The DCS shall use MessagePack (TBC) for the serialization/deserialization of transmitted data packages.

Notes: The GMT Core Framework provides independence of the serialization format and is the recommended way of implementing distributed communications.

Legacy ID: REQ-L3-SCS-0074

REQ-L3-SCS-92061: Programming Languages

The DCS shall use the following programming languages development of its software:

- ANSI C++17 (TBC) for performance sensitive application programming in the DCC
- ANSI C c99 (TBC) for driver programming in the DCC
- Python 3 for general programming
- JavaScript ES6 or Coffeescript 2.5.x for user interface programming and modeling
- IEC 61131-3 for PLC programming

Legacy ID: REQ-L3-SCS-0075

REQ-L3-SCS-92062: User Interface Components

The DCS user interfaces components shall be developed according to the W3C Web Component standard.

Legacy ID: REQ-L3-SCS-0076

4.4.3 Modeling Requirements

System Definition Files (SDF) are used to capture the formal specification of a DCS. For this reason, SDFs play a key role in the specification, testing and validation of the DCS architecture.

SDFs are written in a Domain Specific Language (DSL). A DSL is a computer programming language of limited expressiveness focused on a particular domain. A DSL facilitates productivity and communication with domain experts and DSL stakeholders. SDFs are ASCII files that are parsed and stored in the semantic model database and processed for consistency and completeness. They are hosted in the GMTO private repository in Github, for access by DCS developers and revision control.

The concrete syntax of the SDFs is provided by the DSL, while the semantics are given by a set of models (metamodels) following the Object Management Group Meta Object Facility architecture.

The GMT SDK provides tools for:

- SDF skeleton generation
- DCS component skeletons and scaffolding generation



- DCS build dependencies specification
- Metamodel and Model conformance validation
- Test procedures and test data generation
- Stage-gate document generation
- Project progress reporting
- DCS deployment
- Interface document generation

The SDFs are one of the deliverables of the DCS development phases. Figure [ID 91693] describes an overview of the DCS implementation strategy:

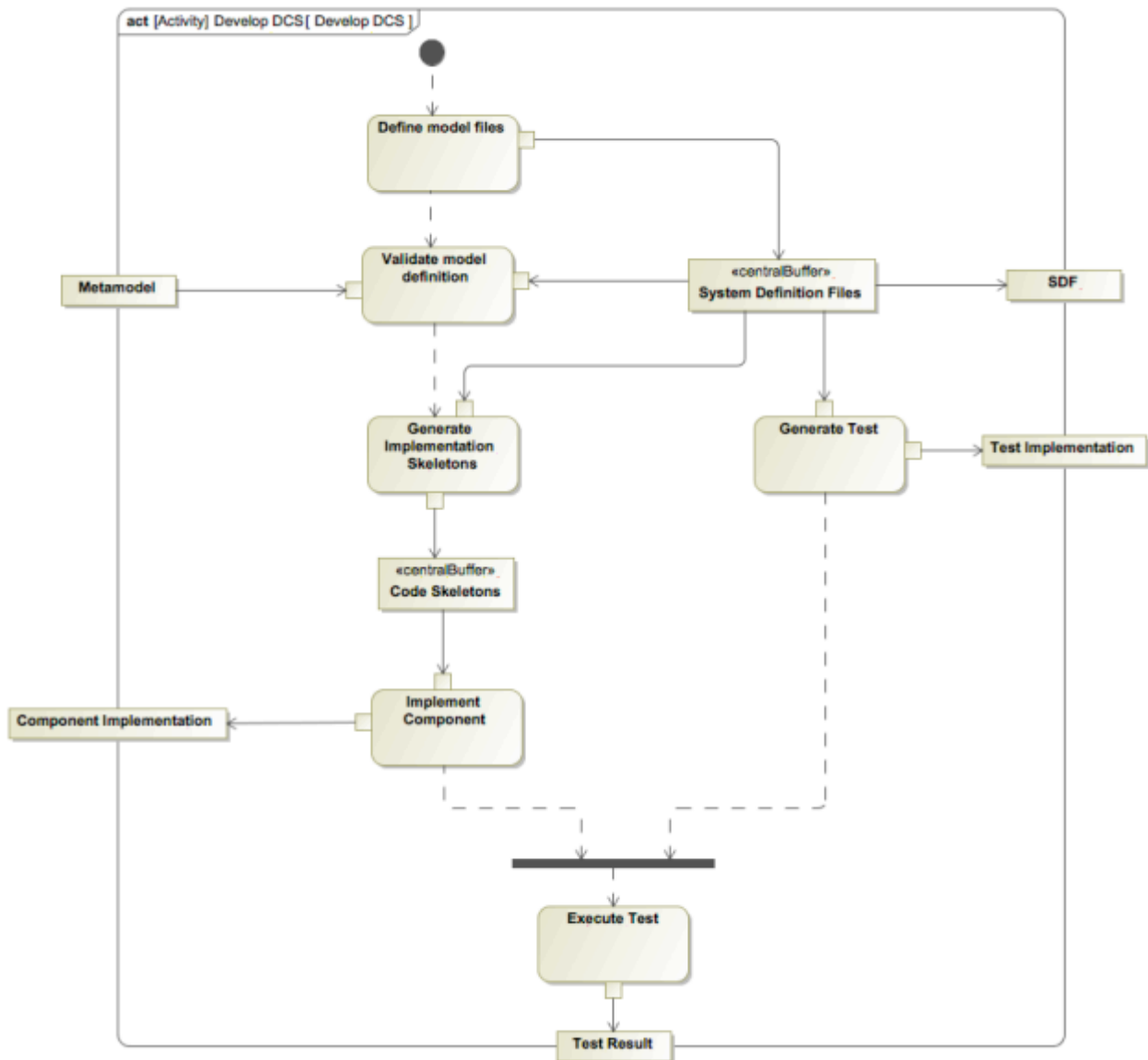


Figure 4-16 [ID 91693]: DCS Implementation Strategy Overview

REQ-L3-SCS-92069: SDF Definition

The SDF of DCS component shall include the following information:

- DCS unique identification
- DCS State Variables specification and corresponding State Machines when applicable.
- DCS Faults specification
- DCS Alarms specification
- DCS Properties specification
- DCS Components Process Inputs and Outputs specification including those of the Hardware Adapters
- DCS Configuration properties specification
 - o DCS constant values
 - o Default values (“factory settings”) for run-time configuration used for DCS start-up
- DCS Logging messages list for messages of level WARNING and ERROR
- DCS specific Data Types
- Definition of the DCS user interface components

Legacy ID: REQ-L3-SCS-0077

REQ-L3-SCS-92070: Visual Modeling

The DCS shall use SysML in the diagrams that provide a visual description of the DCS designs.

Notes: Although SDF provide a formal definition of the DCS that can be validated and it is used to drive the development of the system, visual representations are useful to present high level views of the system structure and behavior (state diagrams, activity diagrams, internal block diagrams).

Legacy ID: REQ-L3-SCS-0077

4.4.4 Development Requirements

GMTO will support the development of all the hardware and software based in these standards, e.g. frameworks, device drivers, development tools. The software developed for the GMTO shall use the frameworks and components provided by GMTO. Other alternatives not included in the standard shall not be considered neither supported.

New specific alternatives may be approved and introduced in the standard only when, according the GMTO criteria:

- The existing standard does not provide the capability to solve a problem.



- The solution of a problem turns out to be too complex with the current standard whereas it could be easily solved by means of alternative technologies.
- Technology evolution during the life of the telescope recommends updating part of the electronic equipment.
- GMTO will release a new version of its software periodically; such software will keep backward compatibility with previous versions to the largest possible extent. The usage of such newer release shall be mandatory for every group developing software for GMTO. In case of incompatible versions, an adaptation time will be considered. Exceptionally, and with GMTO approval, the usage of newer releases can be temporary delayed if critical activities are in progress.
- In the initial stages of the project some frameworks and libraries may have their public interface already specified despite the fact that its implementation is not yet completed. This is to inform the design of some DCSs or to define interfaces between DCSs.

4.4.4.1 Procurement Strategies Related to DCS Development

A DCS can be procured by more than one organization. In some cases the DCS Device Control Package(s) can be developed by the same organization that procures the Controlled Subsystem using a PLC based implementation, while the DCS Operation Support Packages are provided by the PO. In such a case the following considerations apply:

- The Device Control Packages are not required to use the GMT SDK, but instead are required to use the PLC programming environments as defined by these standards.
- The Device Control Packages are required to comply with this specification with the exception of the sections that refer to Operation Support Packages and to the PC-based DCCs.
- The Operation Support Packages are developed using the GMT SDK, which is needed to ensure that the DCS can be integrated with the rest of the OCS.
- The Operation Support Packages are required to comply with the full extent of this specification.

4.4.4.2 OCS Development Runtime

The OCS Development Runtime is a subset of the OCS supplied by GMTO as part of the SDK in order to enable the development and test of the DCS before the Controlled Subsystem is integrated into the GMT system on-site. The main features of OCS Development Runtime are:

- Development and test of the DCS specific user interface components.
- Visualization of DCS detailed status using the generated engineering user interface components
- Handling and visualization of the Alarms generated by the DCS.
- Handling and visualization of the Faults generated by the DCS.
- Handling and visualization of the Logging Messages generated by the DCS.

- Handling and visualization of the Telemetry Data generated by the DCS.
- Management and storage of the Configuration Properties for the DCS.
- Execution of DCS sequences (e.g. calibration or diagnostic sequences) from the OCS.
- Integration of Operation Support Packages components (e.g. Observing tool plugins, calibration pipelines) with the OCS.
- Storage of the data generated by the DCS and access to this data.
- Development and testing of the supervisory functions to be integrated in the OCS.

REQ-L3-SCS-92078: OCS Development Runtime

The DCS shall use the OCS Development Runtime as a tool for DCS software development, support, integration, factory acceptance test and site acceptance test.

Legacy ID: REQ-L3-SCS-0060

4.5 Lifecycle Requirements

GMTO follows an Agile development process with emphasis in early continuous integration of DCS software. DCS software shall be delivered regularly to the GMTO Software Repository. The following requirements define the integration strategy for DCS components.

REQ-L3-SCS-92081: GMT SDK Version

The DCS shall use the latest GMT SDK system version.

Legacy ID: REQ-L3-SCS-0061

REQ-L3-SCS-92083: DCS Release Plan

The DCS shall include a release plan and a product backlog, which shall be delivered to GMTO.

Legacy ID: REQ-L3-SCS-0001

REQ-L3-SCS-92084: Software Deliverables

The DCS software components and updated System Definition Files (SDF) shall be delivered to the GMTO Software Repository.

Notes: See section 4.4.3 for a description of System Definition Files.

Legacy ID: REQ-L3-SCS-00011

REQ-L3-SCS-92085: Software Continuous Integration

The DCS software components shall be delivered periodically during the execution of each phase to enable early and continuous integration of all the GMTO software.

Notes: Each phase shall be divided in a set of periodic iterations aligned with the GMT master release plan. The frequency and number of iterations shall be approved by GMTO.

Legacy ID: REQ-L3-SCS-0002

REQ-L3-SCS-92086: Test Versioning

The DCS shall log for every test (unit testing; system and integration testing; acceptance testing) the version of the equipment being tested, the version of the test specifications being used, and the version of the specification being tested against.

Legacy ID: REQ-L3-SCS-0003

REQ-L3-SCS-92087: Standard compliance

The DCS compliance with these standards shall be assessed on a continuous basis as part of the continuous integration process.

Notes: SDF files shall be tested against the reference architecture well-formedness rules. Continuous integration and testing facilitate early error identification and reduce the time for factory and site acceptance testing.

Legacy ID: REQ-L3-SCS-0004

REQ-L3-SCS-92088: FAT Scope

The DCS shall include the functionality required to support the Factory Acceptance Testing of the Controlled Subsystem

REQ-L3-SCS-92089: FAT DCS Testing

The DCS performance and functionality requirements shall be tested or demonstrated during Factory Acceptance Testing.

Legacy ID: REQ-L3-SCS-0005

REQ-L3-SCS-92090: FAT Environment

The DCS shall use the OCS Development Runtime for Factory Acceptance Testing.

Legacy ID: REQ-L3-SCS-0006

REQ-L3-SCS-92091: Site Testing Scope

The DCS shall include the functionality required to support the Site testing of the Controlled Subsystem.

Legacy ID: REQ-L3-SCS-0006

REQ-L3-SCS-92092: Site Testing Environment

The DCS shall use real OCS for Final Site Acceptance Testing.

Legacy ID: REQ-L3-SCS-0006

REQ-L3-SCS-92093: Interface Automated Testing

The DCS interface acceptance tests will be generated automatically from the specifications contained in System Definition Files.

Notes: The use of generated tests enables automated traceability from implementation to specifications.

Legacy ID: REQ-L3-SCS-0007

REQ-L3-SCS-92094: Acceptance Control System for Low Level Control Systems

The DCS Low-Level Control System, when developed independently of the DCS High-Level Control System shall include an Acceptance Control System that implements the interface with the DCS High-Level Control System and that provides the functionality needed to support the Controlled Subsystem Acceptance Tests.

REQ-L3-SCS-92095: Standards Applicable Version

The latest version available of these standards shall be applicable.

Legacy ID: REQ-L3-SCS-0008

REQ-L3-SCS-92096: Documentation Standards

The DCS documentation shall comply with the *GMT Information and Configuration Management Plan* (GMT-SE-DOC-00003)

Legacy ID: REQ-L3-SCS-0009

REQ-L3-SCS-92097: Documentation for 3rd party and COTS Components

For every DCS item (including 3rd party and COTS), the original documentation shall be delivered.

Legacy ID: REQ-L3-SCS-0010

4.6 Interface Specification

This section defines the interfaces necessary for the successful integration of a Controlled Subsystem with the rest of the Software and Controls. Two groups of interfaces are defined:

- The interfaces related to the GMT Control System
- The interfaces related to the GMT Interlock and Safety System

The following sections describe these two groups of interfaces.

4.6.1 Control System Interfaces

This section defines the interfaces related to the integrated operation of the GMT Control System. These interfaces mostly affect the DCS of each Controlled Subsystem. Two types of interfaces are defined:

- The interface between the OCS and the Controlled Subsystem.
- The interface between the high-level and low-level packages of a DCS when different organizations are involved in its procurement

4.6.1.1 OCS – Controlled Subsystem Interface

This interface defines the interface between the OCS and the DCS of a Controlled Subsystem.

There are three ways to implement data interfaces:

- **Component Based Interfaces:** DCS Component that expose their features (e.g. state variables, faults, etc) to the outside are defined using connection maps that define a set of Connectors. Each Connector specification defines the nominal conditions for the communication of data between Components and parameters for Quality of Services. Connectors are defined in the SDF of each DCS. (e.g. DCS interface with the TCS).
- **OPC-UA Based Interfaces.** Mostly used in external procurements
- **OCS API Based Interfaces:** DCS components that are integrated inside OCS applications must implement an OCS API which is often defined by inheritance from an abstract component definition (e.g. plugins)

Each DCC is considered being a part of the DCS. All DCCs are connected to the GMT Control Network, although one DCS Master Supervisor shall be deployed.

Network interfaces provide the only physical interconnection between DCS and OCS. GMTO manages the GMT OCS networks. The OCS communication system comprises the general-purpose Control Network, the Low Latency Network, the Data Storage Network and the Time Distribution Network.

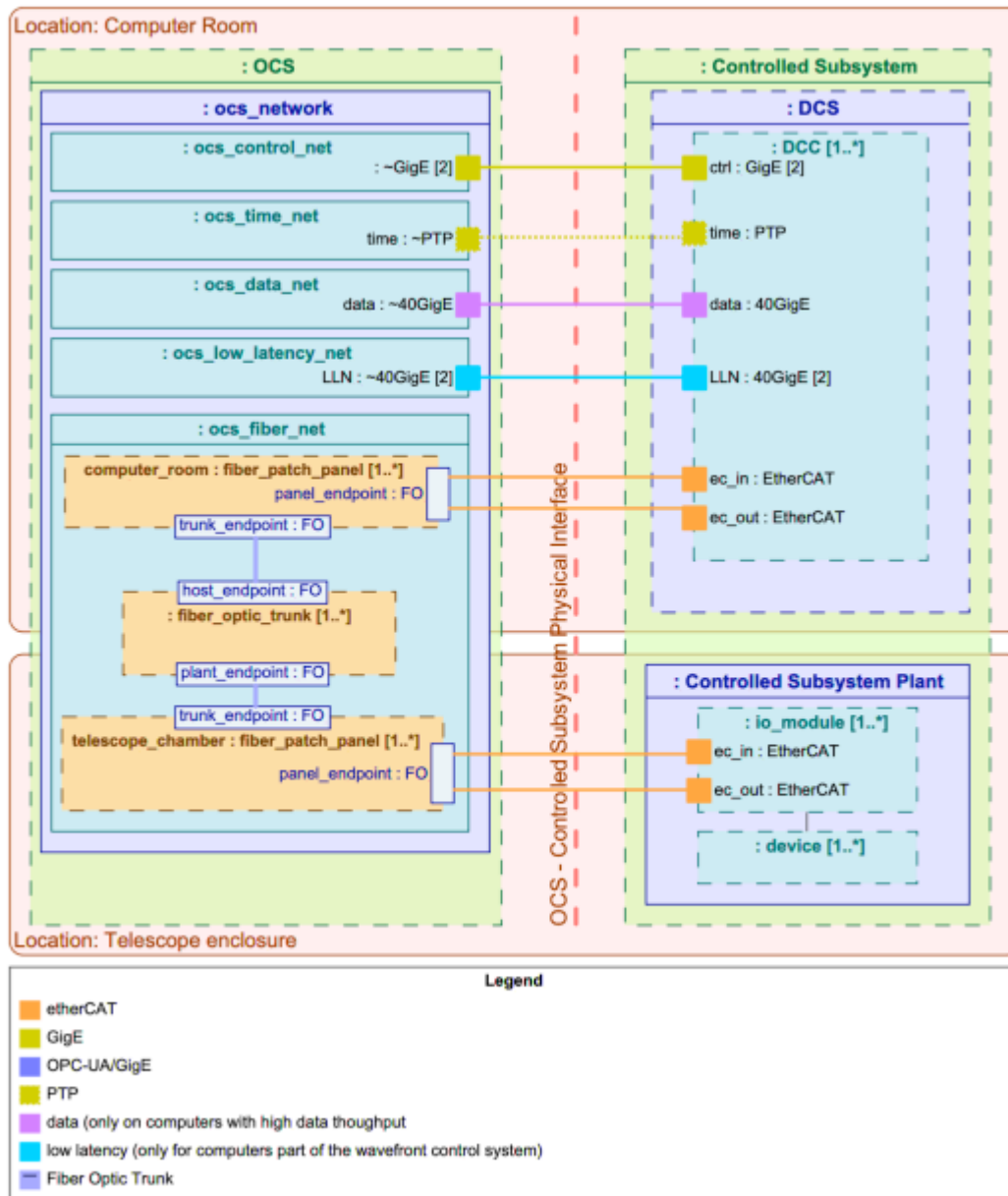


Figure 4-17 [ID 91694]: Controlled Subsystem Physical Interface

4.6.1.1.1 Control Network

REQ-L3-SCS-92112: DCS – OCS Control Network Interface

The DCS shall interface each DCC to the OCS Control Network.

Notes: These interfaces enable access to Observatory Services (i.e. Telemetry, Configuration, Logging, Alarm), the Observatory Operations System and the TCS.

Legacy ID: REQ-L3-SCS-0124

REQ-L3-SCS-92113: DCS – OCS Control Network Interface Redundancy

Each DCS shall have a redundant connection to the OCS Control Network.

Legacy ID: REQ-L3-SCS-0129

4.6.1.1.2 Low Latency Network

The OCS Low Latency Network provides transport for real-time WFC control and telemetry. It guarantees data exchange with latency less than 10 microsec and jitter less than TBD microsec.

Only DCSs participating in fast WFC feedback control shall be connected to the OCS Low Latency Network. DCSs may have multiple OCS Low Latency Network interfaces.

REQ-L3-SCS-92117: DCS – OCS Low Latency Network Interface

The DCS shall interface (read and write data with QoS parameters) to the OCS Low Latency Network for when applicable (e.g. the DCS is part of the wavefront control function).

Legacy ID: REQ-L3-SCS-0125

REQ-L3-SCS-92118: DCS – OCS Low Latency Network Communication Protocol

The DCS shall use UDP Multicast over 40GbE (TBC) to communicate with the OCS Low Latency Network.

REQ-L3-SCS-92119: DCS – OCS Low Latency Network Interface Card

The DCS shall use only GMTO approved OCS Low Latency Network Interface Cards (NIC) to connect to the OCS Low Latency Network.

Legacy ID: REQ-L3-SCS-0130

REQ-L3-SCS-92120: DCS - OCS Low Latency Network Interface Procurement

The DCS shall use GMTO supplied specific hardware and software to interface with the OCS Low Latency Network.

Legacy ID: REQ-L3-SCS-0131

REQ-L3-SCS-92121: OCS Low Latency Network Interface Location

The DCS shall connect its DCC(s) to the OCS Low Latency Network Connection Panel located in the GMT Computer Room.

Legacy ID: REQ-L3-SCS-0132

4.6.1.1.3 Time Distribution Network

The Time Distribution Network provides project-wide time synchronization. It allows the DCS DCCs to be synchronized with an accuracy of 50 us RMS to the GMT Time, which is Universal Coordinated Time (UTC). The TCN network is carrying Precision Time Protocol (PTP version 2, IEEE-11588-2008). Any DCC requiring high accuracy time synchronization and time stamping shall be connected the TCN.

REQ-L3-SCS-92124: DCS – OCS Time Distribution Network

The DCS shall interface to the OCS Time Distribution Network (IEEE 1588) if high accuracy synchronization is required.

Legacy ID: REQ-L3-SCS-0126

REQ-L3-SCS-92125: OCS Time Distribution Network Interface Card

The DCS shall use only GMTO approved Time Distribution Network interfaces to connect to the Time Distribution Network.

Legacy ID: REQ-L3-SCS-0133

REQ-L3-SCS-92126: OCS Time Distribution Network interface Procurement

Specific hardware and software required by the Time Distribution Network interface will be supplied by GMTO.

Legacy ID: REQ-L3-SCS-0134

REQ-L3-SCS-92127: OCS Time Distribution Network Interface Location

The DCS shall connect its DCC(s) to the OCS Time Distribution Network Connection Panel located in the GMT Computer Room.

Legacy ID: REQ-L3-SCS-0135

4.6.1.1.4 Data Storage Network

The Data Storage Network allows transferring the scientific data into the GMT Scientific Data Archiving System. The Data Storage Network is deployed using a dedicated high-throughput Ethernet network infrastructure.

REQ-L3-SCS-92130: Data Storage Network

The DCS shall interface to the OCS Data Storage Network to transfer scientific data, if applicable.

Legacy ID: REQ-L3-SCS-0127

REQ-L3-SCS-92131: Data Storage Network Interface

The DCS shall use only GMTO certified Data Storage Network interfaces to connect to the Data Storage Network.

Legacy ID: REQ-L3-SCS-0136

REQ-L3-SCS-92132: Data Storage Network Interface Procurement

Specific hardware and software required by the Data Storage Network interface will be supplied by GMTO.

Legacy ID: REQ-L3-SCS-0137

REQ-L3-SCS-92133: Data Storage Network Interface Location

The Data Storage Network interface shall be located in the GMT Computer Room hosted by the DCC.

Legacy ID: REQ-L3-SCS-0138

4.6.1.1.5 OCS – DCS Data Interface

The OCS – DCS Interface is formally specify in the corresponding SDF as a set of Connectors between Component Features.

REQ-L3-SCS-92136: Functional Interface

The Device Control Package shall interface the following OCS system functions:

- Supervisory control and monitoring
- Operating states management
- Alarm management
- Fault management
- Configuration management
- Telemetry
- Logging
- Wavefront control (if applicable)
- Telescope control (if applicable)

Legacy ID: REQ-L3-SCS-0122

REQ-L3-SCS-92137: DCS – OCS Data Interface

The DCS shall implement the features needed to support the Connectors defined in the interface.

REQ-L3-SCS-92138: Operation Support Package-OCS Interface

The DCS shall implement the interface between its Operation Support Packages and the rest of the OCS.

Legacy ID: REQ-L3-SCS-0139

REQ-L3-SCS-92139: Device Control Package-OCS Software Interface

The DCS shall implement the software interface between its Device Control Packages and the rest of the OCS.

Legacy ID: REQ-L3-SCS-0123

4.6.1.2 DCS Internal Interfaces

When different parts of a DCS is developed by different institutions, an internal interface is defined to formalize the interface of each part. Figure [ID 91695] shows a diagram describing the High-Level / Low-Level interface when PLC-Based DCCs are used.

REQ-L3-SCS-92142: DCS High Level – Low Level Protocol Interface (PLC-Based DCC)

The DCS shall use OPC-UA for the communication between the High Level Controllers (OPC-UA client) and the Low Level Controllers (OPC-UA server).

REQ-L3-SCS-92143: DCS High Level – Low Level Data Interface (PLC-Based DCC)

The DCS Low Level Controllers OPC-UA servers shall organize the interface data following the hierarchical structure of Packages, Components and Component features, including Inputs, Outputs, State Variables, Faults, Alarms and Properties.

REQ-L3-SCS-92144: DCS High Level – Low Level Physical Interface (PLC-Based DCC)

The DCS shall use the OCS Control Network GigE interface for the communication between the High Level Controllers (PC-Based DCC) and the Low Level Controllers (PLC-Based DCC).

REQ-L3-SCS-92145: DCS High Level – Low Level Protocol Interface (PC-Based DCC)

The DCS shall use SDK Connectors between Component features for the communication between the High Level Controllers and the Low Level Controllers.

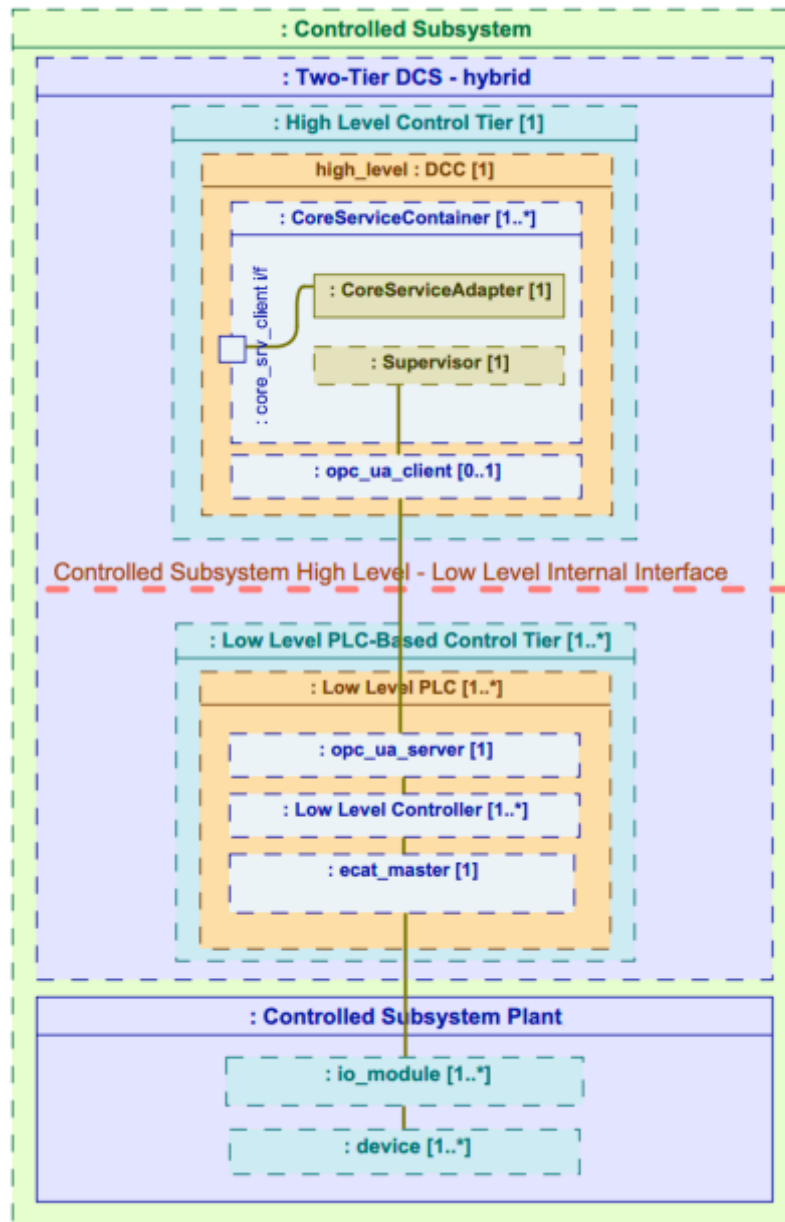


Figure 4-18 [ID 91695]: High-Level – Low-Level Interface (Logical View)

REQ-L3-SCS-92146: DCS High Level – Low Level Physical Interface (PC-Based DCC)

The DCS shall use the OCS Control Network GigE interface for the communication between the High Level Controllers (PC-Based DCC) and the Low Level Controllers (PC-Based DCC).

4.6.2 Interlock and Safety System Interfaces

This section describes the interface between the Global ISS and the Controlled Subsystem Local ISS

4.6.2.1 Safety Interface

REQ-L3-SCS-92150: Global ISS – Controlled Subsystem Safety Interface

The Controlled Subsystem safety input/output modules shall connect to the Global ISS Safety PLC to support the implementation of the global safety functions that involve the Controlled Subsystem.

REQ-L3-SCS-92151: Safety Interface Protocol

The Controlled Subsystem Global Safety IO Modules shall connect to the Global ISS Safety PLC using FSoE.

Legacy ID: REQ-L3-SCS-0128

REQ-L3-SCS-92152: Safety Interface Location

The Controlled Subsystem Global Safety IO Modules shall connect to the Global ISS Safety PLC at the designated panel inside the SEC (TBC).

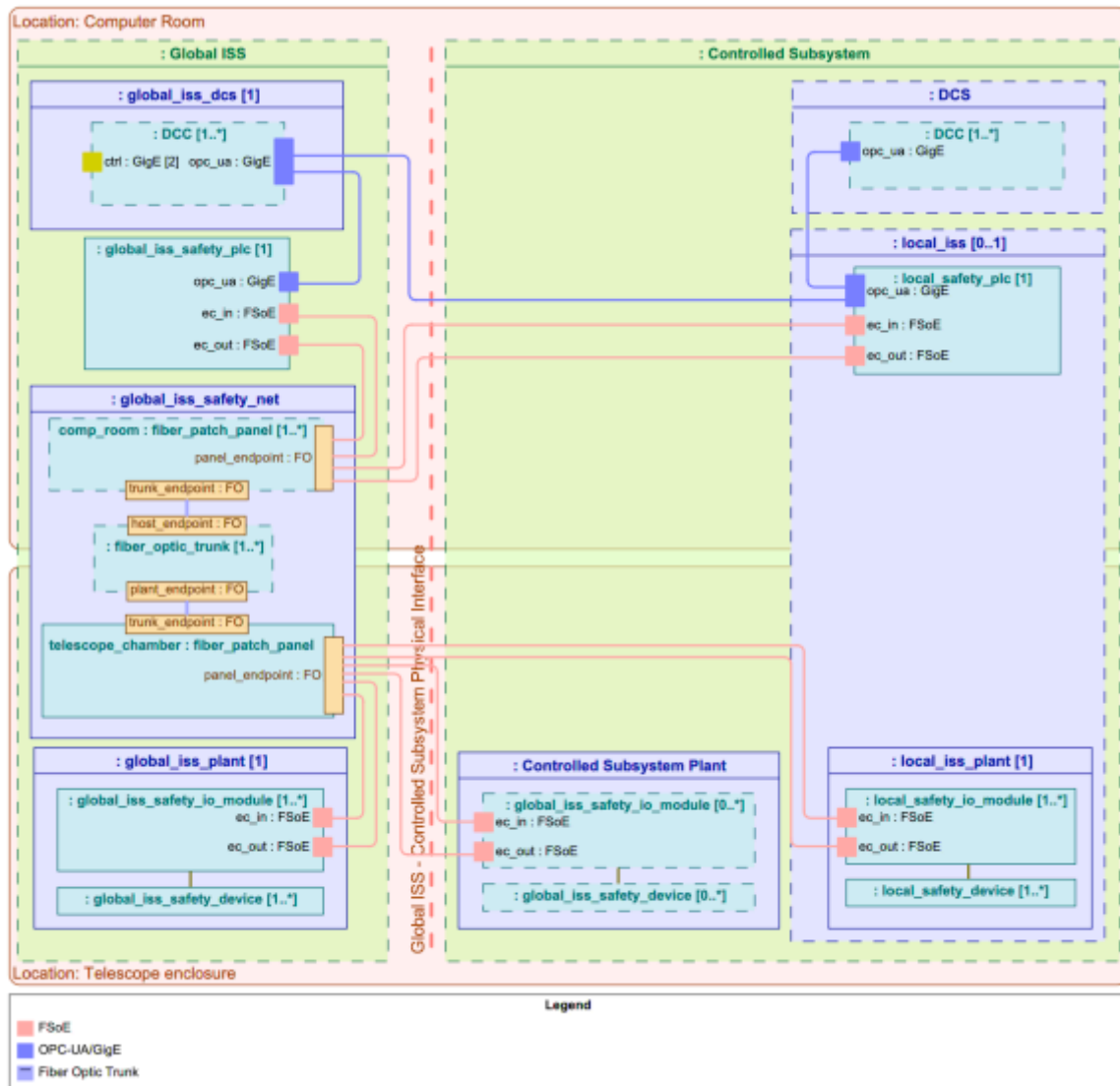


Figure 4-19 [ID 91696]: Global ISS – Controlled Subsystem Physical Interface

4.6.2.2 Local ISS Monitoring Interface

This interface applies when Controlled Subsystem has a Local ISS to implement local safety functions. One of the Global ISS functions is to monitor the overall status of all the GMT safety systems, for this purpose the Global ISS interfaces with the Local ISS Safety PLC using OPC-UA.

REQ-L3-SCS-92155: Global ISS – Controlled Subsystem Monitoring Interface

The Controlled Subsystem Local ISS Safety PLC shall interface with the Global ISS DCC for monitoring purposes.



REQ-L3-SCS-92156: Local ISS Monitoring Protocol

The Controlled Subsystem Local ISS Safety PLC shall implement an OPC-UA server to connect to the Global ISS DCC.

REQ-L3-SCS-92157: Local ISS Monitoring Data

The Controlled Subsystem Local ISS Safety PLC OPC-UA server shall communicate the status of each Local Safety Function as well as the status of the input and output devices associated with it.

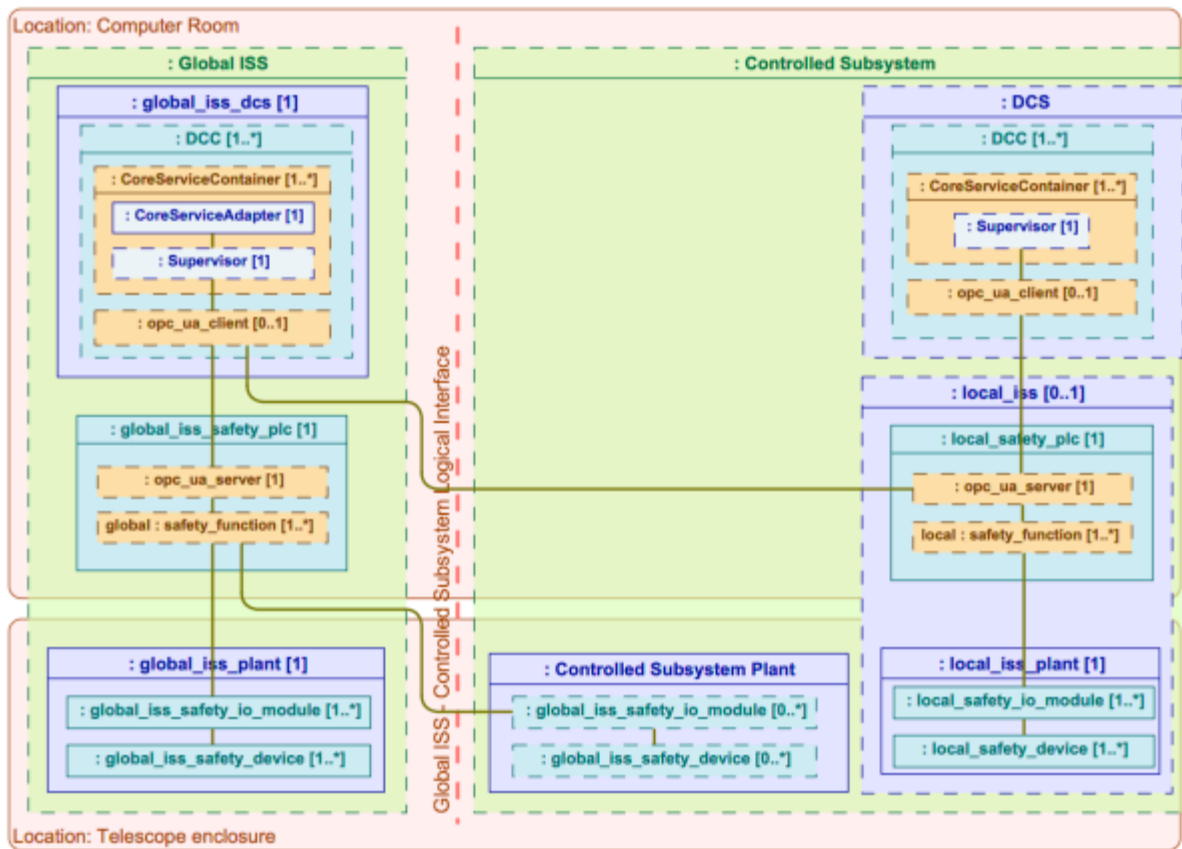


Figure 4-20 [ID 91697]: Global ISS – Controlled Subsystem Logical Interface

REQ-L3-SCS-92158: Local ISS Monitoring Physical Interface

The Controlled Subsystem Local ISS Safety PLC shall connect to the OCS Control Network using a GigE RJ-45 port.

REQ-L3-SCS-92159: Local ISS Monitoring Interface Location

The Controlled Subsystem Local ISS Safety PLC shall connect to the OCS Control Network at the designated panel in the Computer Room.

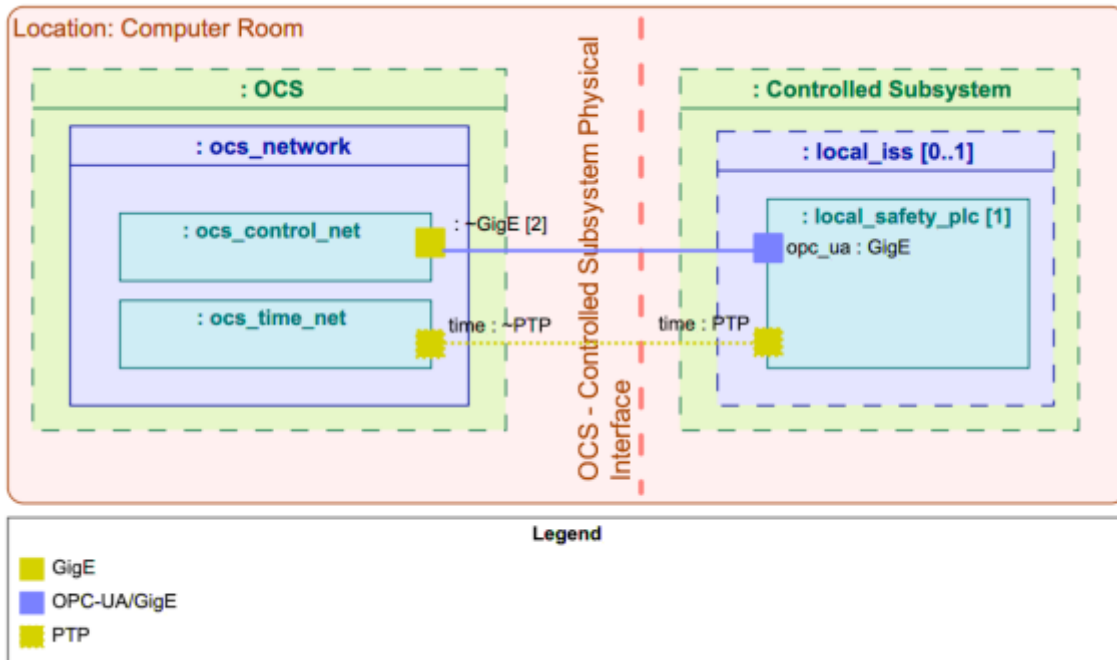


Figure 4-21 [ID 91698]: OCS – Controlled Subsystem Physical Interface

5 Non-Conformances

REQ-L3-SCS-92161: Non-Conformances

Request for non-conformance with the requirements in this document shall be compliant with the waiver process defined in the *GMT Information and Configuration Management Plan* GMT-SE-DOC-00003.

Legacy ID: REQ-L3-SCS-0148

APPENDIX: Accepted EtherCAT Devices

The EtherCAT Technology Group (ETG) is a global organization in which OEM, End Users and Technology Providers join forces to support and promote the further technology development.

The ETG constantly updates the EtherCAT Product Guide. This is a list with EtherCAT products and services as submitted by ETG member companies. The guide contains over 800 entries and many more are in the processes to be evaluated and entered. Technical information is included to help selecting the suitable product. Many entries are enhanced by links to corresponding device description files and datasheets.

For a list of the OCS accepted EtherCAT devices, GMTO strongly suggests searching the ETG webpage: <https://www.ethercat.org/en/products.html>