



European Component Oriented Architecture (ECOIA®) Collaboration Programme: Architecture Specification Change Impact Sheet

Issue: 6

Prepared by
BAE Systems (Operations) Limited and Dassault Aviation

This specification is developed by BAE SYSTEMS, Dassault Aviation, Bull SAS, Thales Systèmes Aéroportés, GE Aviation Systems Limited, General Dynamics United Kingdom Limited and Leonardo MW Ltd and the copyright is owned by BAE SYSTEMS, Dassault Aviation, Bull SAS, Thales Systèmes Aéroportés, GE Aviation Systems Limited, General Dynamics United Kingdom Limited and Leonardo MW Ltd. The information set out in this document is provided solely on an 'as is' basis and co-developers of this specification make no warranties expressed or implied, including no warranties as to completeness, accuracy or fitness for purpose, with respect to any of the information.

Note: *This specification represents the output of a research programme. Compliance with this specification shall not in itself relieve any person from any legal obligations imposed upon them. Product development should rely on the DefStan or BNAE publications of the ECOIA standard.*

1 Scope

This document tracks changes between releases of the ECOA Architecture Specification.

ECOA specifies a uniform method for design, development and integration of complex real time software systems using a service oriented component based approach.

2 ECOA Architecture Specification Volumes which form the current PF Spec

Architecture Specification Part 1	IAWG-ECOA-TR-001 / DGT 144474 Issue 6 Architecture Specification Part 1 – Concepts
Architecture Specification Part 2	IAWG-ECOA-TR-012 / DGT 144487 Issue 6 Architecture Specification Part 2 – Definitions
Architecture Specification Part 3	IAWG-ECOA-TR-007 / DGT 144482 Issue 6 Architecture Specification Part 3 – Mechanisms
Architecture Specification Part 4	IAWG-ECOA-TR-010 / DGT 144485 Issue 6 Architecture Specification Part 4 – Software Interface
Architecture Specification Part 5	IAWG-ECOA-TR-008 / DGT 144483 Issue 6 Architecture Specification Part 5 – High Level Platform Requirements
Architecture Specification Part 6	IAWG-ECOA-TR-006 / DGT 144481 Issue 6 Architecture Specification Part 6 – ECOA [®] Logical Interface
Architecture Specification Part 7	IAWG-ECOA-TR-011 / DGT 144486 Issue 6 Architecture Specification Part 7 – Metamodel
Architecture Specification Part 8	IAWG-ECOA-TR-004 / DGT 144477 Issue 6 Architecture Specification Part 8 – C Language Binding
Architecture Specification Part 9	IAWG-ECOA-TR-005 / DGT 144478 Issue 6 Architecture Specification Part 9 – C++ Language Binding
Architecture Specification Part 10	IAWG-ECOA-TR-003 / DGT 144476 Issue 6 Architecture Specification Part 10 – Ada Language Binding
Architecture Specification Part 11	IAWG-ECOA-TR-031 / DGT 154934 Issue 6 Architecture Specification Part 11 – High Integrity Ada Language Binding

3 Current version

The current version of the Architecture Specification, published on the ECOA website, is:

- Issue 6

4 Overview

The scope of ECOA Architecture Specification Issue 6 can be illustrated in terms of differences with respect to ECOA Architecture Specification Issue 5, as in Figure 1:

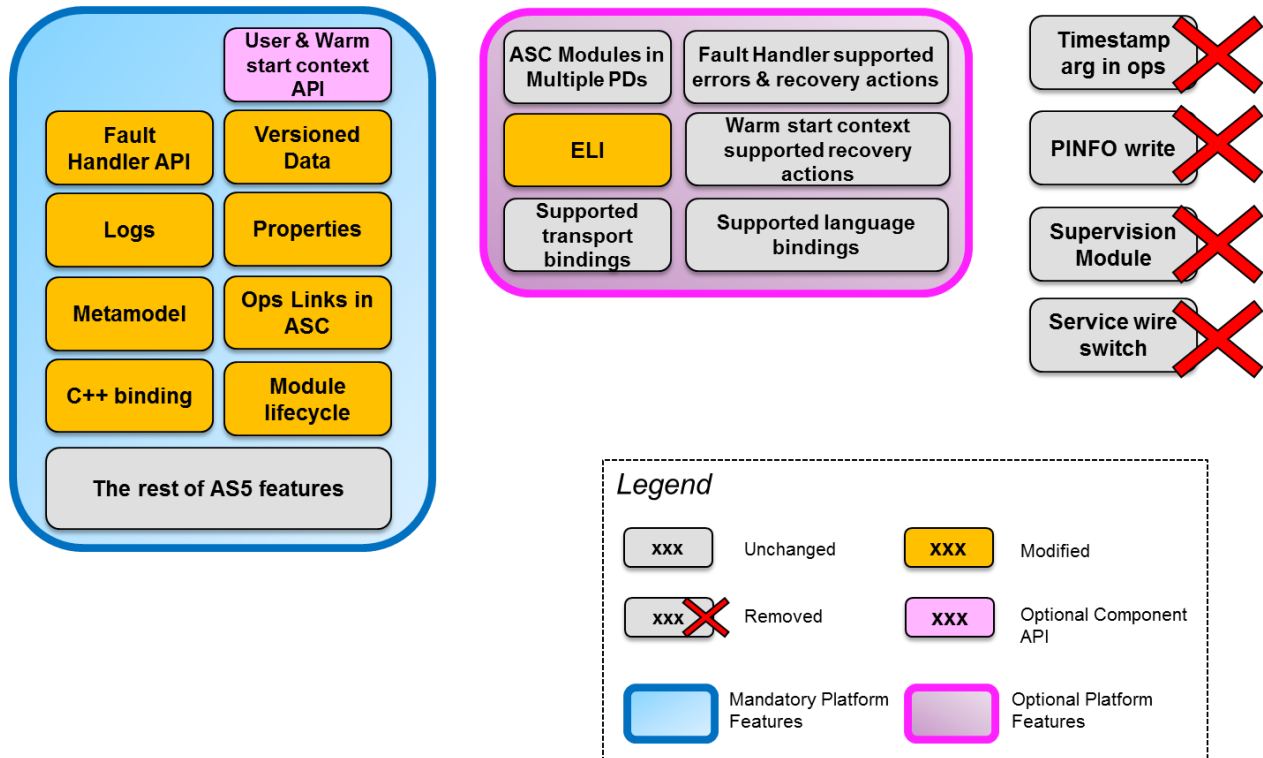


Figure 1: ECOA Architecture Specification Issue 6 overview

A summary of the modifications introduced by Architecture Specification Issue 6 is provided below.

Removed features:

- Supervision Module:
 - Removal of module lifecycle management at ASC level, including the related APIs.
 - Modules are now automatically brought into “RUNNING” state (i.e. initialized and started) by the ECOA infrastructure, previously this was done by the ASC Supervision Module.
 - Removal of error handling at ASC level, including the related APIs.
 - All errors raised by Module Instances now go directly to the ECOA Fault Handler, previously errors were sent to the ASC Supervision Module.
 - In the event of a fatal error being raised by a Module Instance, the ECOA platform shuts down all Module Instances within that ASC.
 - Removal of service availability management by the ECOA infrastructure, including the related APIs, at ASC level.
 - The ECOA Platform now stores incoming operations in the Module Instance’s FIFO queue only provided the Module instance is in the “RUNNING” state, previously this was also determined according to the service availability status.
 - Where service availability information is needed from a functional point of view, it could still be provided as a normal service operation (e.g. versioned data).

- The ELI has been simplified by removing messages related to service availability.
- Wire Switch:
 - Removal of the capability to connect multiple provided Service Instances on the server side to a single required Service Instance on the client side.
- Writable PINFO:
 - Removal of writable PINFO capability from ECOA Platforms, including the related APIs, at ASC level.
 - The capability to read Public and Private PINFO is retained.
- Timestamp argument in all operations:
 - Removal of timestamping by the ECOA Platform of ECOA operations, as well as from the Module Context and ELI messages. Timestamping by the ECOA Platform of Fault Handler error notifications is retained.
 - The Dynamic Trigger API has been changed to replace “delayDuration” with an “expiryTime” argument.
 - The versioned data operation now includes a “stamp” attribute which is changed when the data has been written to.

Modified features:

- Notifying versioned data:
 - The notification when versioned data has been updated is simplified, through the removal of the data handle argument.
- Properties:
 - Properties are now limited to Basic, Simple, Enumerations and Fixed or Variable Arrays of these types.
 - The provision of complex typed data may be implemented using PINFO.
- Operation links within ASCs:
 - Within an ASC Implementation Description, “RequestReceived” and “EventReceived” Module Operations cannot exist in more than one OperationLink.
- Module Lifecycle:
 - The “Reinitialize” entry point has been removed.
 - The User Context will no longer be zeroed by the ECOA Platform.
- Access control setting for versioned data:
 - Possibility to choose to have single copy of the data being accessed by all Module Instances within a single Component. Similar to a “shared memory” to improve performance by not making local copies of data.
- Write-Only mode setting for a writer of a versioned data:
 - Possibility for a writer of a versioned data to have a “writeOnly” copy of the data, thereby avoiding the platform to initialize the local copy of the data with the current value. The purpose is to improve performance when initializing a local copy is not needed.
- Additional error_code for Fault Handler:
 - A parameter is added to errors raised to the ECOA Fault Handler in order to provide functional context information which can be used for deciding recovery actions.
- Log information:
 - Logs have been improved to replace “node name” and “protection domain name” with “component instance name” and “module instance name”.
 - Node / protection domain can be traced back by parsing the deployment schema.
- C++ binding enhancement:
 - The way enumerations are handled in the C++ binding has been improved and the constructor used is now the default C++ one.
- ELI:
 - Constraints on the way ELI IDs are defined have been reduced.

- Capability to specify the transport binding to be used on each logical platform link has been added.
- Composites:
 - Use of composites for designing hierarchical views of systems has been improved in the ECOA Metamodel (new cross platform view schema, new attributes in the deployment schema).

Optional Component APIs:

- “User Context” and “Warm Start Context” data structure and API generation:
 - Generation of these data structures and APIs at ASC level is now optional, using new metamodel attributes associated with ECOA Module Types.
 - ECOA Platforms, however are still required to support “User Context” and “Warm Start Context”.

Optional Platform Features:

- ELI:
 - ECOA Platform support for the ELI is no longer mandatory. Where the system under development is hosted on a single ECOA platform, the platform may now be procured without ELI support.
- ASC module deployment in multiple protection domains:
 - ECOA Platform support for distributing Module Instances of the same ASC across different Protection Domains is no longer mandatory.

Note: “Supported language bindings”, “Supported transport bindings”, “Fault Handler supported errors and recovery actions” and “Warm start context supported recovery actions” were already optional features in Architecture Specification Issue 5, which is why they are not detailed here.

5 Change Tracking

The following table tracks major changes brought by each Architecture Specification Issue (compared to the previous version).

AS Issue 6 (December 2017)	<p>Removed features :</p> <ul style="list-style-type: none"> ● <u>Supervision Module:</u> <ul style="list-style-type: none"> ○ Abstract: <ul style="list-style-type: none"> ▪ Removal of module lifecycle management at ASC level, including the related APIs. ▪ Removal of error handling at ASC level, including the related APIs. ▪ Removal of service availability management by the ECOA infrastructure, including the related APIs, at ASC level. ○ Modified volumes: <ul style="list-style-type: none"> ▪ Part 1 - §7.7; §7.9; §8.2; §8.3.2 ▪ Part 2 ▪ Part 3 - §7.6; §7.7; §9.1; §9.3 ; §10.2 ▪ Part 4 - §6; §9.4; §10; §11.2.3; §11.3; §11.5; §11.6; §13 ▪ Part 6 - §6.1.2; §6.3 ▪ Part 7 - §6.1; §8.7; §8.8 ▪ Part 8 - §6; §9.4; §10; §11.2.3; §11.3; §11.5; §11.6; §13; §17 ▪ Part 9 - §6; §9.4; §10; §11.2.3; §11.3; §11.5; §11.6; §13; §17 ▪ Part 10 - §6; §9.4; §10; §11.2.3; §11.3; §11.5; §11.6; §13; §17 ▪ Part 11 - §6; §9.4; §10; §11.2.3; §11.3; §11.5; §11.6; §12; §13; §17
-----------------------------------	--

- Wire Switch:
 - Abstract:
 - Removal of the capability to connect multiple provided Service Instances on the server side to a single required Service Instance on the client side.
 - Modified volumes:
 - Part 2
 - Part 3 - §10.2
 - Part 7 - §6.1.4; §7.5; §8.2; §8.18; §9.3

- Writable PINFO:
 - Abstract:
 - Removal of writable PINFO capability from ECOA Platforms, including the related APIs, at ASC level.
 - The capability to read Public and Private PINFO is retained.
 - Modified volumes:
 - Part 1 - §8.4
 - Part 2
 - Part 3 - §7.10
 - Part 4 - §11.5
 - Part 6 - §6.1.2; §6.3
 - Part 7 - §6.1.5; §8.8
 - Part 8 - §11.5
 - Part 9 - §11.5
 - Part 10 - §11.5
 - Part 11 - §11.5

- Timestamp argument in all operations:
 - Abstract:
 - Removal of timestamping by the ECOA Platform of ECOA operations, as well as from the Module Context and ELI messages. Timestamping by the ECOA Platform of Fault Handler error notifications is retained.
 - The Dynamic Trigger API has been changed to replace “delayDuration” with an “expiryTime” argument.
 - The versioned data operation now includes a “stamp” attribute which is changed when the data has been written to.
 - Modified volumes:
 - Part 3 - §7.7
 - Part 4 - §10.3; §11.1.2
 - Part 8 - §6.2; §8; §10.3; §12; §16; §17
 - Part 9 - §6.2; §10.3; §12; §16; §17
 - Part 10 - §6.2; §8; §10.3; §12; §16; §17
 - Part 11 - §6.2; §8; §10.3; §11.1.2; §12; §16; §17

- Modified features:**
 - Notifying versioned data:
 - Abstract:
 - The notification when versioned data has been updated is simplified, through the removal of the data handle argument.
 - Modified volumes:
 - Part 3 - §7.5.3
 - Part 4 - §10.1.2

- Part 8 - §10.1.2
- Part 9 - §10.1.2
- Part 10 - §10.1.2
- Part 11 - §10.1.2
- Properties:
 - Abstract:
 - Properties are now limited to Basic, Simple, Enumerations and Fixed or Variable Arrays of these types.
 - The provision of complex typed data may be implemented using PINFO.
 - Modified volumes:
 - Part 4 - §11.2
- Operation links within ASCs:
 - Abstract:
 - Within an ASC Implementation Description, “RequestReceived” and “EventReceived” Module Operations cannot exist in more than one OperationLink.
 - Modified volumes:
 - Part 7 - §6.1.5; §8.8
- Module Lifecycle:
 - Abstract:
 - The “Reinitialize” entry point has been removed.
 - The User Context will no longer be zeroed by the ECOA Platform.
 - Modified volumes:
 - Part 3 - §7.10.5; §9.1; §9.4
 - Part 4 - §10.2
 - Part 8 - §10.2
 - Part 9 - §10.2
 - Part 10 - §10.2
 - Part 11 - §10.2
- Access control setting for versioned data:
 - Abstract:
 - Possibility to choose to have single copy of the data being accessed by all Module Instances within a single Component. Similar to a “shared memory” to improve performance by not making local copies of data.
 - Modified volumes:
 - Part 3 - §7.1; §7.5 §8.2.3
 - Part 4 - §11.1.2
 - Part 7 - §6.1.6; §8.8
 - Part 8 - §11.1.2
 - Part 9 - §11.1.2
 - Part 10 - §11.1.2
 - Part 11 - §11.1.2
- Write-Only mode setting for a writer of a versioned data:
 - Abstract:
 - Possibility for a writer of a versioned data to have a “writeOnly” copy of the data, thereby avoiding the platform to initialize the local copy of the data with the current value. The purpose is to

improve performance when initializing a local copy is not needed.

- Modified volumes:
 - Part 3 - §7.5.1; §7.5.2
 - Part 4 - §11.1.2
 - Part 7 - §6.1.5; §8.8
- Additional error code for Fault Handler:
 - Abstract:
 - A parameter is added to errors raised to the ECOA Fault Handler in order to provide functional context information which can be used for deciding recovery actions.
 - Modified volumes:
 - Part 4 - §9.4.7; §10.3; §11.3
 - Part 8 - §9.4.7; §10.3; §11.3
 - Part 9 - §9.4.7; §10.3; §11.3
 - Part 10 - §9.4.7; §10.3; §11.3
 - Part 11 - §9.4.7; §10.3; §11.3
- Log information:
 - Abstract:
 - Logs have been improved to replace “node name” and “protection domain name” with “component instance name” and “module instance name”.
 - Node / protection domain can be traced back by parsing the deployment schema.
 - Modified volumes:
 - Part 4 - §11.3
- C++ binding enhancement:
 - Abstract:
 - The way enumerations are handled in the C++ binding has been improved and the constructor used is now the default C++ one.
 - Modified volumes:
 - Part 9 - §6.3; §9.3; §9.4; §17
- ELI:
 - Abstract:
 - Constraints on the way ELI IDs are defined have been reduced.
 - Capability to specify the transport binding to be used on each logical platform link has been added.
 - ECOA Platform support for the ELI is no longer mandatory. Where the system under development is hosted on a single ECOA platform, the platform may now be procured without ELI support.
 - Modified volumes:
 - Part 1 - §7.10.2
 - Part 3 - §15
 - Part 5
 - Part 6 - §6; §6.6
 - Part 7 - §6.1.8; §8.7; §8.11; §8.13; §8.14
- Composites:
 - Abstract:
 - Use of composites for designing hierarchical views of systems has been improved in the ECOA Metamodel (new cross platform view schema, new attributes in the deployment schema).

- Modified volumes:
 - Part 3 - §15
 - Part 7 - §6; §6.1.1; §6.1.2; §6.1.9; §6.2; §7.4; §7.8; §7.9 §8.13; §8.14
 - “User Context” and “Warm Start Context” data structure and API generations:
 - Abstract:
 - Generation of these data structures and APIs at ASC level is now optional, using new metamodel attributes associated with ECOA Module Types.
 - ECOA Platforms, however are still required to support “User Context” and “Warm Start Context”.
 - Modified volumes:
 - Part 3 - §9.4
 - Part 4 - §8
 - Part 7 - §6.1.5; §8.5; §8.8
 - ASC module deployment in multiple protection domains:
 - Abstract:
 - ECOA Platform support for distributing Module Instances of the same ASC across different Protection Domains is no longer mandatory.
 - Modified volumes:
 - Part 5
 - Clarification of Metamodel version:
 - Abstract:
 - All Metamodel schemas versions have been updated to 2.0.
 - Modified volumes:
 - Part 7
 - Clarification of ELI version:
 - Abstract:
 - ELI messages version has been updated to 2.0.
 - Modified volumes:
 - Part 6
- Clarifications of existing features:**
- Clarification of the Metamodel:
 - Abstract:
 - ECOA XML filenames convention is clarified.
 - Include char and hexadecimal values besides integer and floating point values for constants.
 - Legality rules added regarding the scope of types.
 - Removed statement “*All names of ModuleInstances hosted by a ProtectionDomain shall be unique within the ProtectionDomain scope*”.
 - Modified volumes:
 - Part 7 - §6.1.8; §6.2.3; §6.2.4; §7.1; §8.12
 - Clarification of ELI:
 - Abstract:
 - ELI platform ID added in the logical system.

- Platform level message ID 0x00000000 has been defined.
 - ELI message discard cases have been specified.
 - ELI is not required to be used between Computing Nodes and Container instances.
 - Logical platform ID now 32 bits instead of 4 bits.
 - Statement "Do not transmit NaN and infinity values" is removed.
- Modified volumes:
 - Part 1 - §7.10.2
 - Part 6 - §6.1.1.1; §6.1.2; §6.4
 - Part 7 - §8.11
- Clarification of Request-Responses:
 - Abstract:
 - Clarification about failure cases (no response).
 - Modified volumes:
 - Part 3 – §7.4
- Clarification of C & C++ bindings:
 - Abstract:
 - Replace all guards in C and C++ language bindings with 'ECO_A_H' and 'ECO_A_ASSETS_H' forms as per C (C standard (ISO/IEC 9899) section 7.1.3/1) and C++ (C++ standard (ISO/IEC 14882) section 17.4.3.1.2/1) standards recommendations.
 - The user context header is not included by the container header.
 - Modified volumes:
 - Part 8
 - Part 9
- Clarification of Ada binding:
 - Abstract:
 - Missing types added to the example ECOA types package.
 - Modified volumes:
 - Part 10 - §17
- Clarification of High Integrity Ada binding:
 - Abstract:
 - "Asset" parameter is renamed to "Asset_Type" to fall inline with all other bindings.
 - Missing types added to the example ECOA types package.
 - Highlight differences versus other bindings has been added.
 - Recommendation to use only one of the Ada bindings for any Components within a Protection Domain as this will avoid any namespace clashes.
 - Modified volumes:
 - Part 11 - §6; §9.2; §11.6; §17
- Clarification of Platform procurement requirements:
 - Abstract:
 - New requirements have been added.
 - Modified volumes:
 - Part 5