European Component Oriented Architecture (ECOA®) Collaboration Programme: Guidance Document: ECOA® Logical Interface Example Data Distribution Service (DDS) Binding

Date: 27/11/2017

Prepared by
BAE Systems (Operations) Limited

Contents

Figures

Tables

No table of figures entries found.

0 Executive Summary

ECOA platforms communicate using the ECOA Logical Interface (ELI). The ELI defines a generic message format which is independent of the underlying transport mechanism. This guidance document provides an example of mapping this generic message format onto the Data Distribution Service (DDS) transport mechanism/middleware.

It is not in any way a “normative”, part of ECOA, or even definitive. The discussions here are purely examples of how ELI may be mapped on the DDS transport mechanism.

1 Scope

This document is intended to provide an example of mapping the ELI onto the DDS transport mechanism/middleware.

The document is structured as follows:

Section 2 gives a brief introduction to the topic.

Section 3 expands abbreviations used in this report.

Section 4 provides definitions for the key terms used in this report.

Section 5 lists key documents referenced by this report.

Section 6 discusses an example binding of ELI onto DDS.

Section 7 provides a summary of the guidance presented within this document.

2 Introduction

ECOA platforms communicate using the ECOA Logical Interface (ELI). The ELI defines a generic message format which is independent of the underlying transport mechanism. This guidance document provides an example of mapping this generic message format onto the Data Distribution Service (DDS) transport mechanism/middleware.

3 Abbreviations

API	Application Programming Interface
COTS	Commercial Off-The-Shelf
DGA	Direction Générale de l'Armement
Dstl	Defence Science and Technology Laboratory
ECOA	European Component Oriented Architecture
IP	Internet Protocol
IDL	Interface Definition Language
MOD	Ministry of Defence
SOA	Service-oriented Architecture
TCP	Transmission Control Protocol
XML	eXtensible Markup Language
XSD	XML Schema Definition

4 Definitions

For the purpose of this document, the definitions given in the ECOA Architecture Specification *(ref. [AS])* Part 2 and those given below apply.

Term	***Definition***
(currently none)	

5 References

AS	European Component Oriented Architecture (ECOA) Collaboration Programme: Architecture Specification (Parts 1 to 11) "ECOA" is a registered mark.

6 Data Distribution Service Binding

6.1 Domains and Partitions

DDS provides two mechanisms for scoping information:

Domain – A domain is a "virtual" network linking all DDS applications which have joined it;

Partition – A partition allows further organisation within a domain, where each partition can represent a logical grouping of topics.

These DDS concepts could be mapped onto the following ECOA Concepts:

Domain – A domain could be mapped to an ECOA Assembly. Each ECOA platform within the assembly would join the DDS domain which represents the ECOA Assembly. The name of the domain could be that of the composite (attribute "name" of the root element of the actual implemented assembly) as defined in the ELI Binding for the EUID of the Composite.

Alternatively, the domain does not need to be mandated; it could be left to the system integrator to split the ECOA system into a number of domains as appropriate for a given deployment.

Partition – A partition could be mapped to an ECOA Service Instance. Each ECOA Component would create a partition representing each provided service. Each ECOA Component requiring a service would join the partition. The name of a partition could be that of the "component_instance_name/provided_service_instance_name" as defined in the ELI Binding for the EUID of a Service Instance.

Figure 1 illustrates this mapping by showing how components and services are mapped onto DDS concepts of domains and partitions.

A single ECOA Assembly maps to a single DDS Domain.

"Component B" provides the service "1". "Component A" requires this service and is connected via a service link. In DDS, the (Container of) a Component providing a service will create a partition. The (Container of) a Component requiring the service will join this partition.

"Component C" provides two services, "2" and "3". Component A requires both of these services and is connected via two service links.

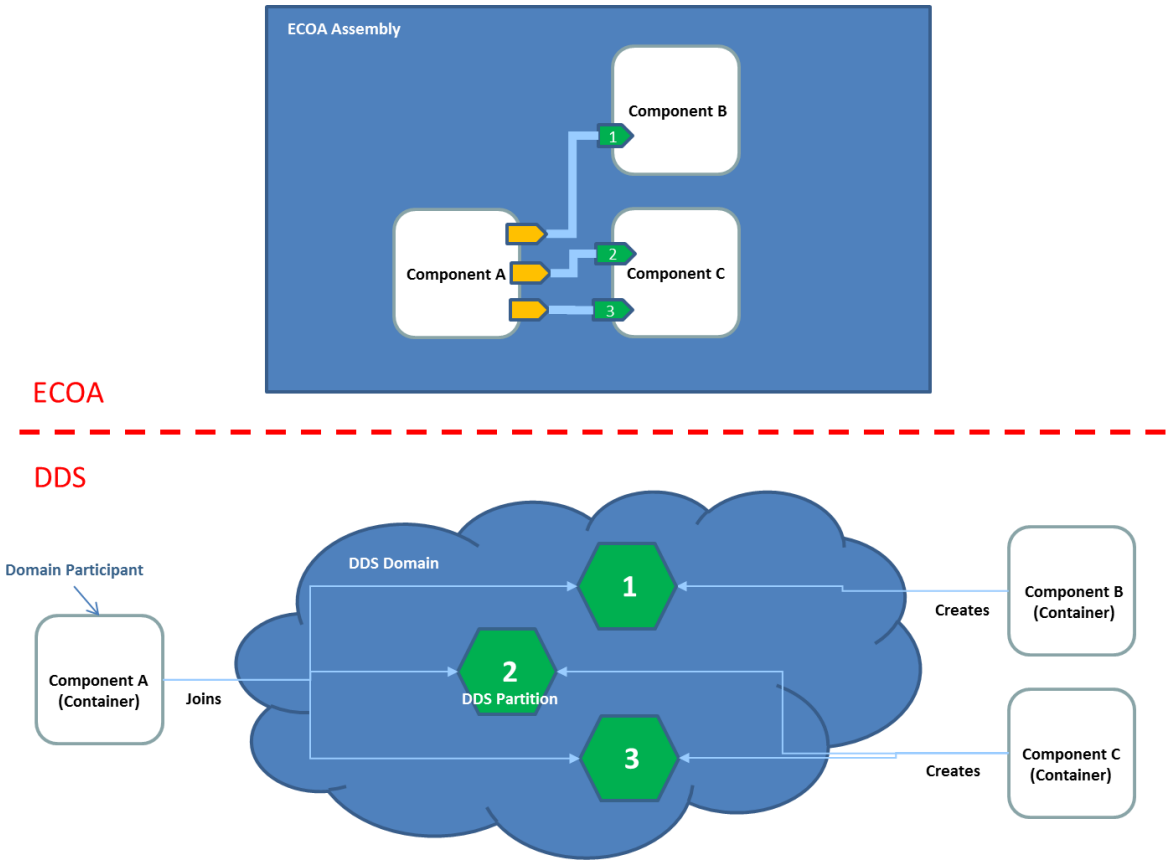

Figure 1 - Mapping Domains and Partitions

6.2 Topics

A topic in DDS represents the unit for information that can be produced or consumed by a DDS application. A topic is defined by a name, type and a set of Quality of Service (QoS) policies.

A topic type (defined as an IDL structure) is a structure which can contain as many fields as required. Additionally, a key can be defined for a topic (see section 6.2.1 for further detail).

A topic could be mapped to an ECOA Service Operation. Each (Container of) an ECOA Component that provides/requires a service would create/join a topic (within the applicable partition) for each service operation. Note that Request-Response operations are in effect two operations (each possibly containing a set of parameters) so this could be mapped as two separate topics; one topic for the request operation and another topic for the response operation.

Depending upon the type of operation the (Container of) the Component would either subscribe or publish to the topic.

Figure 2 illustrates this concept and shows how the 3 different types of ECOA Service Operation could be mapped to the topic.

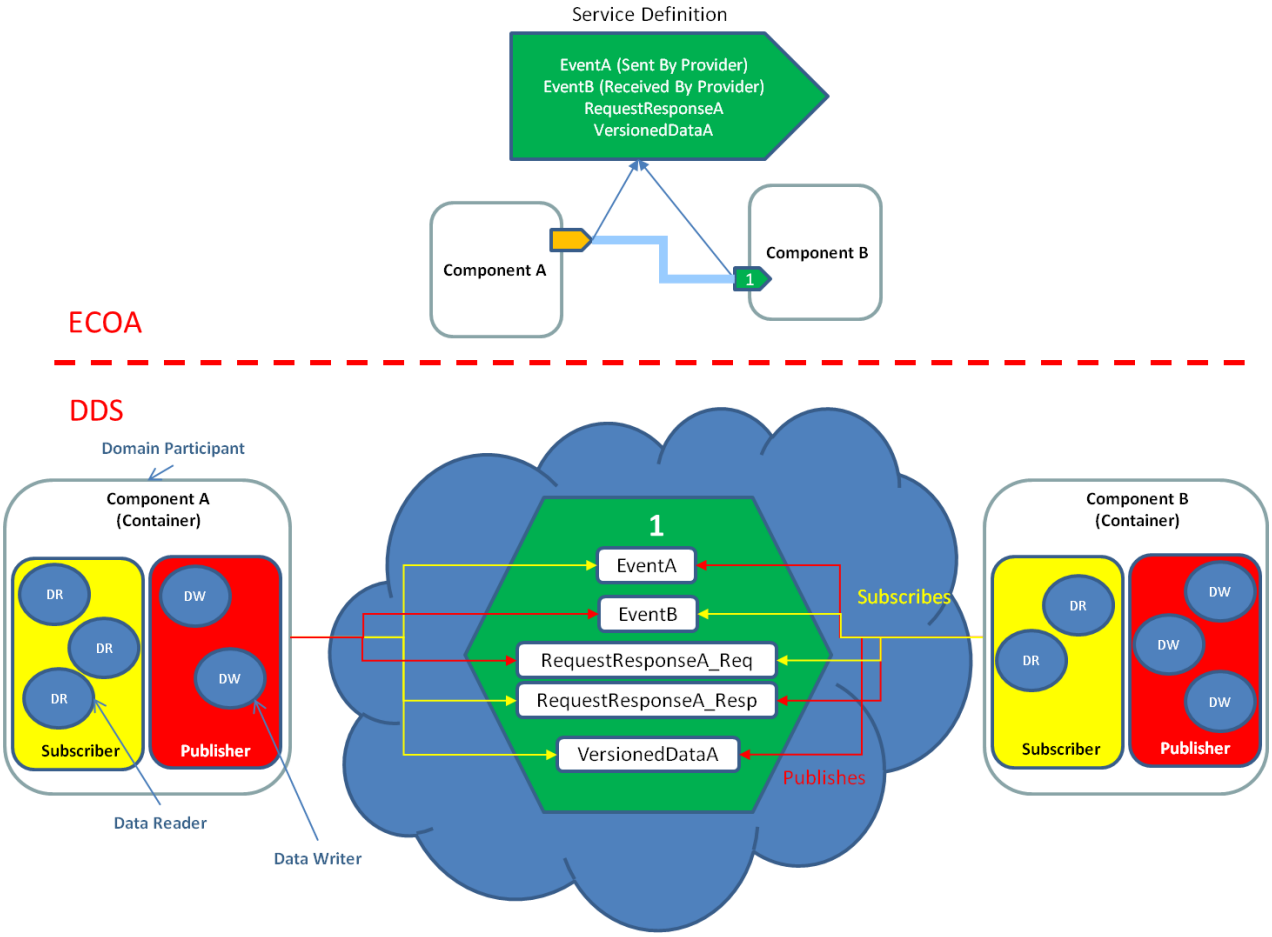

Figure 2 - Mapping Topics

6.2.1 Topic Instances (using Topic Keys)

Topic Keys can be used to create separate instances of a topic without defining multiple separate topics. This allows the system to contain multiple instances of a given topic, with each instance uniquely identified by a "key". The key can be made up of one or more data-types within the topic.

This mechanism could be used in the ELI DDS Binding to determine the client of a request (in order to know who to respond to). In this scenario, a topic instance would exist for each client connected. Equally, the response topic could use the same method to ensure only the requesting client receives the response to the request. To achieve this, the key of the topic could be the service operation UID as defined in the ELI Binding Specification.

6.3 RTPS Protocol

The Real-time Publish-Subscribe Protocol provides interoperability between different implementation of the DDS specification. This protocol can be used to ensure that ECOA platforms can interoperate. It performs much of the functionality defined in the ECOA ELI specification including serialization and deserialization of data.

7 Summary

Although it is possible to use DDS as the transport mechanism for ELI, it is likely not the most efficient solution. A lower-level transport mechanism is likely to be a more appropriate solution in most situations; such as UDP or TCP. This is effectively the layer below DDS, as DDS itself can use TCP or UDP as its underlying transport mechanism.