The DDS Security Standard

Angelo Corsaro, PhD

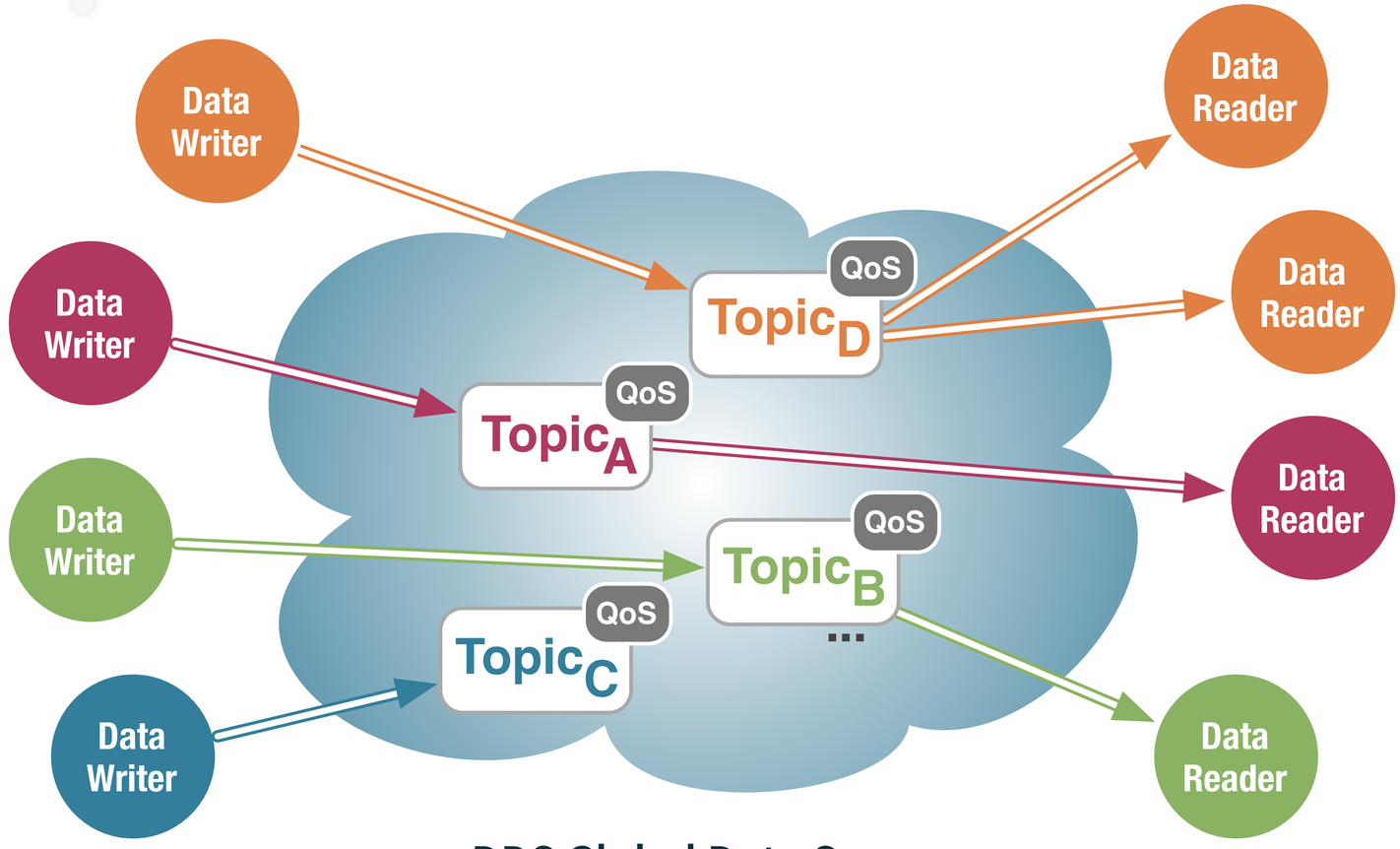
CTO, ADLINK Tech. Inc.
Co-Chair, OMG DDS-SIG
angelo.corsaro@adlinktech.com

PRISMTECH™
AN ADLINK COMPANY

DDS Refresher

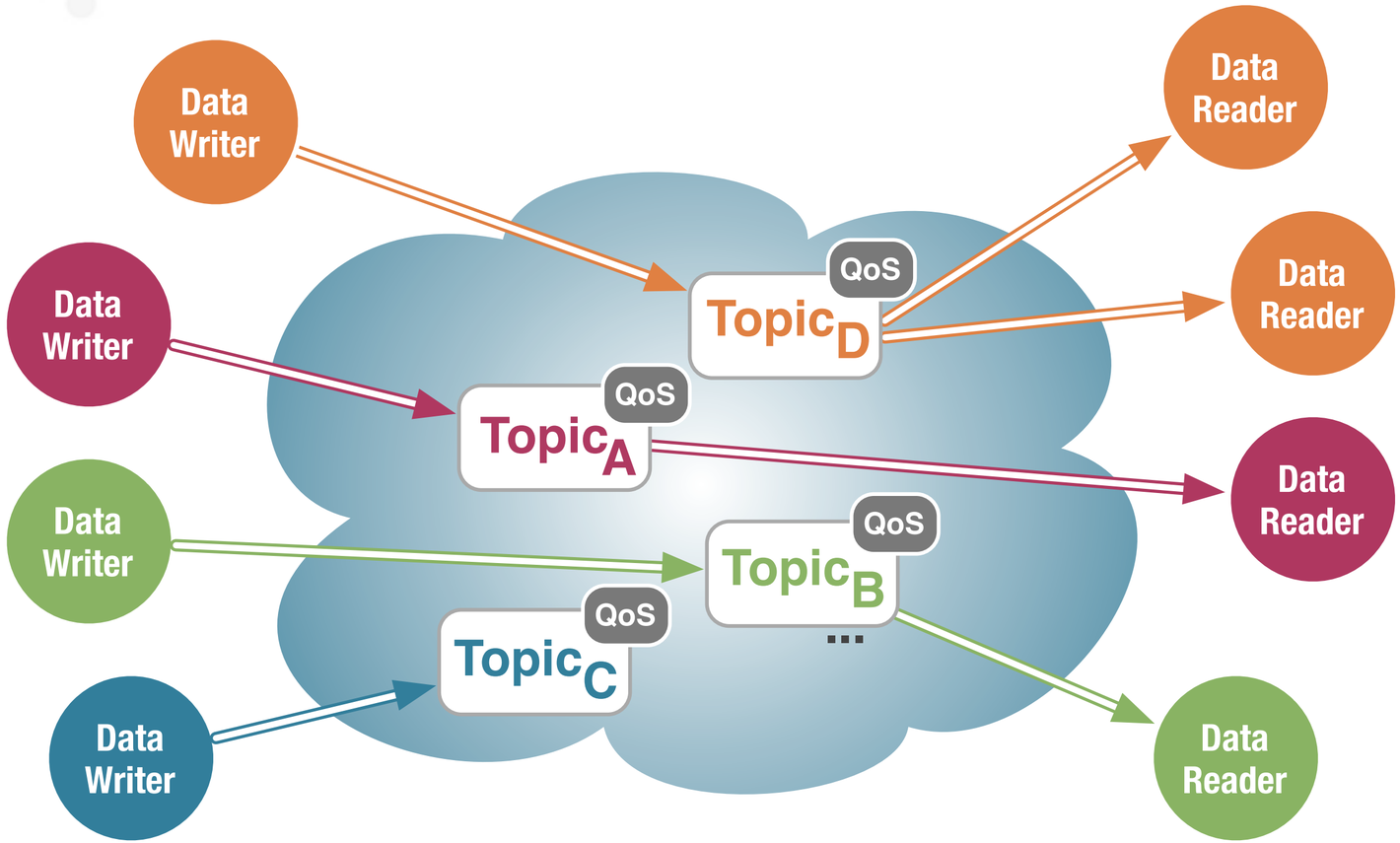
DDS Abstraction

DDS provides applications with a Virtual **Global Data Space** abstraction



DDS Abstraction

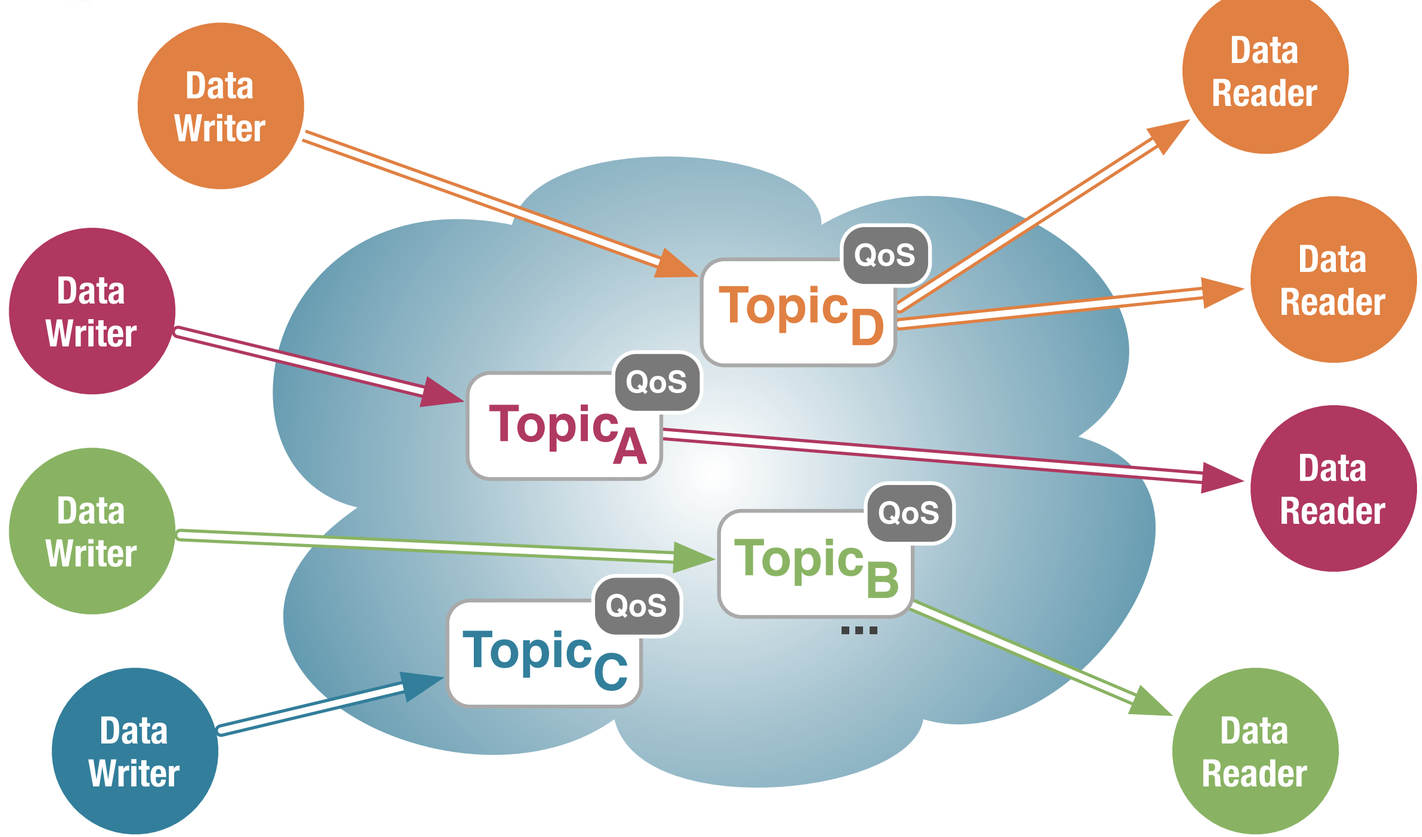
Applications coordinate by **autonomously** and **asynchronously reading** and **writing** data in the Data Space enjoying **spatial** and **temporal decoupling**



DDS Global Data Space

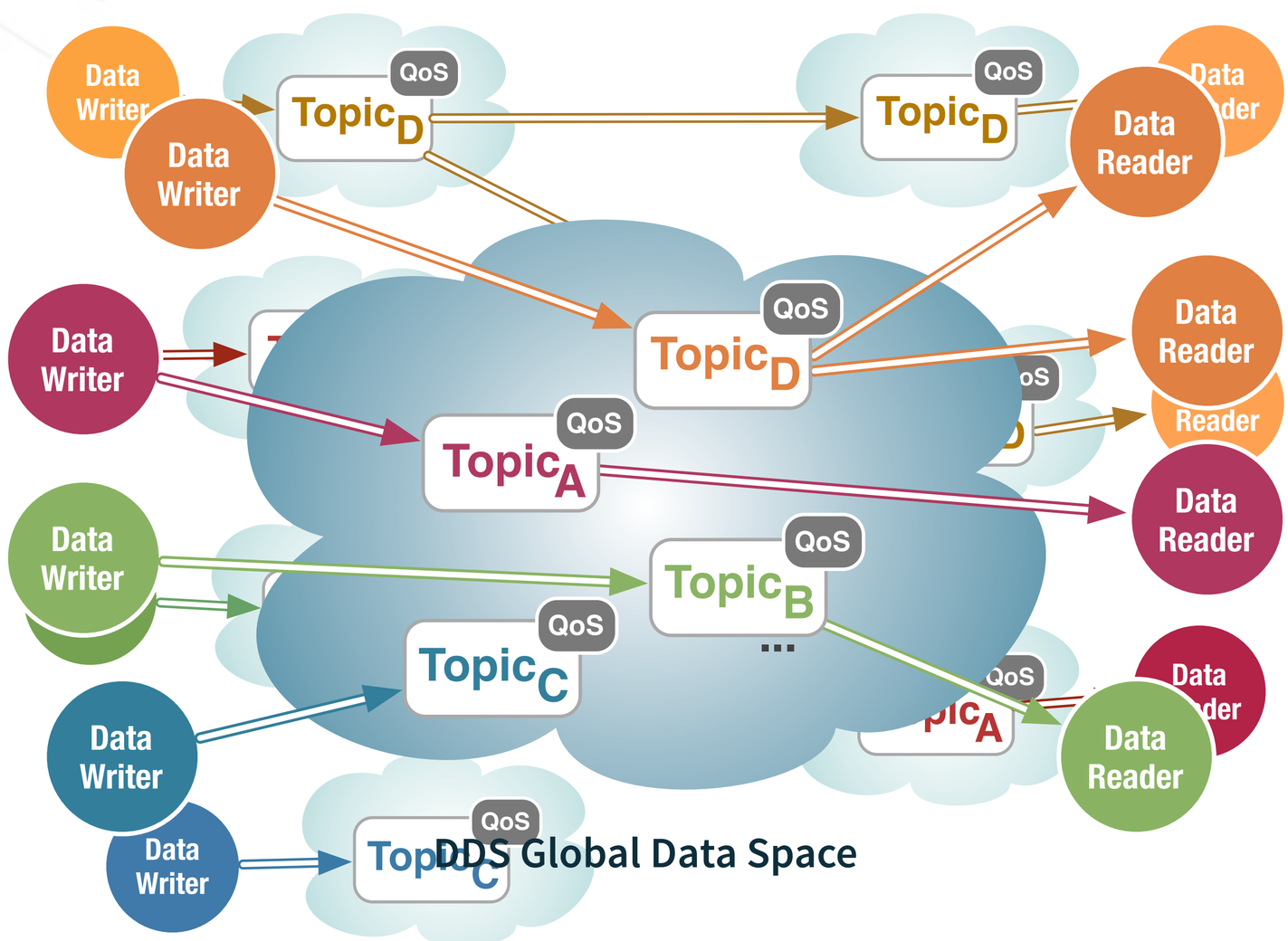
Dynamic Discovery

DDS has built-in dynamic discovery that **automatically matches interest** and **establishes data path isolating applications from network topology** and **connectivity** details



Decentralised Data-Space

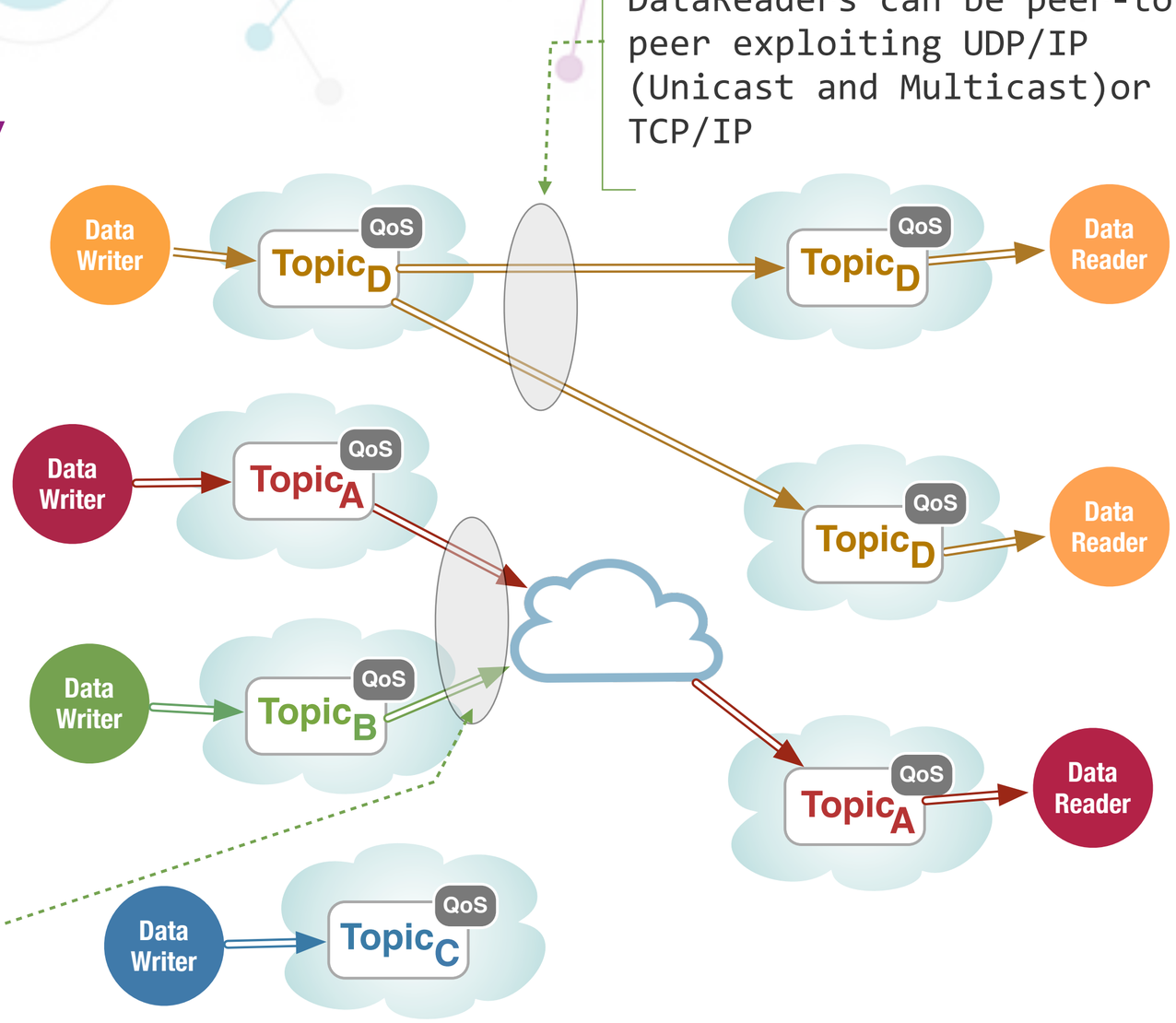
DDS global data space implementation is **decentralised** and does not suffer of single point of failure or bottleneck



Adaptive Connectivity

Connectivity is **dynamically adapted** to chose the most effective way of sharing data

The communication between the DataWriter and matching DataReaders can be peer-to-peer exploiting UDP/IP (Unicast and Multicast)or TCP/IP

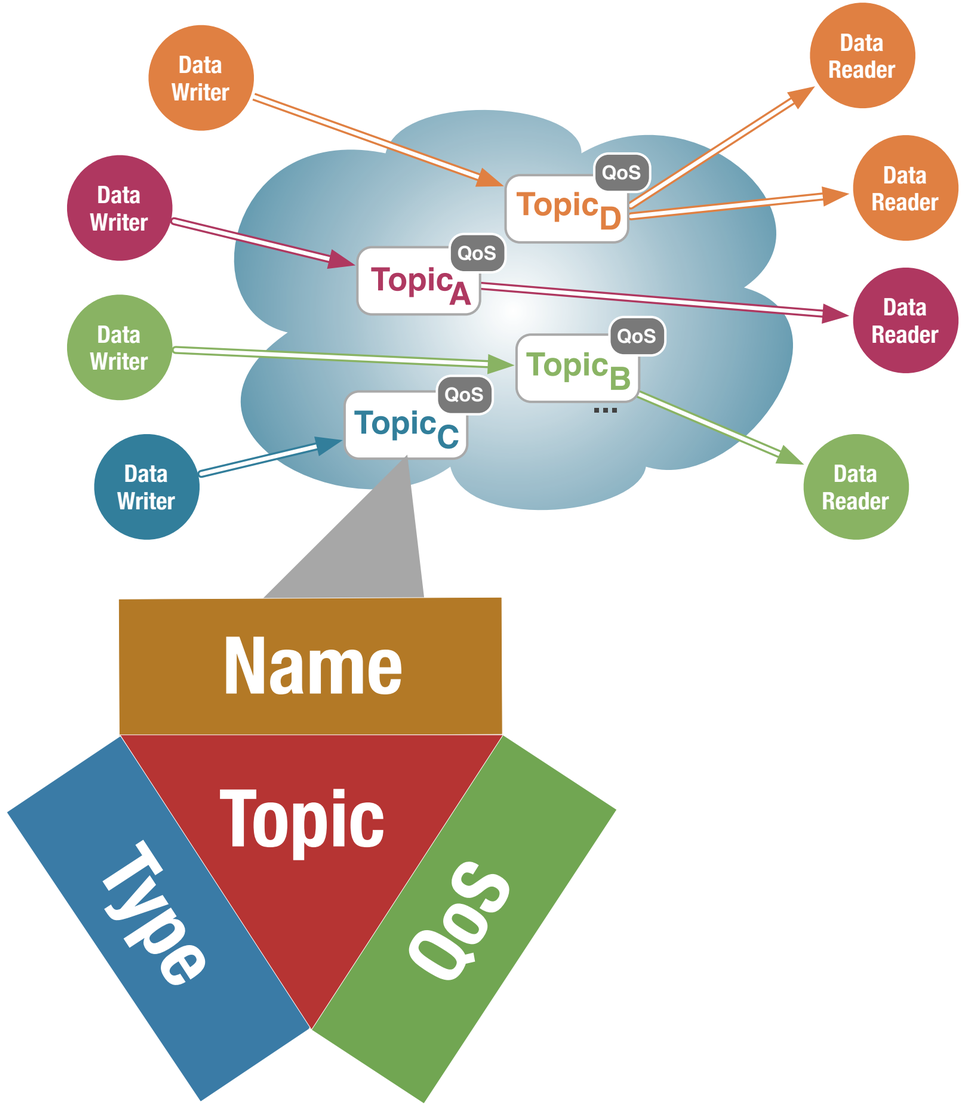


The communication between the DataWriter and matching DataReaders can be “brokered” but still exploiting UDP/IP (Unicast and Multicast)or TCP/IP

Topics Organisation

DDS data streams are defined by means of **Topics**

A **Topic** represented is by means of a <name, type, qos>

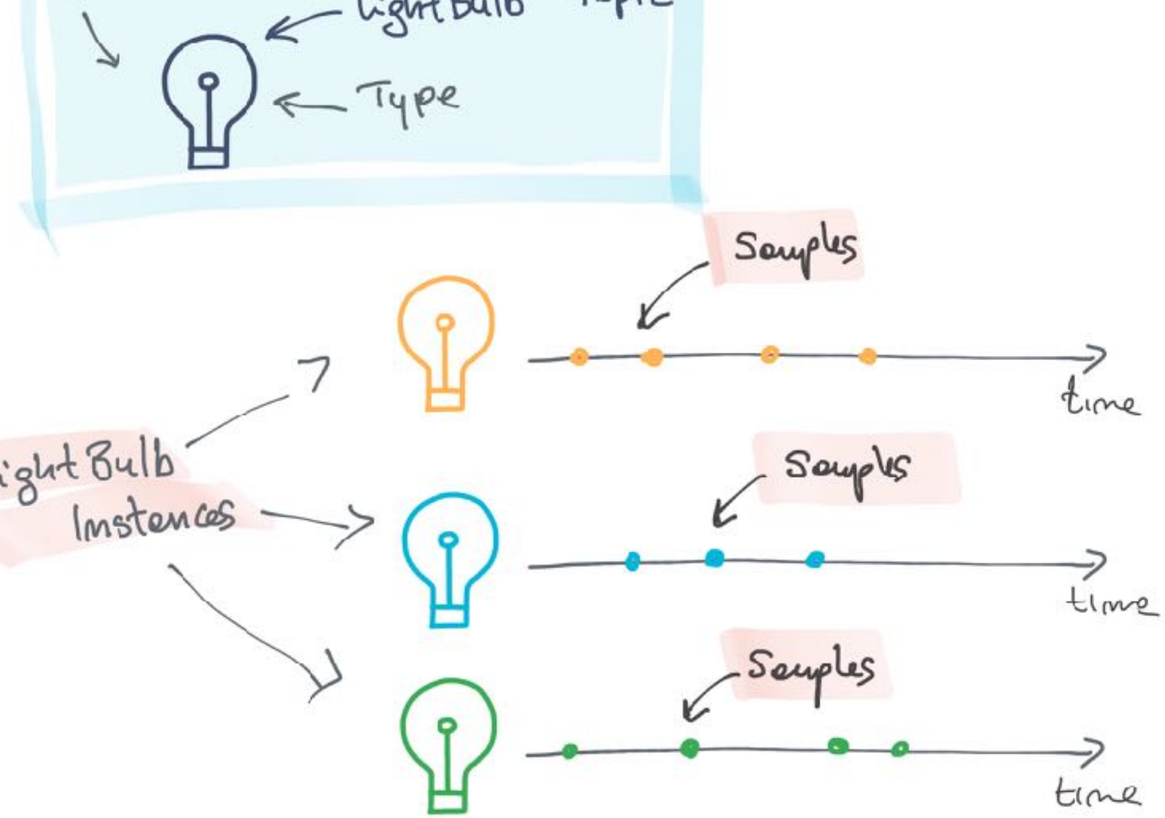


Topic Instances

Topic may mark some of their associated type attributes as key-fields

Each unique key value (tuple of key attributes) identifies a Topic Instance. Each Topic Instance has associated a FIFO ordered stream of samples

DDS provides useful **instance life-cycle management** and samples demultiplexing

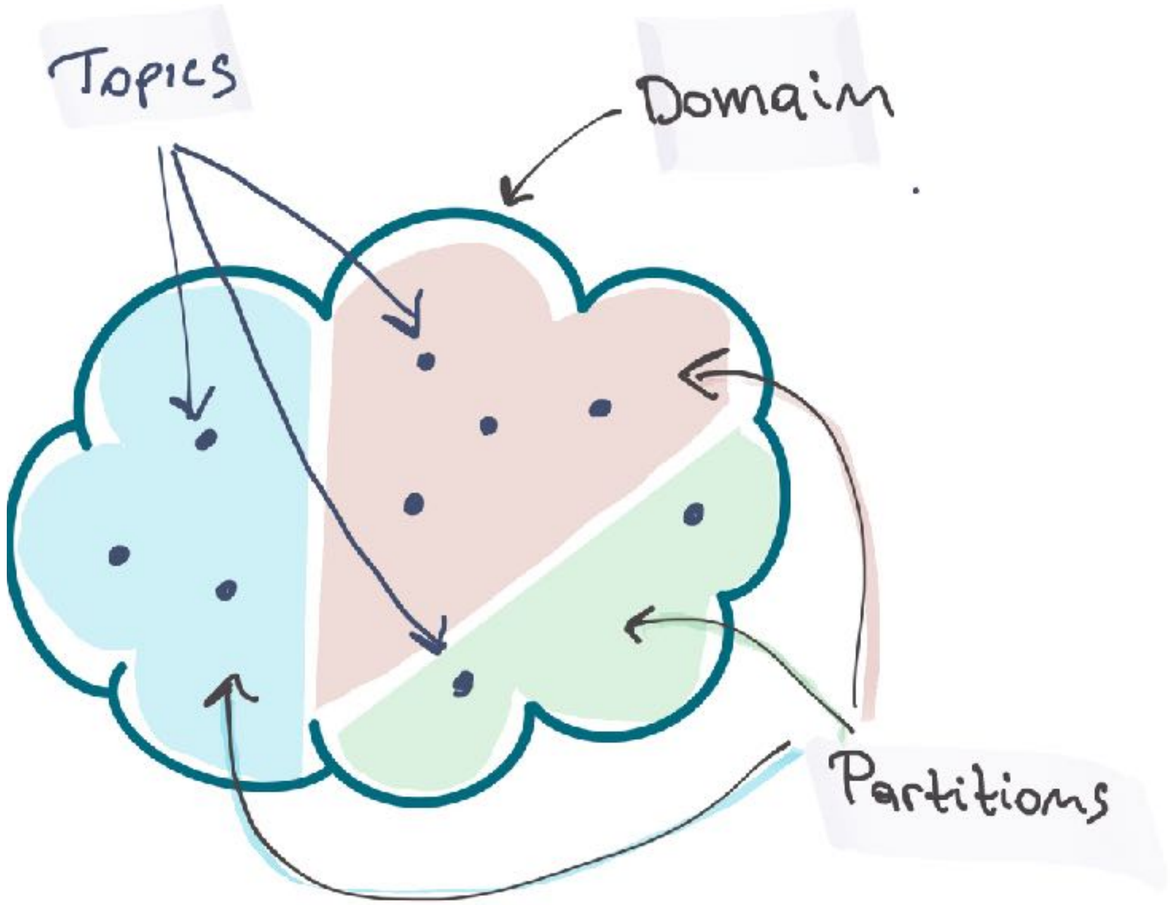


Information Scopes

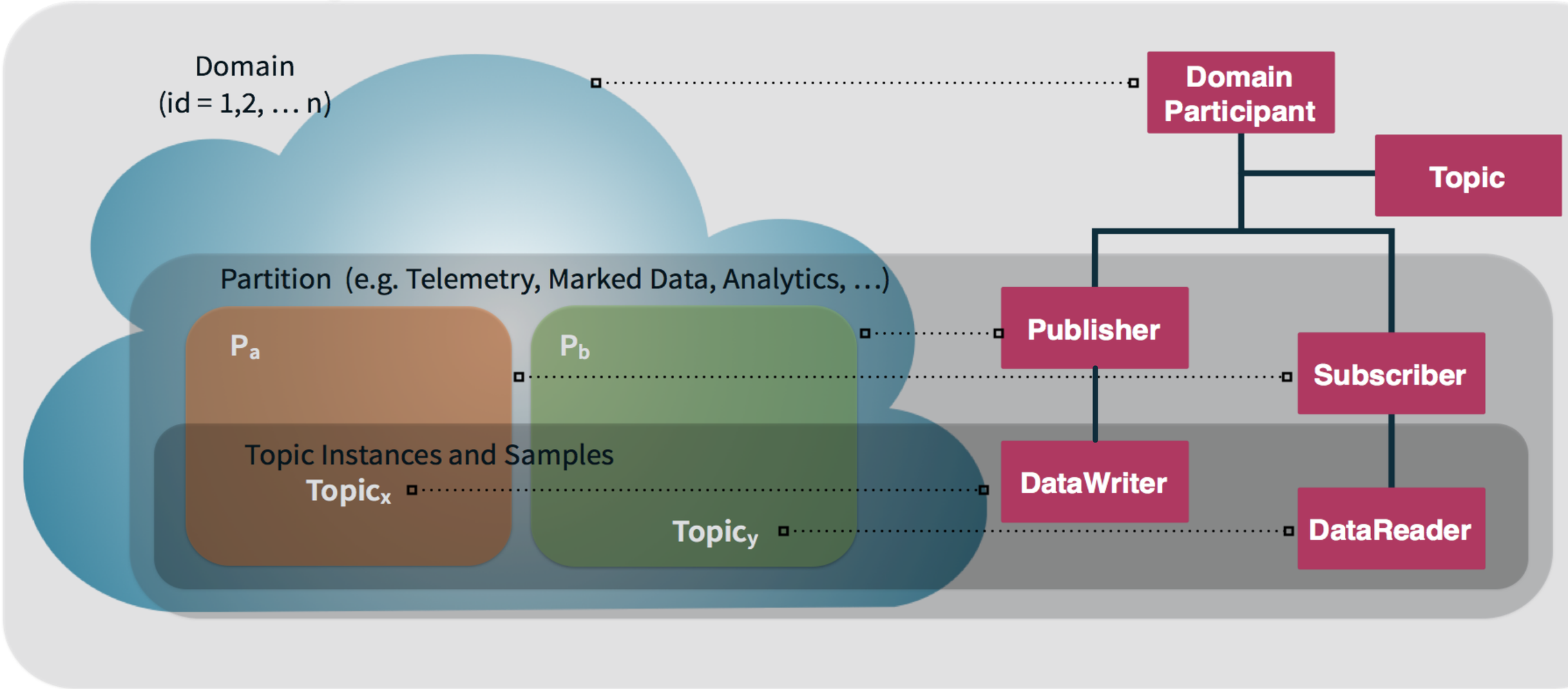
DDS information lives within a **domain**

A domain can be thought as organised in **partitions**

Samples belonging to a given **Topic Instance** are read/written from/in one or more **partitions**



Anatomy of a DDS Application



Security Models

Network/Transport Security

Network Security implements authentication and encryption at Layer 3 of the ISO/OSI reference model, *e.g.*, IPSec

Transport Security implements a secure channel above a Layer-4 abstraction, *e.g.*, TLS for TCP/IP and DTLS for UDP/IP

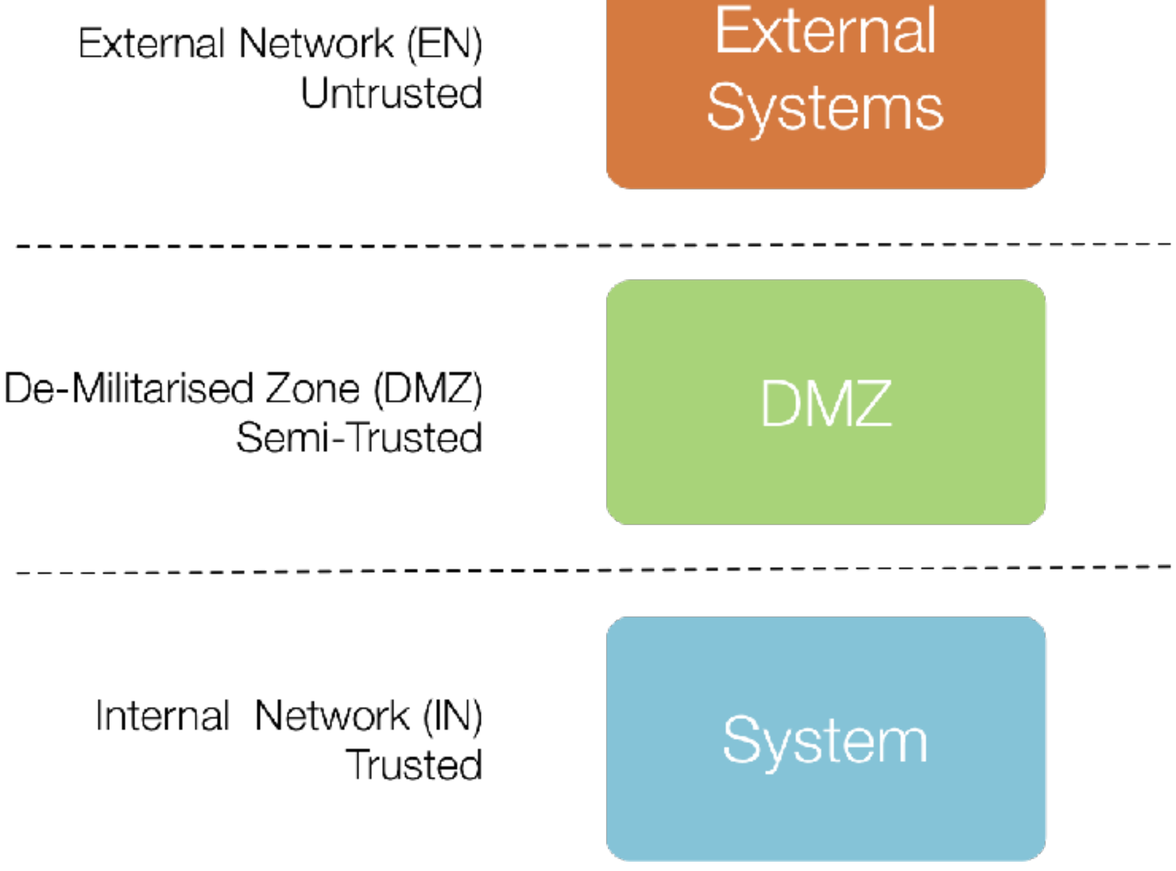
Notice that network security and transport security are concerned with the creation of secure channels, but no access control is performed over exchanged information

Boundary Security

Boundary security is concerned with securing the interconnection of a system with an untrusted network.

The **system network**, is considered **trusted**. In any case security within the system is addressed using traditional LAN-level security techniques

DMZ is a common way of implementing boundary security. Notice that access control can be executed as part of the DMZ

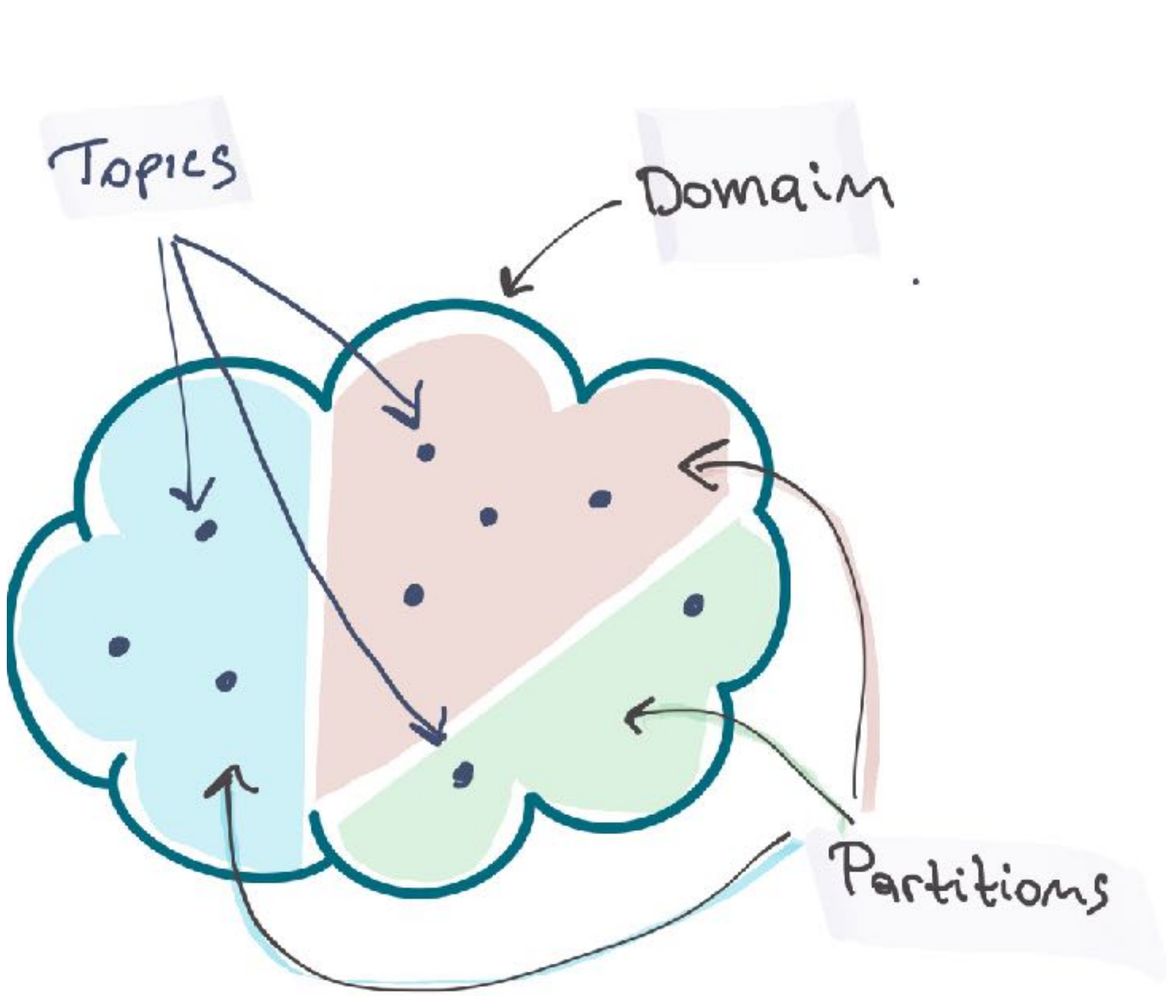


Toward DDS Security

DDS vendors have been providing transport security solution for some time.

Network/transport security solutions based on TLS/DTLS are not suitable for high-fan-out group communication nor for real-time data flows

Additionally Network/Transport security solutions don't provide any access control

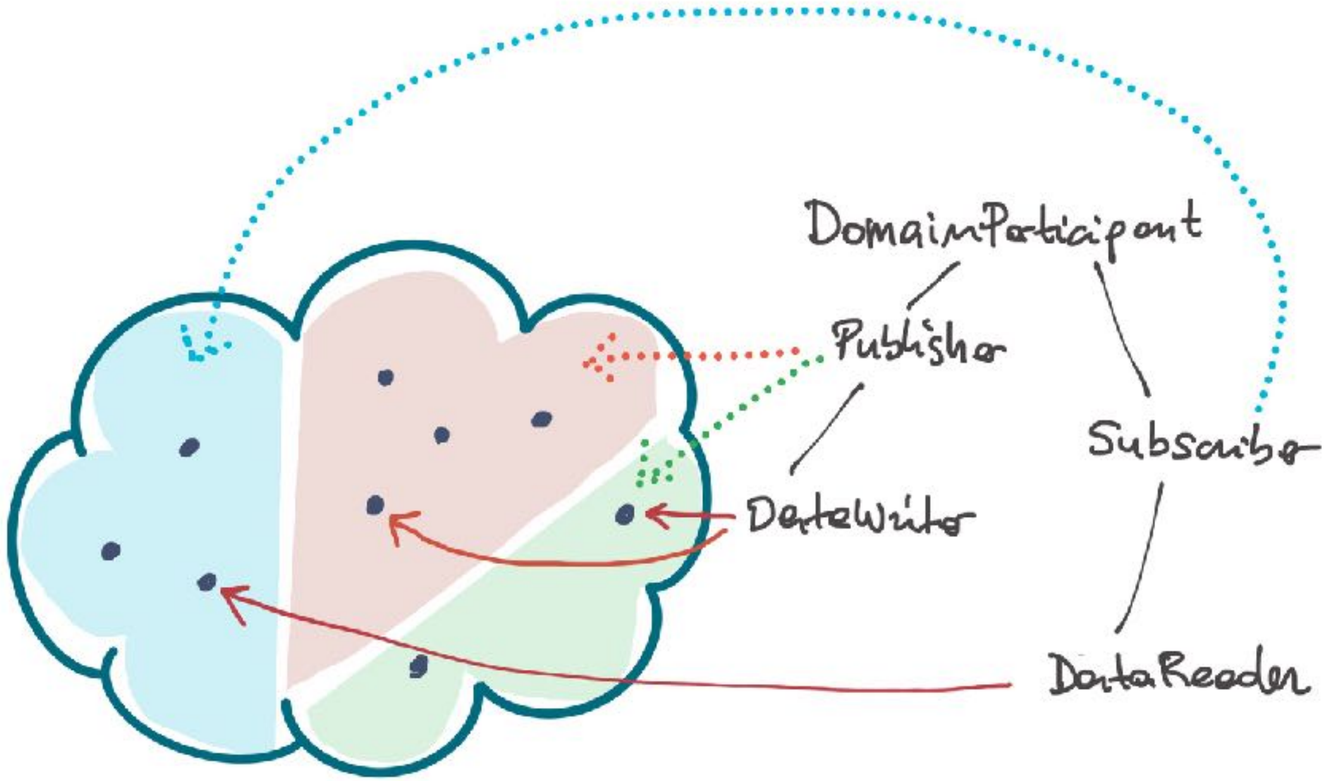


DDS Security Goals

Provide a **data-centric security** that allows to control access to the DDS Global Data Space

Ensure that the security solution is multicast-friendly

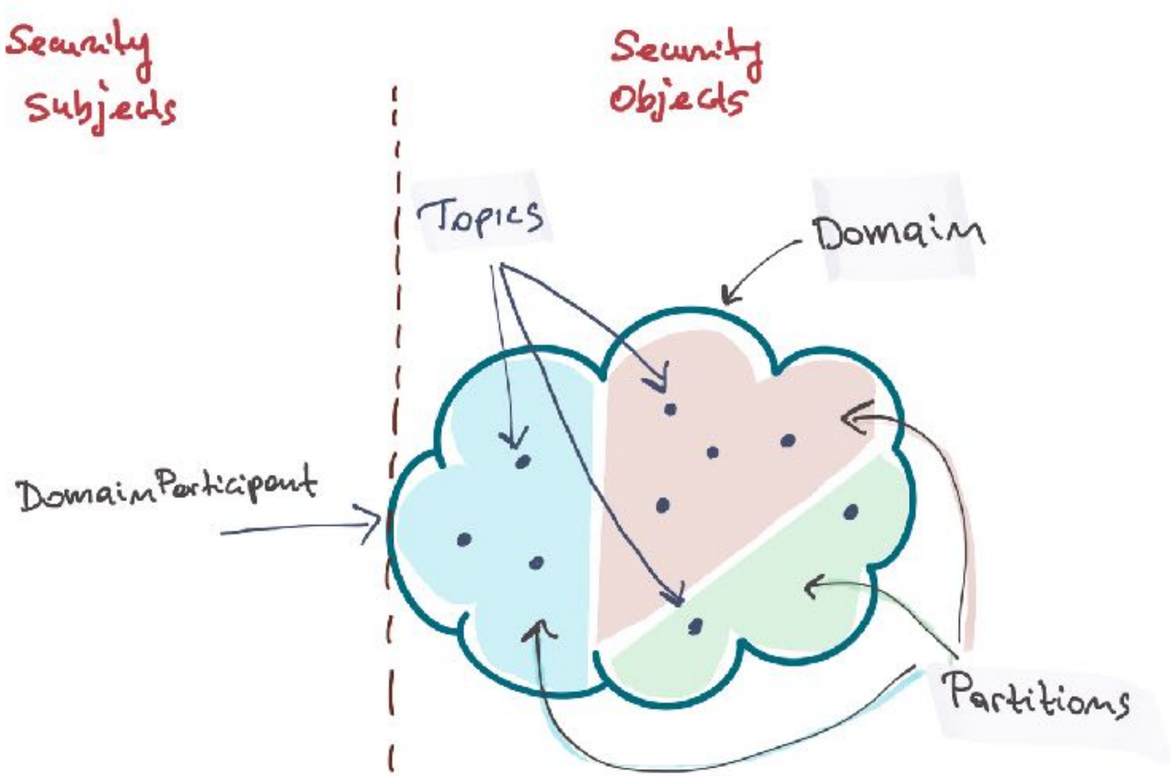
Design for extensibility and customisability



DDS Security Model

The DDS Security Model **defines** the **security principals**, the **objects** that are being **secured**, and the **operations** on the objects that are to be restricted

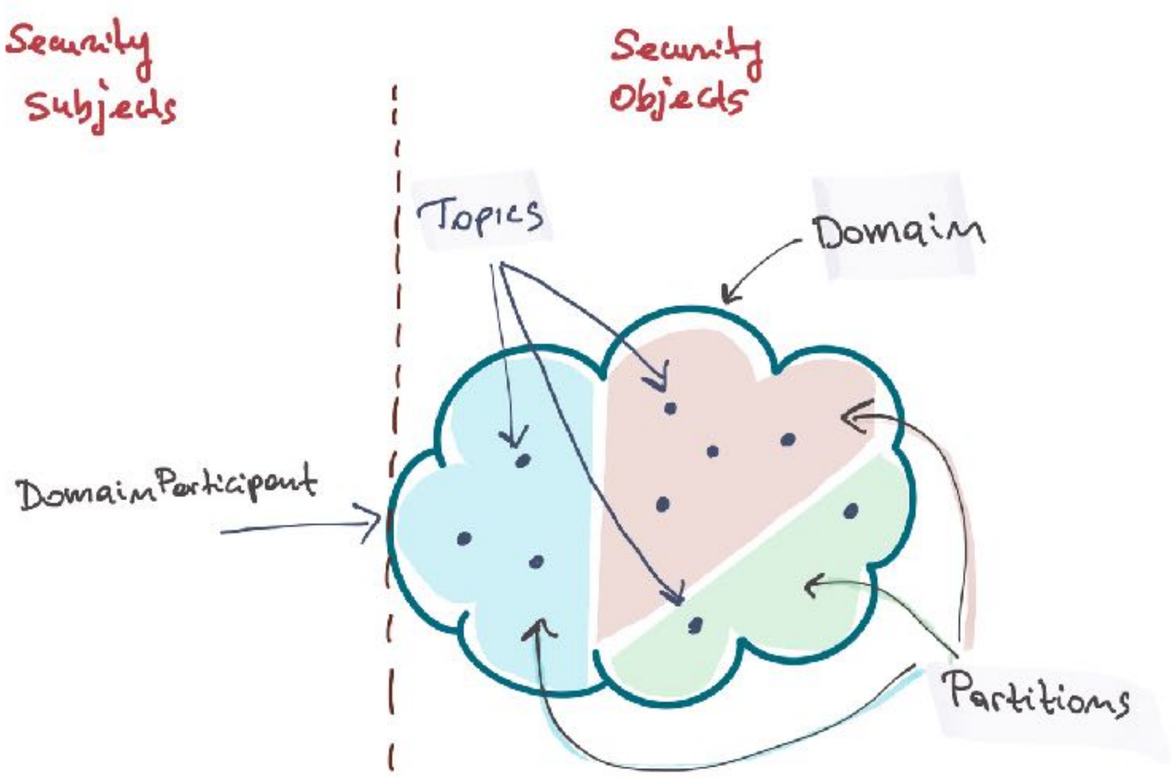
DDS applications share information on DDS Global Data Spaces (DDS Domains) where the information is organised into **Partitions, Topics** and **Instances** and accessed by means of **read** and **write** operations on data-instances of those Topics.



DDS Security Model

The DDS Security provides

- **Confidentiality** of the data samples
- **Integrity** of the data samples and the messages that contain them
- **Authentication** of DDS writers and readers
- **Authorisation** of DDS writers and readers
- **Non-repudiation** of data

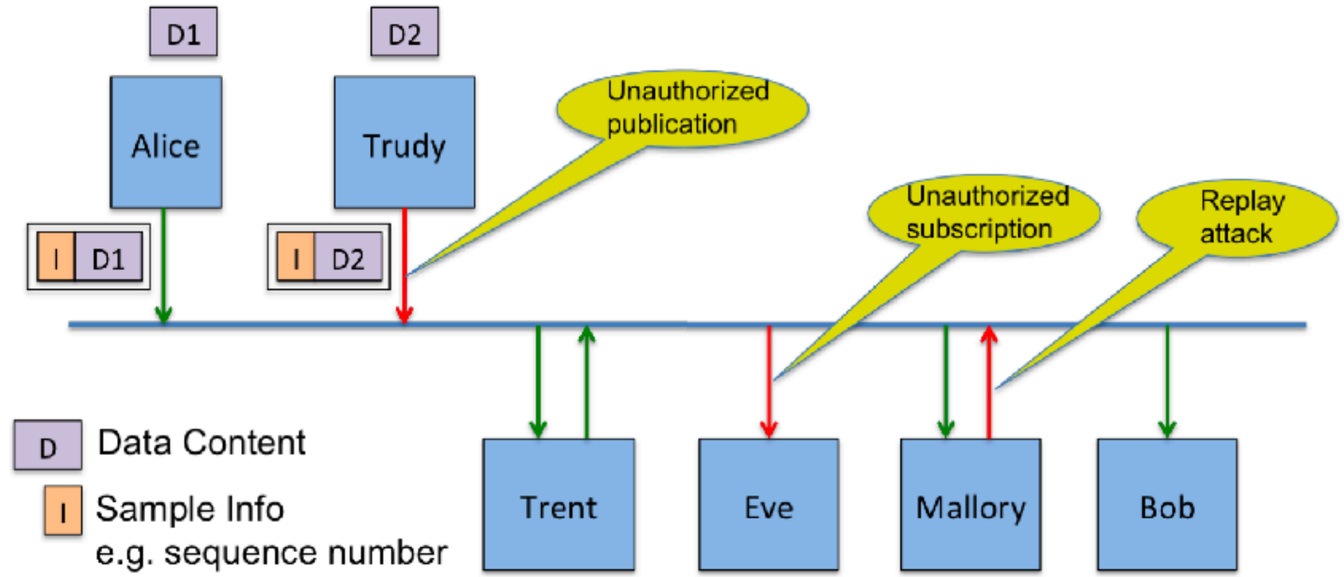


Security Threats

Security Threats

The DDS Security specification provide protection against:

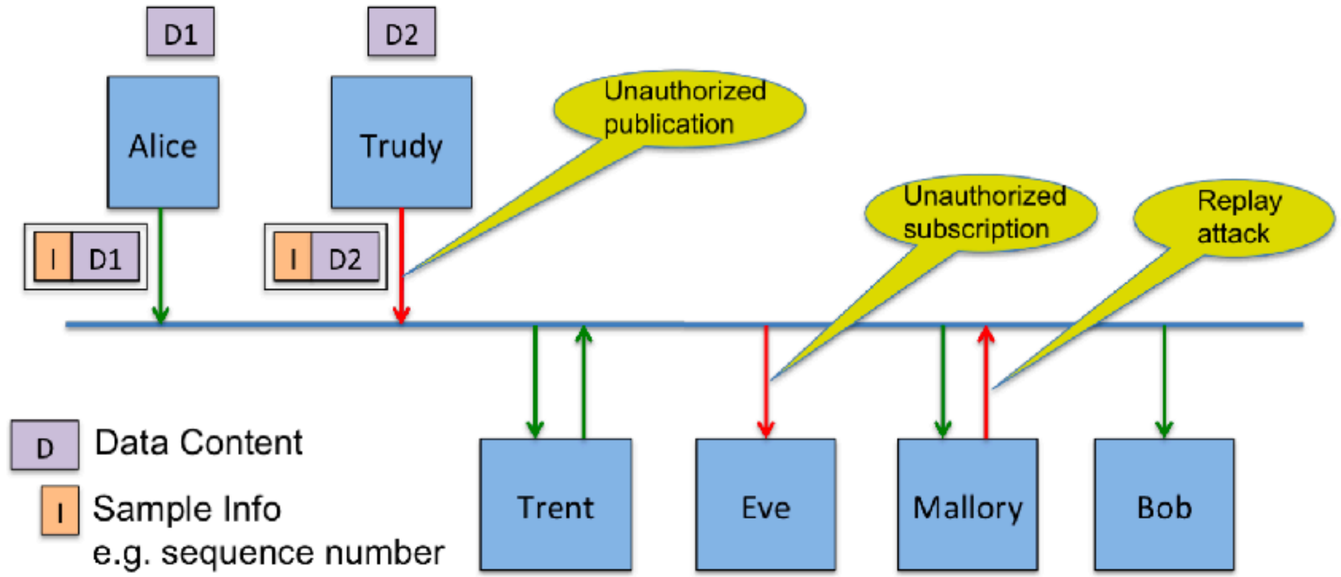
- **Unauthorised Subscription**
- **Unauthorised Publication**
- **Tampering and Replay**
- **Unauthorised data access**



[See DDS Security Specification v1.0 p.9]

Unauthorised Subscription

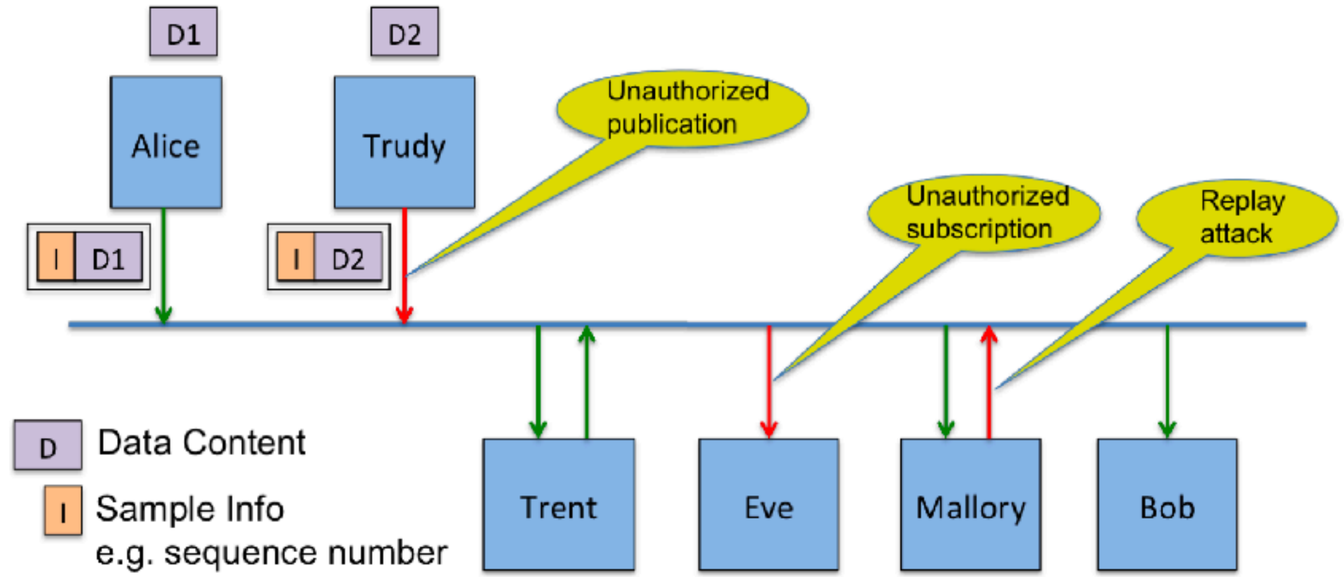
Eve has managed to connected to the same network as the rest of the applications and is able to see DDSI packets not addressed to him



[See DDS Security Specification v1.0 p.9]

Unauthorised Subscription

To protect **Alice** against **Eve's** unauthorised subscription, DDS Security uses a secret key shared only with authorised readers such as **Bob**, **Trent** and **Mallory**

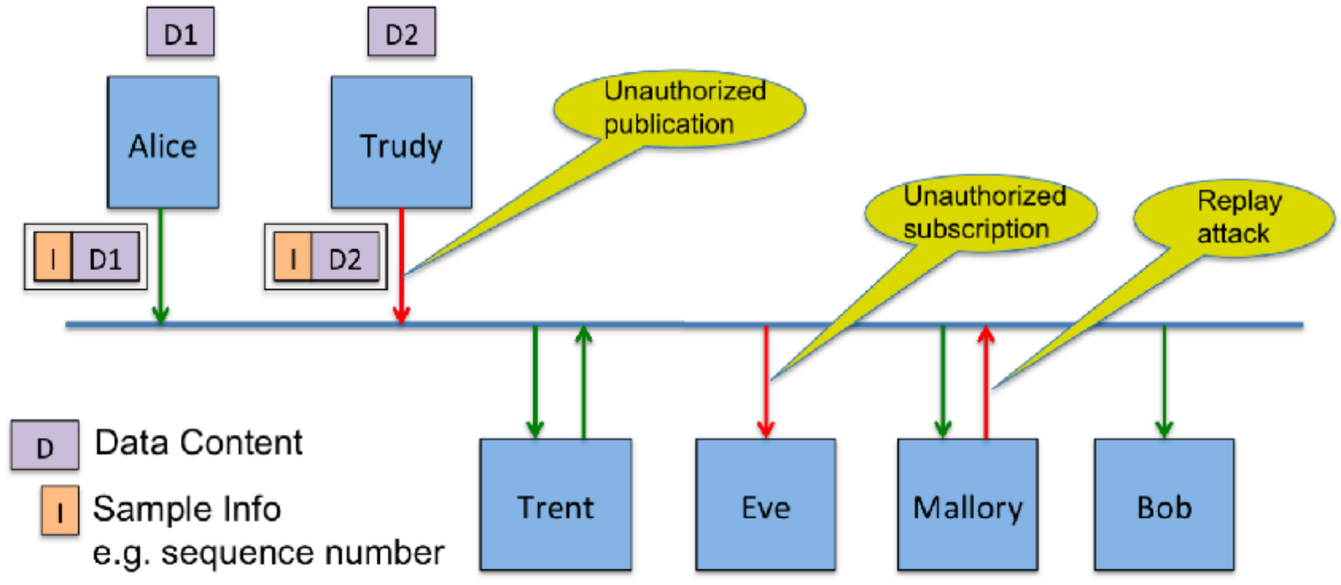


[See DDS Security Specification v1.0 p.9]

Unauthorised Publication

Trudy is connected to the same network infrastructure as the rest of the agents and **is able to inject network packets with any data contents**, headers and destination she wishes (e.g., Bob).

The network infrastructure will route those packets to the indicated destination

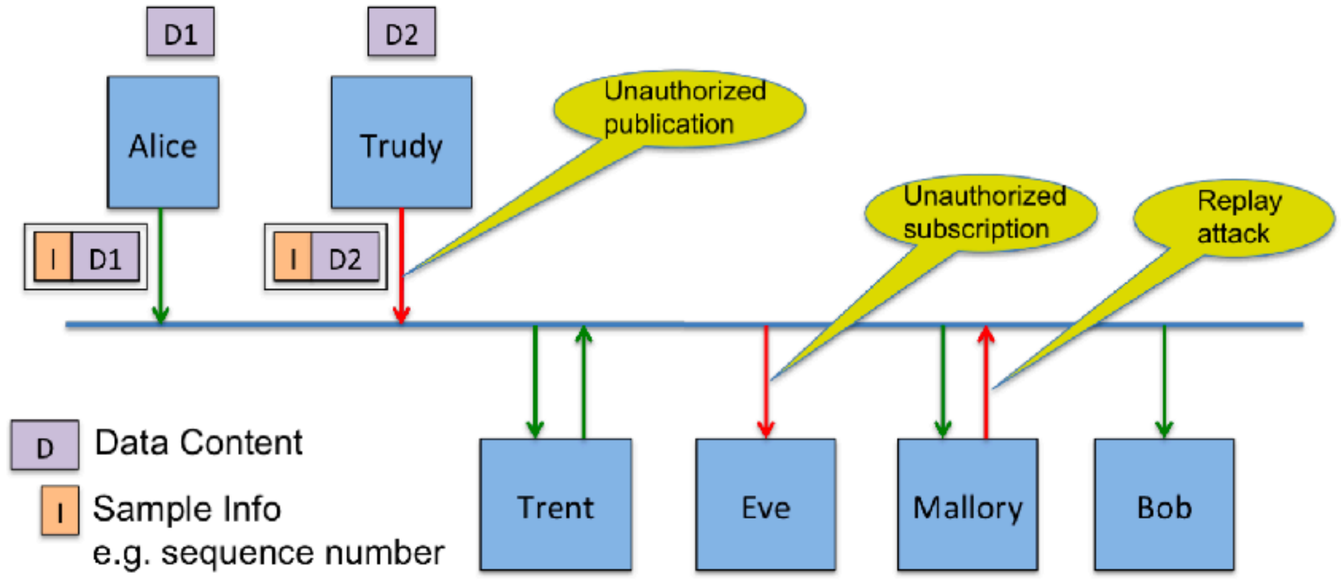


[See DDS Security Specification v1.0 p.9]

Unauthorised Publication

To protect against Trudy, Bob, Trent and Mallory need to realise that the data is not originating from Alice.

They need to realise that the data is coming from someone not authorised to send data or Topic T and therefore reject the packet

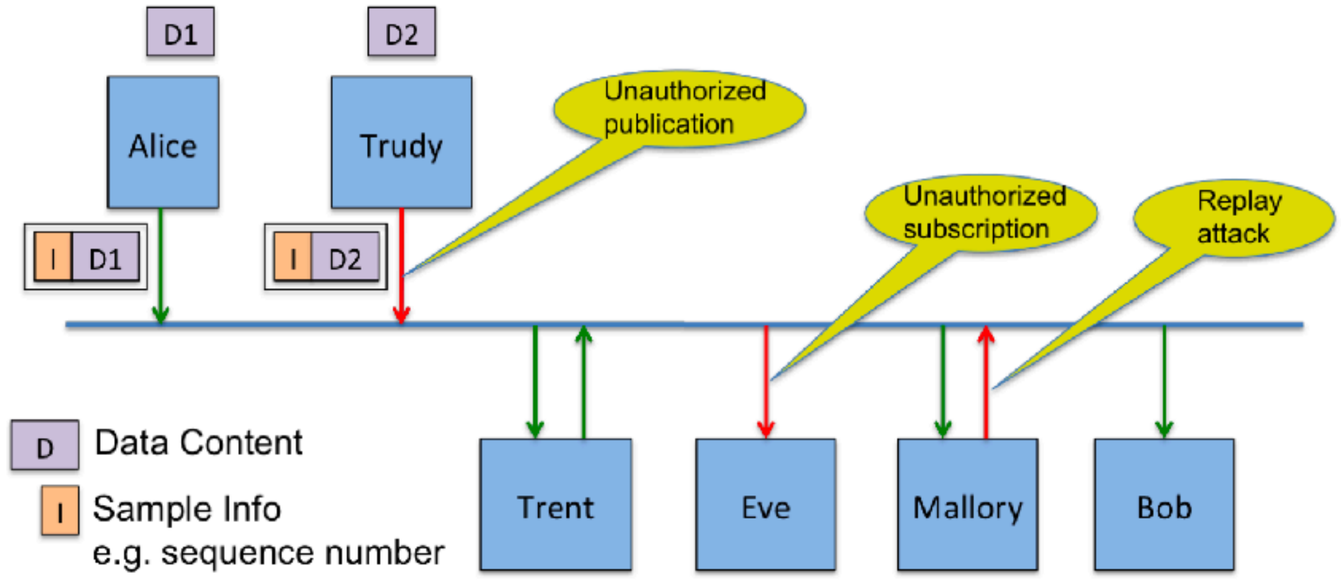


[See DDS Security Specification v1.0 p.9]

Unauthorised Publication

To protect against Unauthorised Publications the DDS Security standard allows writers to sign their messages

Messages can be signed with either a hash-based message authentication code (HMAC) or digital signature.

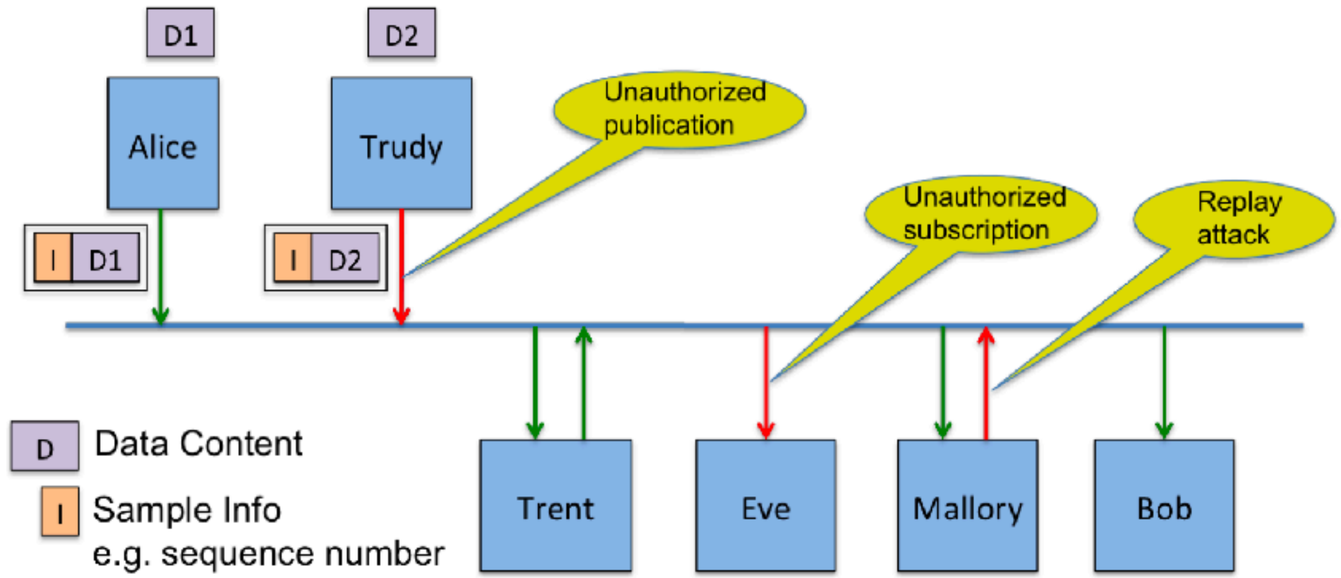


[See DDS Security Specification v1.0 p.9]

Tampering and Relay

Mallory is authorised to subscribe to Topic T

Therefore **Alice** has shared with Mallory the secret key to encrypt the topic and also, if an HMAC is used, the secret key used for the HMAC



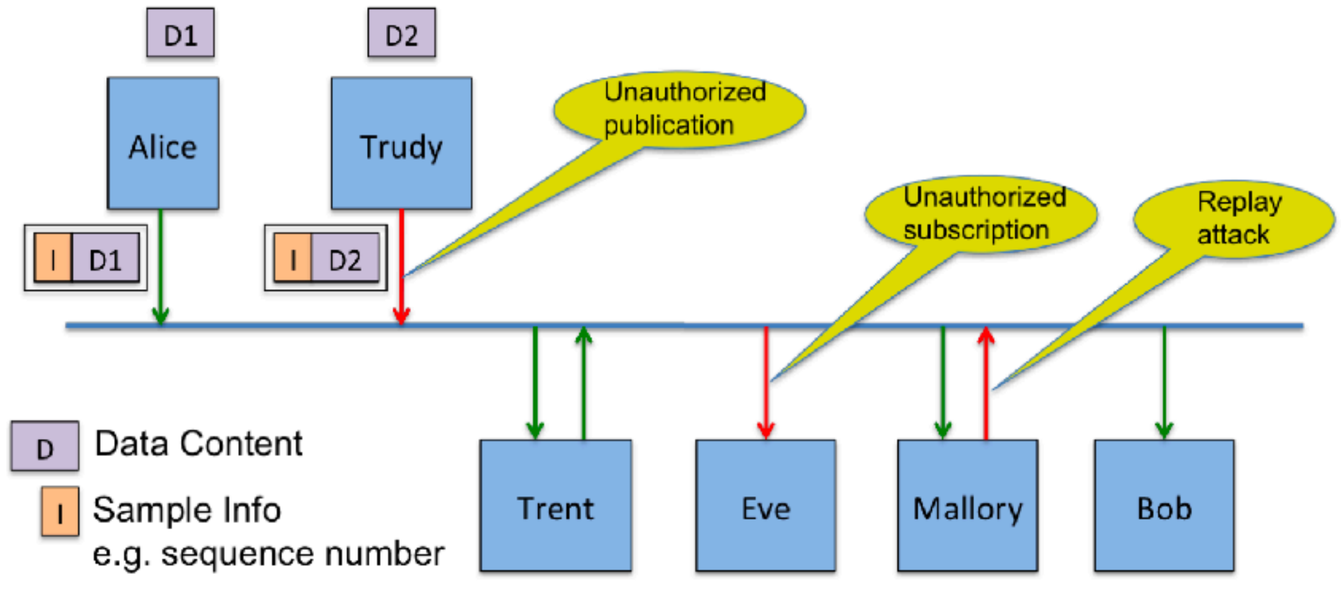
[See DDS Security Specification v1.0 p.9]

Assume Alice used HMACs instead of digital signatures. Then Mallory can use her knowledge of the secret keys used for data encryption and the HMACs to create a message on the network and pretend it came from Alice.

Tampering and Relay

If **Alice** used an HMAC, the only solution to the problem is to have a different HMAC secret key for each matching reader.

Then Mallory will not have the HMAC key that Bob expects from Alice and the messages from Mallory to Bob will not be misinterpreted as coming from Alice.

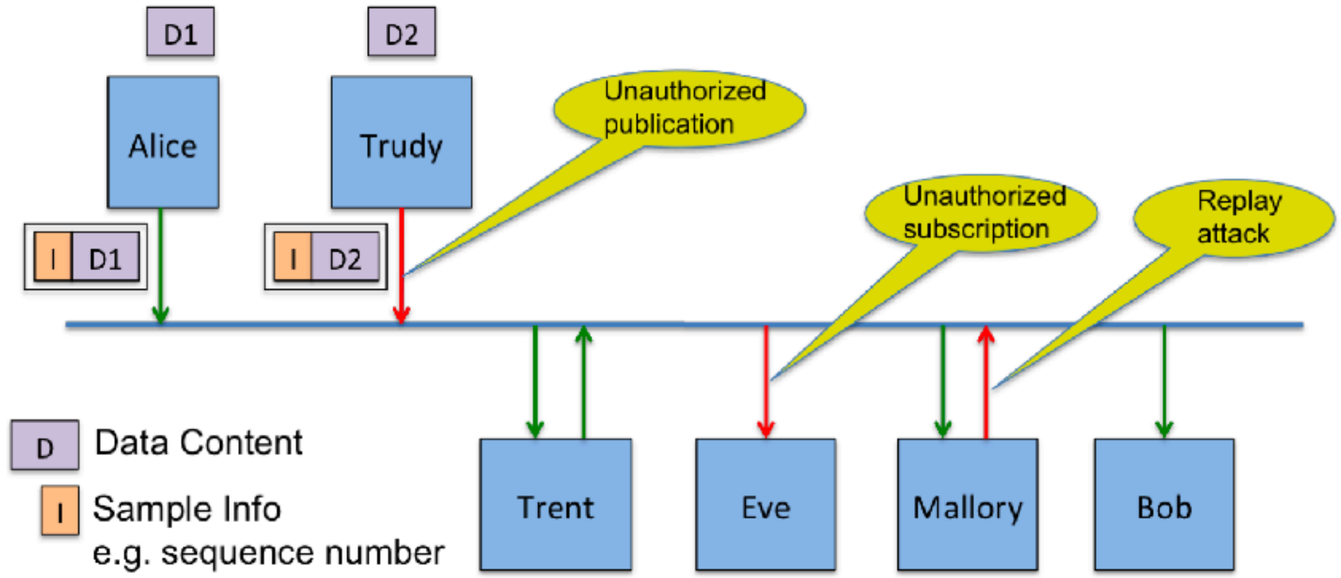


[See DDS Security Specification v1.0 p.9]

Tampering and Relay

If **Alice uses multicast** to communicate efficiently with multiple receivers, she will have to send an **HMAC** with a **different key for every receiver**. In other terms multiple HMACs will be appended to the multicast message along with some key-id that allows the recipient to select the correct HMAC

Signing messages using a **digital signature based** on **public key cryptography removes this challenge** at the cost of running a **more time-consuming** signing algorithm



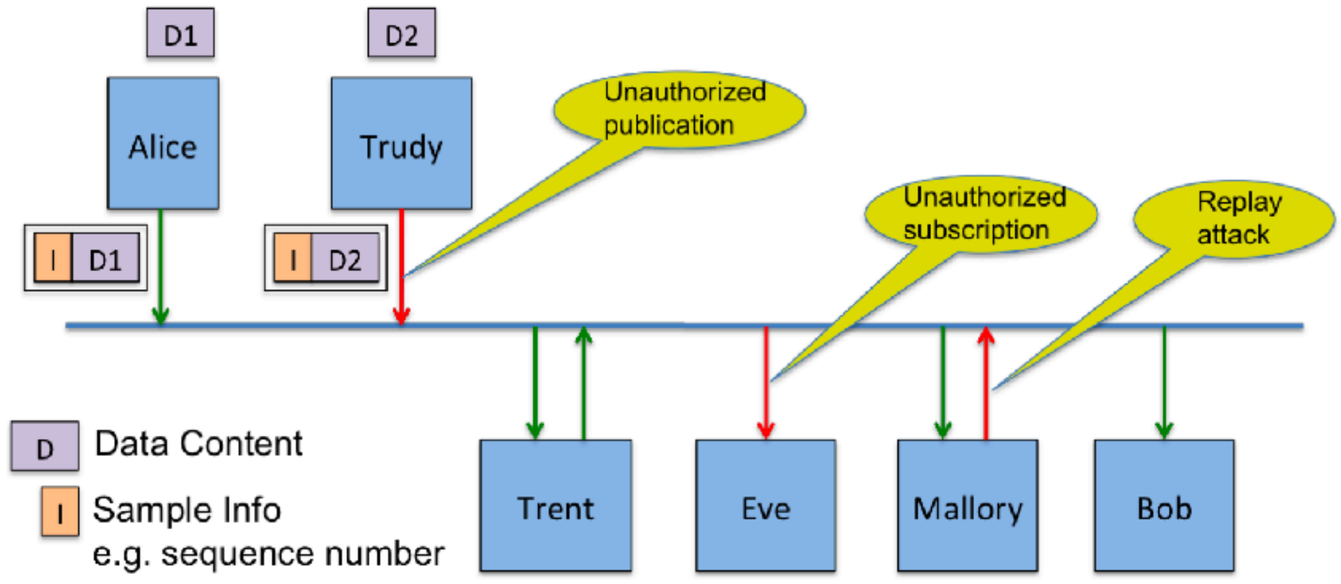
[See DDS Security Specification v1.0 p.9]

Unauthorised Access by Services

Infrastructure services, such as the DDS Durability Service or DDS-middle-boxes need to be able to receive messages, verify their integrity, store them, and send them to other participants on behalf of the original application

These services can be trusted not to be malicious; however, often it is not desirable to let them understand the contents of the data.

They are **allowed to store and forward** the **data**, but **not to see inside** the **data**

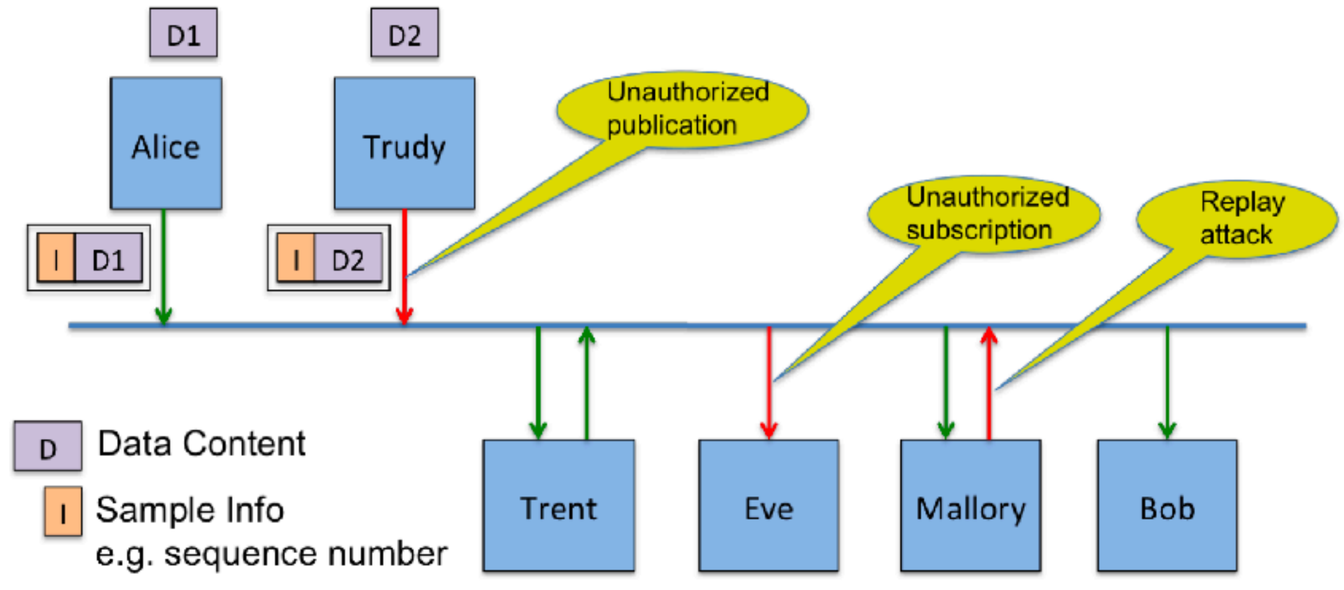


[See DDS Security Specification v1.0 p.9]

Unauthorised Access by Services

The DDS Security Standard addresses this problem by supporting independent configuration for protocol and user data encryption

This way, infrastructure services only need to be trusted with the secrets required to process the protocol data, but not with the secrets required to read the user data

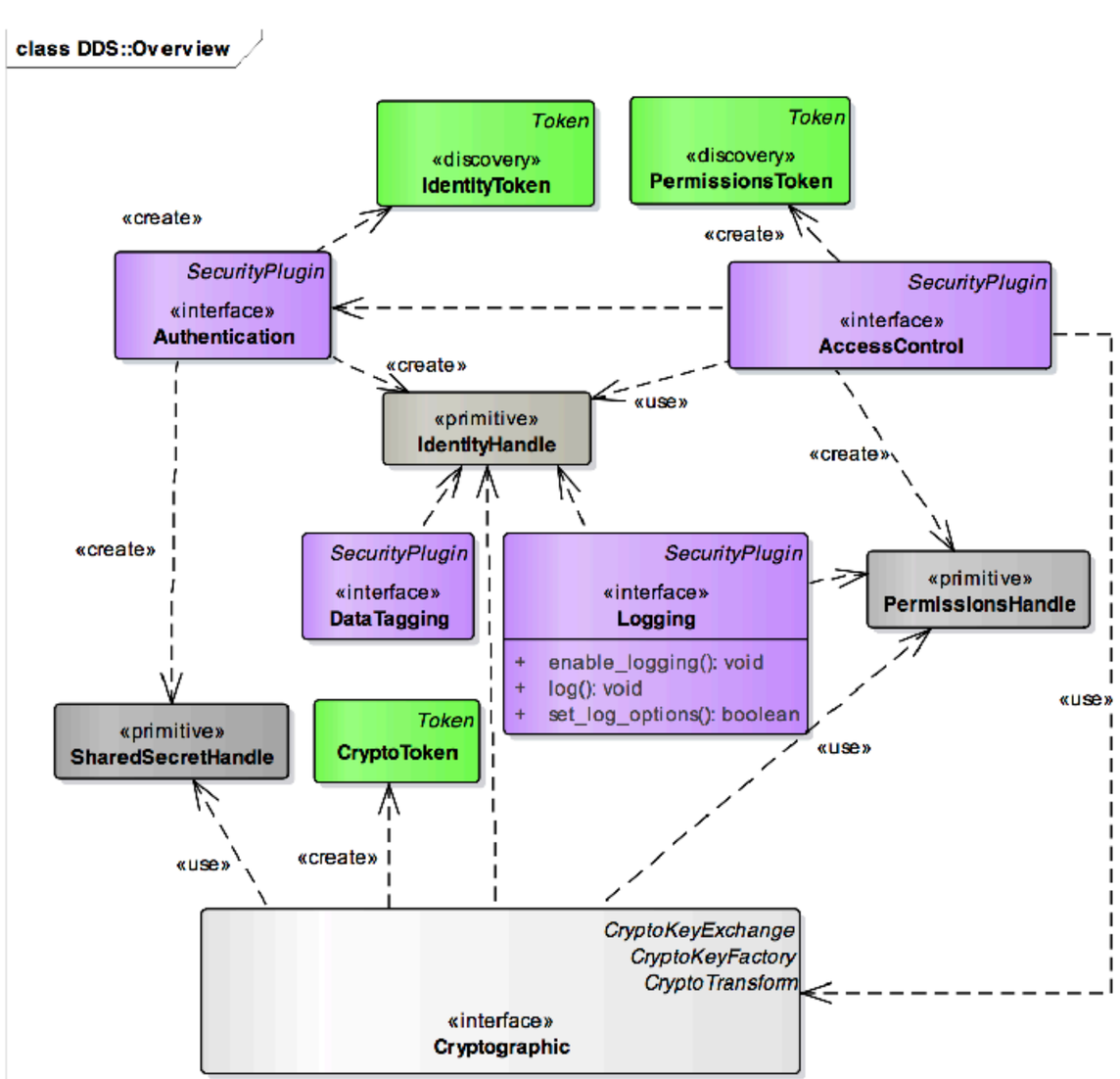


[See DDS Security Specification v1.0 p.9]

DDS Security Architecture

Plug-in Architecture

The DDS Security standard has a modular and **plug-in architecture** that allows for pluggable **Authentication**, **Access Control**, **Logging**, **Cryptography** and **Data Tagging**

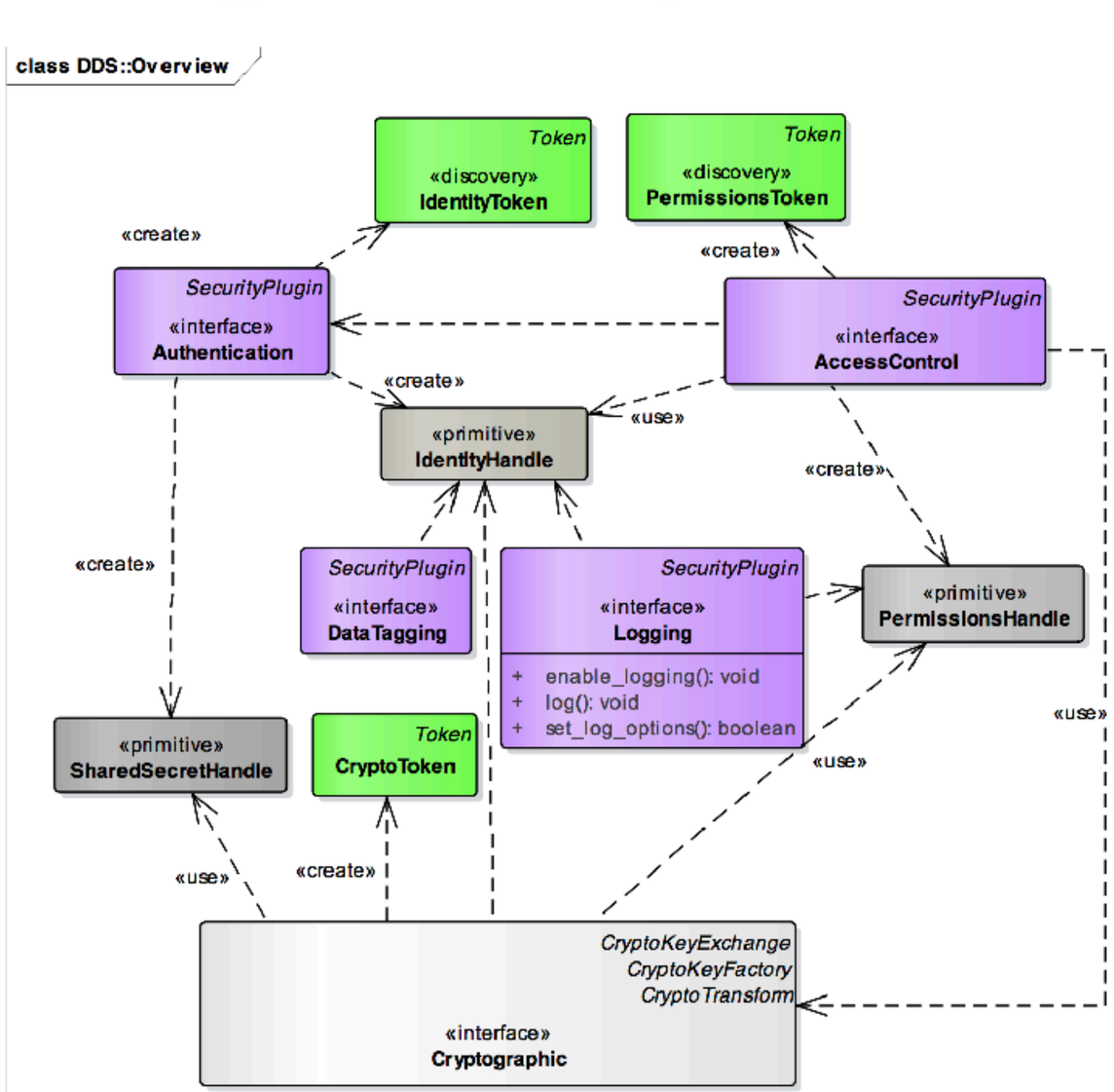


[*See DDS Security Specification v1.0 p.47*]

Authentication Control Plugin

Authenticate the principal that is joining a DDS Domain

Support mutual authentication between participants and establish a shared secret

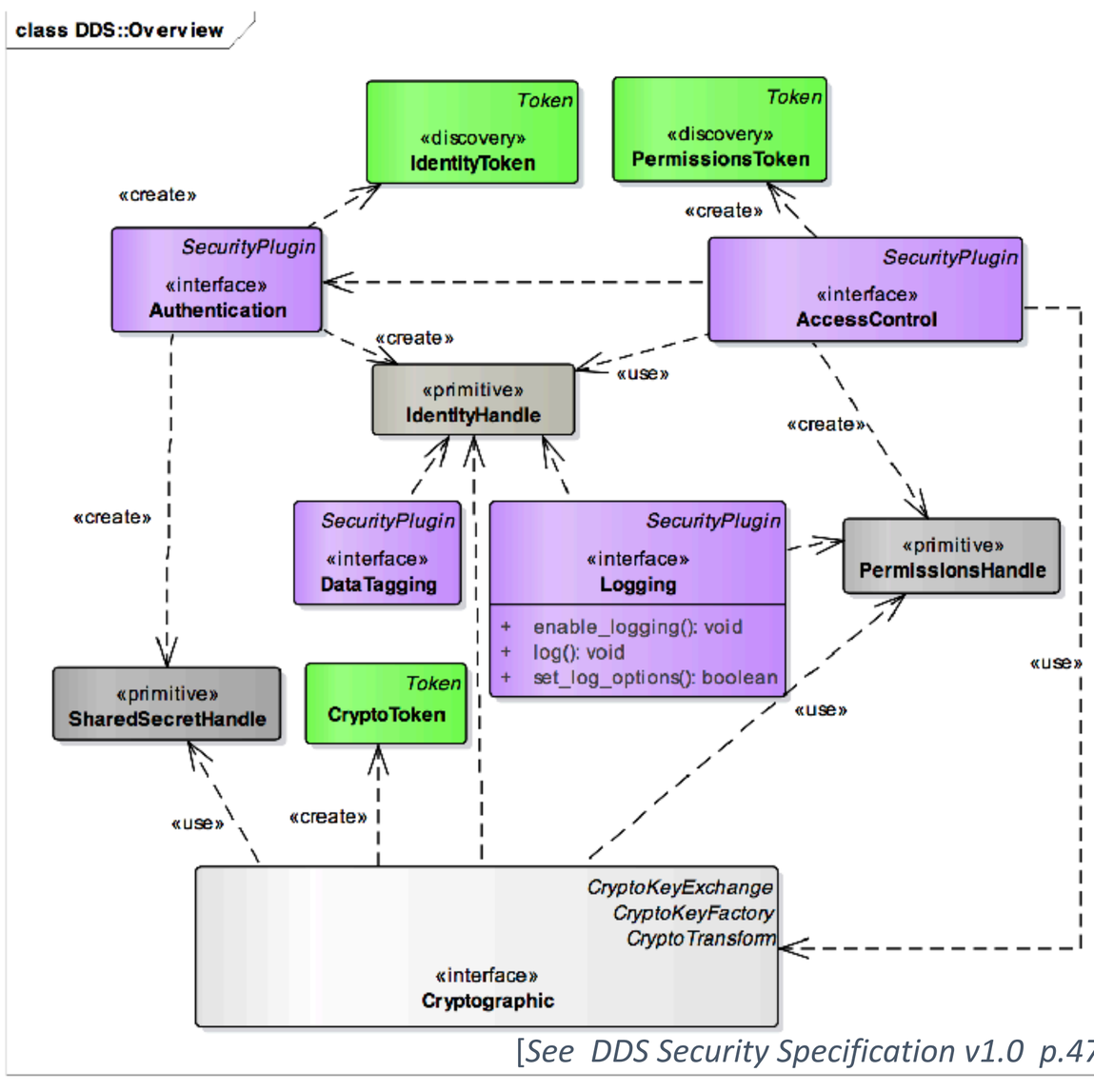


[*See DDS Security Specification v1.0 p.47*]

Authentication Control Plugin

The DDS Security **discovery is enhanced with an authentication protocol**

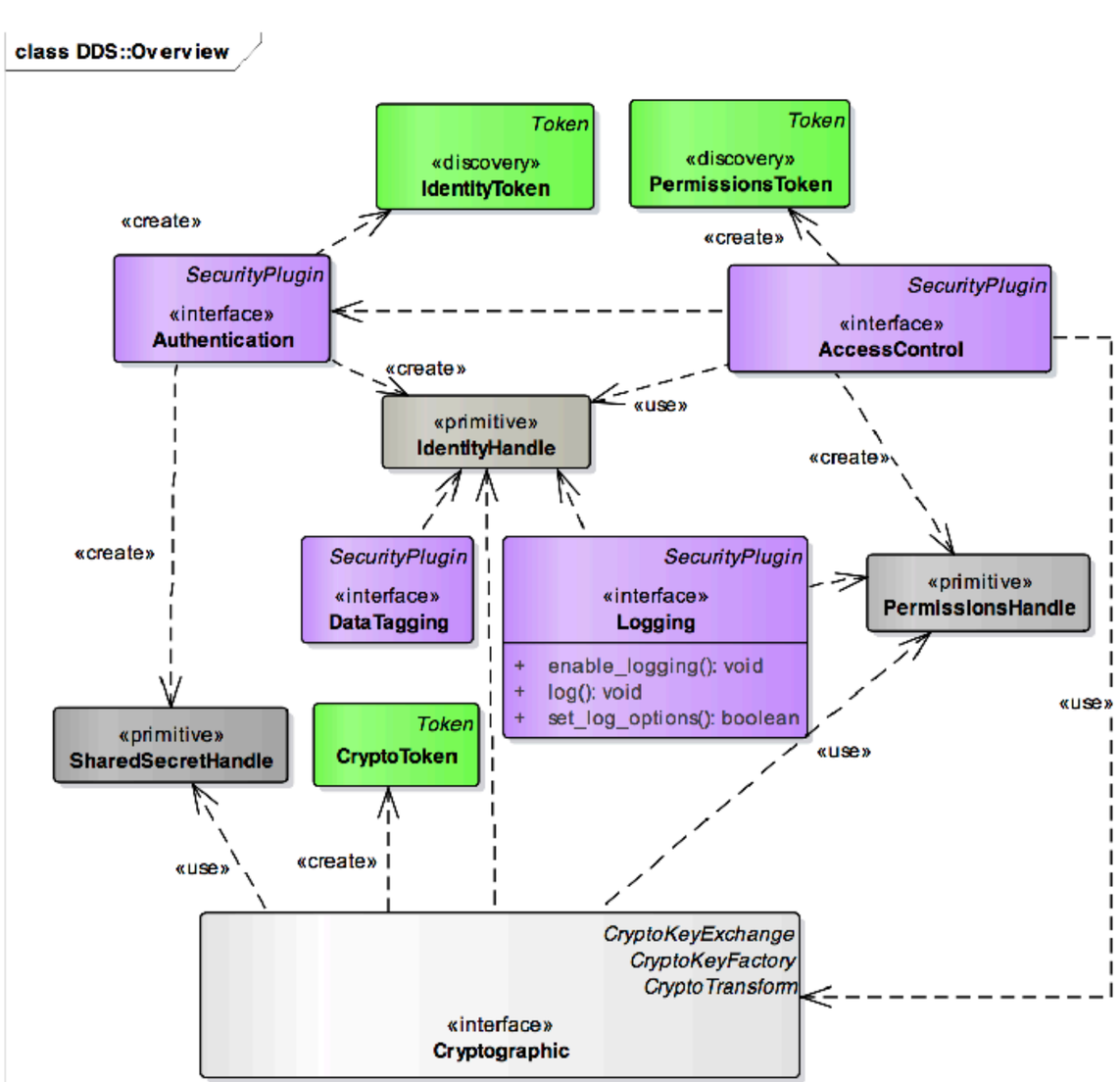
For protected DDS Domains a DomainParticipant that enables the authentication plugin will only communicate with another Domain Participant that has the authentication plugin enabled



[*See DDS Security Specification v1.0 p.47*]

Access Control Plugin

Decide whether a principal is allowed to perform a protected operation.



[*See DDS Security Specification v1.0 p.47*]

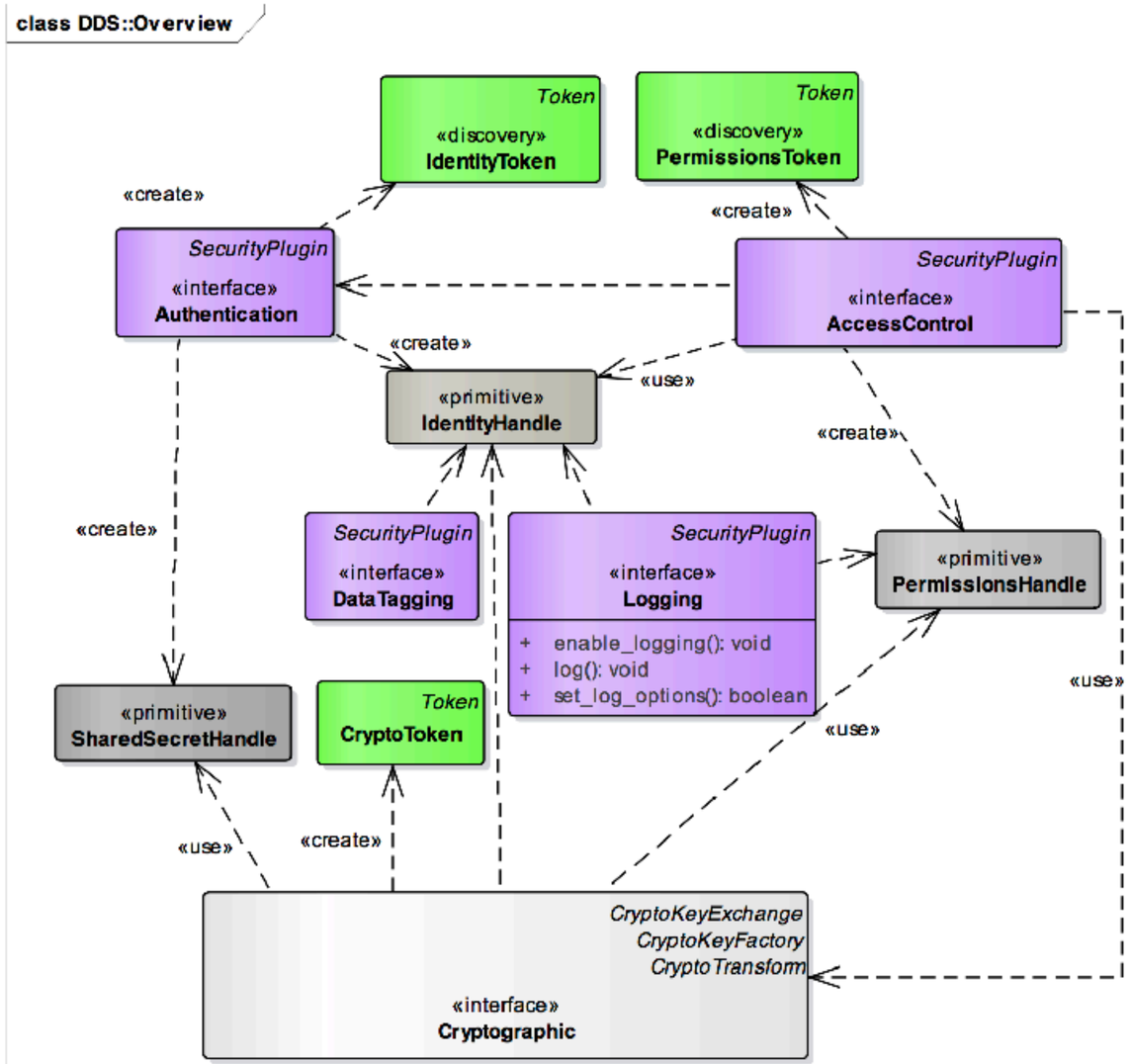
Access Control Plugin

Once a DomainParticipant is authenticated, its permissions need to be validated and enforced

Access rights are often described using an access control matrix where the rows are subjects (i.e., users), the columns are objects (i.e., resources), and a cell defines the access rights that a given subject has over a resource

Typical implementations provide either a **column-centric view**, i.e., **access control lists**, or a **row-centric view**, i.e., a set of **capabilities** stored with each subject

The DDS Access Control plugin supports both approaches



[*See DDS Security Specification v1.0 p.47*]

Cryptography Plugin

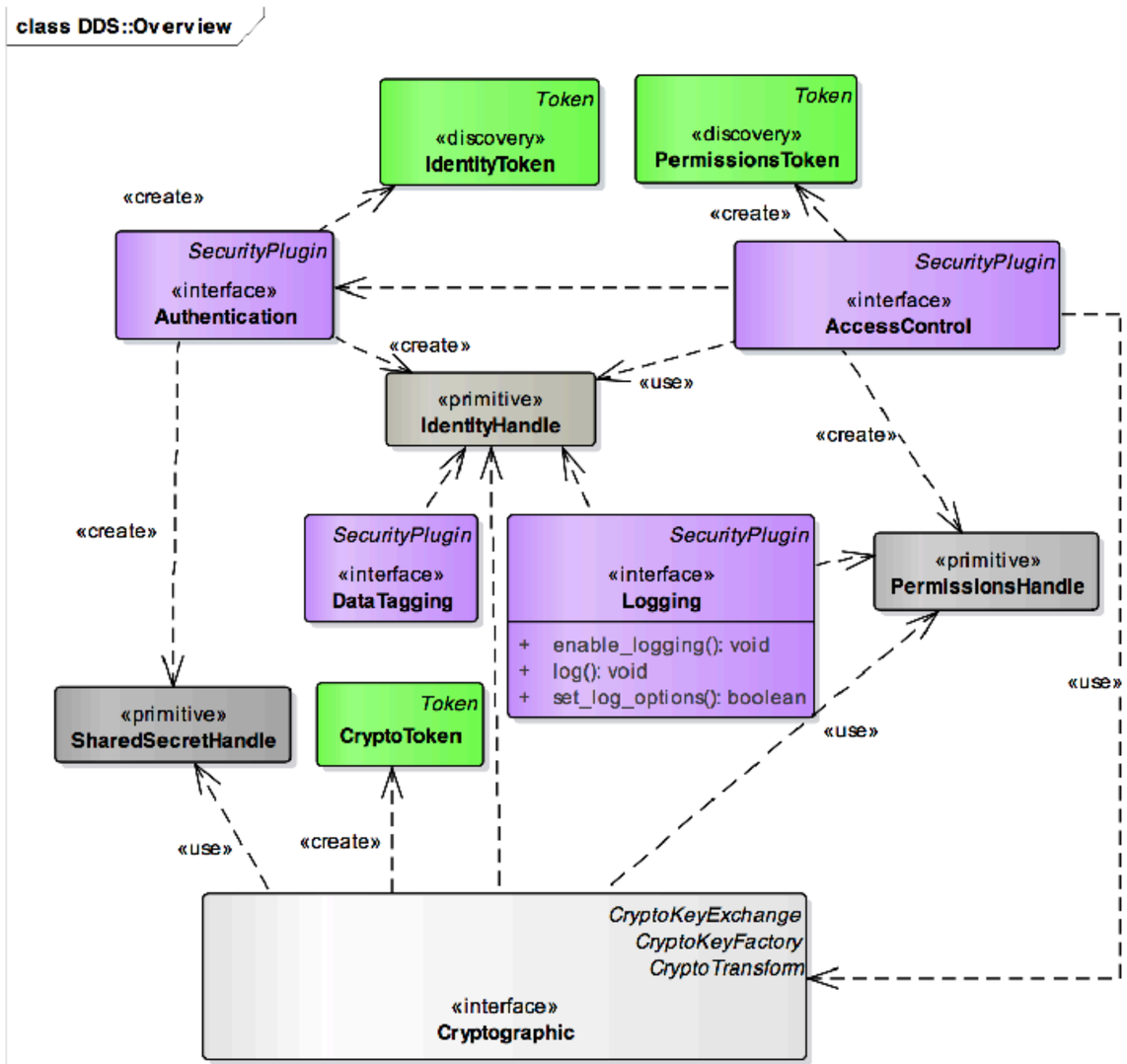
Generate keys.

Perform Key Exchange.

Perform the encryption and decryption operations.

Compute digests, compute and verify Message Authentication Codes.

Sign and verify signatures of messages.



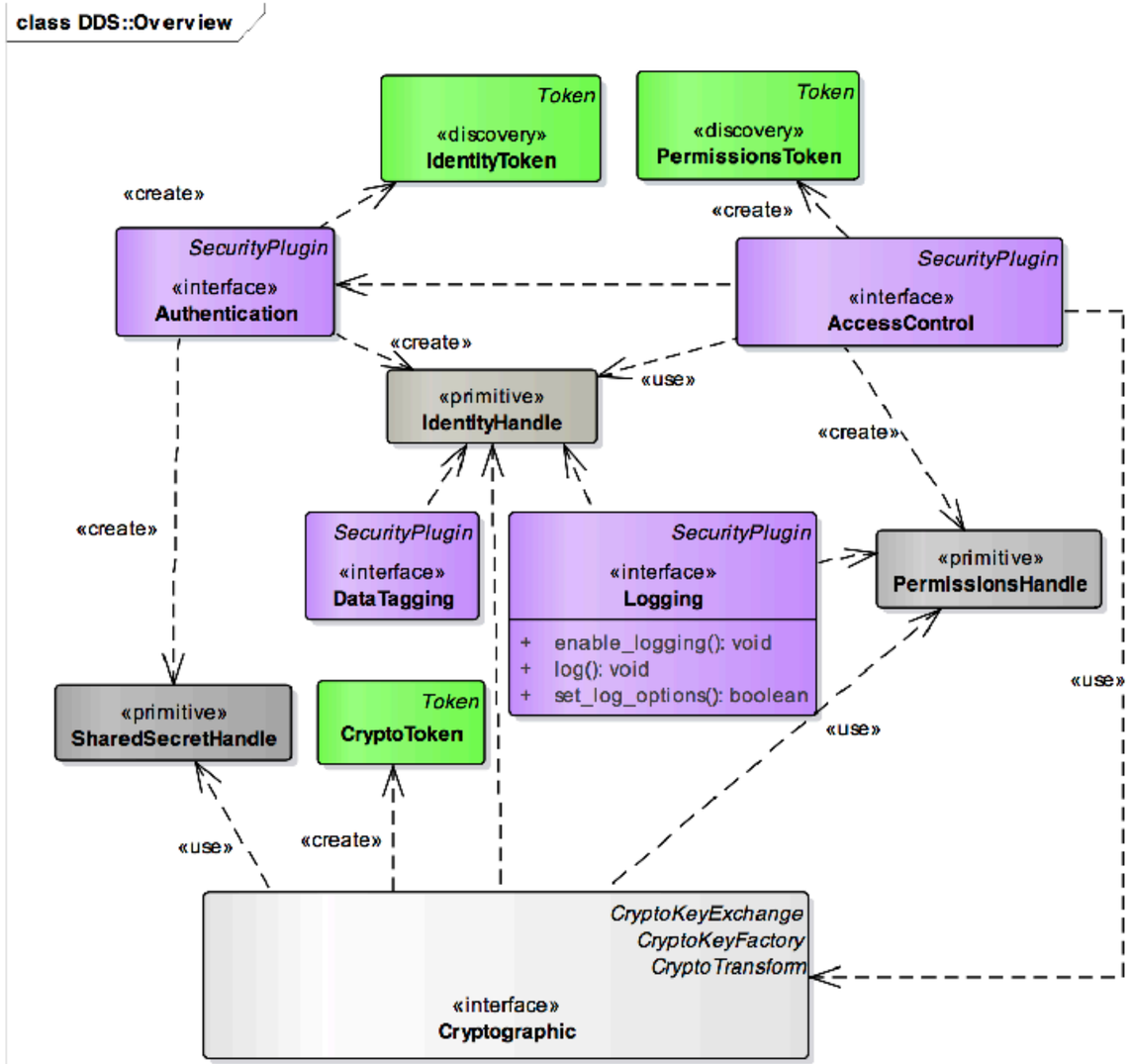
[*See DDS Security Specification v1.0 p.47*]

Cryptography Plugin

The Cryptographic plugin defines the operations necessary to support **encryption**, **digest**, **message authentication codes**, and **key exchange** for DDS DomainParticipants, DataWriters andDDSDataReaders

Users of DDS may have specific cryptographic libraries they use for encryption, as well as, specific requirements regarding the algorithms for digests, message authentication, and signing.

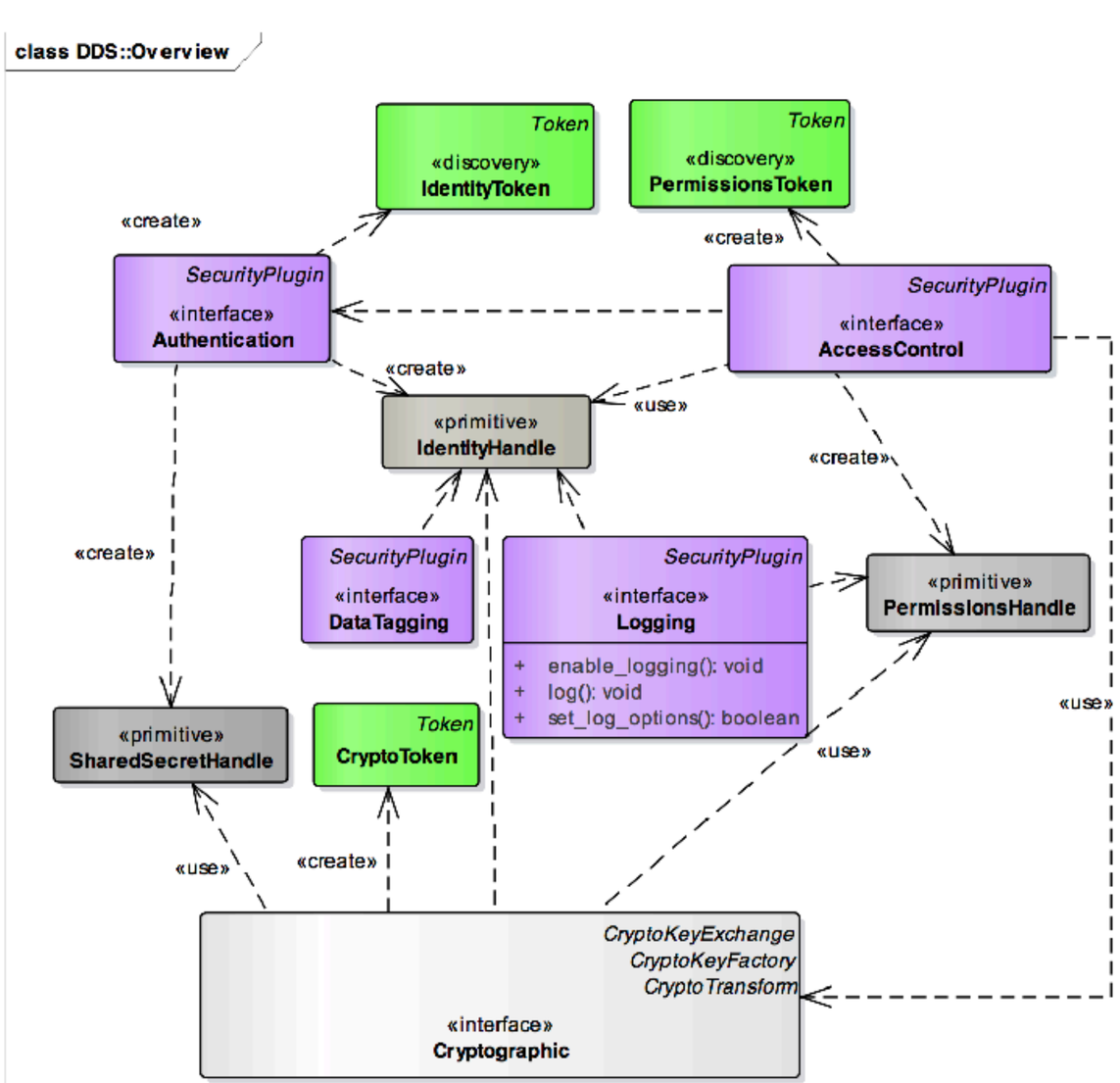
In addition, applications may require having only some of those functions performed, or performed only for certain DDS Topics and not for others.



[*See DDS Security Specification v1.0 p.47*]

Logging Plugin

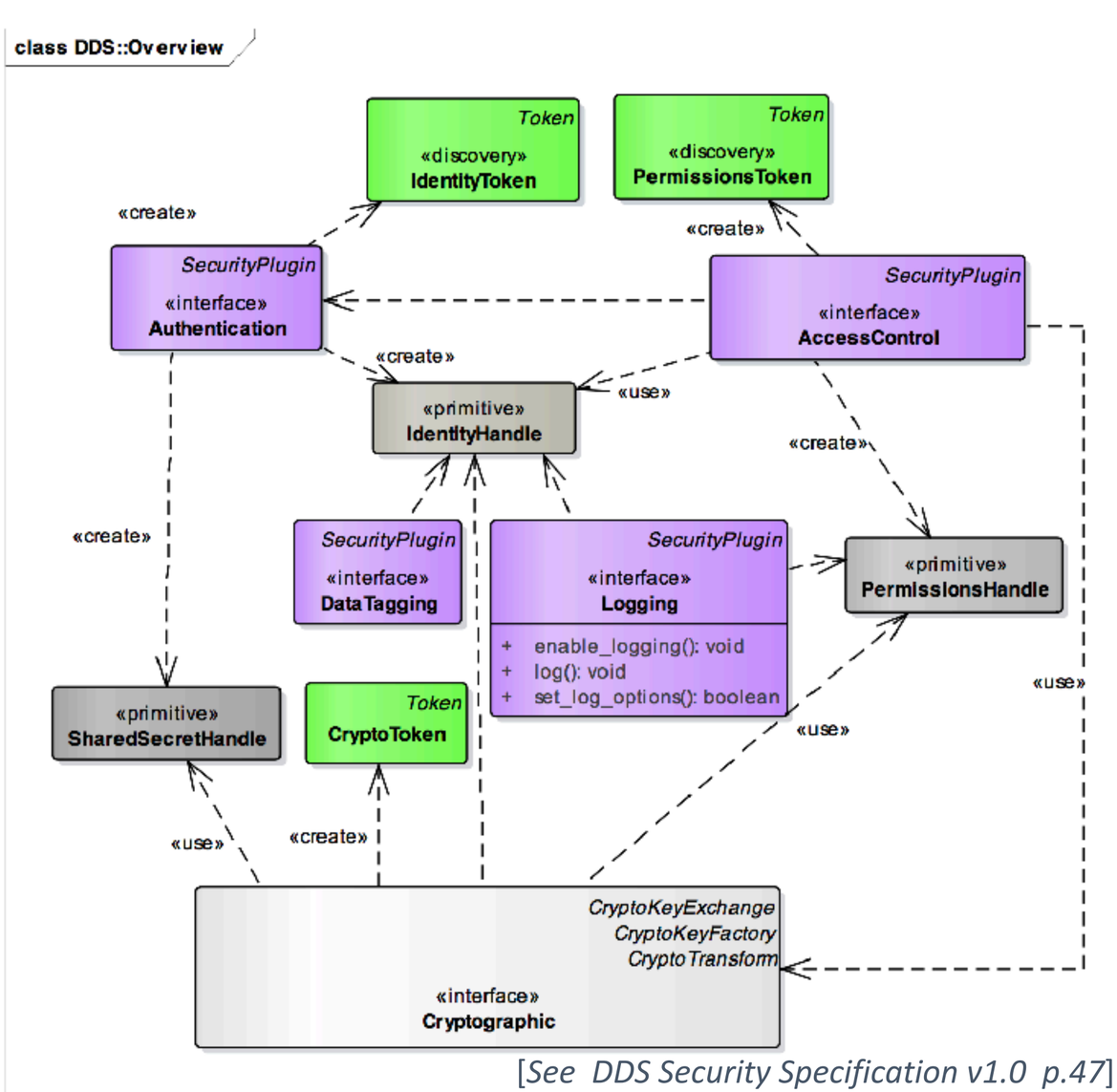
Log all security relevant events



[*See DDS Security Specification v1.0 p.47*]

Data Tagging Plugin

Add a data tag for each data sample



[*See DDS Security Specification v1.0 p.47*]

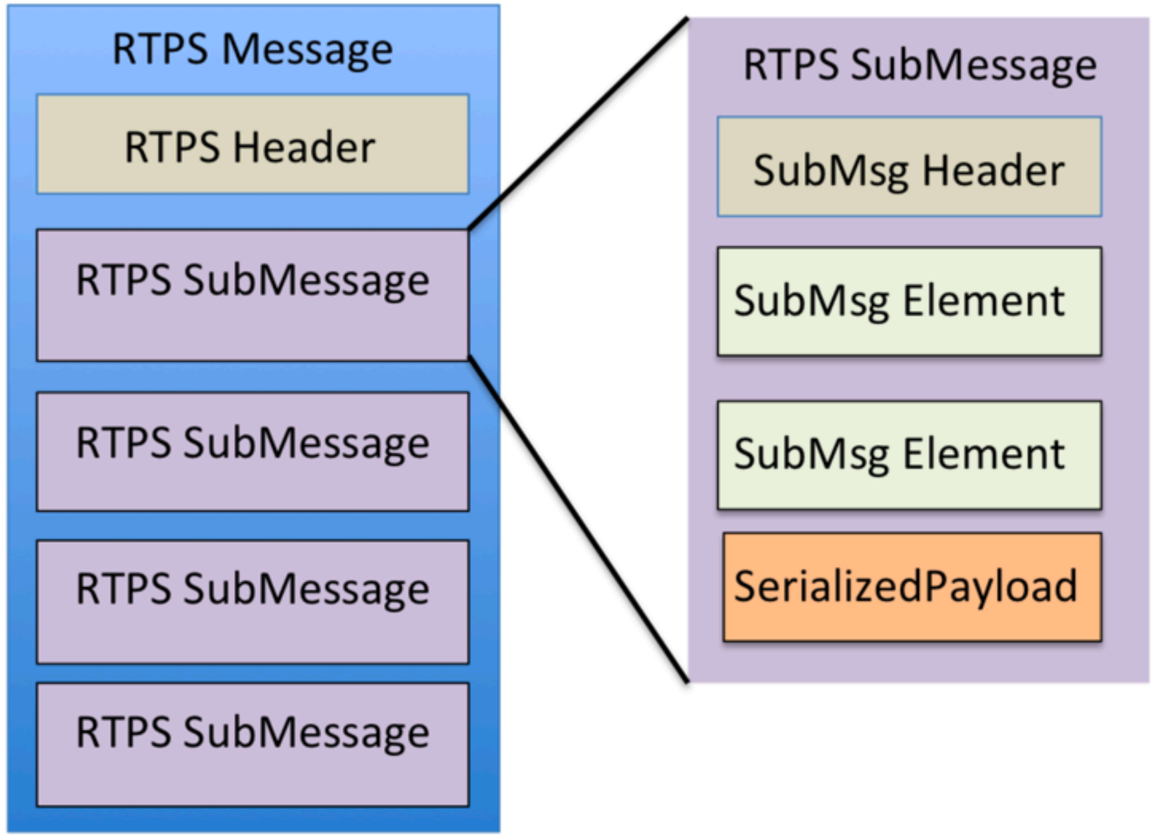
Securing DDSI-RTPS Messages

DDSI-RTPS Message

An DDSI-RTPS Message is composed of a leading **Header** followed by a variable number of **Submessages**.

Each **Submessage** is composed of a **SubmessageHeader** followed by a variable number of **SubmessagElements**.

There are various kinds of SubmessageElements to communicate things like sequence numbers, unique identifiers for DataReader and DataWriter entities, SerializedKeys or KeyHash of the application data, source timestamps, QoS, etc.



[See DDS Security Specification v1.0 p.18]

Default Plugins

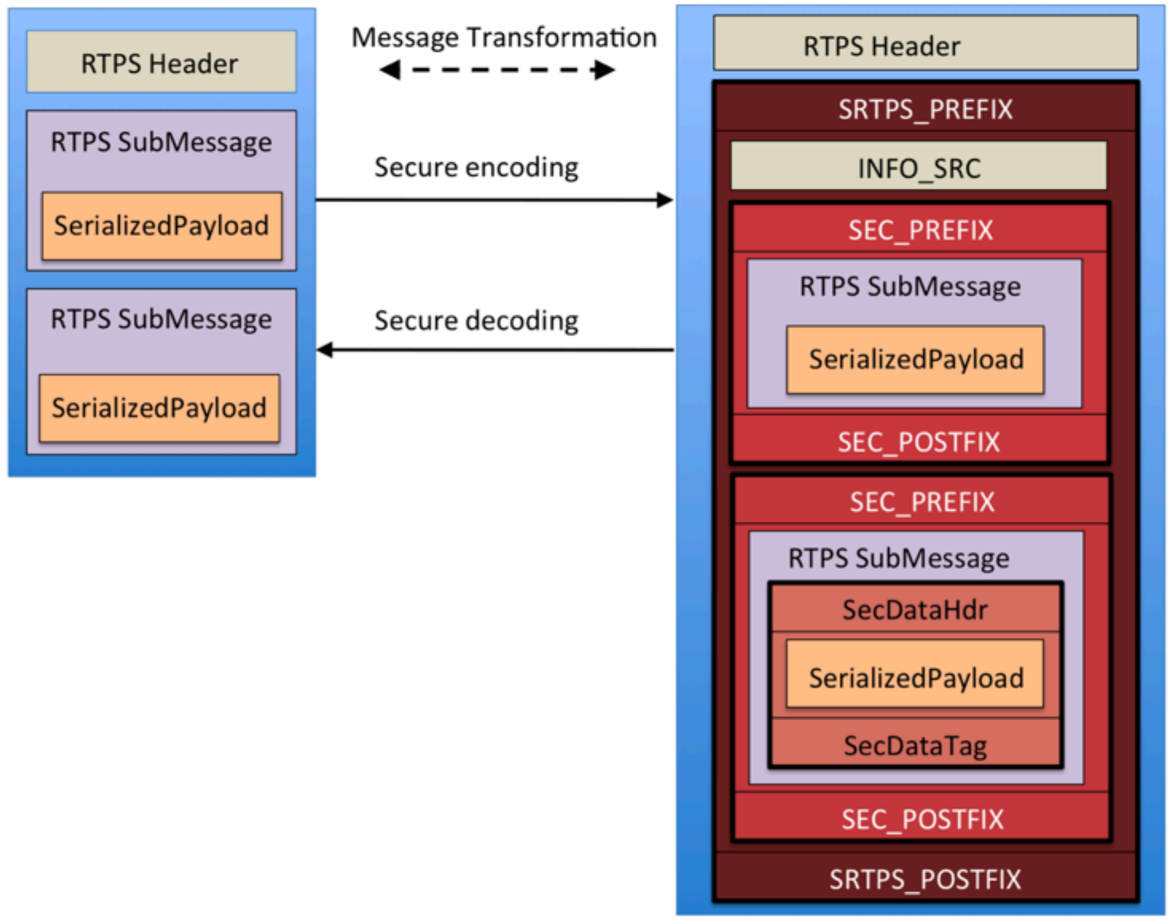
	Name	Description
Authentication	DDS:Auth:PKI-DH	*Uses PKI with a pre- configured shared Certificate Authority. RSA or DSA and Diffie- Hellman for authentication and key exchange.*
Access Control	DDS:Access:Permissions	*Permissions document signed by shared Certificate Authority*
Cryptography	DDS:Crypto:AES-GCM-GMAC	*AES-GCM (AES using Galois Counter Mode) for encryption. AES-GMAC for message authentication*
Data Tagging	DDS:Tagging:DDS_Discovery	*Send Tags via endpoint discovery*
Logging	DDS:Logging:DDS_LogTopic	*Logs security events to a dedicated DDS Log Topic*

Message Transformation

DDS Security is implemented as a "bump" in the DDSI-RTPS Protocol stack

In other terms, the plugins transform regular DDSI-RTPS messages into secure DDSI-RTPS messages before sending on the wire.

Likewise secure DDSI-RTPS messages are transformed into regular messages before being passed to the protocol state machine

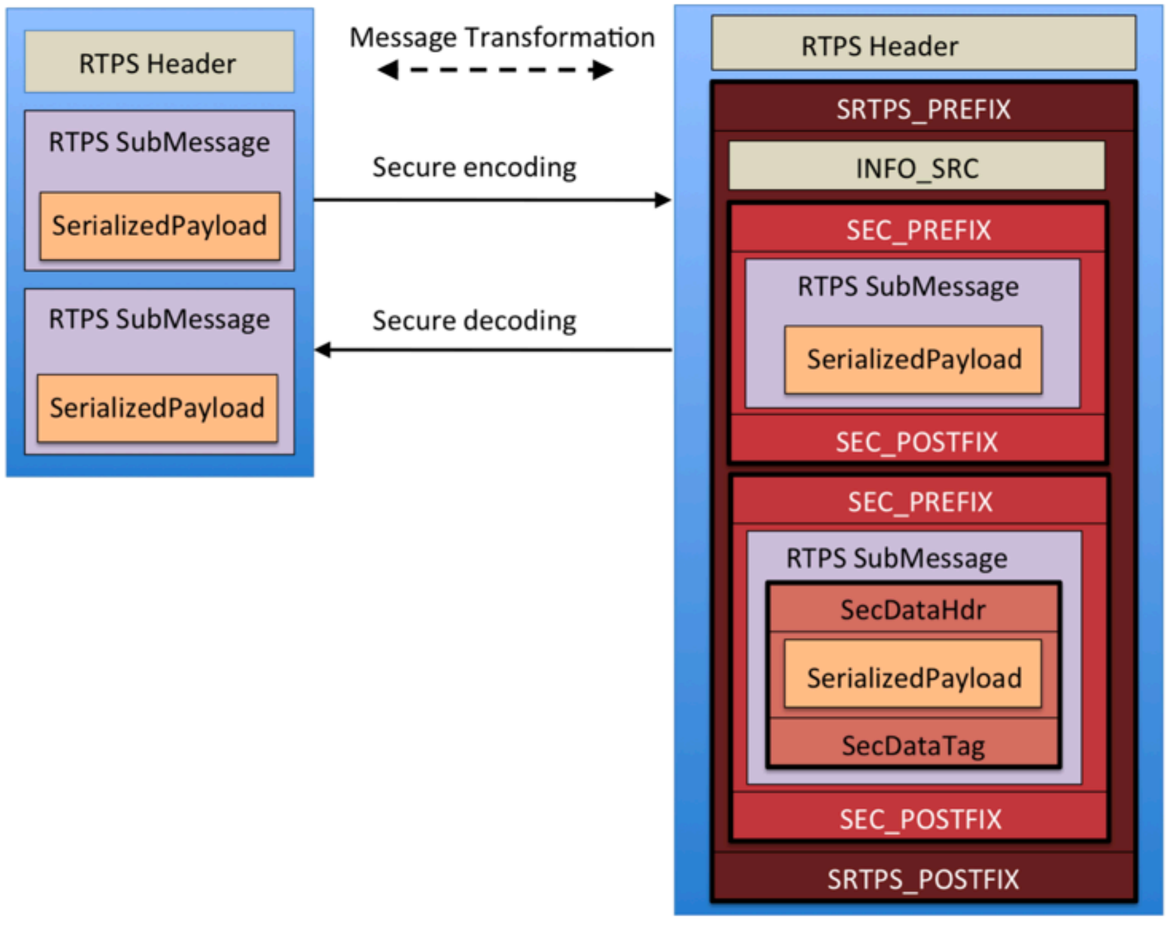


[See DDS Security Specification v1.0 p.25]

Message Transformation

Prefix and Postfix "markers" are defined for both the DDSI-RTPS message as well as its sub-messages

These sub-messages provide the required information to decrypt and authenticate the sources



[See DDS Security Specification v1.0 p.25]

Interoperability

Interoperability

The DDS Security Standard ensures that compliant implementation can securely interoperate

Additionally a DDS implementation compliant with the DDS Security standard is capable of interoperating with a non-secure DDS implementation over non-secured Topics

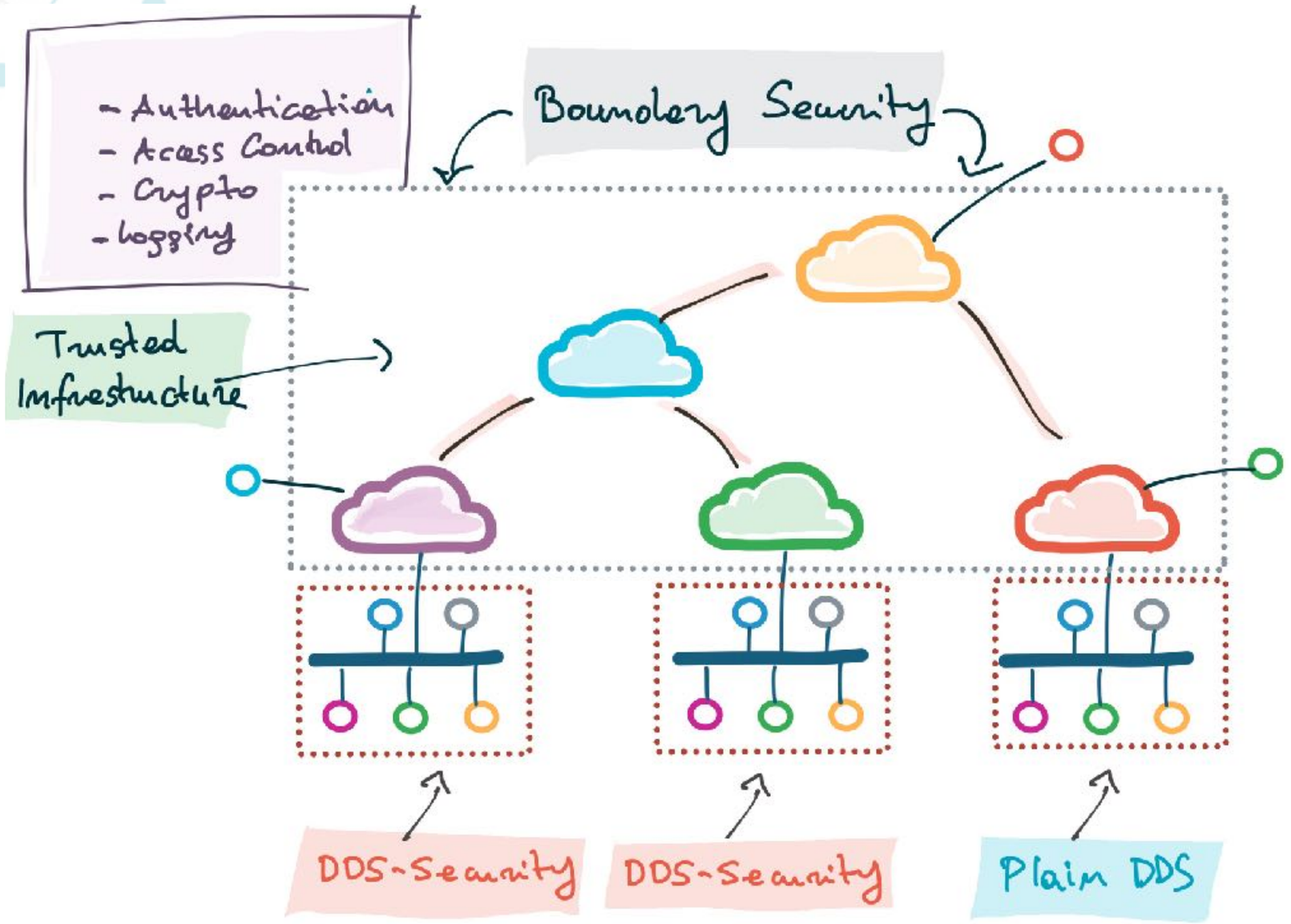
Scaling Security

Securing at Scale

To build Scalable and Secure DDS systems a combination of regular DDS Security and Boundary Security will be necessary

Introducing boundaries allow to group entire subsystems behind a single (or a few) subjects, thus making it easier to scale

The DDS Security mechanism are suitable for Boundary Securing implementation



Concluding Remarks

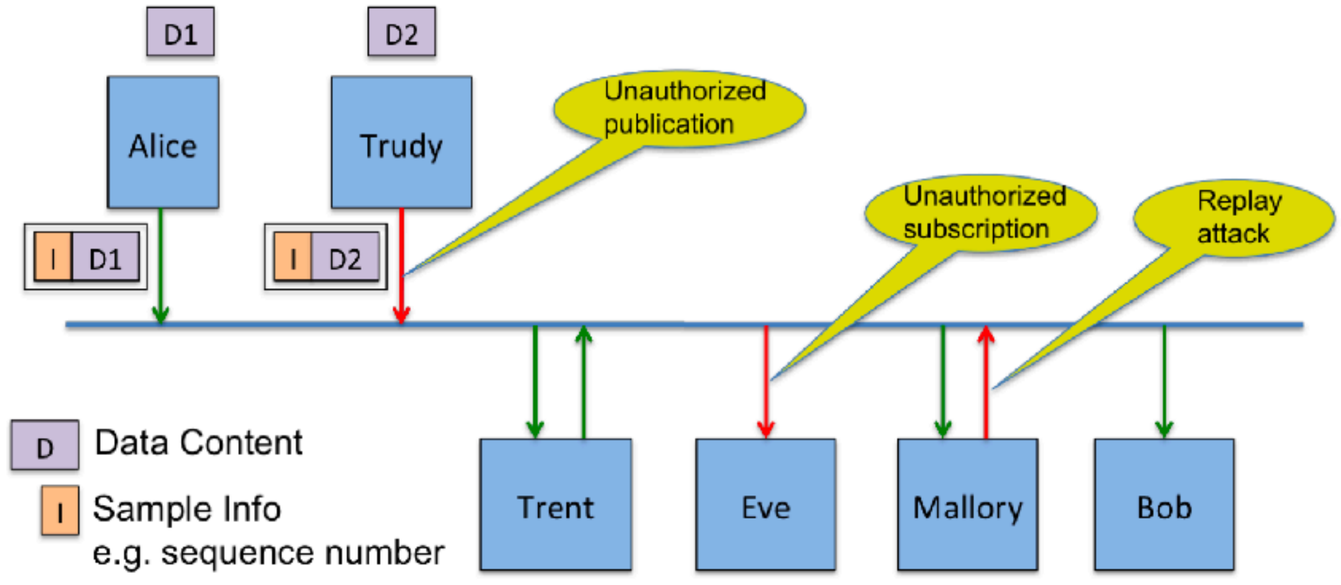
Summing Up

The DDS Security Standard provides protection against:

- **Unauthorised Subscription**
- **Unauthorised Publication**
- **Tampering and Replay**
- **Unauthorised data access**

The specific implementation supports multicast and real-time communication

The DDS-Security mechanism can be used for implementing sub-system as well as boundary security



[See DDS Security Specification v1.0 p.9]