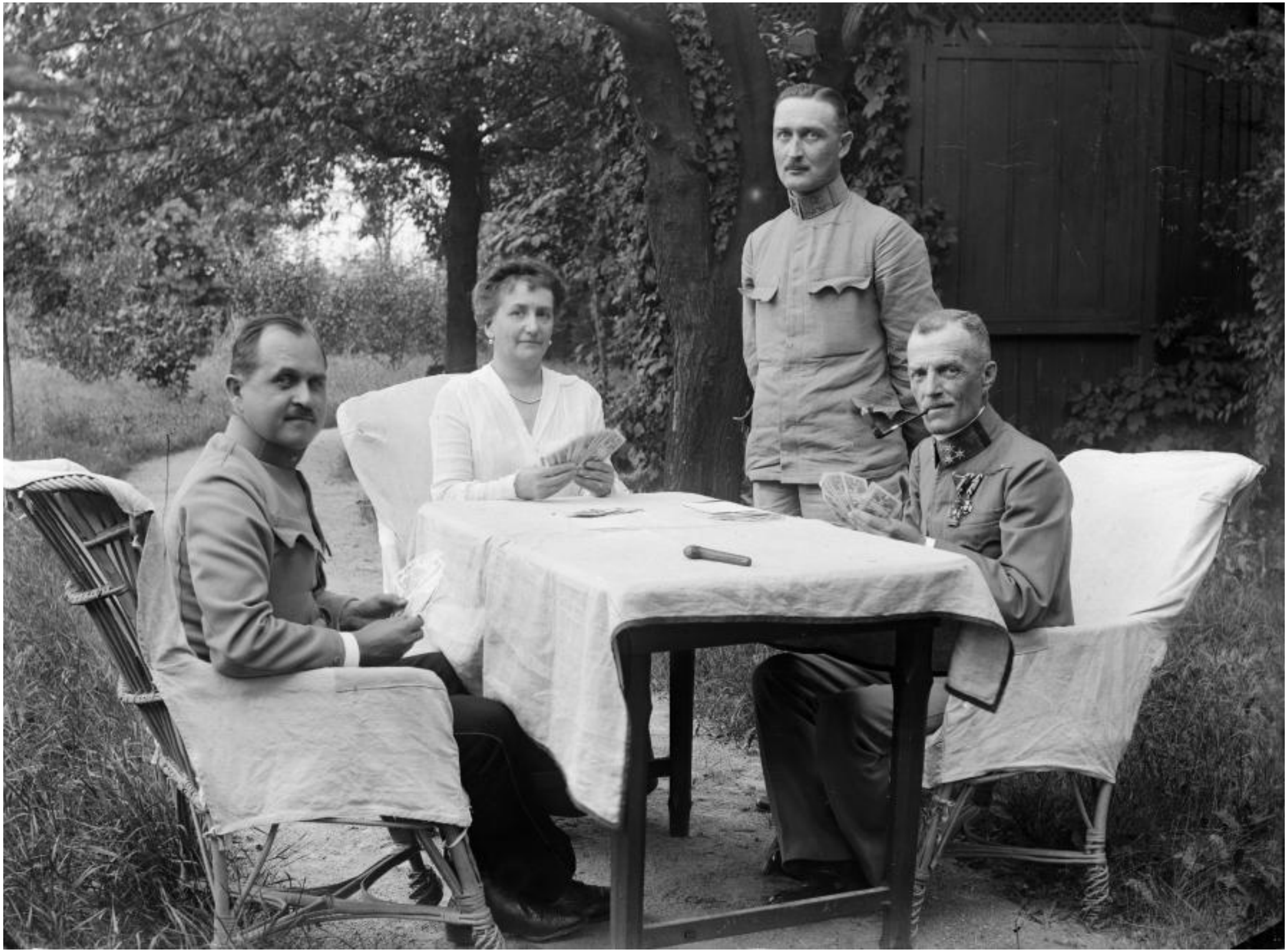
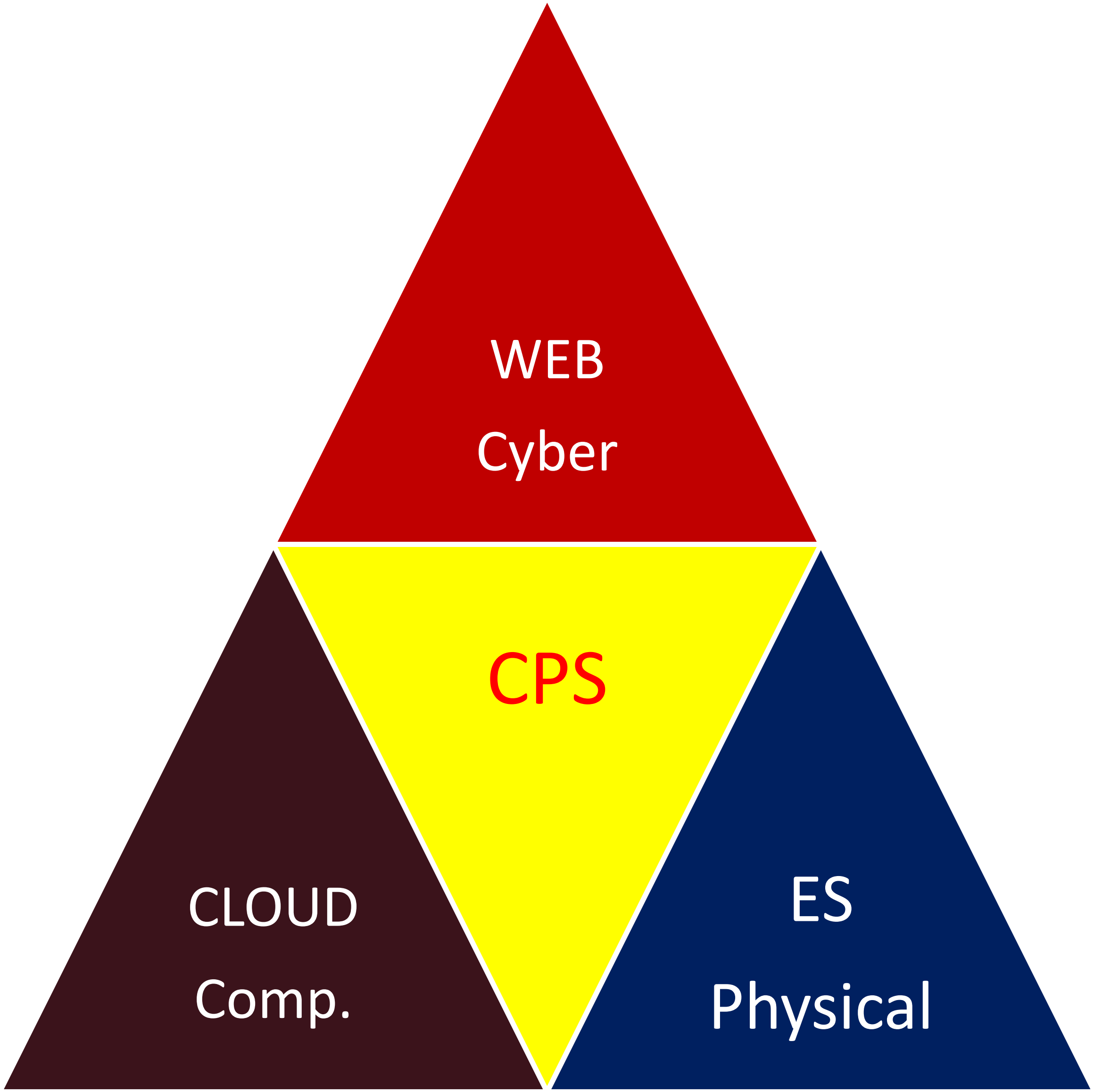
Distributed Technologies for Cyber-Physical Systems

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group

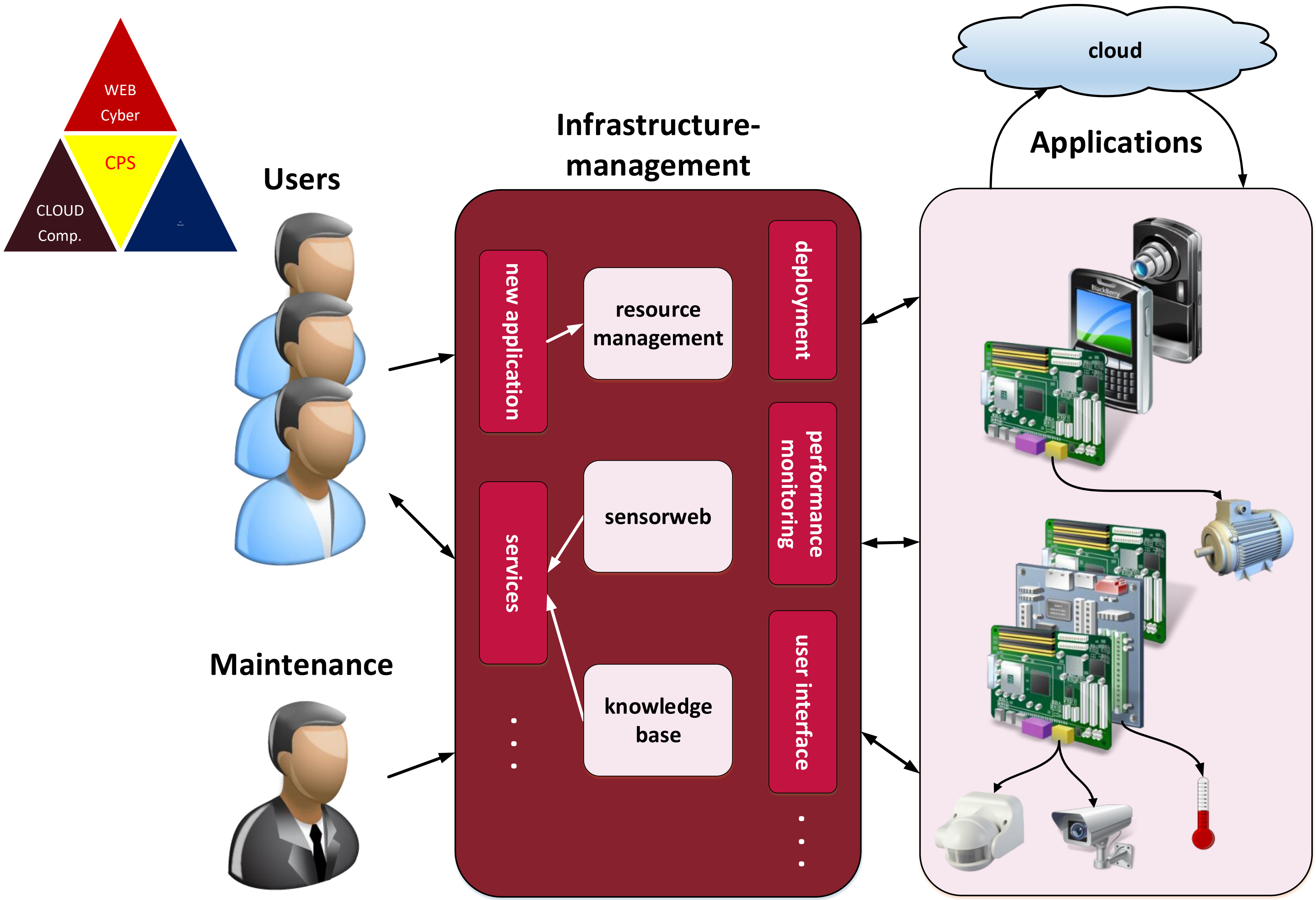
A distributed, collaborative system



The design triangle



Dynamic composition of cyber-physical systems

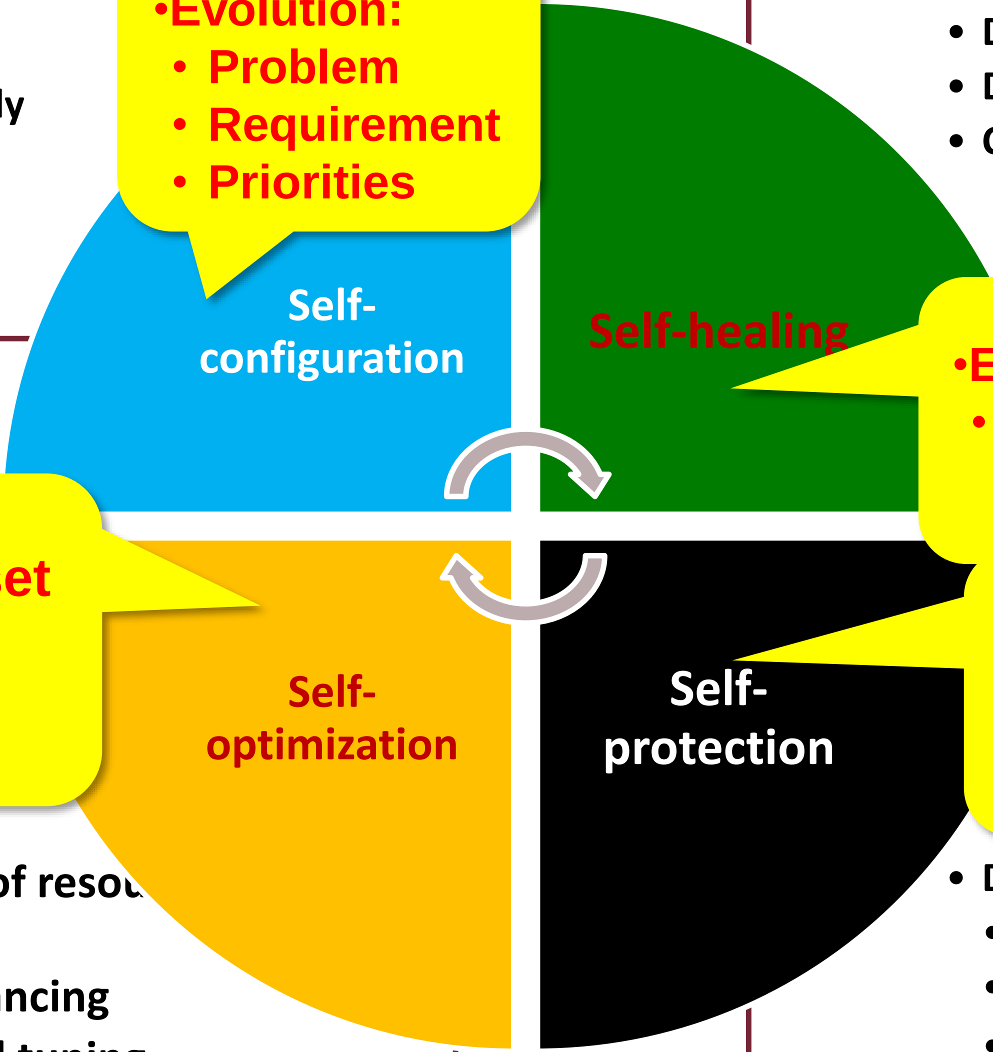


Self-* properties – dynamic challenges

- Reactiveness
- Adaptation to the dynamically changing enviroment

- Mobility
- Evolution:
  - Problem
  - Requirement
  - Priorities

- FT
- Detection
- Diagnostics
- Compensation



- Evolution:
  - Fault/failure modes

- Evolution:
  - Resource set
  - Capability
  - Capacity
  - Workload

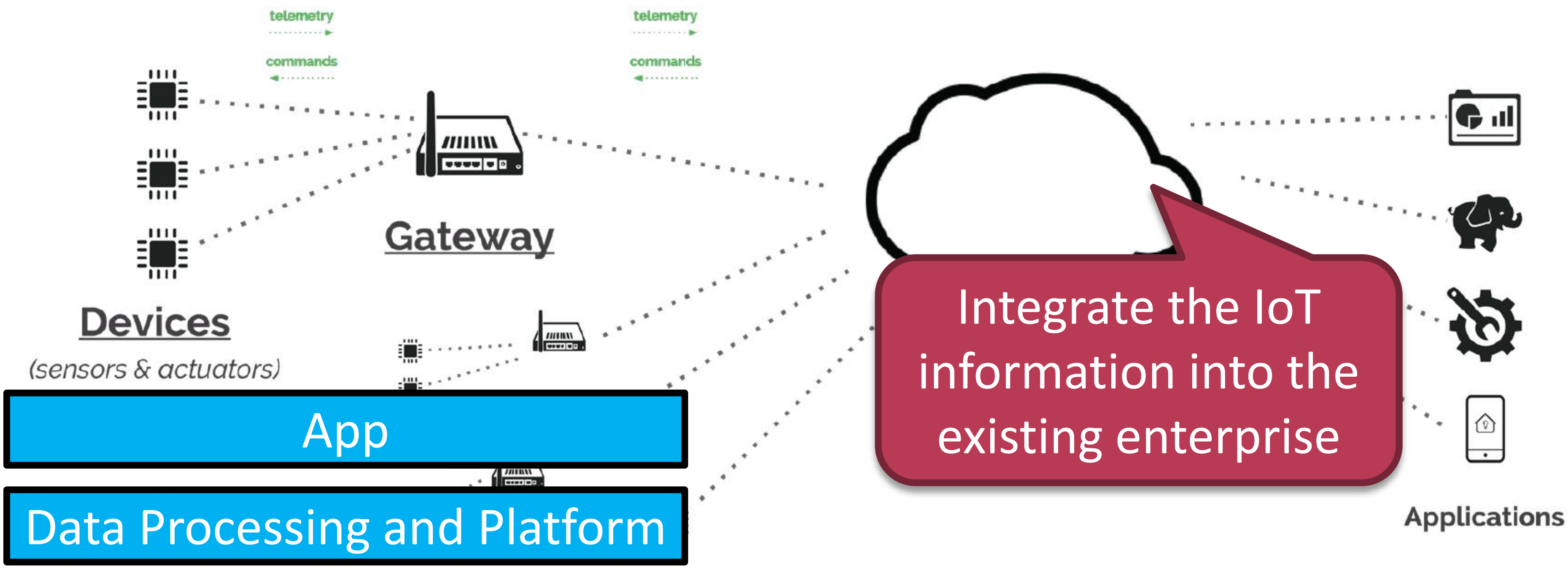
- Evolution:
  - Requirements
  - Threads

- Effectivity of reso use
  - Load balancing
  - Workload tuning

- Data protection
  - Detection ,
  - Identification ,
  - Reaction

IOT FUNCTIONS

IoT Architectures



Integrate the IoT information into the existing enterprise

App

Data Processing and Platform

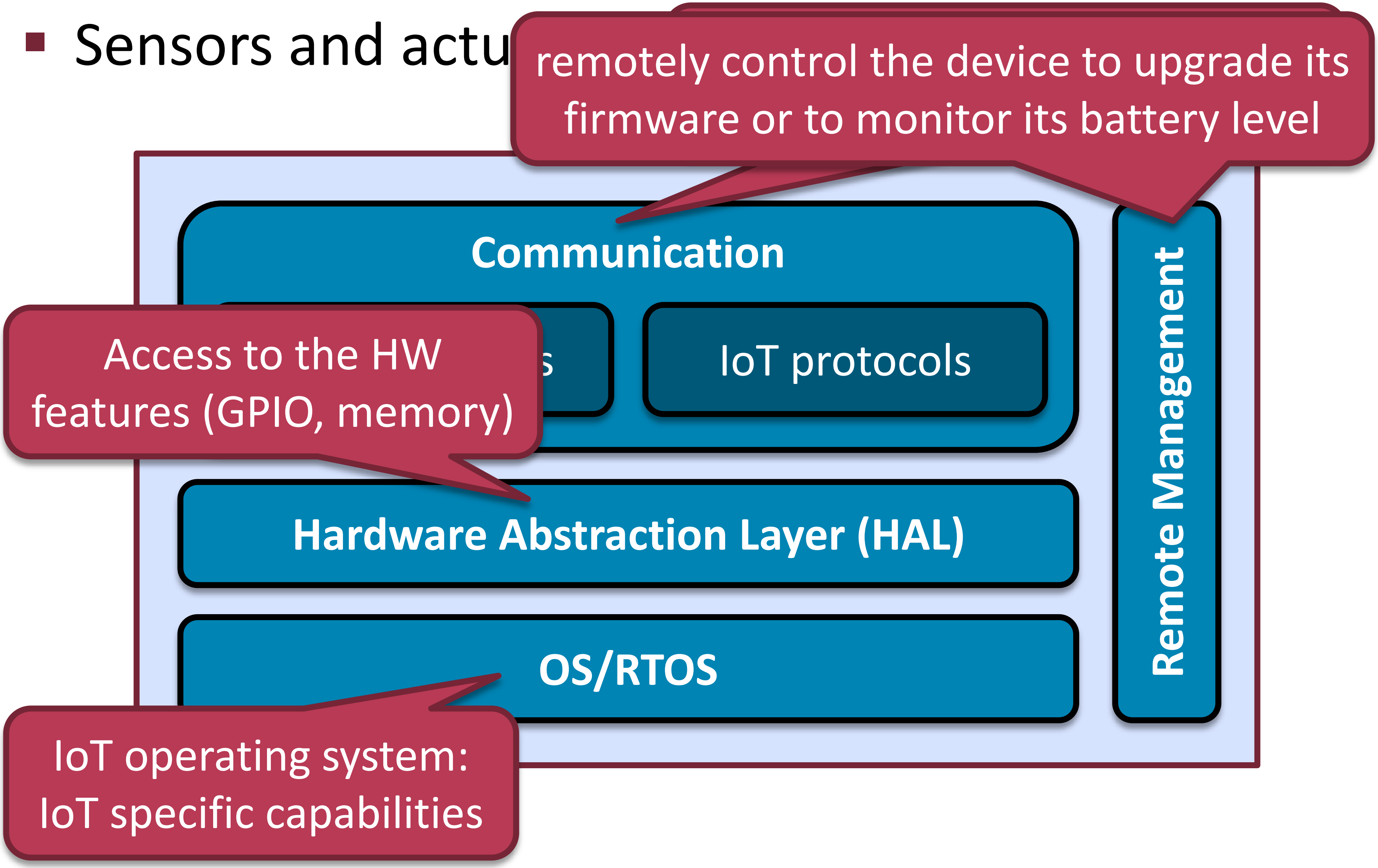
Edge

Thing/Device

Source: http://iot.eclipse.org/

Software stack for Constrained Devices

- Sensors and actu



Communication

s

IoT protocols

Hardware Abstraction Layer (HAL)

OS/RTOS

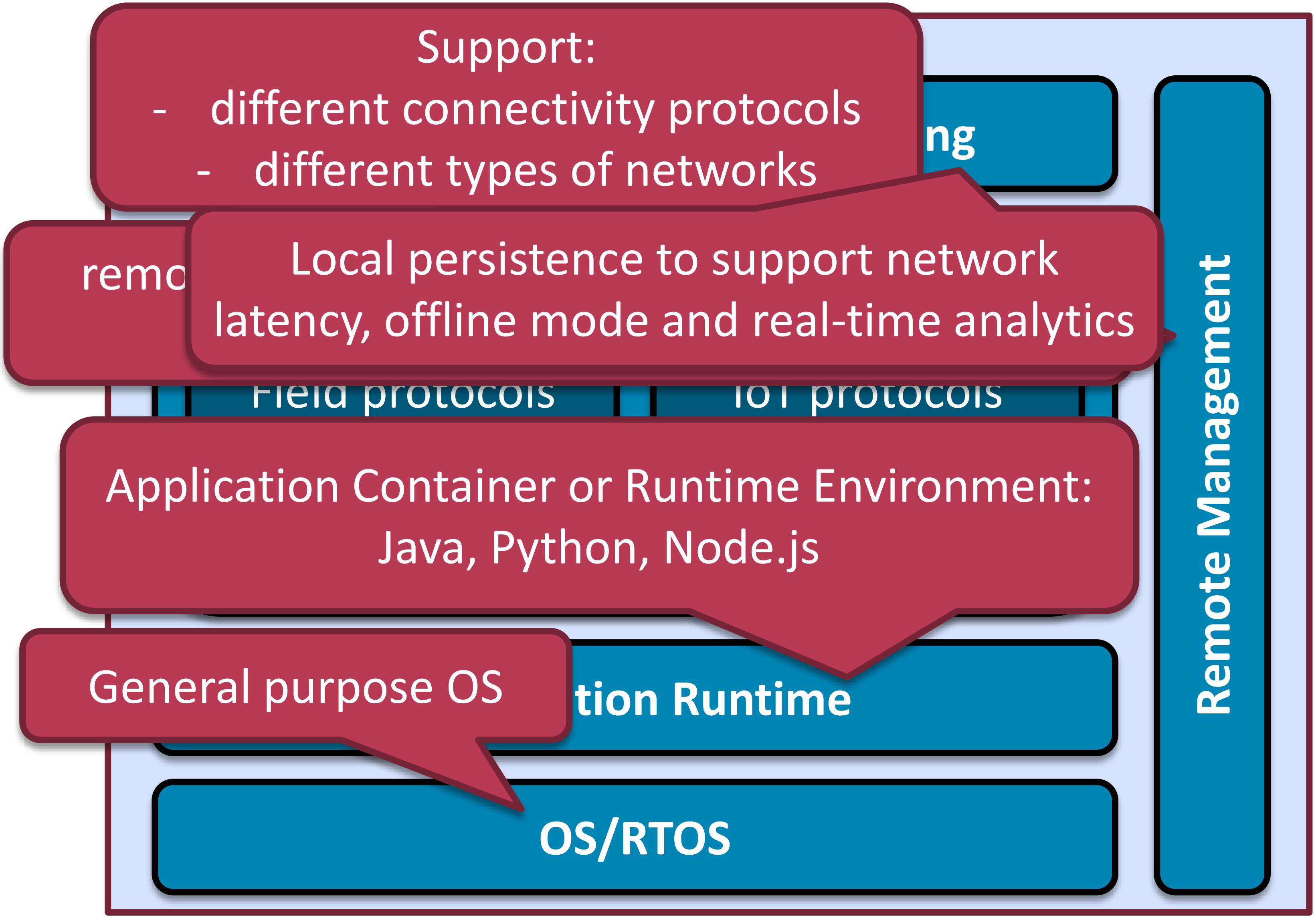
Remote Management

remotely control the device to upgrade its firmware or to monitor its battery level

Access to the HW features (GPIO, memory)

IoT operating system: IoT specific capabilities

Software stack for Gateways



Support:

- different connectivity protocols
- different types of networks

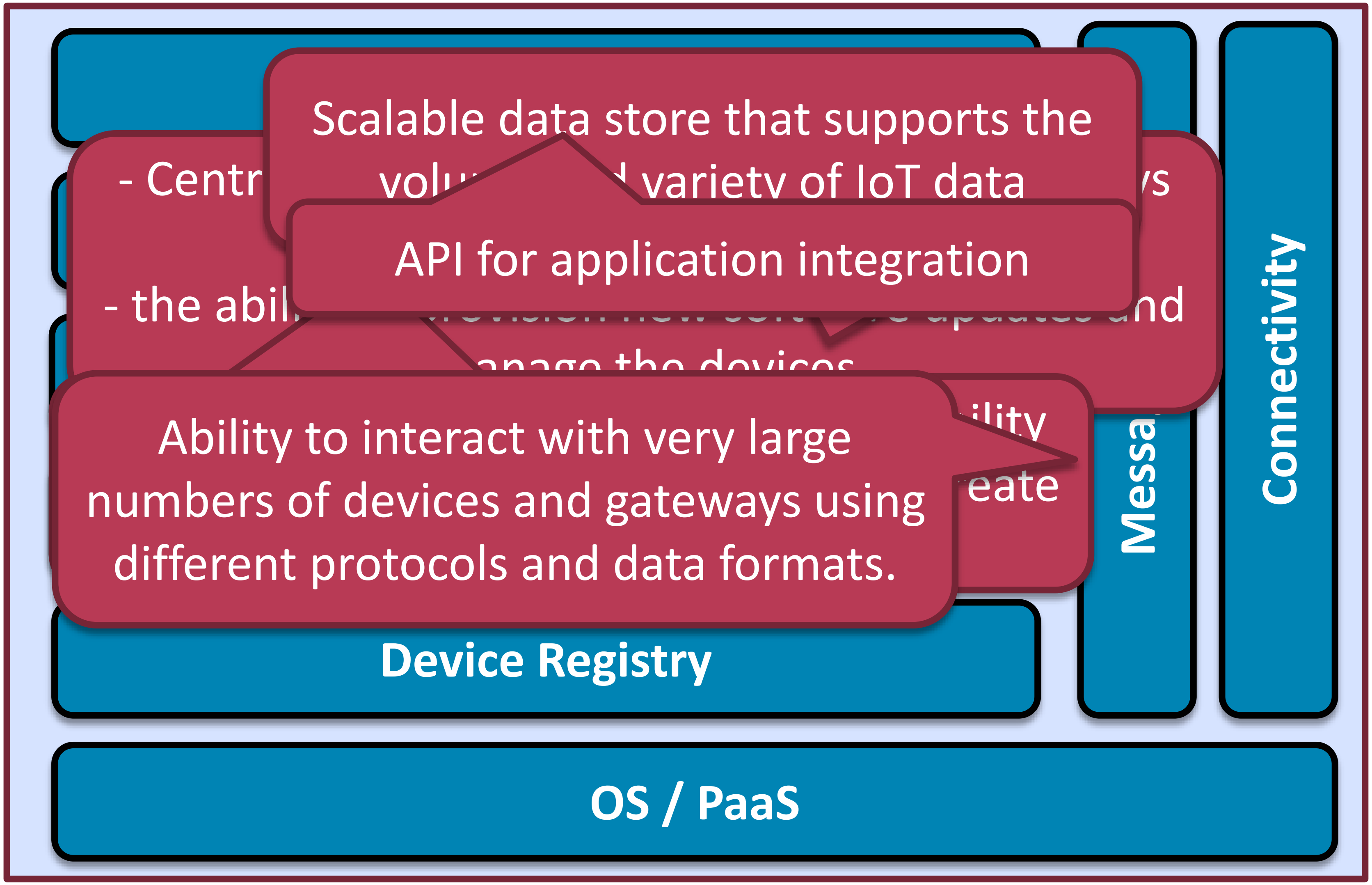
remo

Local persistence to support network latency, offline mode and real-time analytics

Application Container or Runtime Environment: Java, Python, Node.js

General purpose OS

Software stack for IoT Cloud Platforms



Scalable data store that supports the volu… l variety of IoT data

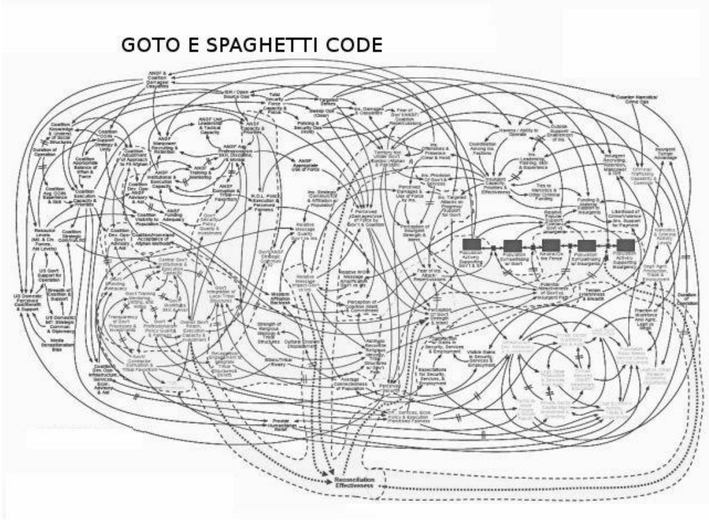
API for application integration

- Centr… …ys

- the abil… …nd …anage the devices

Ability to interact with very large numbers of devices and gateways using different protocols and data formats.

…ility …eate

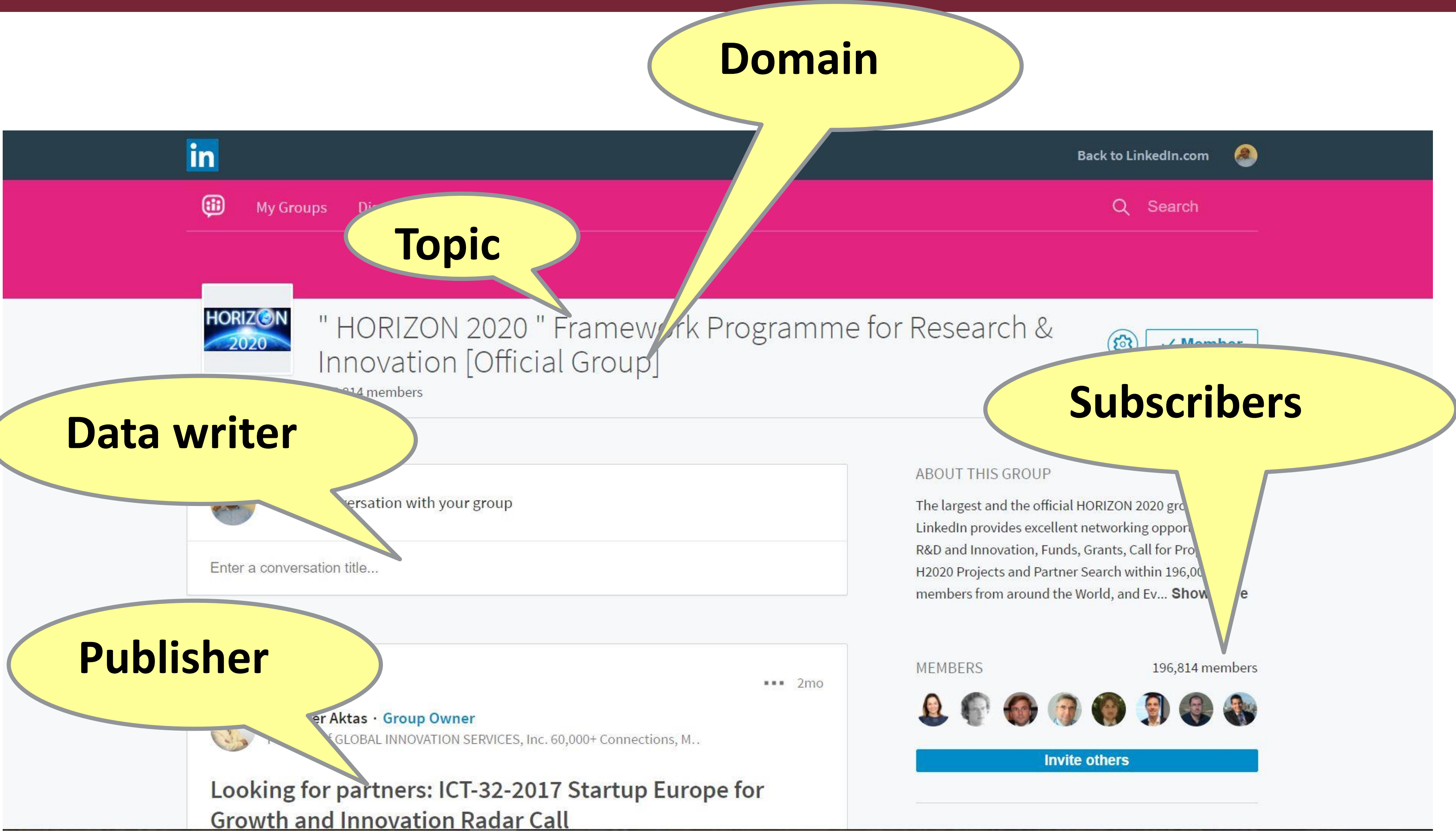


http://ziogeek.com/wp-content/uploads/2013/07/spaghetti.jpg

PUBLISH-SUBSCRIBE

How to avoid spaghetti code?

Publish-subscribe



Domain

Topic

Data writer

Subscribers

Publisher

" HORIZON 2020 " Framework Programme for Research & Innovation [Official Group]

814 members

ersation with your group

Enter a conversation title...

••• 2mo

er Aktas · Group Owner

f GLOBAL INNOVATION SERVICES, Inc. 60,000+ Connections, M..

Looking for partners: ICT-32-2017 Startup Europe for Growth and Innovation Radar Call

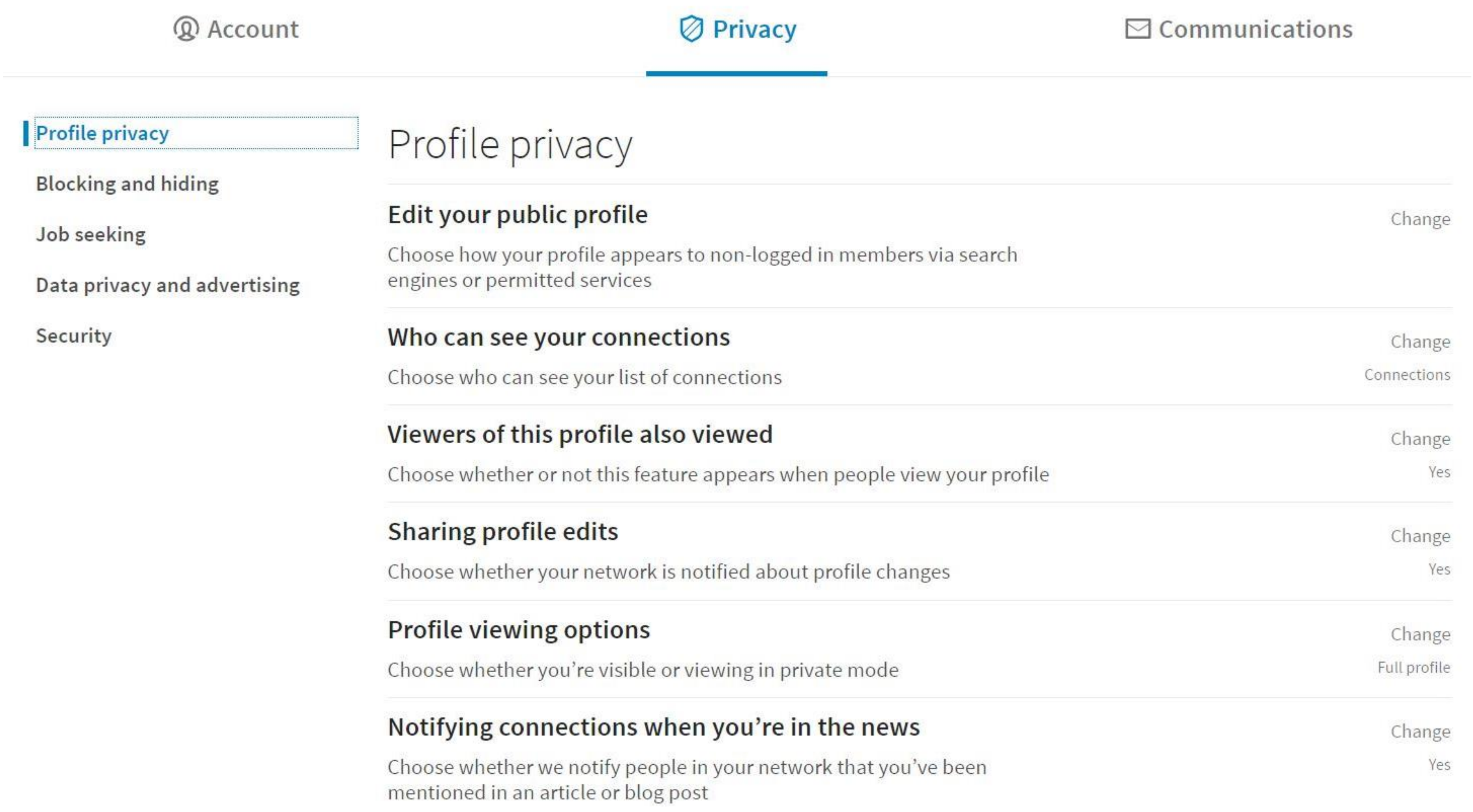
ABOUT THIS GROUP

The largest and the official HORIZON 2020 gro
LinkedIn provides excellent networking oppor
R&D and Innovation, Funds, Grants, Call for Pro
H2020 Projects and Partner Search within 196,00
members from around the World, and Ev... Show e

MEMBERS 196,814 members

Invite others

Control layer



Profile privacy

Edit your public profile — Change

Choose how your profile appears to non-logged in members via search engines or permitted services

Who can see your connections — Change

Choose who can see your list of connections — Connections

Viewers of this profile also viewed — Change

Choose whether or not this feature appears when people view your profile — Yes

Sharing profile edits — Change

Choose whether your network is notified about profile changes — Yes

Profile viewing options — Change

Choose whether you're visible or viewing in private mode — Full profile

Notifying connections when you're in the news — Change

Choose whether we notify people in your network that you've been mentioned in an article or blog post — Yes

COMMUNICATION

MQTT =
MQ TELEMETRY TRANSPORT

publish/subscribe,
extremely simple and lightweight
designed for constrained devices and low-bandwidth, high-latency or unreliable networks.
minimise network bandwidth , device resource requirements
attempting reliability and some assurance of delivery.

MQTT

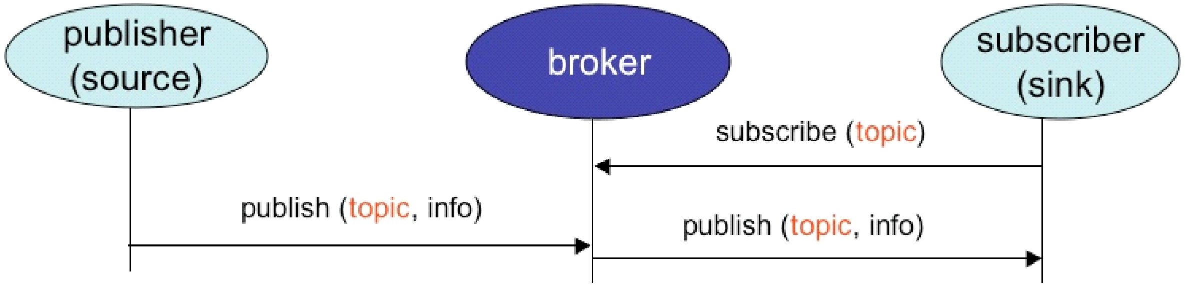
- Message Queuing Telemetry Transport
  - machine-to-machine (M2M)/"Internet of Things" connectivity protocol
  - Publish-subscribe
  - Asynchronous

- Message broker:
  - Receives subscriptions from clients on topics
  - Receives messages and distributes messages

- On top of TCP/IP

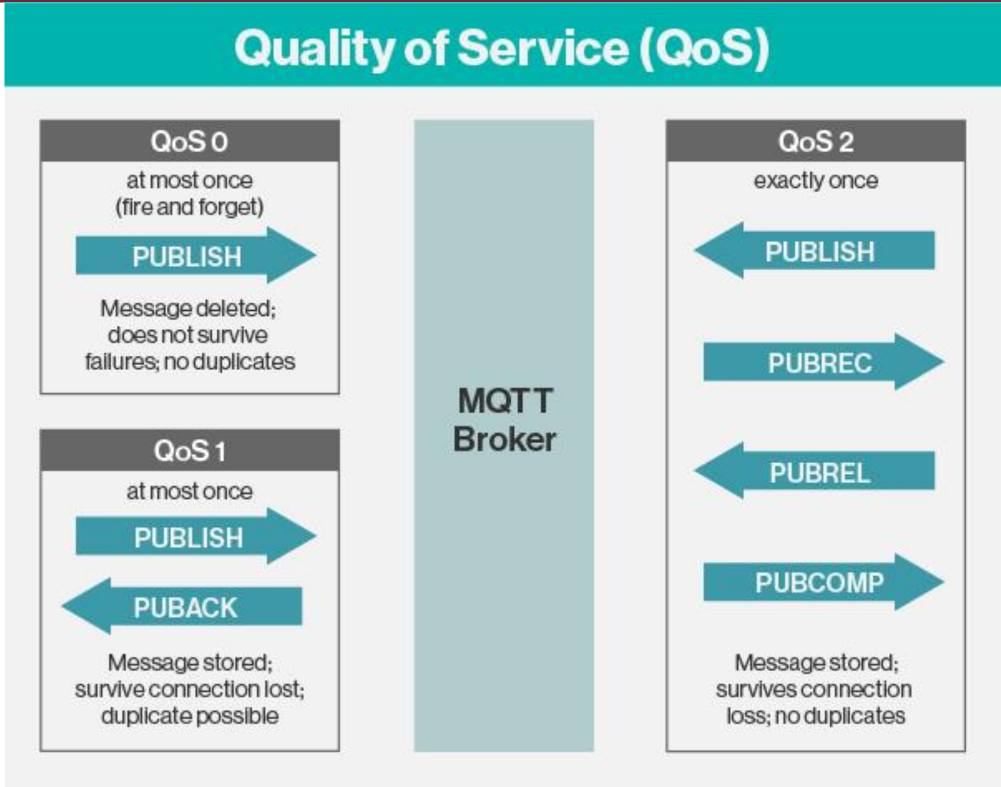
MQTT session

1. Connection
   - TCP/IP client-server (broker) connection
2. Authentication
   - Client: validates the server certificate
   - Optional: client certificate
     (e.g. SSL/TLS client-side certificates, clear-text username and password)
3. Communication
   - *publish*, *subscribe*, *unsubscribe*, *ping*
4. Termination



MQTT: lightweight protocol

- Message
  - Fixed header (2 bytes),
  - Opt: variable header,
  - Payload: <=256 MB
  - QoS
    - QoS0: at most once Unacknowledged Service (PUBLISH)
    - QoS1: at least once Acknowledged Service (PUBLISH/PUBACK)
    - QoS2: exactly once Assured Service (PUBLISH/PUBREC, PUBREL/PUBCOMP)



http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

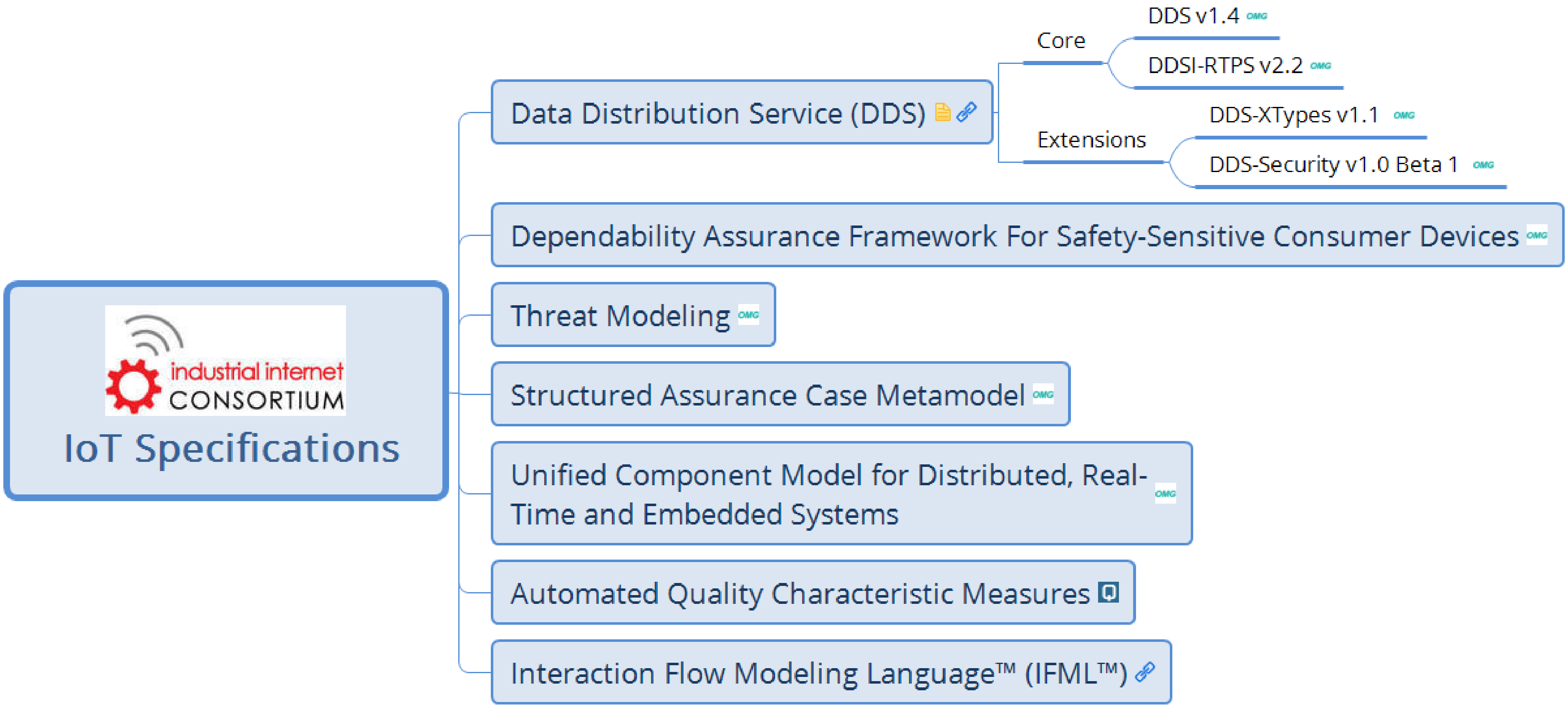
OMG

OBJECT MANAGEMENT GROUP

industrial internet CONSORTIUM

OMG DATA DISTRIBUTION SYSTEM FOR REAL-TIME SYSTEMS

Scalability → Flexibility → Publish-subscribe
Real-time → high-performane (latency: 25-30 µsec, throughput>10G…)
Interoperability
Platform independent,
Polyglot (Ada, C, C++, C#, .Net, Java, JavaScript, Scala, Lua, Ruby)



industrial internet CONSORTIUM

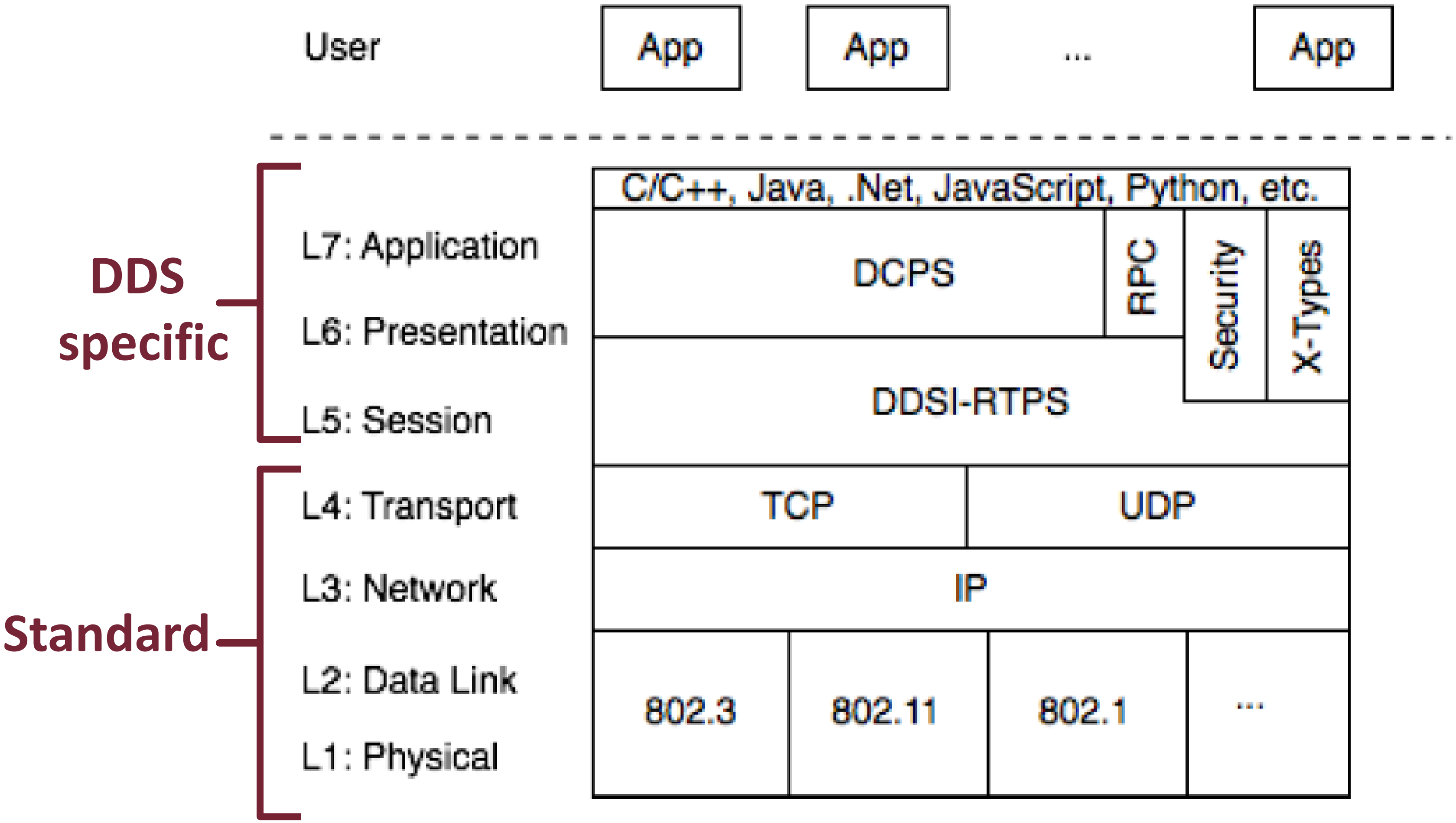
IoT Specifications

- Data Distribution Service (DDS)
  - Core
    - DDS v1.4
    - DDSI-RTPS v2.2
  - Extensions
    - DDS-XTypes v1.1
    - DDS-Security v1.0 Beta 1
- Dependability Assurance Framework For Safety-Sensitive Consumer Devices
- Threat Modeling
- Structured Assurance Case Metamodel
- Unified Component Model for Distributed, Real-Time and Embedded Systems
- Automated Quality Characteristic Measures
- Interaction Flow Modeling Language™ (IFML™)

Introducing DDS DDS™ – The Proven Data Connectivity Standard for IIoT™

FOUNDING & CONTRIBUTING MEMBERS

DDS standards



User

App

App

...

App

DDS specific

Standard

L7: Application

L6: Presentation

L5: Session

L4: Transport

L3: Network

L2: Data Link

L1: Physical

C/C++, Java, .Net, JavaScript, Python, etc.

DCPS

RPC

Security

X-Types

DDSI-RTPS

TCP

UDP

IP

802.3

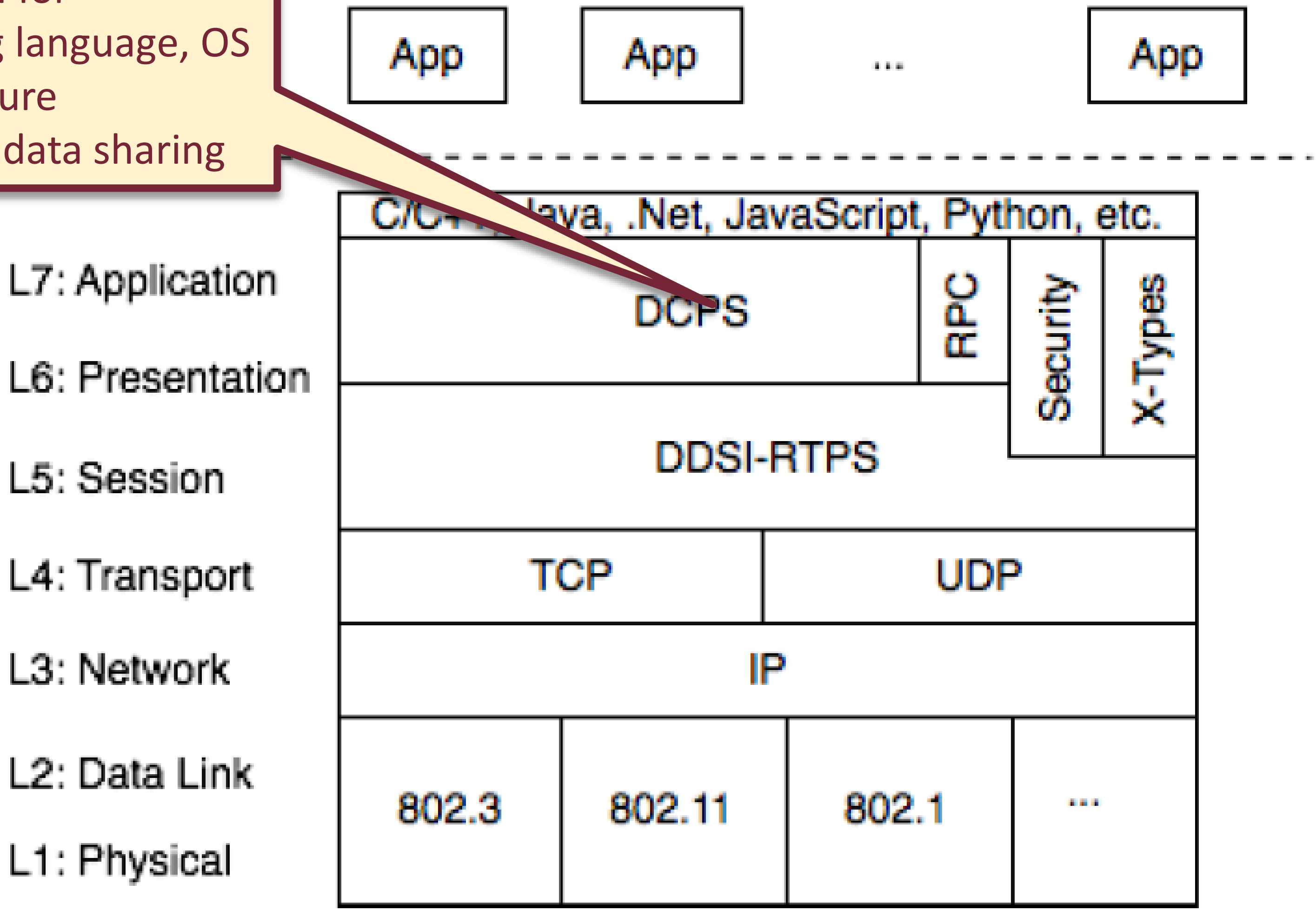
802.11

802.1

...

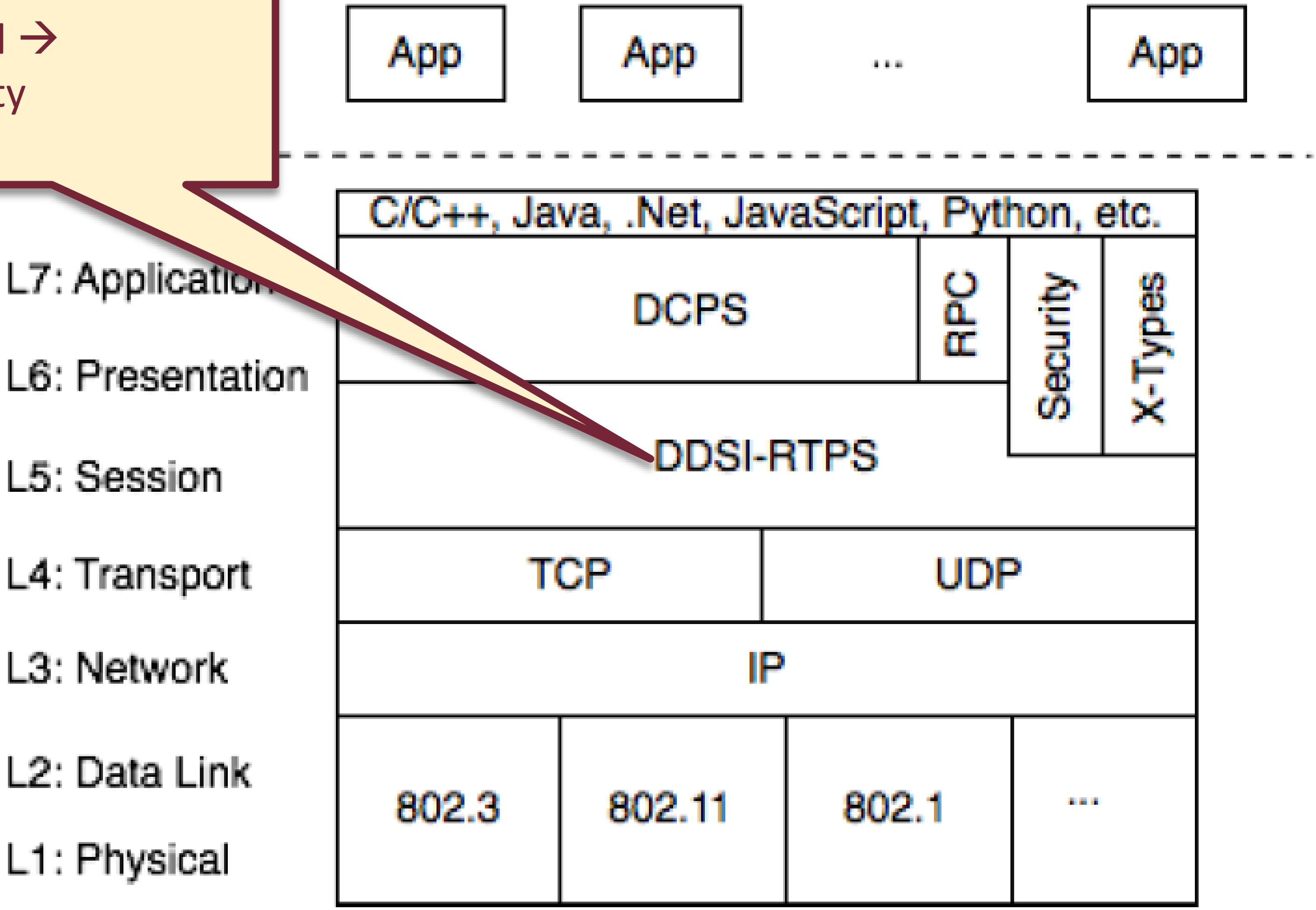
DDS standards

High level API for programming language, OS and architecture independent data sharing



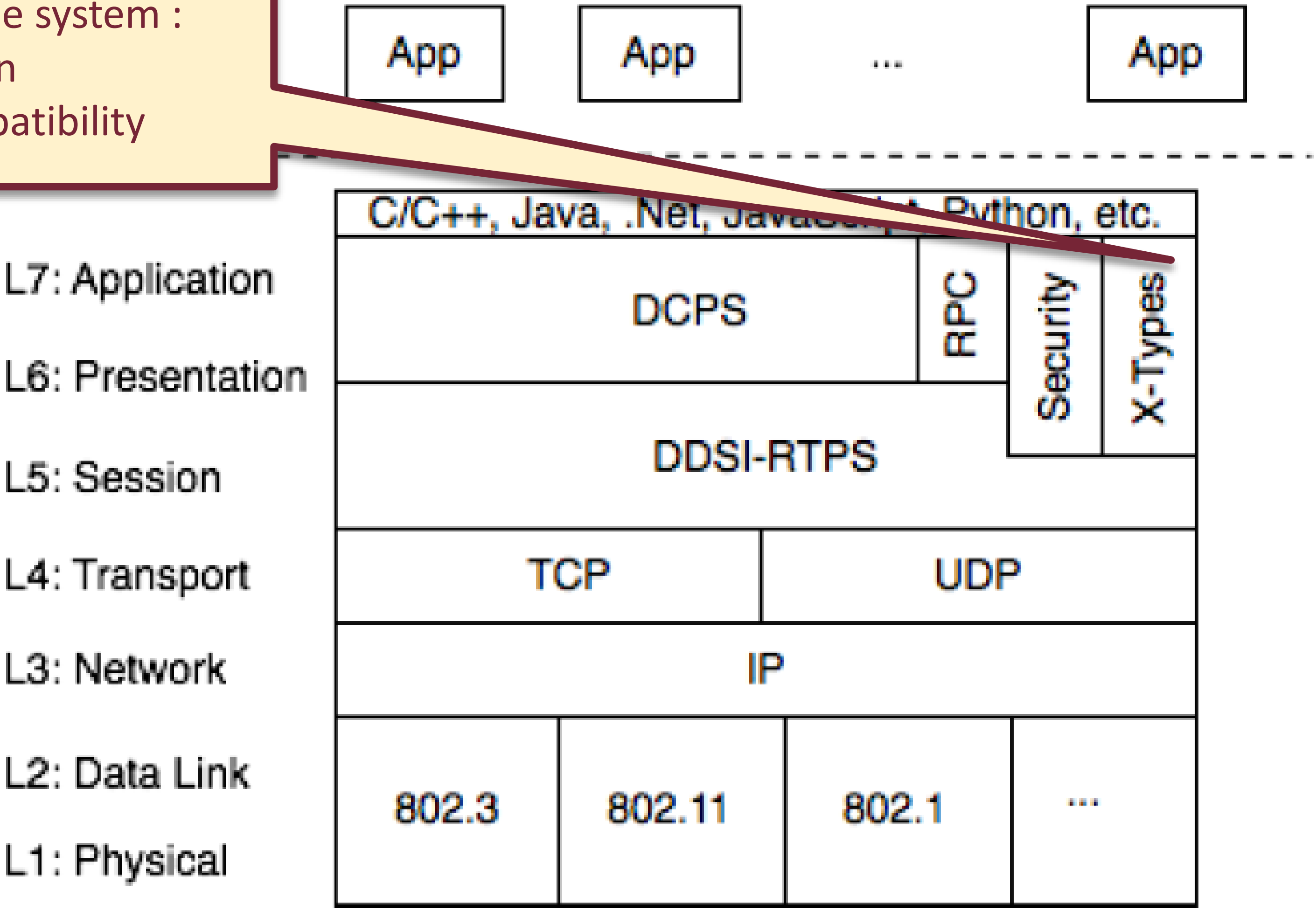
DDS Interoperability Wire Protocol
Real-time Publish-Subscribe Protocol

Wire protocol → interoperabilty



Type system

Structural type system :
type evolution
forward compatibility



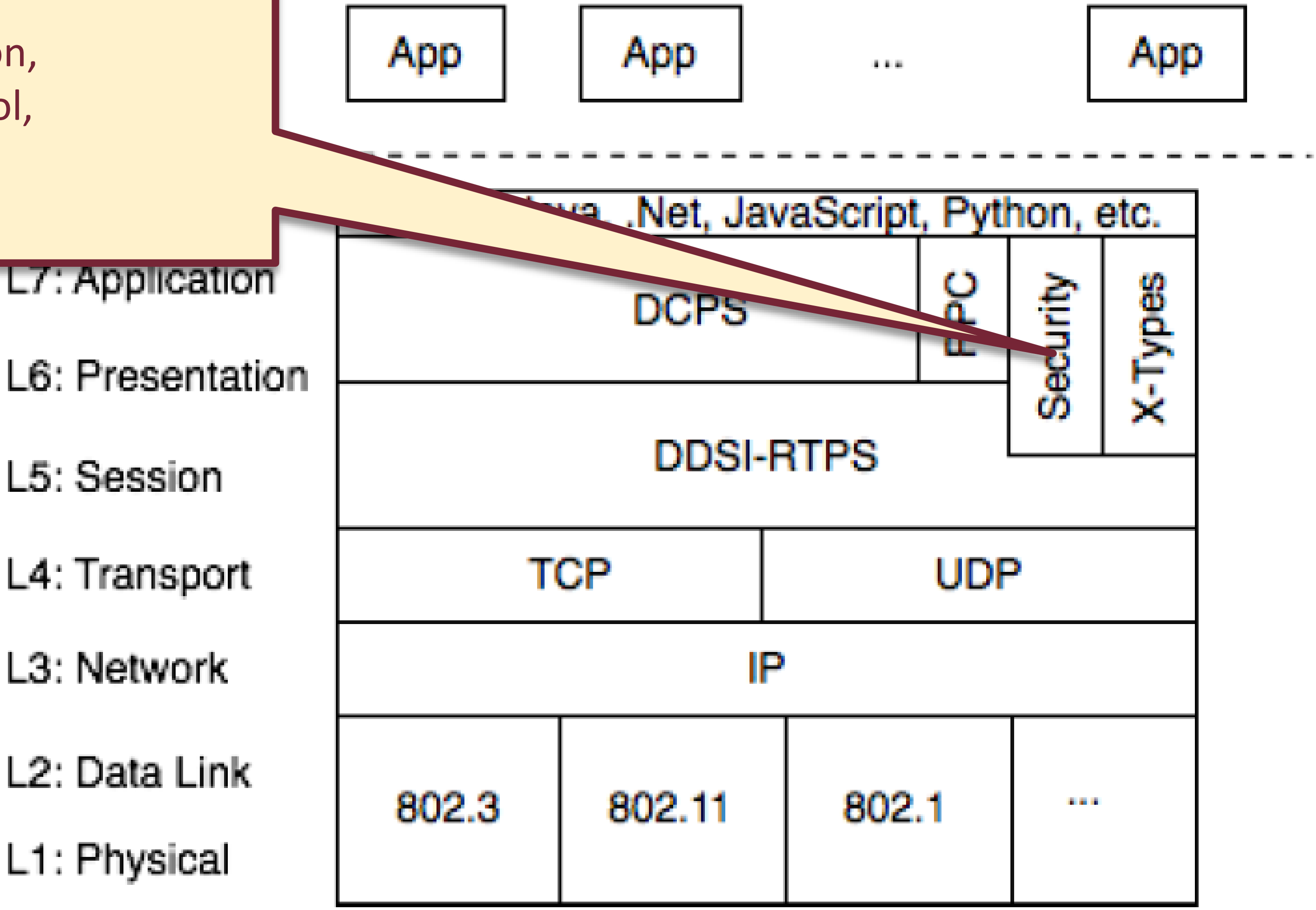
App | App | ... | App

C/C++, Java, .Net, JavaScript, Python, etc.

L7: Application L6: Presentation	DCPS		RPC	Security	X-Types
L5: Session	DDSI-RTPS				
L4: Transport	TCP	UDP			
L3: Network	IP				
L2: Data Link L1: Physical	802.3	802.11	802.1	...	

Security

Pluggable :
Authentication,
Access Control,
Crypto a
Logging



App
App
...
App

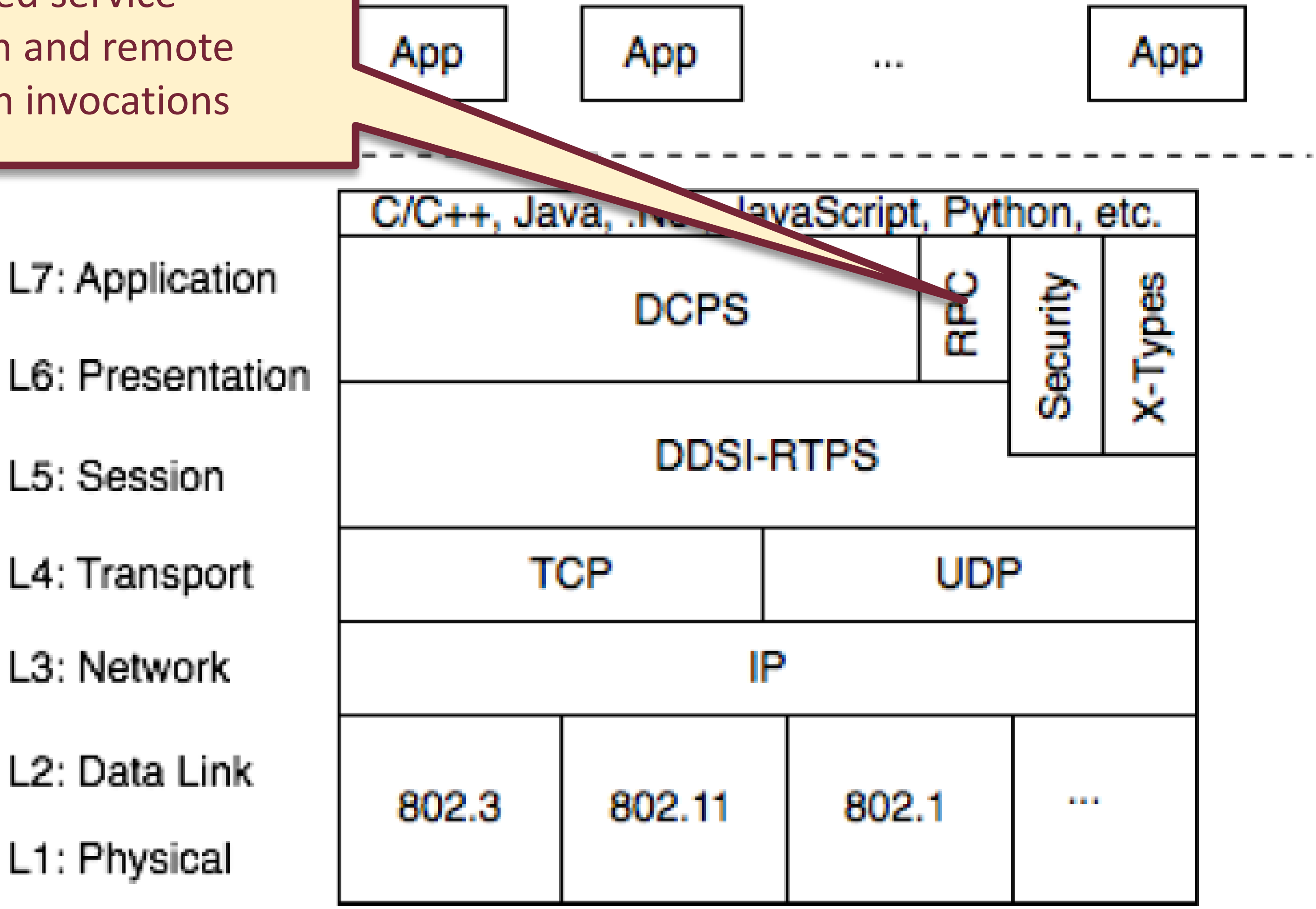
va, .Net, JavaScript, Python, etc.

L7: Application
L6: Presentation
L5: Session
L4: Transport
L3: Network
L2: Data Link
L1: Physical

DCPS
RPC
Security
X-Types
DDSI-RTPS
TCP
UDP
IP
802.3
802.11
802.1
...

RPC

Distributed service definition and remote operation invocations



App

App

...

App

C/C++, Java, .Net, JavaScript, Python, etc.

L7: Application

L6: Presentation

L5: Session

L4: Transport

L3: Network

L2: Data Link

L1: Physical

DCPS

RPC

Security

X-Types

DDSI-RTPS

TCP

UDP

IP

802.3

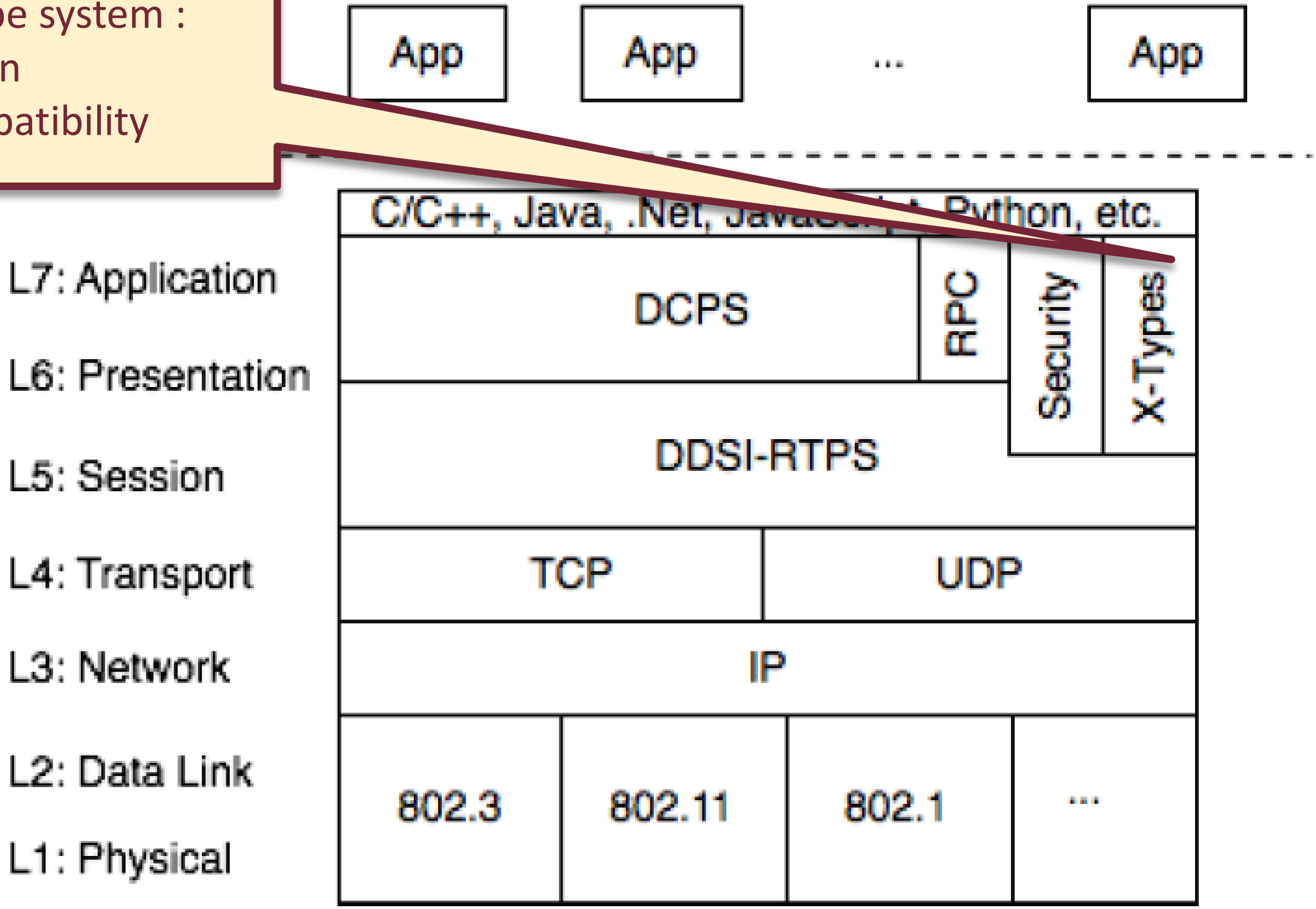
802.11

802.1

...

X-types

Structural type system :
type evolution
forward compatibility



User Datagram Protocol

Packet-switched network

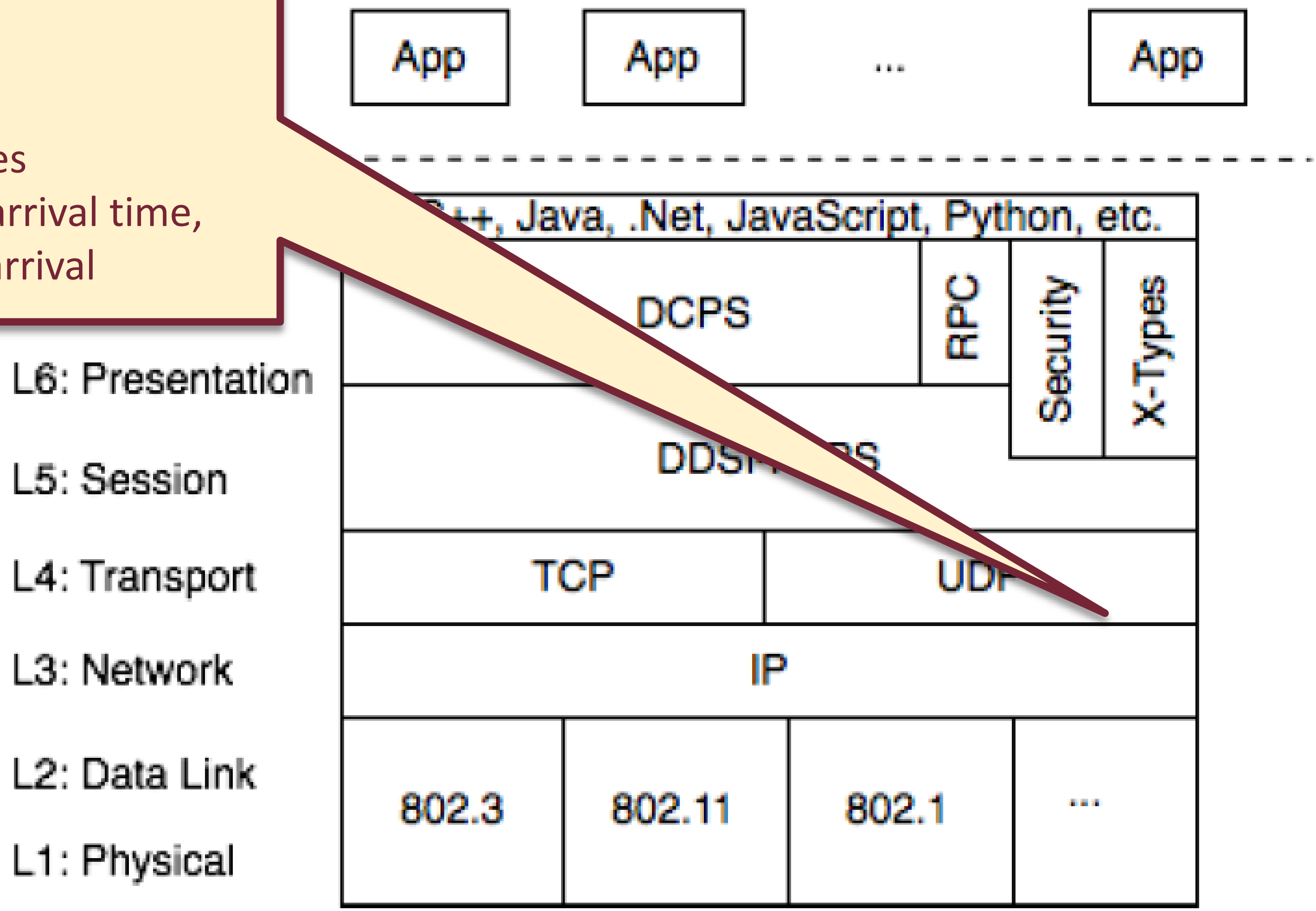
- Connectionless

Sections

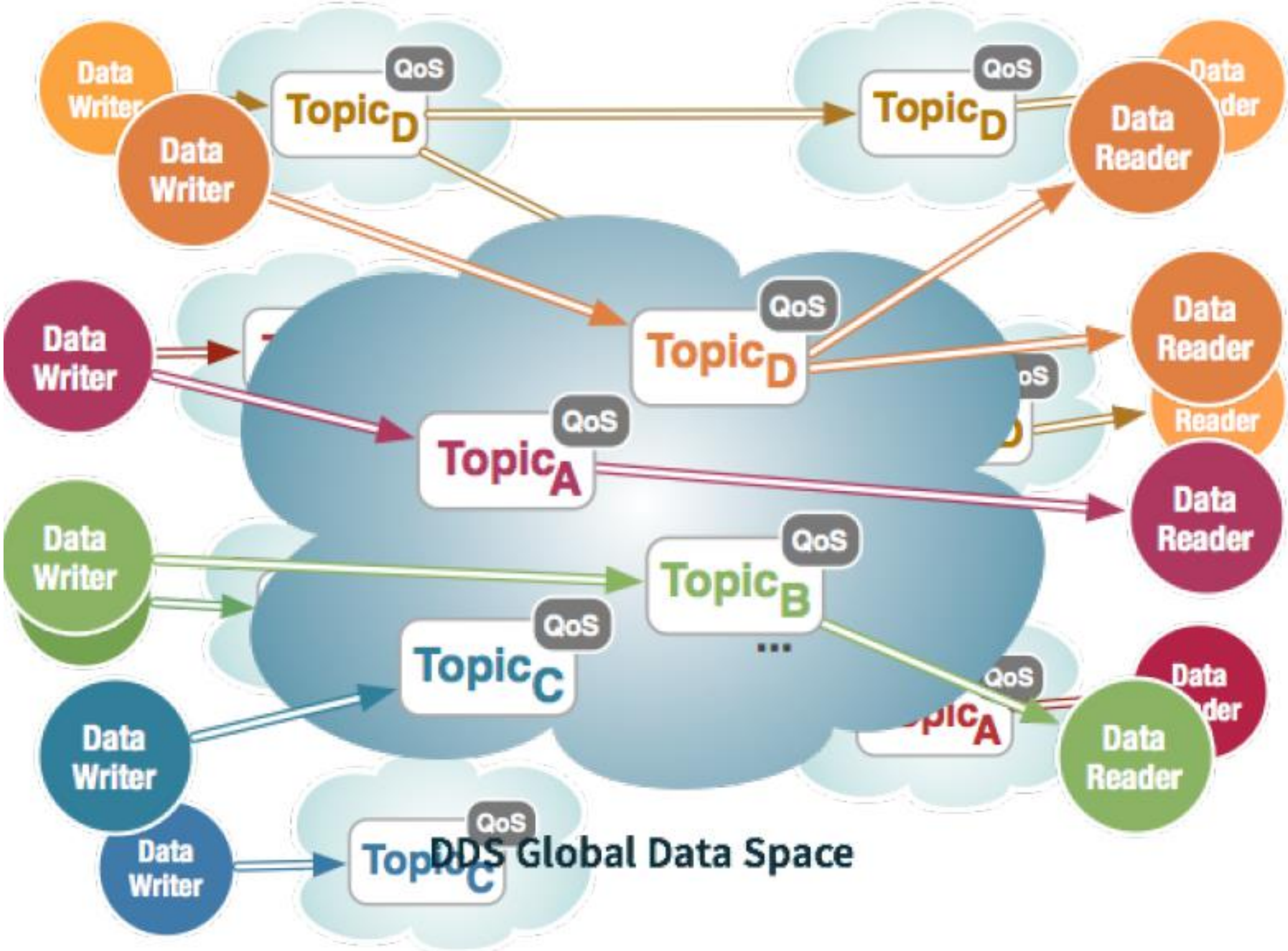
- Header
- Payload

No guarantees

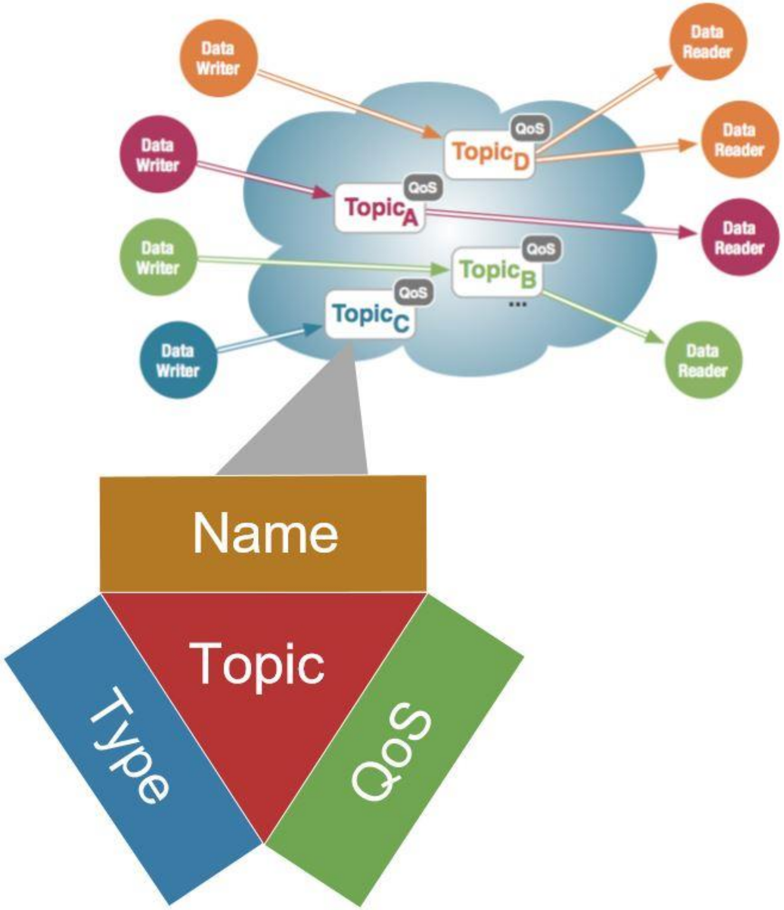
- Delivery, arrival time,
- Order of arrival

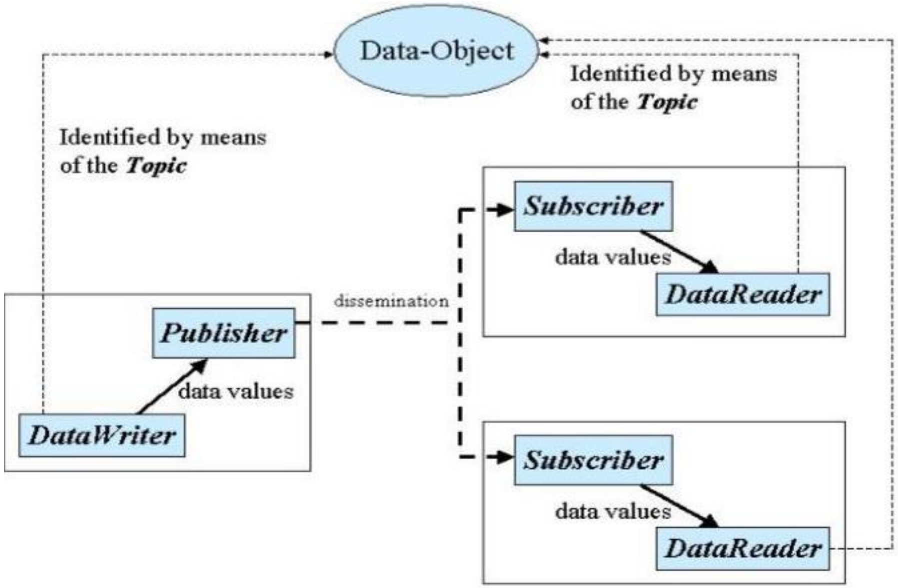


Decentralised Data-Space



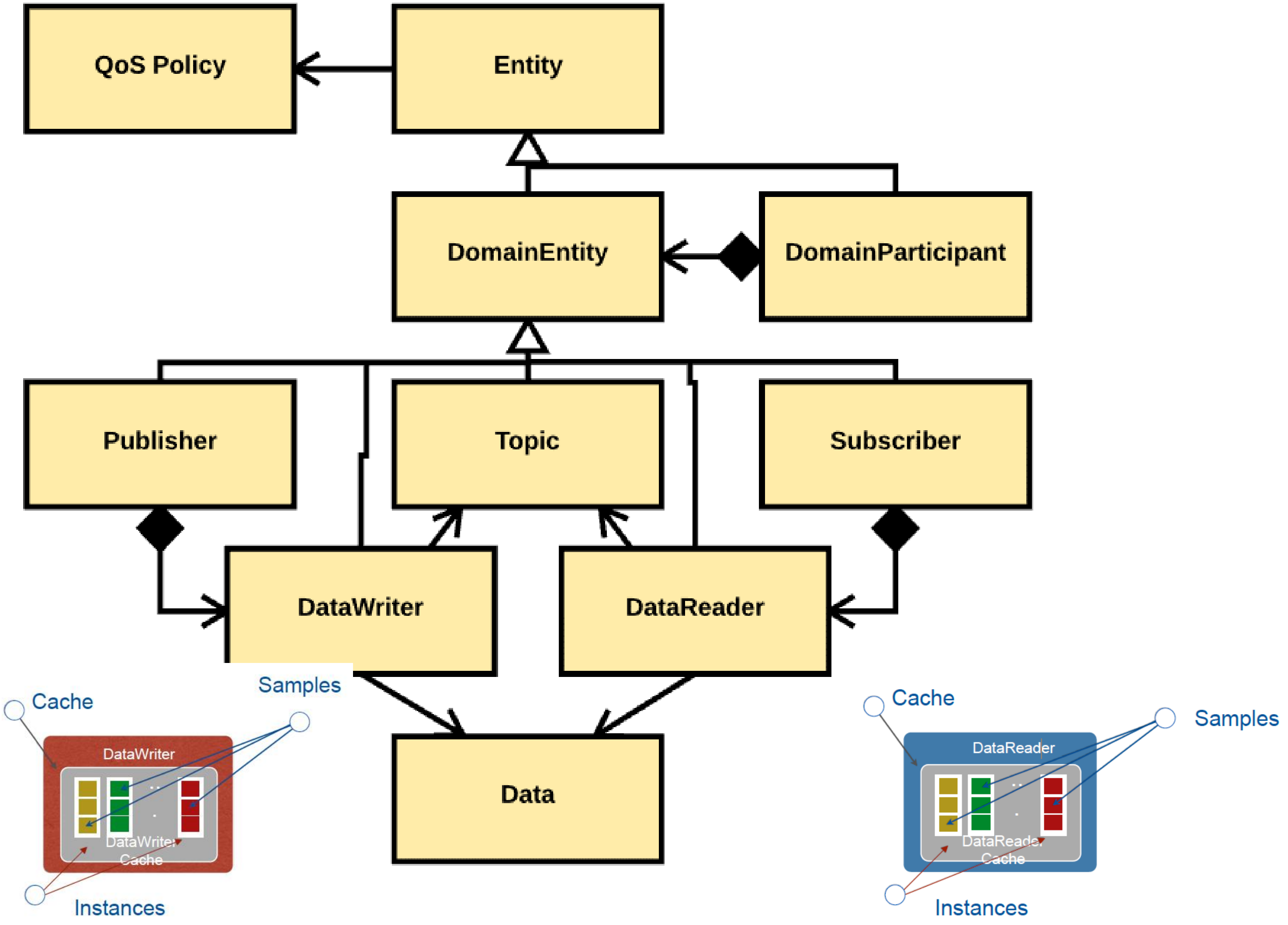
OMG DDS Core notions



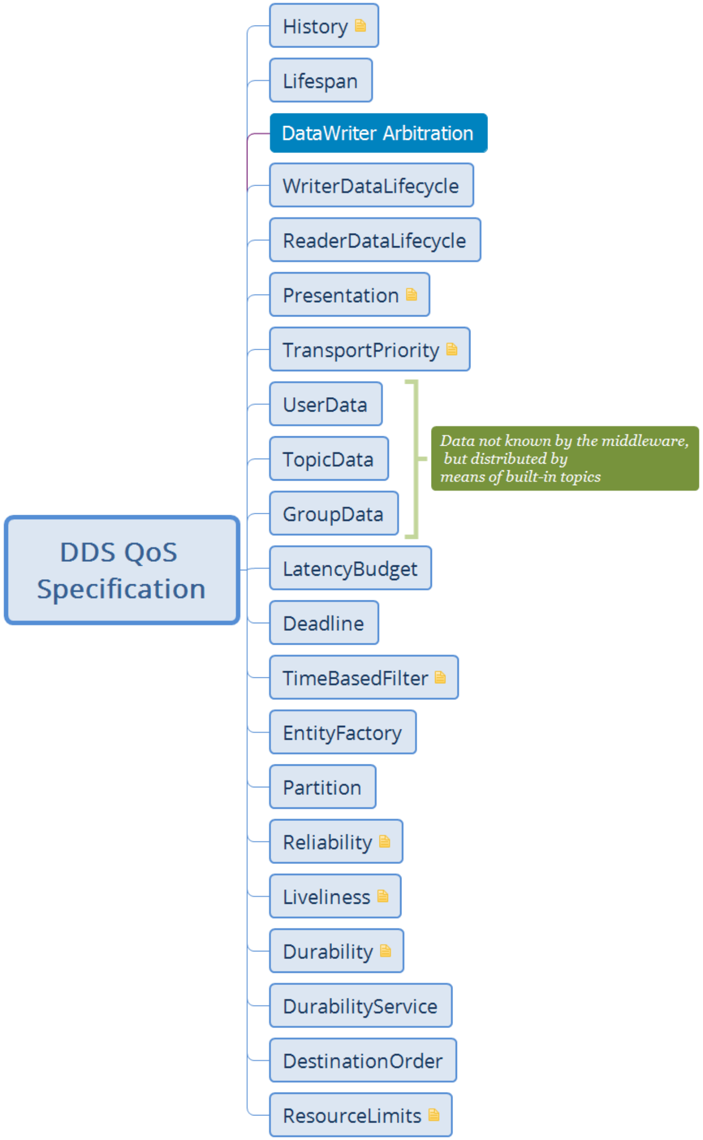


Introducing DDS DDS™ – The Proven Data Cbonnectivity Standard for IIoT™

DDS notions


QoS Policy
Entity
DomainEntity
DomainParticipant
Publisher
Topic
Subscriber
DataWriter
DataReader
Data
Cache
Samples
DataWriter
DataWriter Cache
Instances
Cache
Samples
DataReader
DataReader Cache
Instances

DDS QoS

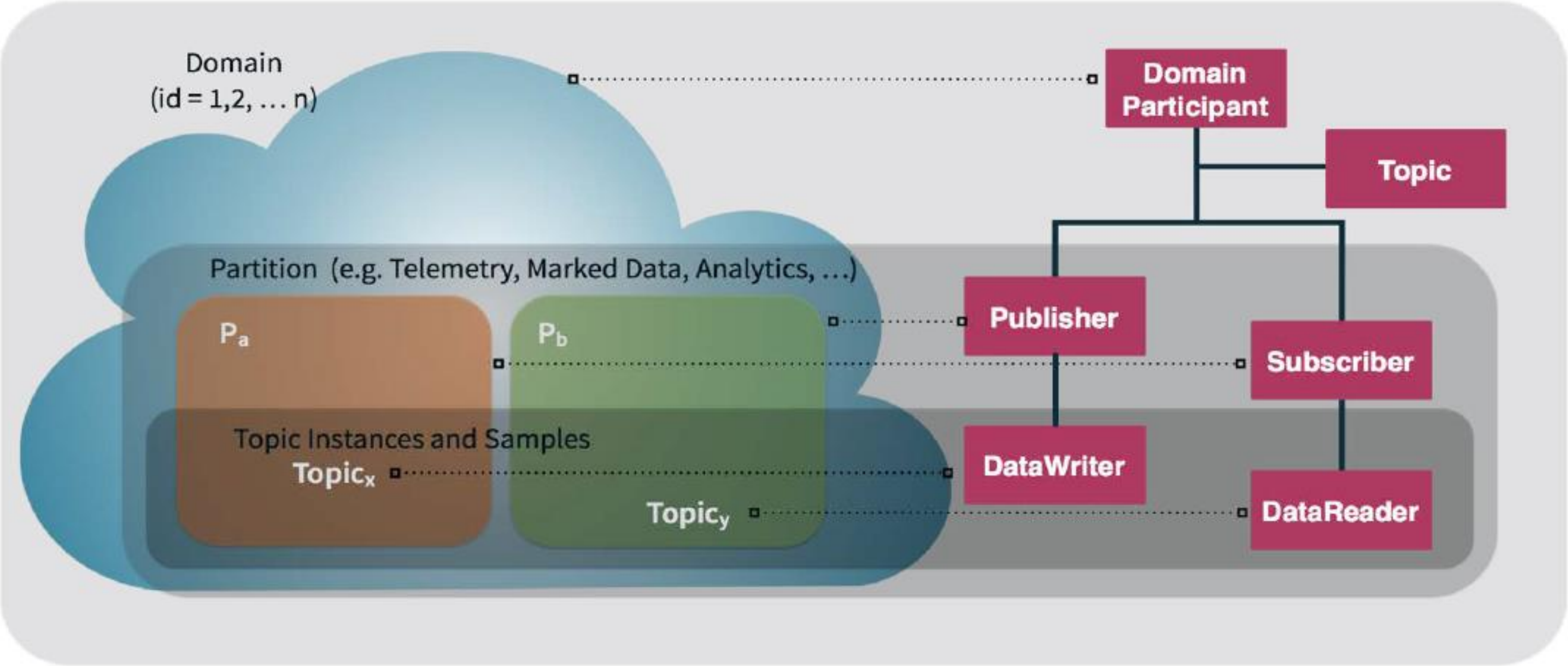


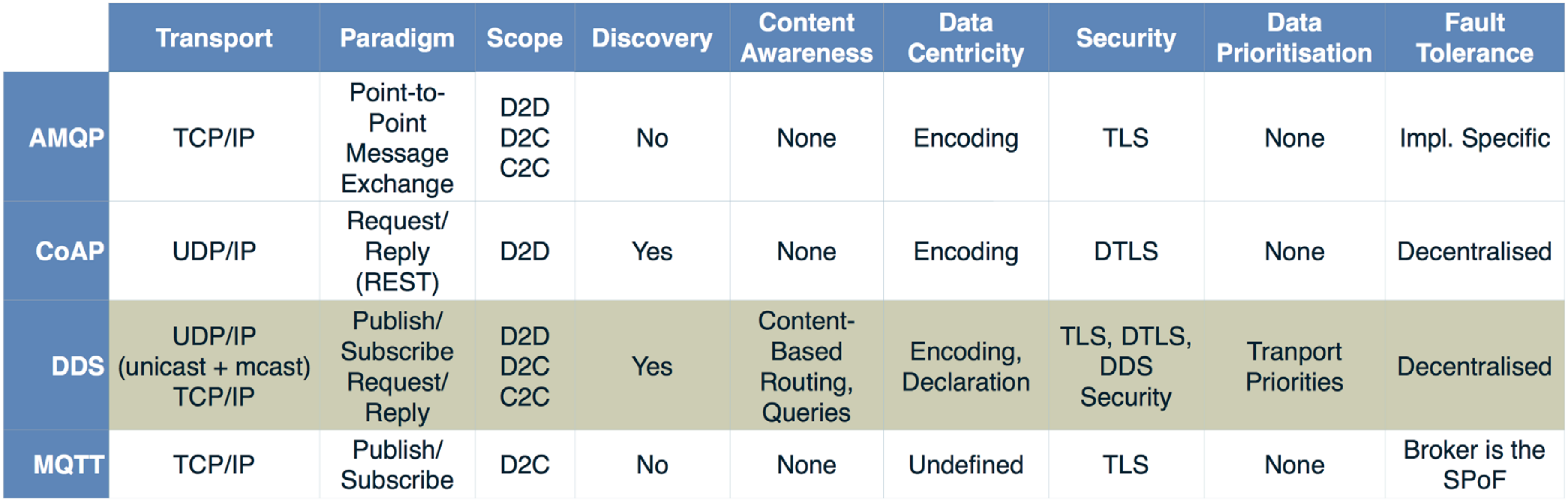
DDS QoS Specification

- History
- Lifespan
- DataWriter Arbitration
- WriterDataLifecycle
- ReaderDataLifecycle
- Presentation
- TransportPriority
- UserData
- TopicData
- GroupData
- LatencyBudget
- Deadline
- TimeBasedFilter
- EntityFactory
- Partition
- Reliability
- Liveliness
- Durability
- DurabilityService
- DestinationOrder
- ResourceLimits

Data not known by the middleware, but distributed by means of built-in topics

Anatomy of a DDS Application

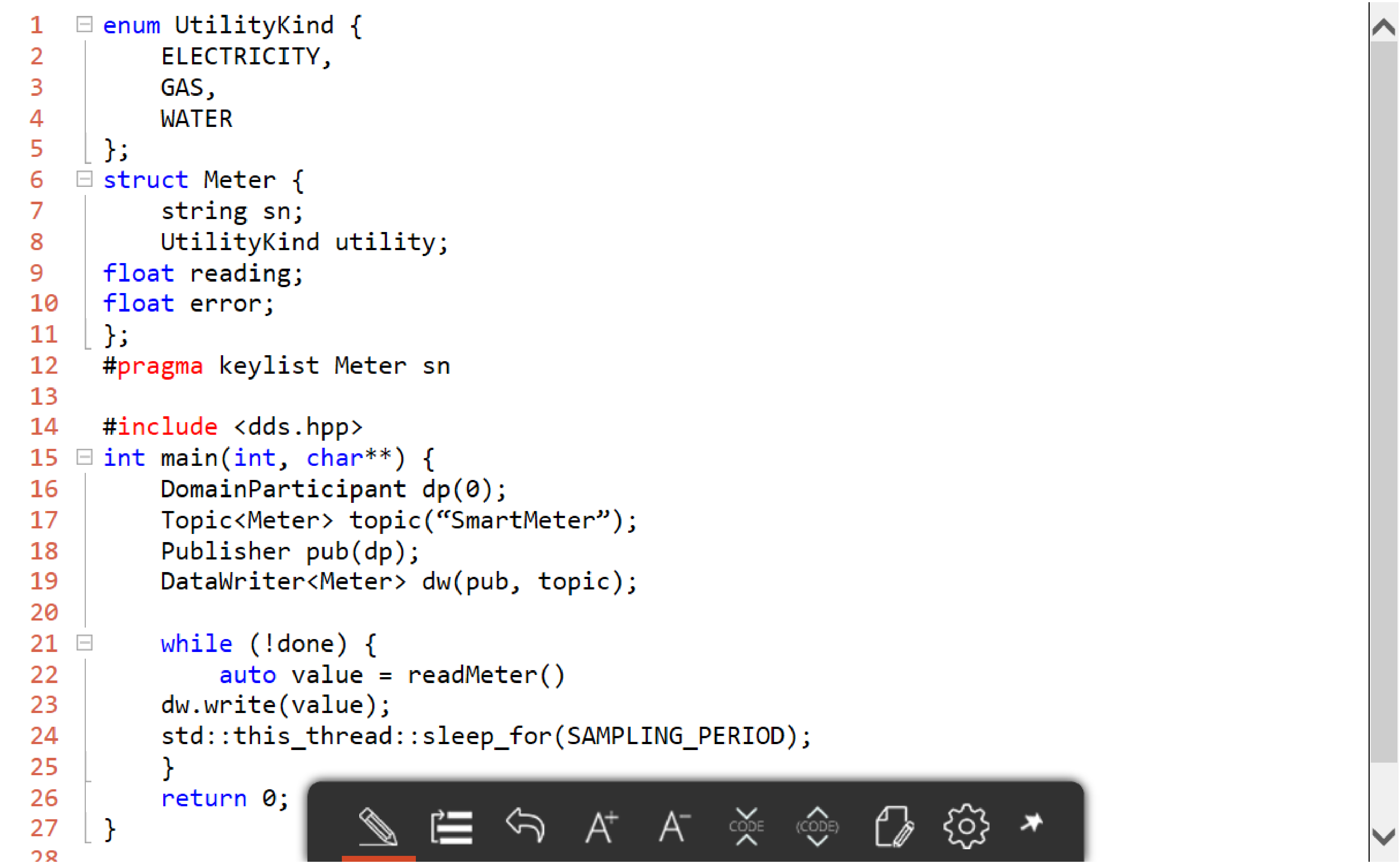




	Transport	Paradigm	Scope	Discovery	Content Awareness	Data Centricity	Security	Data Prioritisation	Fault Tolerance
AMQP	TCP/IP	Point-to-Point Message Exchange	D2D D2C C2C	No	None	Encoding	TLS	None	Impl. Specific
CoAP	UDP/IP	Request/ Reply (REST)	D2D	Yes	None	Encoding	DTLS	None	Decentralised
DDS	UDP/IP (unicast + mcast) TCP/IP	Publish/ Subscribe Request/ Reply	D2D D2C C2C	Yes	Content-Based Routing, Queries	Encoding, Declaration	TLS, DTLS, DDS Security	Tranport Priorities	Decentralised
MQTT	TCP/IP	Publish/ Subscribe	D2C	No	None	Undefined	TLS	None	Broker is the SPoF

[Ref: A Comparative Study of Data-Sharing Standards for the Internet of Things, Cutter Journal, Dec 2014

Writing data

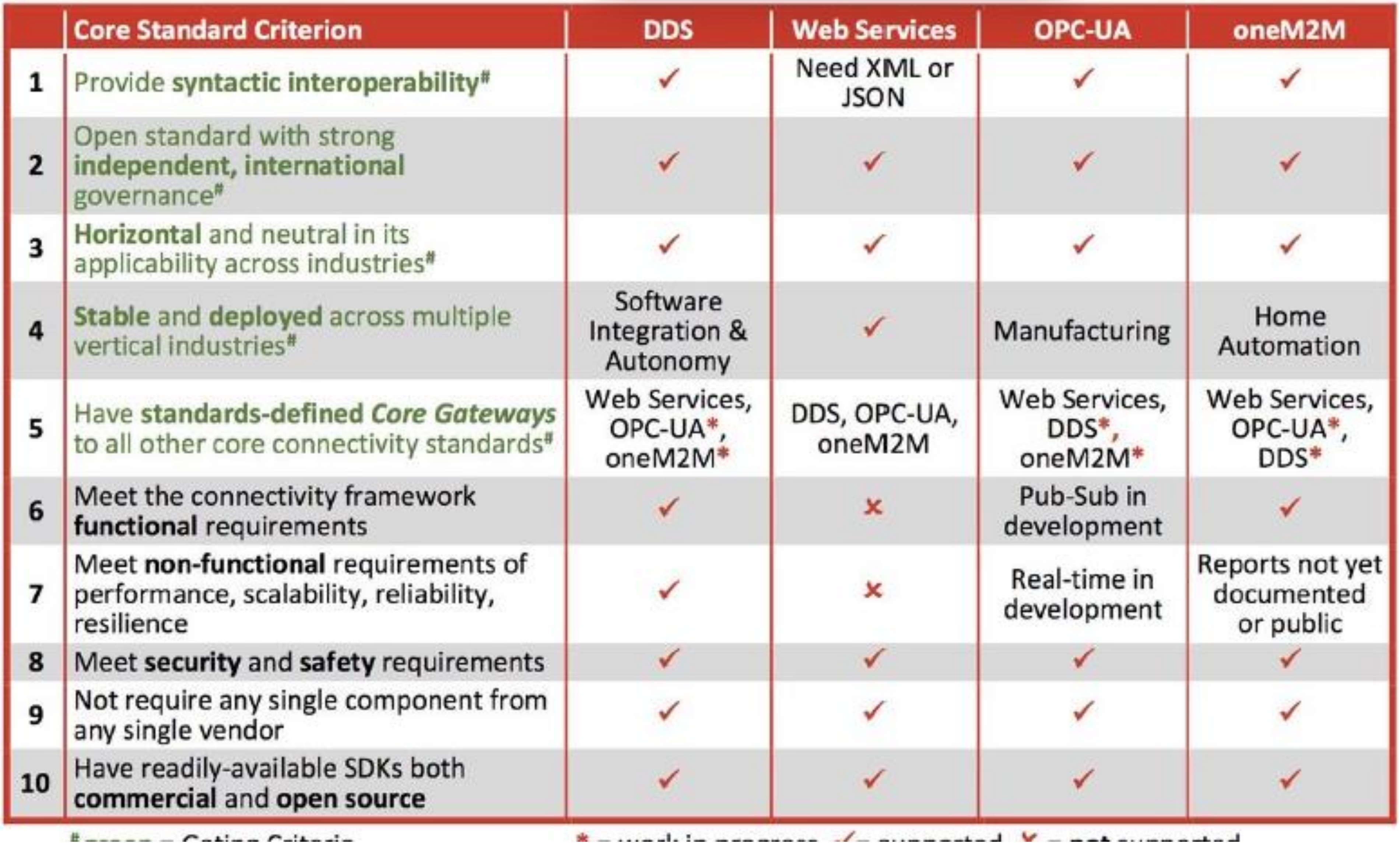


```
enum UtilityKind {
    ELECTRICITY,
    GAS,
    WATER
};
struct Meter {
    string sn;
    UtilityKind utility;
float reading;
float error;
};
#pragma keylist Meter sn

#include <dds.hpp>
int main(int, char**) {
    DomainParticipant dp(0);
    Topic<Meter> topic(“SmartMeter”);
    Publisher pub(dp);
    DataWriter<Meter> dw(pub, topic);

    while (!done) {
        auto value = readMeter()
    dw.write(value);
    std::this_thread::sleep_for(SAMPLING_PERIOD);
    }
    return 0;
}
```

IIC Connectivity Framework

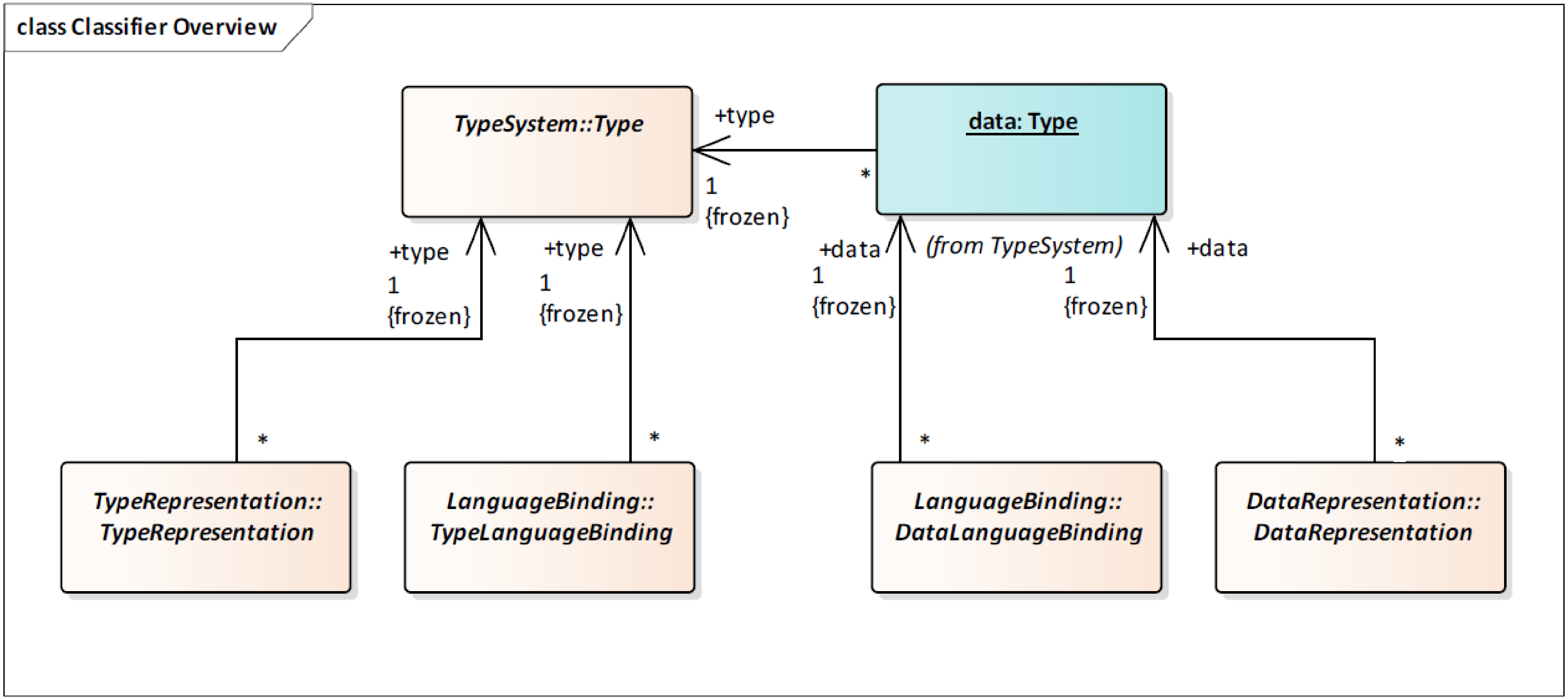


	Core Standard Criterion	DDS	Web Services	OPC-UA	oneM2M
1	Provide **syntactic interoperability**#	✓	Need XML or JSON	✓	✓
2	Open standard with strong **independent, international** governance#	✓	✓	✓	✓
3	**Horizontal** and neutral in its applicability across industries#	✓	✓	✓	✓
4	**Stable** and **deployed** across multiple vertical industries#	Software Integration & Autonomy	✓	Manufacturing	Home Automation
5	Have **standards-defined *Core Gateways*** to all other core connectivity standards#	Web Services, OPC-UA*, oneM2M*	DDS, OPC-UA, oneM2M	Web Services, DDS*, oneM2M*	Web Services, OPC-UA*, DDS*
6	Meet the connectivity framework **functional** requirements	✓	✗	Pub-Sub in development	✓
7	Meet **non-functional** requirements of performance, scalability, reliability, resilience	✓	✗	Real-time in development	Reports not yet documented or public
8	Meet **security** and **safety** requirements	✓	✓	✓	✓
9	Not require any single component from any single vendor	✓	✓	✓	✓
10	Have readily-available SDKs both **commercial** and **open source**	✓	✓	✓	✓

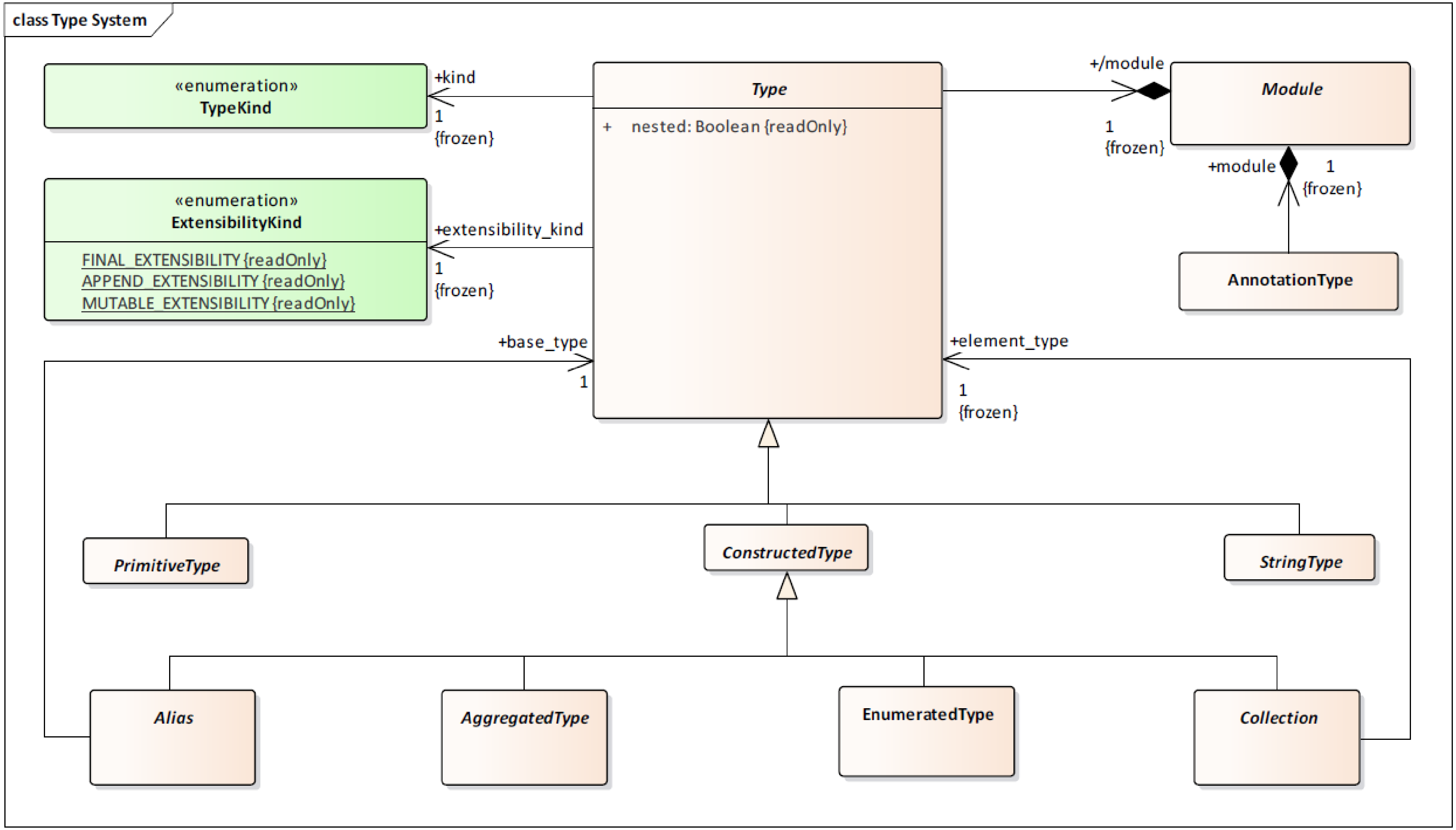
#green = Gating Criteria * = work in progress ✓ = supported ✗ = not supported

EXTENSIBLE AND DYNAMIC TOPIC TYPES FOR DDS

Classifier


class Classifier Overview
TypeSystem::Type
+type
1
{frozen}
*
data: Type
(from TypeSystem)
+type
1
{frozen}
+type
1
{frozen}
+data
1
{frozen}
1
{frozen}
+data
*
*
*
*
TypeRepresentation::
TypeRepresentation
LanguageBinding::
TypeLanguageBinding
LanguageBinding::
DataLanguageBinding
DataRepresentation::
DataRepresentation

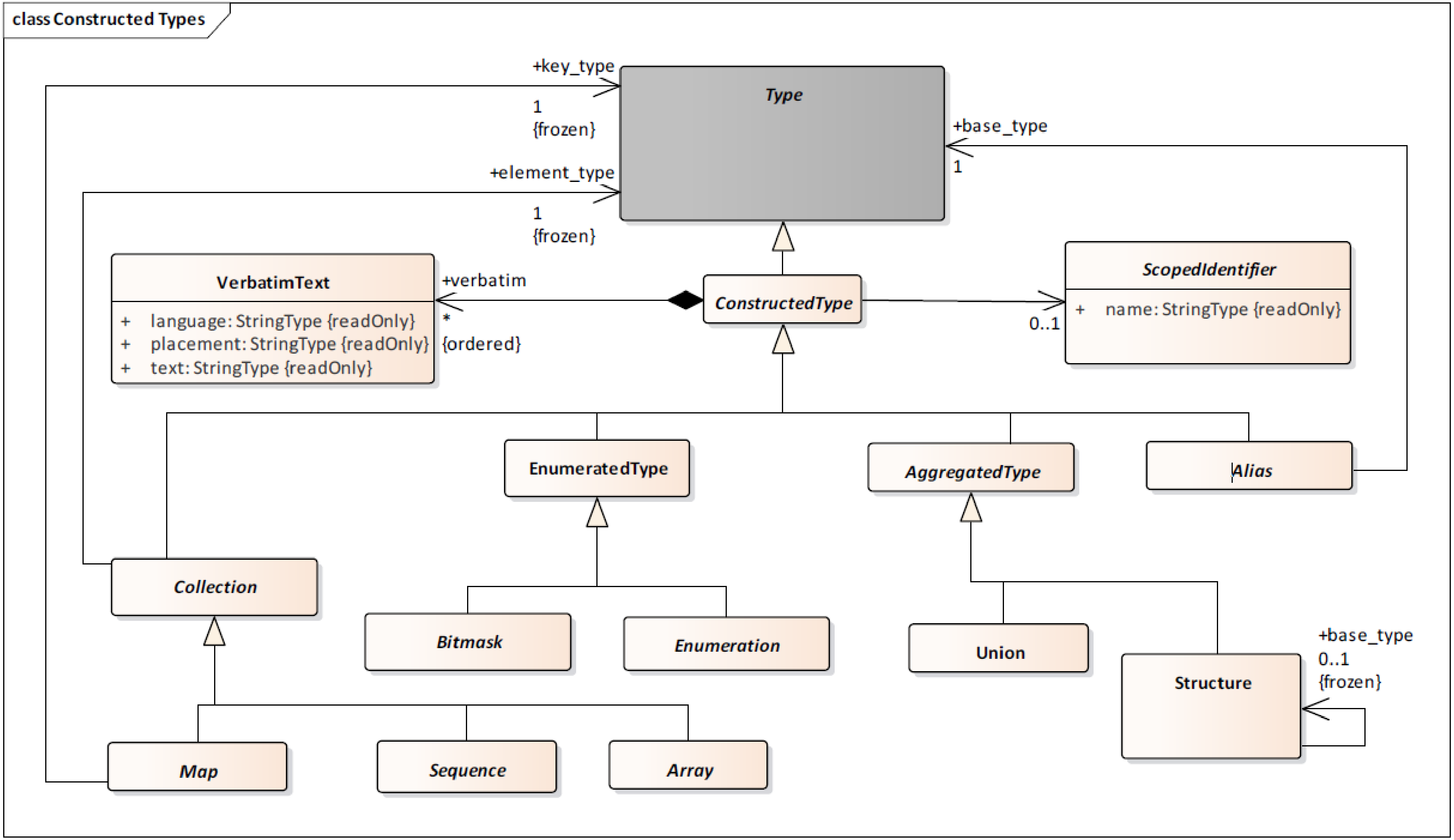
Type System Model



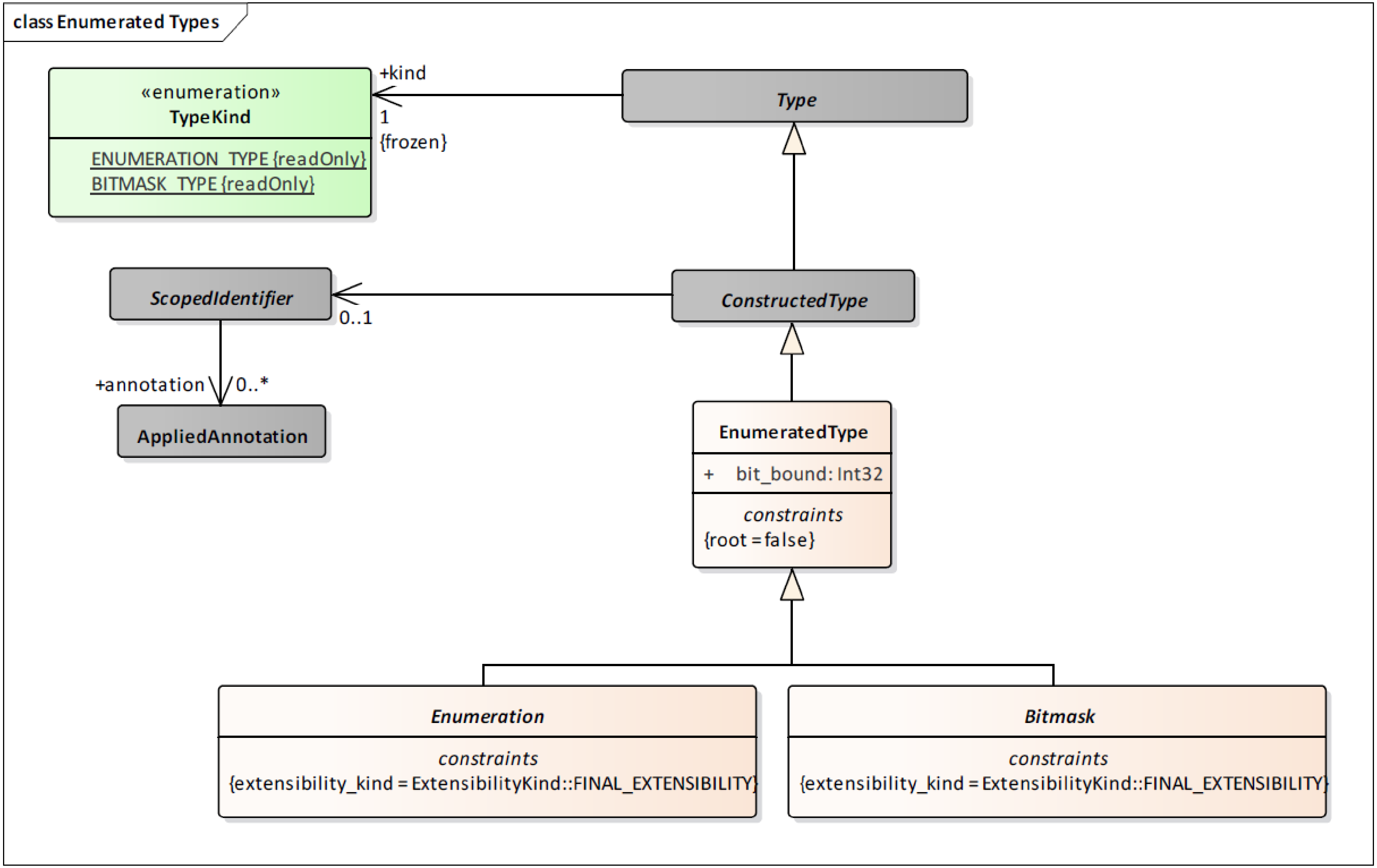
Primitive types (read only)

  - Integer (16,32,64)
  - INT_xx_TYPE
  - UINT_xx_TYPE
  - Float (32,64,128)
  - FLOAT_xx_TYPE
- Byte
  - BYTE_TYPE
- Boolean
  - BOOLEAN_TYPE
- Char and string (8,16)
  - CHAR_xx_TYPE

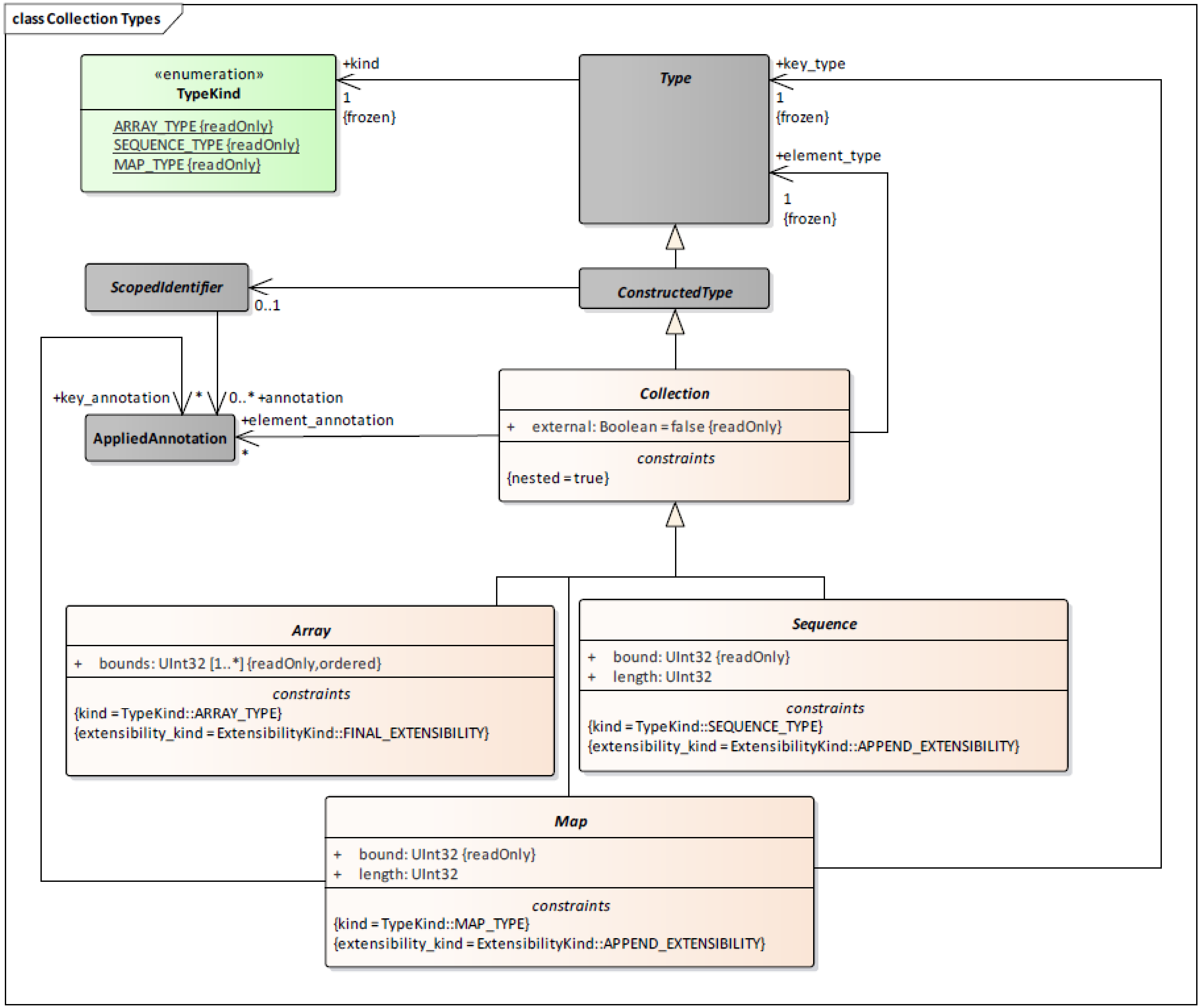
Constructed Types



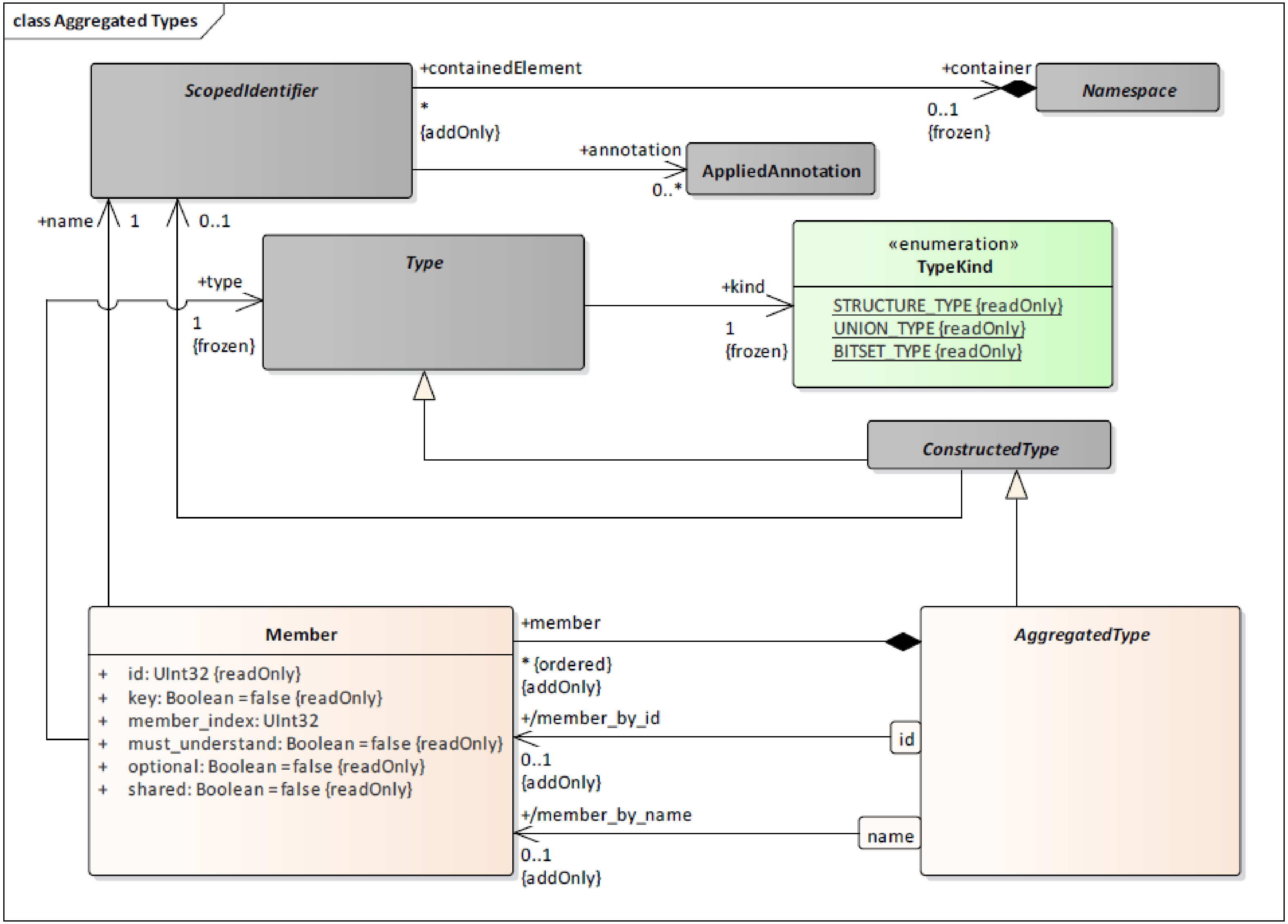
Enumerated Types



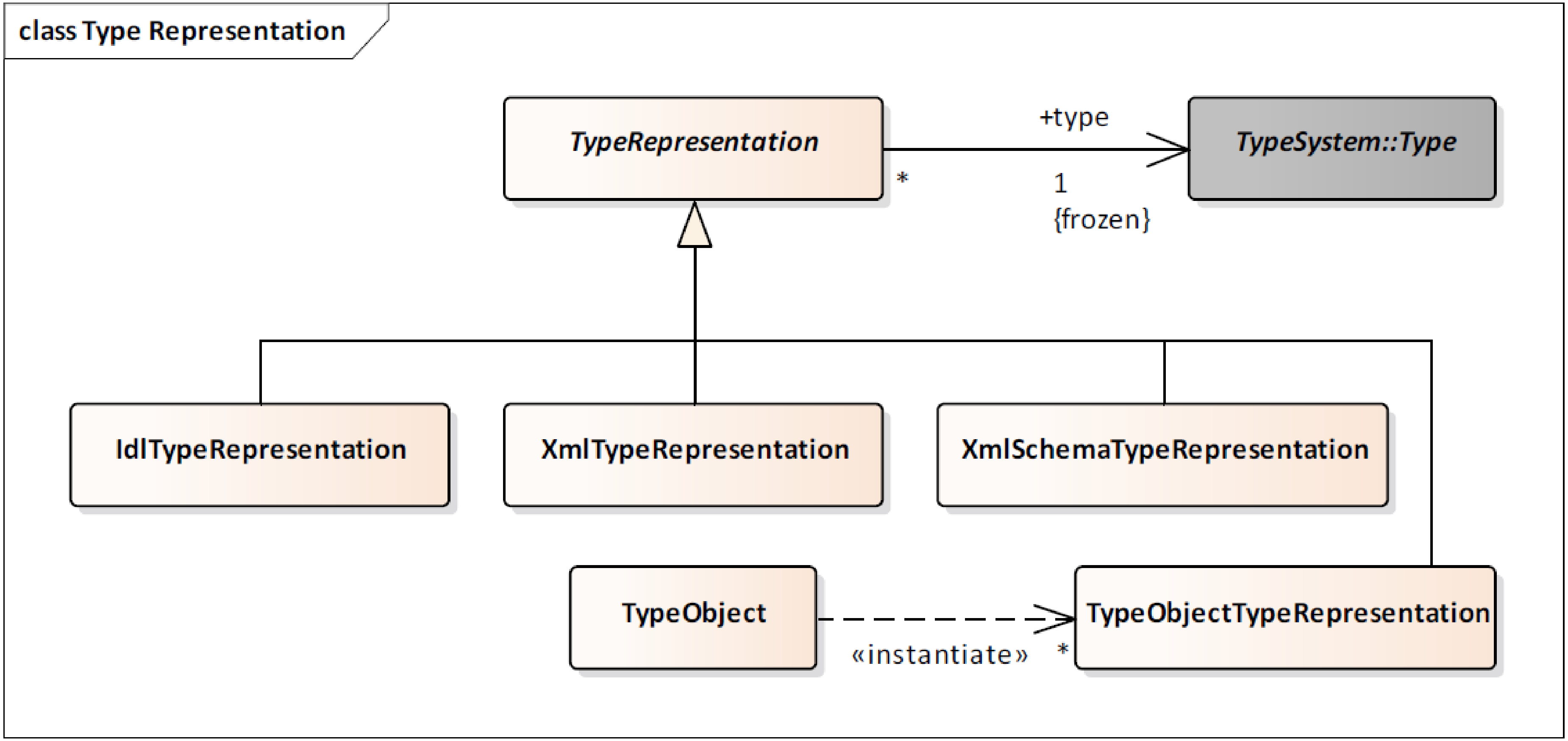
Collection Types


class Collection Types
«enumeration»
TypeKind
ARRAY_TYPE {readOnly}
SEQUENCE_TYPE {readOnly}
MAP_TYPE {readOnly}
+kind
1
{frozen}
Type
+key_type
1
{frozen}
+element_type
1
{frozen}
ScopedIdentifier
0..1
ConstructedType
+key_annotation
*
0..*
+annotation
+element_annotation
*
AppliedAnnotation
Collection
+ external: Boolean = false {readOnly}
constraints
{nested = true}
Array
+ bounds: UInt32 [1..*] {readOnly,ordered}
constraints
{kind = TypeKind::ARRAY_TYPE}
{extensibility_kind = ExtensibilityKind::FINAL_EXTENSIBILITY}
Sequence
+ bound: UInt32 {readOnly}
+ length: UInt32
constraints
{kind = TypeKind::SEQUENCE_TYPE}
{extensibility_kind = ExtensibilityKind::APPEND_EXTENSIBILITY}
Map
+ bound: UInt32 {readOnly}
+ length: UInt32
constraints
{kind = TypeKind::MAP_TYPE}
{extensibility_kind = ExtensibilityKind::APPEND_EXTENSIBILITY}

Aggregated Types



Type Representation


class Type Representation
TypeRepresentation
+type
TypeSystem::Type
*
1
{frozen}
IdlTypeRepresentation
XmlTypeRepresentation
XmlSchemaTypeRepresentation
TypeObject
TypeObjectTypeRepresentation
«instantiate»
*