



Lorenzo Gagliardini
Politecnico di Torino,
ITALY

Packet Utilisation Standard and Mission
Operations: a Service Frameworks
Comparison

Politecnico di Torino- STAR Team

STAR Team: Undergraduate and PhD students led by
Prof. Sabrina Corpino

Main Topic: CubeSat

Research areas:

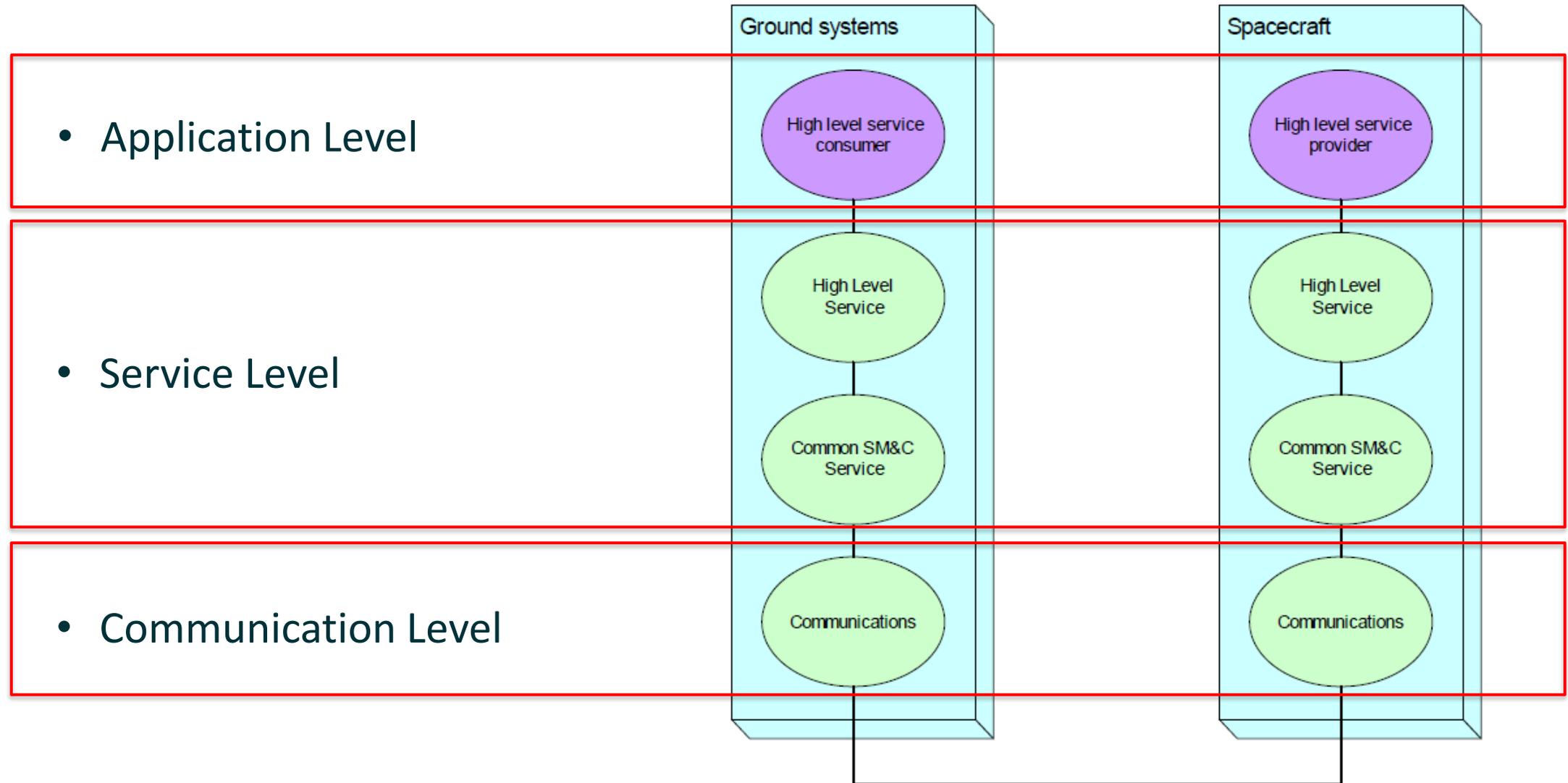
- Aerospace mission and system design
- Space system Assembly integration
- Functional verification
- Technology development

Main facilities:

- CubeSat Concurrent Design Facility
- Clean room ISO 7
- Laminar Flow Bench ISO 5
- Ground Control Station for CubeSat



Service Oriented Architectures (SOA)



Application Level



- ❑ Each Application Process constitutes a separate entity
- ❑ The applications/components rely on the service level to exchange data
- ❑ No common standard for conceiving the On-board elements

Service Level



Both ECSS Packet Utilisation Standard and CCSDS Mission Operations define the features of this layer, in terms of:

- ❑ The type of information
- ❑ The intrinsic semantic of the information
- ❑ The communication behaviours with other application processes

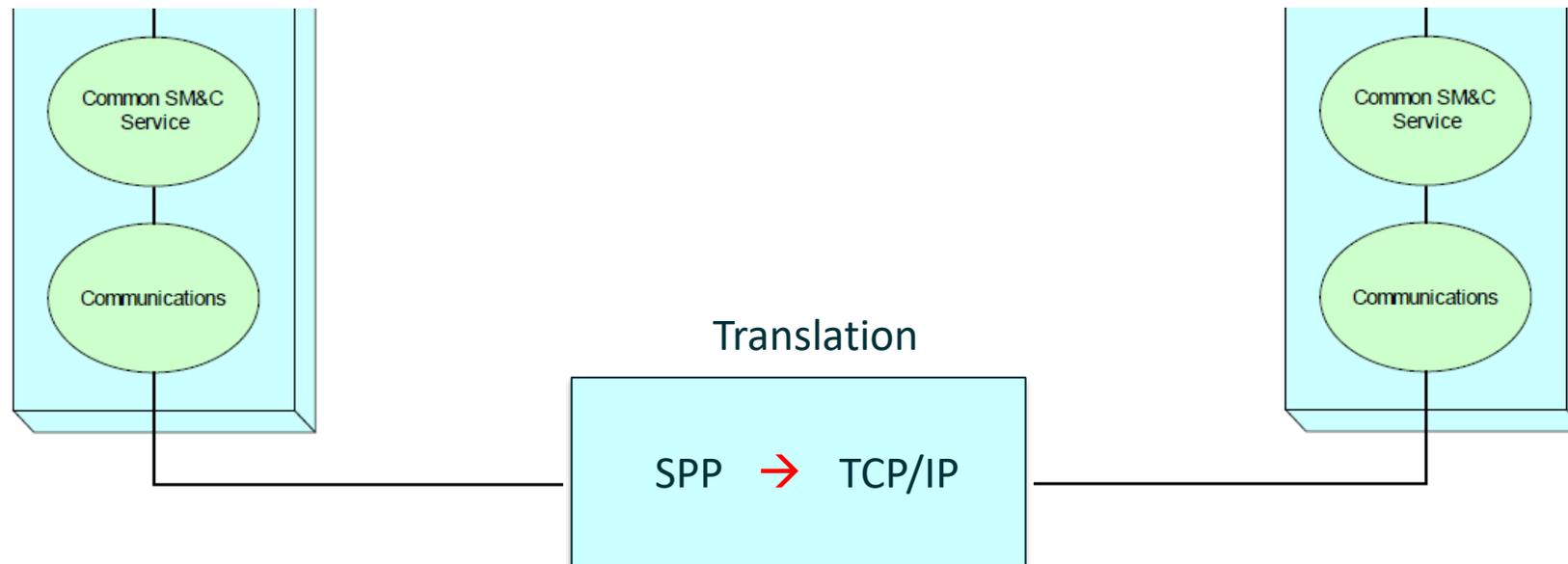
Communication Level



The Communication level allows the translation:

Service specific information
(e.g. Parameter Setting)

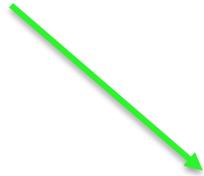
Protocol consistent structure



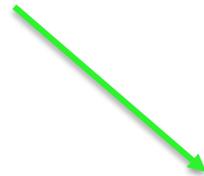
ECSS Packet Utilisation Standard (PUS)



ECSS-E-ST-70-41A



ECSS-E-ST-70-41B



ECSS-E-ST-70-41C

service type	
name	ID
request verification	1
device access	2
housekeeping	3
parameter statistics reporting	4
event reporting	5
memory management	6
(reserved)	7
function management	8
time management	9
(reserved)	10
time-based scheduling	11
on-board monitoring	12
large packet transfer	13
real-time forwarding control	14
on-board storage and retrieval	15
(reserved)	16
test	17
on-board control procedure	18
event-action	19
parameter management	20
request sequencing	21
position-based scheduling	22
file management	23

Note: The reserved service type identifiers were used in previous versions of this Standard. This Standard no longer promotes the use of these service types but does not preclude that existing implementations are reused for new missions.

ECSS Packet Utilisation Standard (PUS)



- The service message structure reflects the semantic of the service-specific information

PUS Data Types:

- Boolean
- Enumerated
- Unsigned Integer
- Signed Integer
- Real
- Bit-string
- Octet-string
- Character-string
- Absolute Time
- Relative Time
- Deduced
- Packet

Set Parameter Value Service:

- How many parameters?
- Which parameters?
- To which value?

repeated N times

N	parameter ID	value
unsigned integer	enumerated	deduced

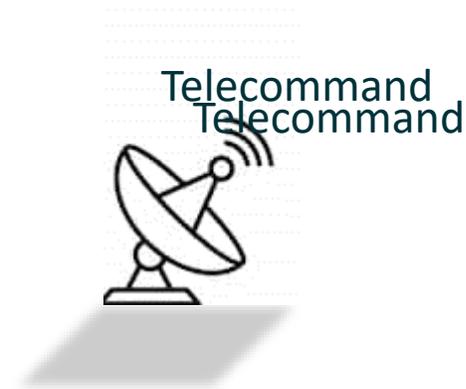
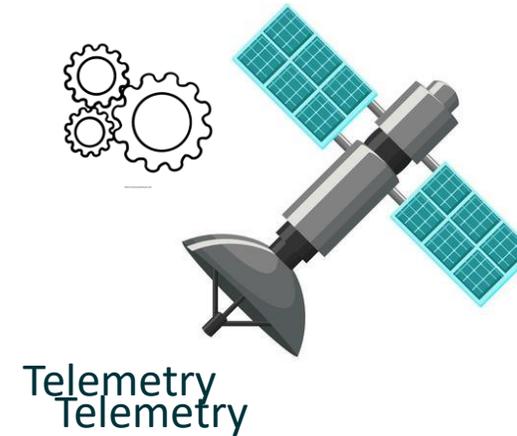
NOTE The format of the value field is specific to the parameter identified by the associated parameter ID field.

ECSS Packet Utilisation Standard (PUS)



Marked distinction between:

- ❑ On-board processes
- ❑ Ground-based processes



ECSS Packet Utilisation Standard (PUS)



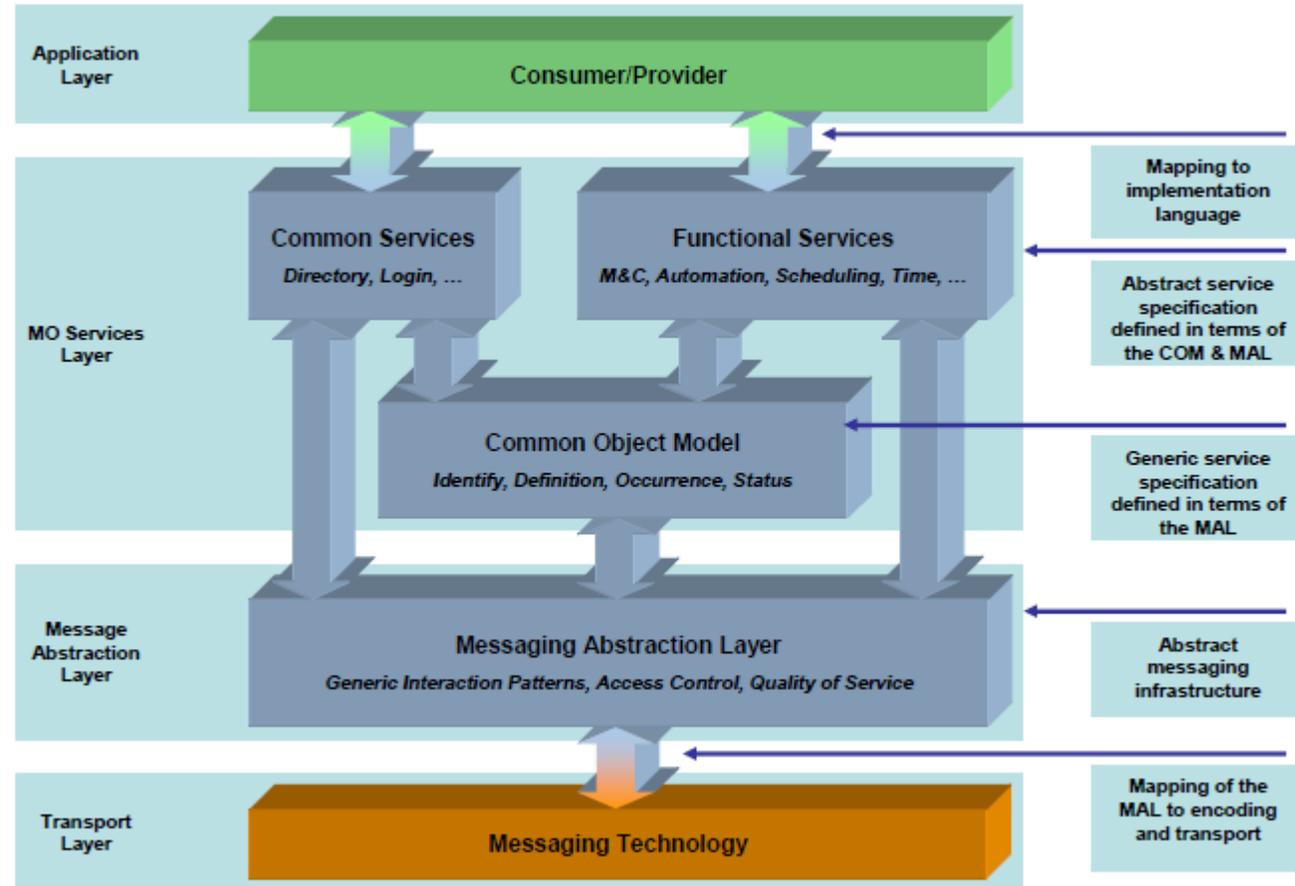
This Standard promotes using space packets compliant to the CCSDS space packet protocol to transport the PUS messages. It does not prescribe the protocol used to transport requests initiated on-board and reports destined for on-board.

- ❑ A "**telecommand packet**" is the data unit that is used to carry a service request from an application process on the ground to an application process on-board
- ❑ A "**telemetry packet**" is the data unit that is used to carry a service report from an application process on board to an application process on the ground.

CCSDS Mission Operations (MO)



- ❑ Reduced heritage if compared with PUS
- ❑ Marked subdivision within the service layer



CCSDS Mission Operations (MO)

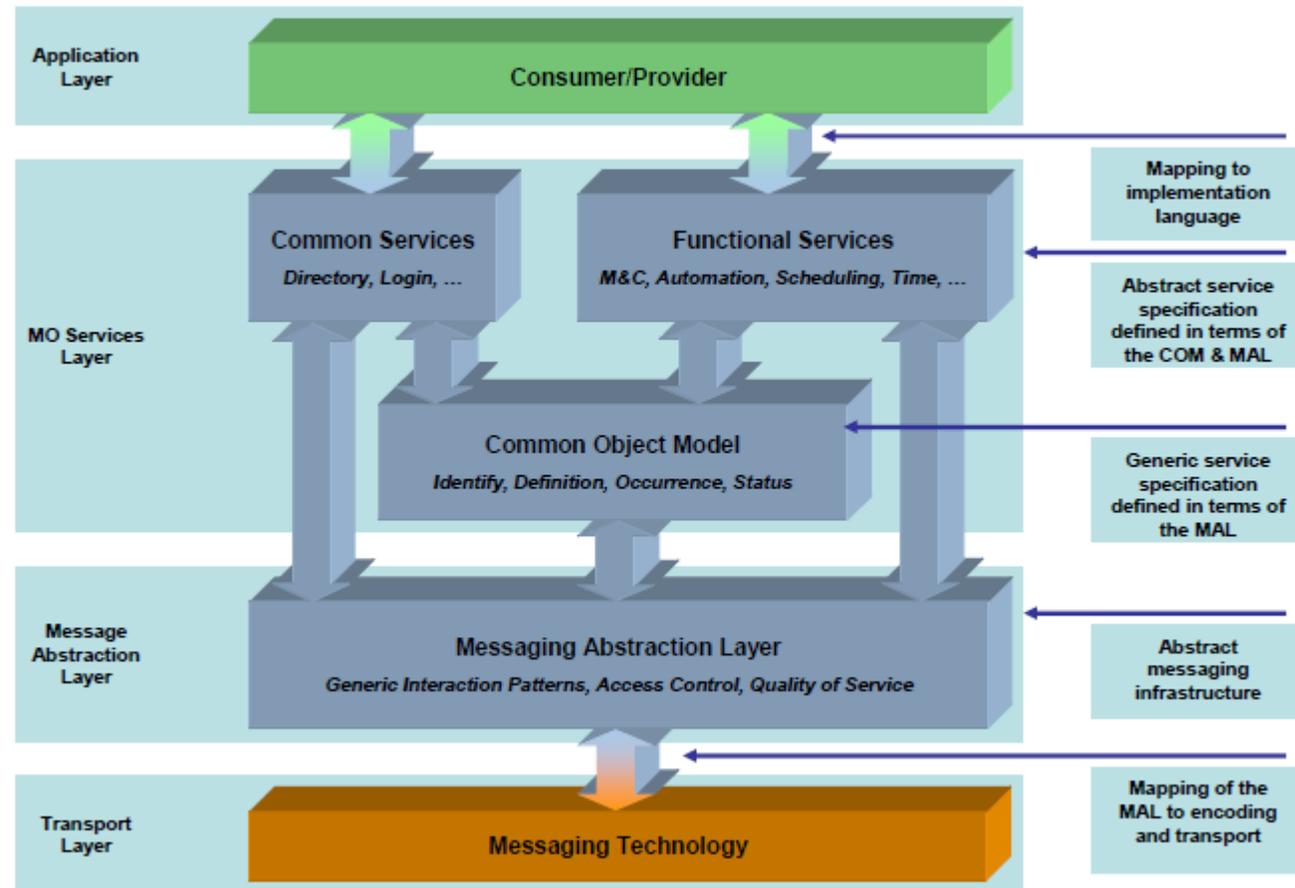


The Message Abstraction Layer (MAL) is responsible for the definition of:

❑ Interaction patterns

❑ Data types as basis for data structures

❑ Abstract definition of the message header

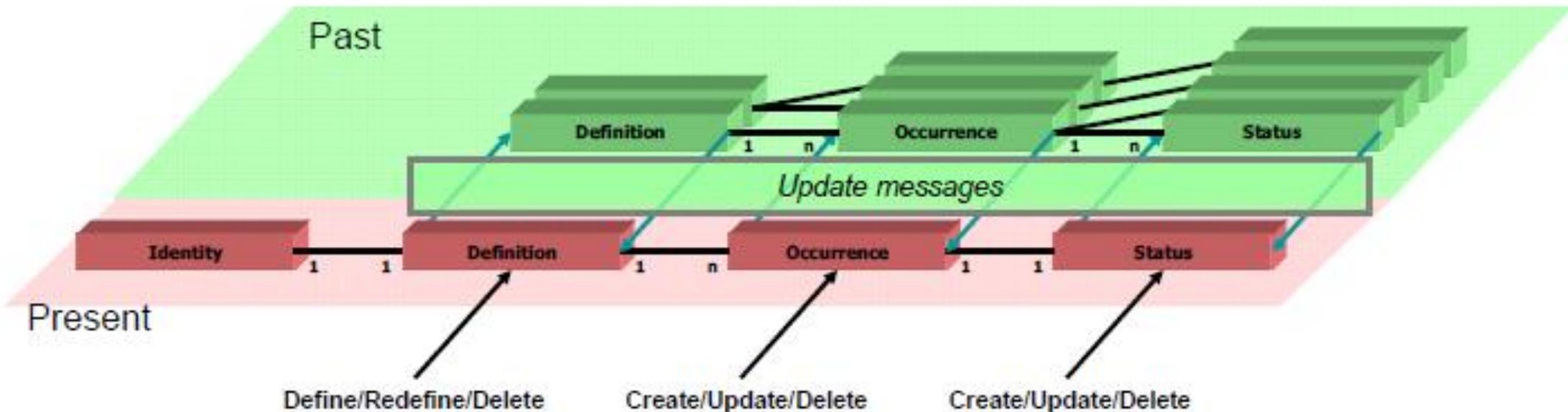


CCSDS Mission Operations (MO)



The Common Object Model (COM) defines:

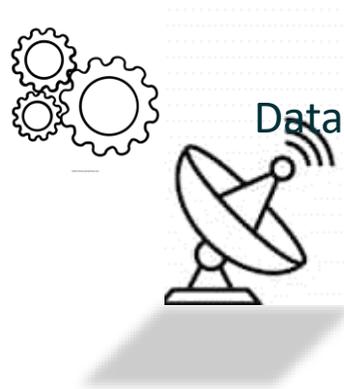
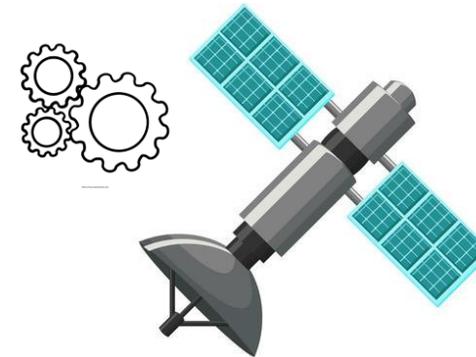
- How object refer to each other
- The rules for managing service objects



CCSDS Mission Operations (MO)



- ❑ No distinction between on-board and ground-based processes
- ❑ Interaction patterns composed by message sequences



CCSDS Mission Operations (MO)



❑ Each service specification defines its data structures

❑ The data structure rely on the MAL data types

Operation Identifier	setValue					
Interaction Pattern	SUBMIT					
Pattern Sequence	Message	List	Field	Body Type	Field	Body Type
IN	SUBMIT	List	newRawValues	ParameterRawValue	paramInstId	MAL::Long
					rawValue	MAL::Attribute

CCSDS Mission Operations (MO)



Service message
structure →

Field	Type	Value
URI From	URI	Message Source URI
Authentication Id	Blob	Source Authentication Identifier
URI To	URI	Message Destination URI
Timestamp	Time	Message generation timestamp
QoSlevel	QoSLevel	The QoS level of the message
Priority	UInteger	The QoS priority of the message
Domain	List<Identifier>	Domain of the message
Network Zone	Identifier	Network zone of the message
Session	SessionType	Type of session of the message
Session Name	Identifier	Name of the session of the message
Interaction Type	InteractionType	Interaction Pattern Type
Interaction Stage	UOctet	Interaction Pattern Stage
Transaction Id	Long	Unique to consumer
Service Area	UShort	Service Area
Service	UShort	Service
Operation	UShort	Service Operation
Area version	UOctet	Areaversion
Is Error Message	Boolean	'True' if this is an error message; else 'False'

Space Packet Protocol

TCP/IP

HTTP

..

Conclusions



ECSS PUS:

- ❑ High heritage
- ❑ Limited standardisation span
- ❑ Too prescriptive in single parts
- ❑ Heavy documentation

Which leads to:

- ❑ Difficulties in adaptation
- ❑ Need for very high expertise

CCSDS MO:

- ❑ Limited presence in the market

Independency from:

- ❑ Transport protocol
- ❑ Implementation language

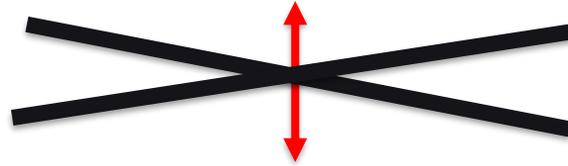
Which allows:

- ❑ Migration of platform
- ❑ Software reuse
- ❑ Separation of concerns

Next Steps



- Application Level



- Service Level: Standard service interface

Service-Protocol Mapping



- Communication Level:
SPP, HTTP, TCP/IP

Implementation into a specific programming language



Thank You