CCSDS Mission Operations

Sam Cooper (SciSys), Mario Merri (ESA)

Overview

- Primer
- Integration with on-board architectures
- MO Services
  - Mapping to and from PUS
- Status and Future




Primer

Packet Utilisation Standard

- The PUS continues to serve us well and will also be improved in the next update, However:
  - Only used in Europe
  - Only really extends as far as the edge of the MCS
  - Fixed to CCSDS Space Packets

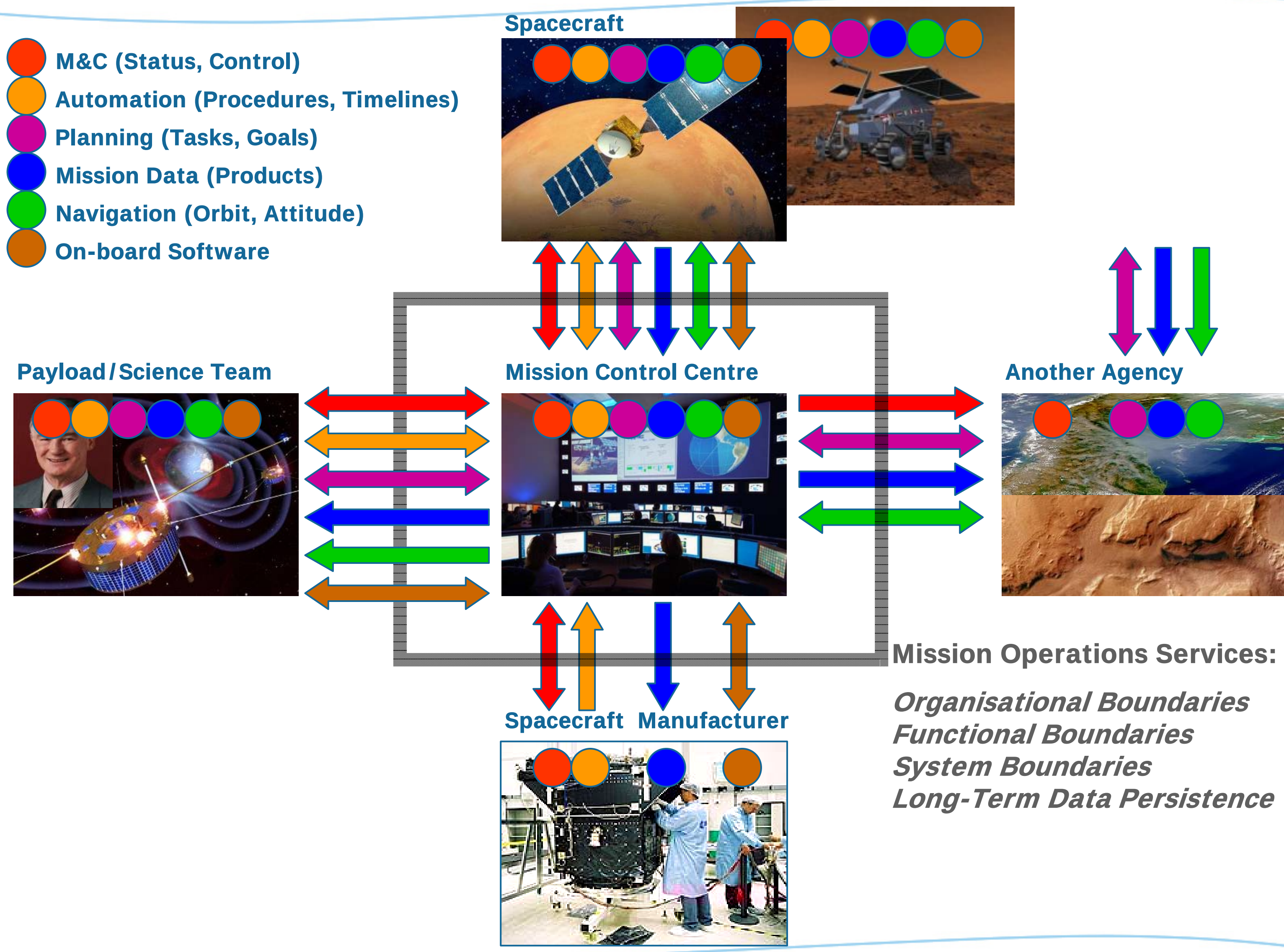
- The CCSDS Spacecraft Monitor & Control WG is defining a standard that
  - Extends from the onboard applications right through the ground systems to the end users
  - Is compatible with all CCSDS Agencies
  - With support of standardised architectures such as SAVOIR-FAIRE will provide a single solution for the M&C of any spacecraft by any agency

  - Known as the CCSDS Mission Operations (MO) Services

Relationship to PUS

- MO Services are fundamentally based on PUS
  - Refactored to make them self-consistent and transport independent
- MO Services expand PUS Services
  - MO Services cover more functions than PUS
- MO Services improve PUS Services
  - The specifications are independent of transport and encoding technology
    - A Message Abstraction Layer (MAL) ensures this
  - They are design to be used on-ground, on-board and across the spacelink

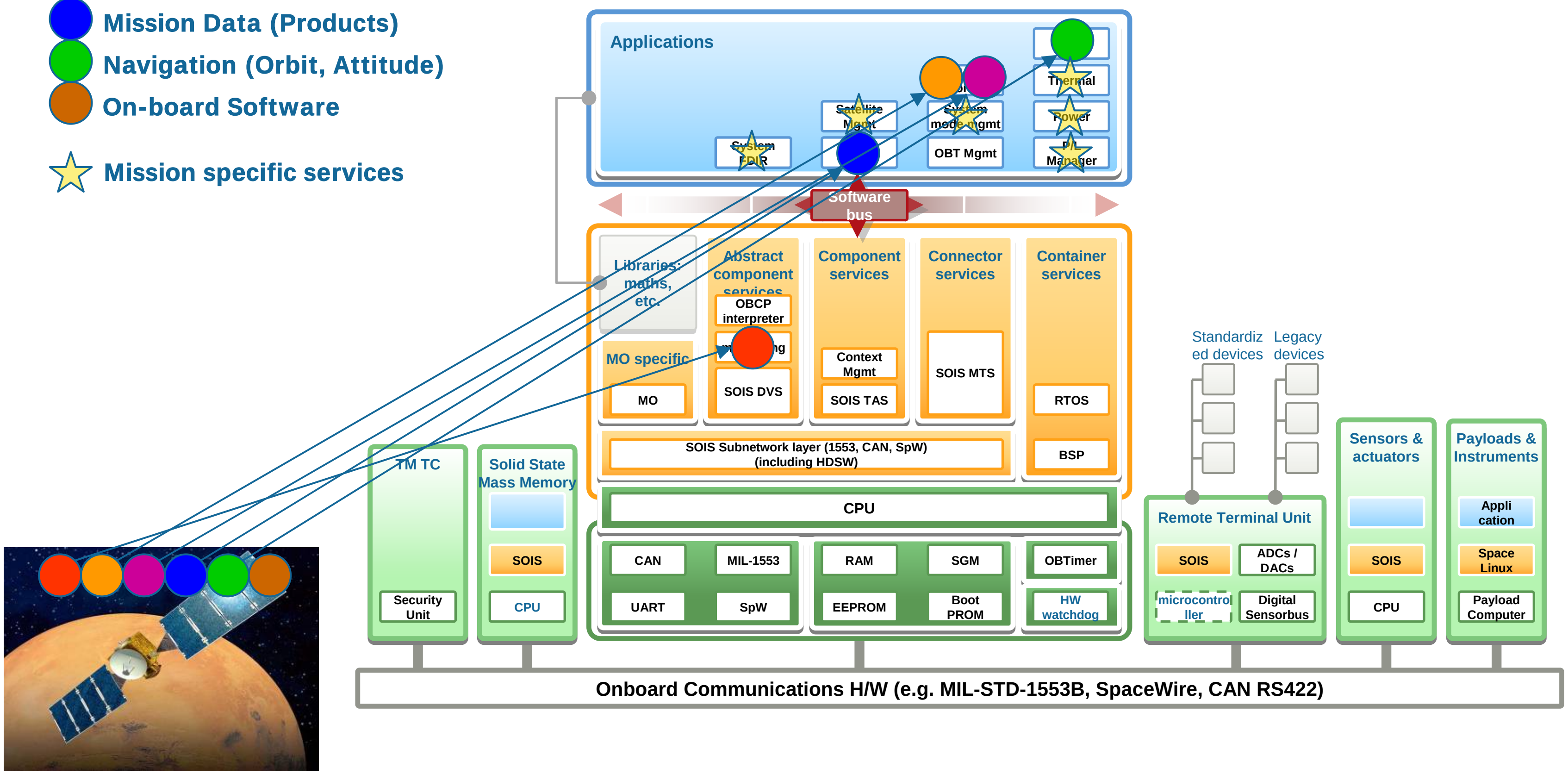
Distributable Mission Operations Functions



Distributable Mission Operations Functions

- M&C (Status, Control)
- Automation (Procedures, Timelines)
- Planning (Tasks, Goals)
- Mission Data (Products)
- Navigation (Orbit, Attitude)
- On-board Software

☆ Mission specific services



Technology Independence

- Service specifications defined in XML
  - Provides a machine readable version of the service specifications

- Standardised mappings define transformations from the MAL representation
  - Language mappings for specific programming languages
  - Technology mappings for ‘on-the-wire’ transports/encodings
    - Such as SOIS MTS
  - Private mappings are also supported

- Mappings are not service specific they work for all services
  - Services are defined in terms of the MAL
  - Mappings are defined in terms of the MAL

- Code generators can be used to auto-generate service APIs
  - Standard transformations from the XML to languages are defined
  - Allows high level APIs for applications

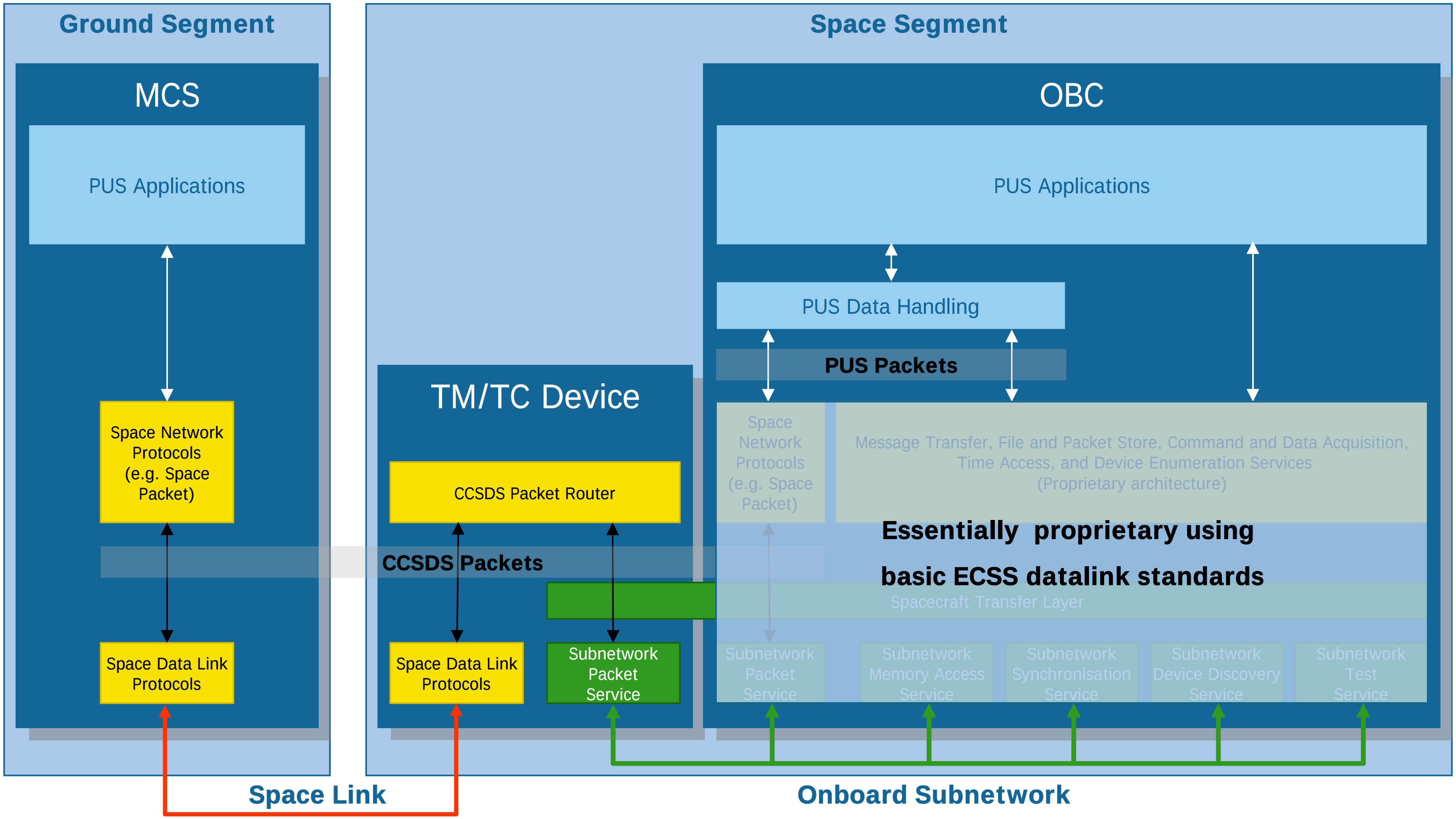



Integration with on-board architectures

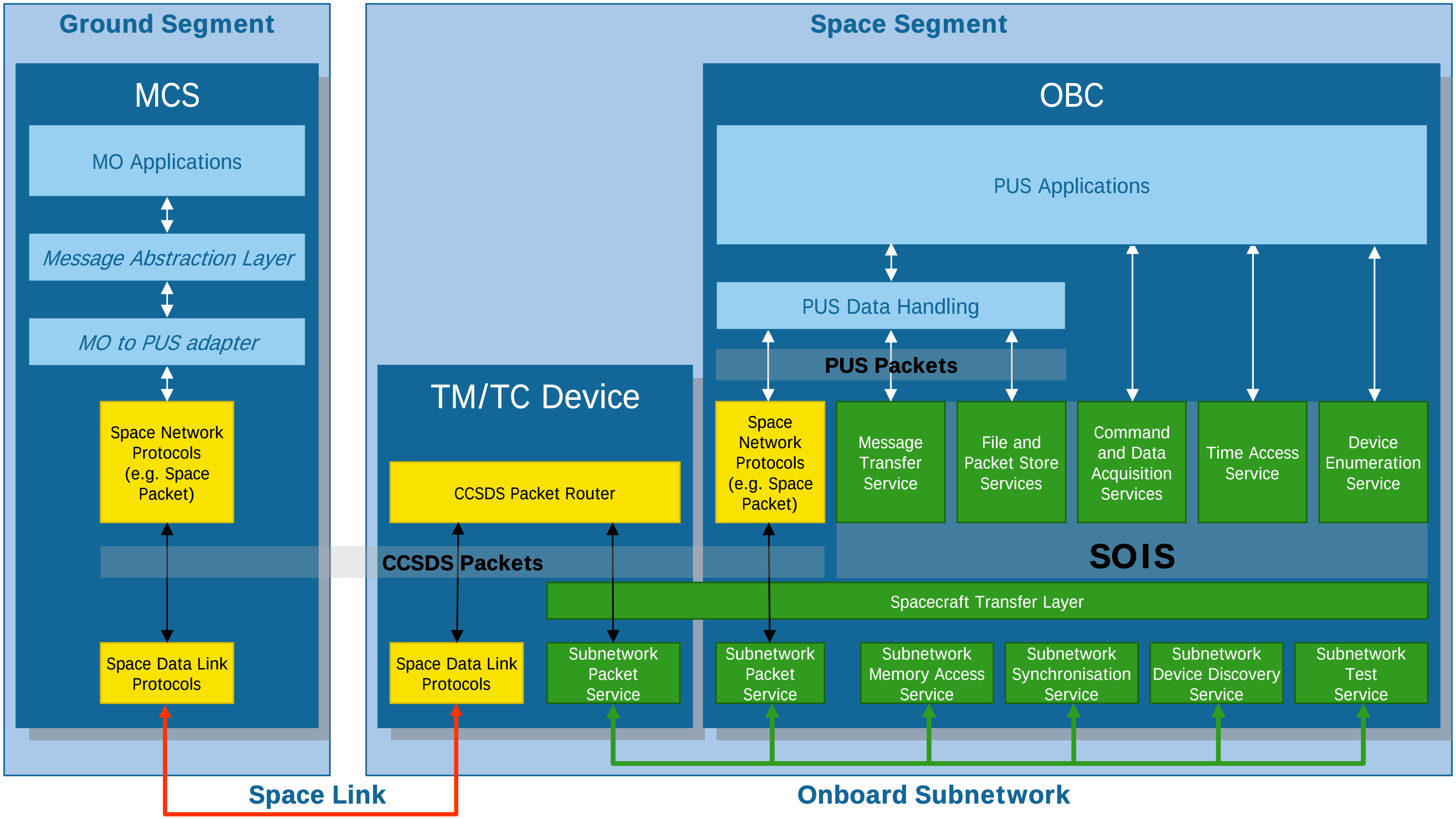
Implications of Onboard MO Implementation

- The layered approach of MO aims to reduce implementation complexity
  - Enables development of components that can be reused
  - To be effective requires
    - Appropriate architecture and infrastructure design
    - Enforcement of policies to strictly adhere to standards
  - But does NOT come at the cost of efficiency
    - Layers are conceptual
    - Code auto-generation can merge layers and optimise code for selected target platform
- Establishment of a widely accepted and supported reference architecture and API's for onboard SW does help
  - SAVOIR-FAIRE

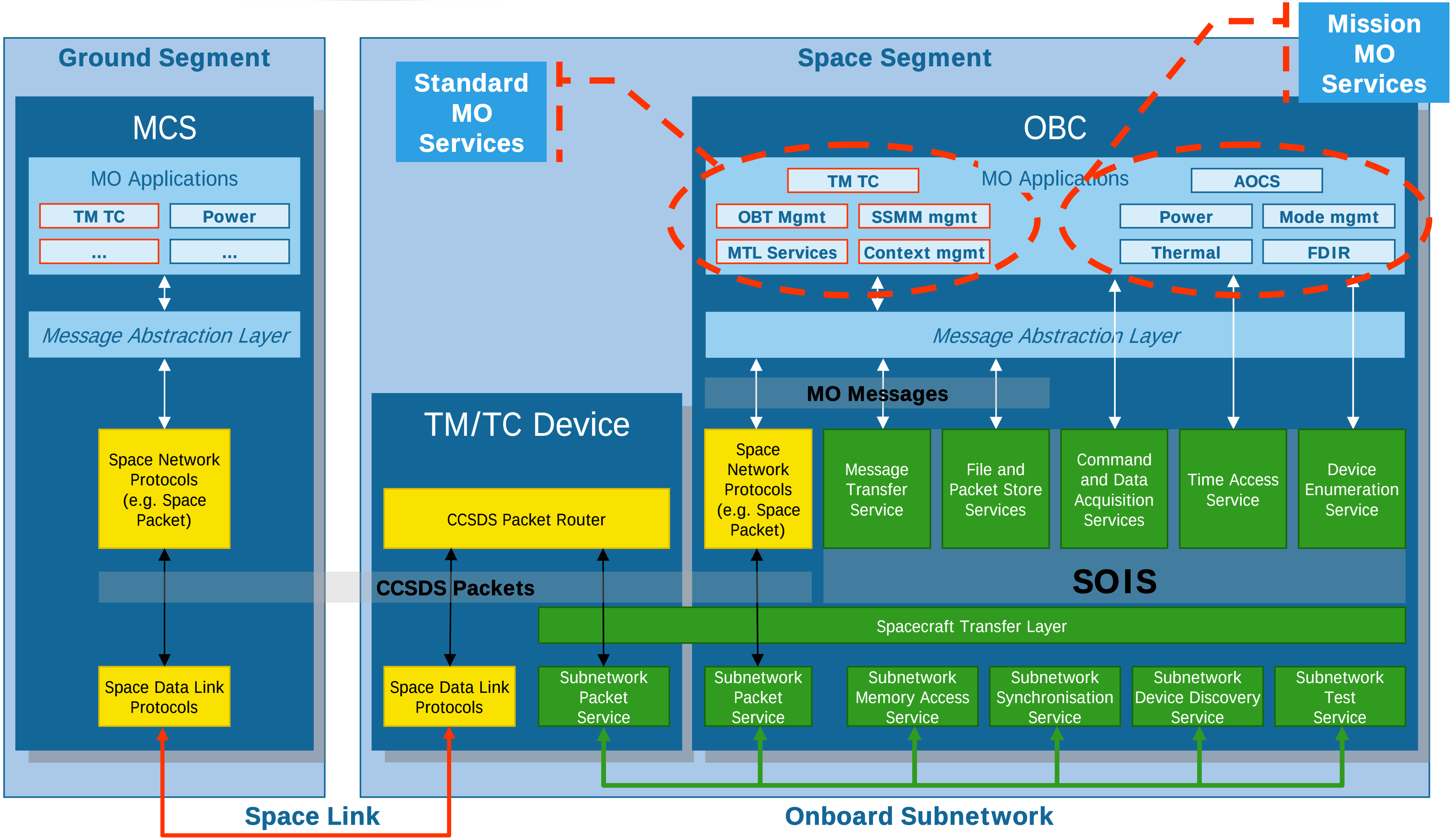
Current architecture



Transitional architecture



Future architecture






CCSDS MO Services

Candidate MO Services

- Operationally meaningful information exchange:

Status [Monitoring Parameter] Control Directive Event Notification	M&C
Automated Activity Scheduled Timeline or Activity List	Automation
Planning Request	Planning
Trajectory and Pointing Predicted Events	Navigation
Mission Product	Data Products
Time	Time
On-board Software Image	Remote Software

  - Identified and prioritised by CCSDS space agencies

PUS to MO Service mapping

	PUS Service	MO Service
1	Telecommand verification	COM / Activity
2	Device command distribution	M&C / Action
3	Housekeeping & diagnostic data reporting	COM / Status
4	Parameter statistics reporting	M&C / Statistics
5	Event reporting	COM / Status
6	Memory management	*Software management*
8	Function management	*Automation*
9	Time management	*Time*
11	On-board operations scheduling	*Scheduling*
12	On-board monitoring	M&C / Check
13	Large data transfer	*Data product management*
14	Packet forwarding control	*Remote buffer management*
15	On-board storage and retrieval	*Remote buffer management*
17	Test	M&C / Action
18	On-board operations procedure	*Automation*
19	Event-action	*Automation*

Service only planned by CCSDS, but not yet developed

MO to PUS Service mapping

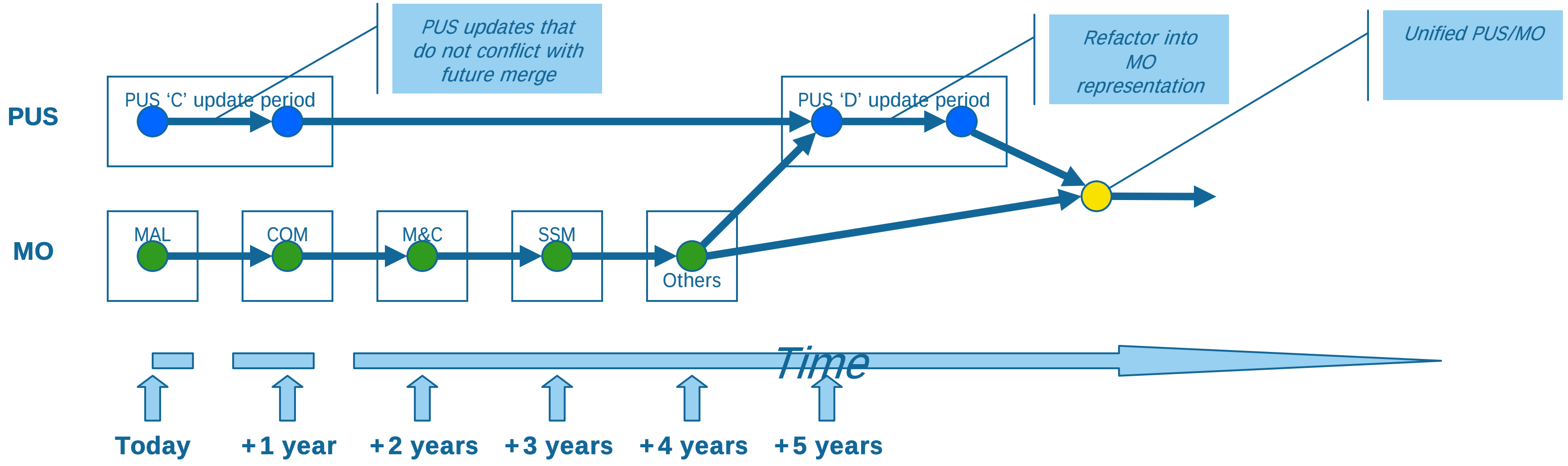
MO Service	PUS Service
Monitoring and Control	Telecommand verification / Device command distribution / Housekeeping & diagnostic data reporting / Parameter statistics reporting / Event reporting / Function management / On-board monitoring / Test
Time	Time management
Software Management	Memory management
Planning	
Scheduling	On-board operations scheduling (subset of MO service)
Automation	On-board operations procedure (subset of MO service) Event-Action
Data product management	Large data transfer (subset of MO service)
Navigation	
Remote Buffer management	On-board storage and retrieval Packet forwarding control
Common Services: *Directory, Login, Interaction, Replay, Configuration*	

Service only planned by CCSDS, but not yet developed

Proposed MO/PUS Unification Roadmap

- The timescale of the two standards are very different
  - PUS has a revision ‘C’ due next
  - PUS has a revision ‘D’ due 5 years after that
  - MO is expecting to produce our first version of the M&C service in the next 5 years
  - At this point the two may be able to unify:
    - PUS gets the MO protocol independence
    - MO gets flight proven PUS services

- This leads to the MO view of the roadmap:



Status and Future

CCSDS SM&C Working Group

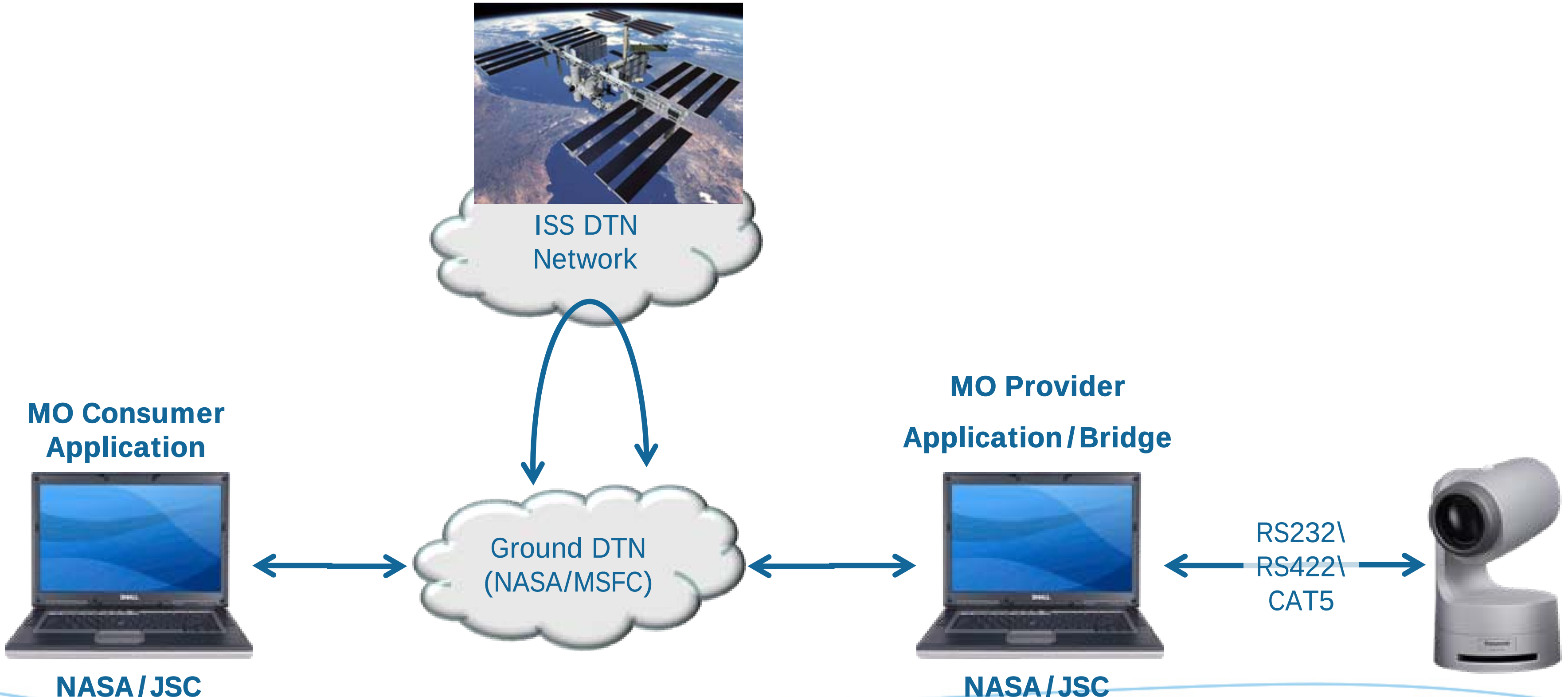
- 8 year lifetime (started in Dec 2003)
- ESA lead activity with excellent team work among agencies!
- 10 Space Agencies actively involved
- Very active (15 workshops, 60+ telecons)
- Quite productive (... and ready for more)
  - **Green Books**
    - 2 published (XTCE, MO)
    - 1 under preparation (XTCE Core)
  - **Blue Books**
    - 2 published (XTCE, MAL)
    - 3 under finalisation (COM, M&C, Common)
    - 1 under preparation (Space Packet Binding)
  - **Magenta Books**
    - 1 published (RM)
    - 1 under finalisation (Java API)
    - 2 under preparation (C++ API, XTCE CCSDS)
  - **Yellow Books**
    - 1 published (MAL testing)
  - **White Books**
    - several in early draft

Prototypes

- **Prototype 1 (ESA/CNES)** Goal: validation of MAL specification as required by CCSDS
  - Results: Done! Complete automation of 16840 individual tests, the two implementations interoperated perfectly!
- **Prototype 2 (NASA/JSC)** Goal: validation of MAL + M&C & Common services
  - Results: Validity of MAL concept proven (3 different underlined technologies used: SOAP/Java, AMS/C, AMS/Python).
- **Prototype 3 (CNES)** Goal: evaluate feasibility of migration of the CNES mission control system infrastructure, Octave, to MO services and performance
  - Results: Migration possible and economically convenient. Performances depends on architecture, but in general satisfactory (14,000 parameters/s in typical Octave operational configuration)
- **Prototype 4 (Eumetsat)** Goal: Stack implications and performances
  - Results: Several configurations (Point-2-Point, Pub/Sub, packet size, 1-n clients) and communication technologies (RMI, JMS, AMQP (two broker implementations: Java, C++)) were tested and compared. Performances depend on configuration (range 35,000-2,000 parameters/s). The MAL layer does not add noticeable overhead
- **Prototype 5 (DLR-NASA/JSC)** Goal: concept and feasibility of MO interoperability between DLR's MCS and JSC Simulator.
  - Results: All tests have successfully completed. Overall objective of an interoperability test where DLR monitor and control a Simulated NASA spacecraft was successful.
- **Prototype 6 (CNES)** Goal: Validation of the M&C service over CCSDS Space Packets
  - Results: The prototype is still on-going, but so far very successful. Please refer to the paper "Space Packet encoding : Reduce the design effort to zero?" included in the proceedings of SpaceOps 2010
- **Prototype 7 (ESA)** Goal: Validation of the MAL using Web Services specifications
  - Results: The prototype is still on-going.

Prototypes

- **Prototype 8 (NASA / JSC)**
  - Goal: Prototype a camera service leveraging the CCSDS integrated protocol stack (MO/AMS/DTN) via the ISS
  - Results: the first step has successfully completed. Many lessons learned about the relevant CCSDS protocols and their use together. Feedback being provided to relevant working groups.



Future developments

- **Implementation 1 (CNES)** ISIS
  - ISIS is a completely new generic onboard and on ground system based on PUS, SSM, and MO. Specification is expected to start in 2012 with development starting in 2013.
- **Implementation 2 (ESA) Ground SW infrastructure**
  - ESOC Service Management Framework will be ported to the MO service framework as soon as it is available.

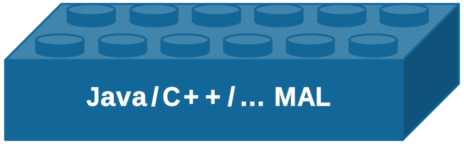
On-board software

- **Implementation 3 (ESA) SAT-OPS IOD**
  - Proposal from ESOC to deploy demonstration MO applications on the SAT-OPS In Orbit Demonstrator CDF activity.

Future plans

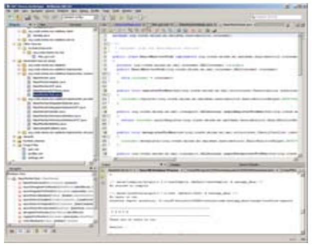
- The strategy is that a future version of the PUS and MO will unify to form a single coherent standard
  - MO brings a layered, technology agnostic, structure
  - PUS has the flight proven service set
  - Aligned with SOIS specifications

- Make available reference implementations of MO MAL



- Make available auto-generators for:
  - Java
  - C/C++
  - MS Word
  - CCSDS XTCE
  - …

- Produce tools that support development of service specifications






End




Backup slides

CCSDS Specification Status

- MO Concept GB (ESA): published
- MO Reference Model MB (ESA): published
- MO MAL BB (ESA): published
  - can be downloaded free-of-charge from www.ccsds.org

BB: Blue Book
GB: Green Book
MB: Magenta Book

- Java API MB (CNES): completed Agency review

- COM BB (ESA): in Agency review

- MO M&C Service BB (ESA): Agency review and prototyping commencing Q1 2012
- MO Common Service BB (ESA): Agency review and prototyping commencing Q1 2012

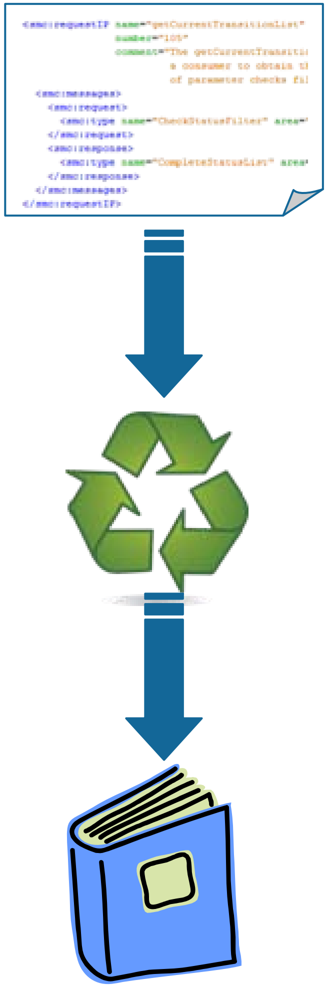
- C++ API MB (NASA/JSC): in preparation
- CCSDS Space Packet mapping BB (CNES): in preparation (Q2 2012)
- CCSDS AMS and DTN mapping BB (NASA/JSC): in preparation

Perceived Complexity of the Standard

- MO concept implies use of multiple layer specific specifications
  - Implies a relatively high level of abstraction in individual specifications
  - Not easy to see the “full picture” when studying one document
  - Not easy to understand what information is actually included in the data structures transferred

- With proper infrastructure in place, operators would not need to concern themselves with all details
  - Requires a shift in mindset, which may be difficult and take time

- It is possible to generate tailored documentation from the XML
  - This is just another output option of the auto-generation tools
  - It can be very PUS like if so desired, it is just a ‘display’ choice



Example: An MO Operation …

- An operation in the MO XML format:

```
<smc:requestIP name="getCurrentTransitionList"
               number="105"
               comment="The getCurrentTransitionList operation allows
                        a consumer to obtain the status of a number
                        of parameter checks filtering on check state.">
  <smc:messages>
    <smc:request>
      <smc:type name="CheckStatusFilter" area="MC" service="Check"/>
    </smc:request>
    <smc:response>
      <smc:type name="CompleteStatusList" area="COM" service="COM"/>
    </smc:response>
  </smc:messages>
</smc:requestIP>
```

Example: An MO Operation …

- Is used to generate tables in the MO specifications:

OPERATION: getCurrentTransitionList

General

The getCurrentTransitionList operation allows a consumer to obtain the status of a number of parameter checks filtering on check state.

Operation Name	getCurrentTransitionList	
Interaction Pattern	REQUEST	
IP Sequence	Message	Field Type
IN	Request	CheckStatusFilter
OUT	Response	CompleteStatusList

Example: ... in PUS style

- But it is simple to generate alternate representations from the XML:

Operation: getCurrentTransitionList

The getCurrentTransitionList operation allows a consumer to obtain the status of a number of parameter checks filtering on check state.

Telecommand application data:

checkFilter	stateFilter
OccurrenceKeyList	CheckStateList

Response telemetry report application data:

N	element
Integer	CompleteStatus
	⊢ Repeated N times ⊣

Onboard Implementation Complexity

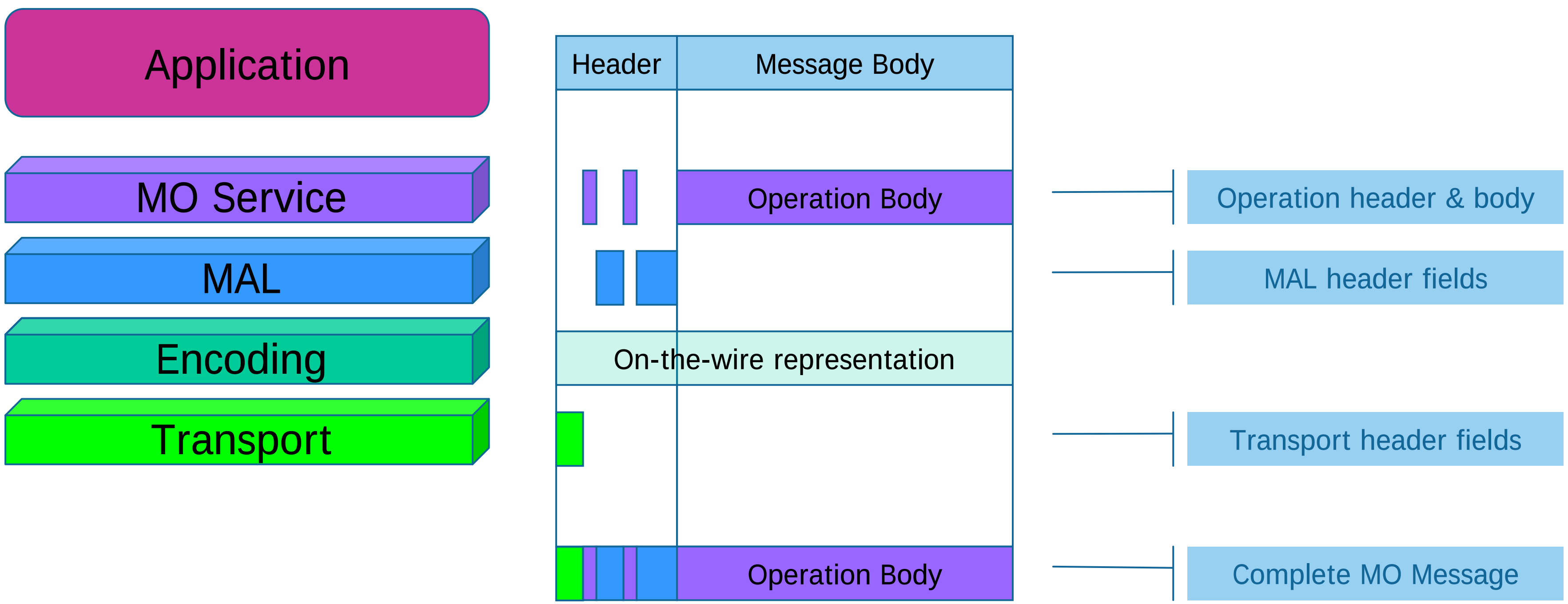
- Auto-generation of interface code from service specifications will help
  - Feasible as MO service specifications provided in XML

- Use in on-board implementation needs to be considered
  - Pros
    - Minimises errors when frequent changes are made during development lifecycle
    - Auto generation of mission databases and documentation
  - Cons
    - Still requires validation of the generator or generated code

- This is currently done internally at SciSys for on-board projects using a proprietary technology
  - Standardisation of this would increase the reuse benefits
  - Experience gained from this should be taken into account

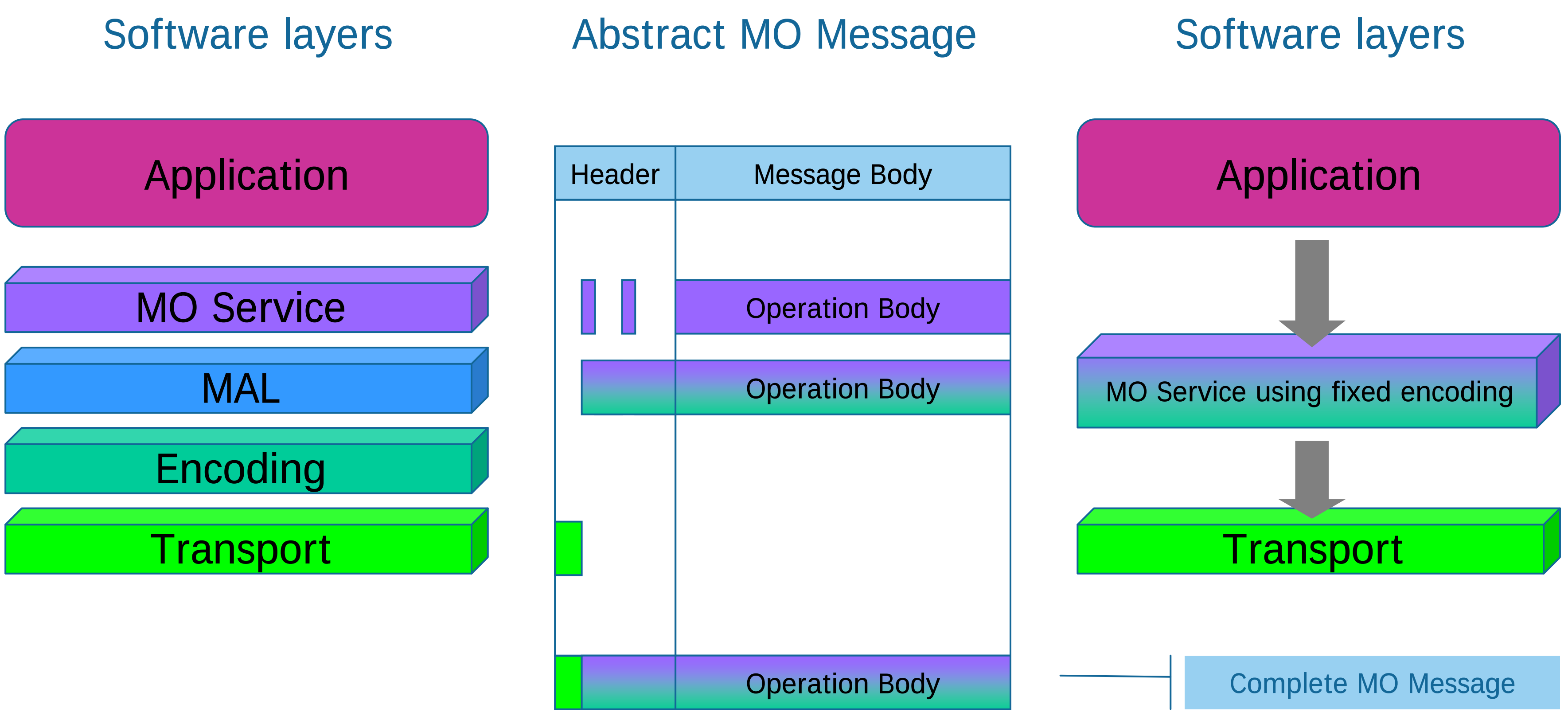
How is an MO Message built?

Software layers | Abstract MO Message | Layer provides



- The more options you can fix the more you can optimise

For a fixed encoding…



- The more options you can fix the more you can optimise

Efficiency of Bandwidth Usage

- Current MO Service Specifications
  - are considered verbose
  - data structures include a superset of all information
  - data encoding efficiency has been “second priority”

- CNES are currently defining a mapping of MO to CCSDS Space Packets
  - From the abstract model of MO to an efficient binary encoding
  - Also maps to using CCSDS Space Packets as a transport
  - Uses context information to optimise the information actually carried on the space link
  - The goal is near PUS efficiency

Relationship to the Space System Model

- Plan is to produce an SSM-like specification for system modelling
- Would closely follow ECSS SSM concepts
  - ECSS SSM model is good but PUS specifics preclude its direct use
- Currently looking at using XTCE, maybe with extensions
  - XTCE and ECSS SSM are well aligned conceptually
    - Extensions to XTCE may be required to capture the high level system information that an SSM requires

Technical Characteristics

- Initially based on the PUS for service specification but abstracted
- Services defined in terms of a set of Operations:
  - Each operation is a “conversation” between Consumer and Provider
  - The pattern of the exchange are common to many Operations
  - Generic Interaction Patterns simplify specification
- Message Abstraction Layer (MAL) provides 6 Patterns:
  - SEND, SUBMIT, REQUEST, INVOKE, PROGRESS, & PUBLISH-SUBSCRIBE
- MAL also gives independence from underlying representation
  - Concerned with information exchange rather than ‘bits and bytes’