CCSDS MO Training

Module 2 - Medium

Overview

- Introduction
- Part 1 - Primer
  - Introduction to Mission Operations Services
  - Goals and Benefits
- Part 2 – MO Framework and Services
  - The MAL and COM
  - Common services
  - Monitor and Control services
  - Forthcoming services
- Part 3 – A worked example
  - A simple mission
  - Service boundaries
  - MO and our mission
- Summary

Introduction

- CCSDS and MO
- About the presenter
- Aims of the training module

CCSDS and the MO Standards

Images © the relevant Agency

Mission Operations

About the presenter

Aims of the training module

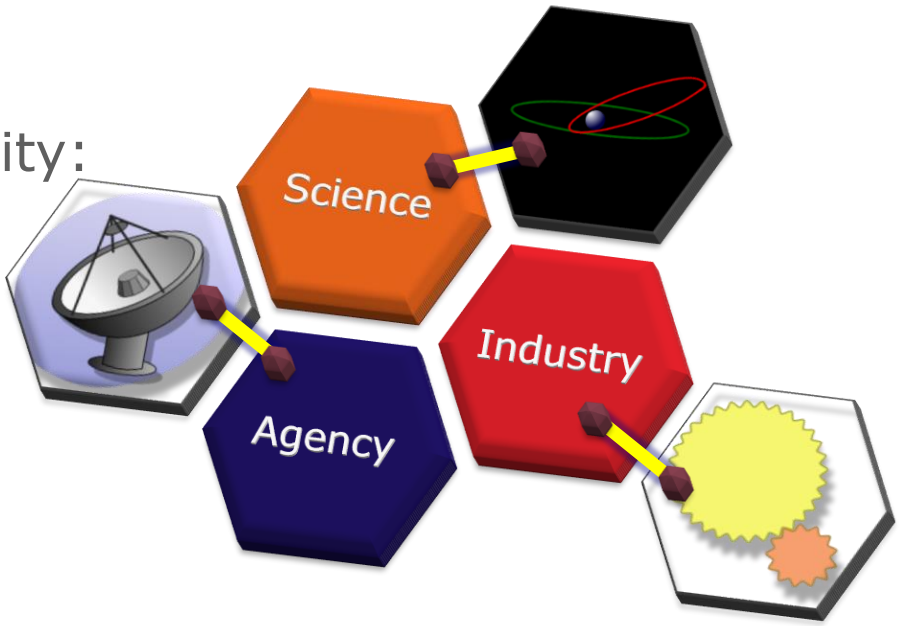
- The medium level training module is a partial follow on module from the Primer
  - It will cover some of the same items as the primer but only as a recap
- The module will focus more on the:
  - An overview of the MAL and COM layers
  - Services defined by MO
  - Discuss their intended usage and how they can be used together to facilitate operational needs
- It will also cover other envisaged MO Services such as:
  - Mission Data Product Distribution Service
  - Mission Planning and Scheduling Services
  - Automation Service
  - File Management Service
- The aim of the training is that attendees leave feeling that
  - They can see how the services fit together
  - How they could be used in, or to support, their area

Part 1

- Primer

Issues that exist today

- Two main trends are becoming apparent in current and future missions
- An increase in mission complexity:
  - More organisations
  - More functions
  - More interfaces
  - More technologies
- With an increasing pressure to reduce costs



Issues that exist today

- In today's systems a lack of
  - Standardisation between and inside organisations
  - Combined with lack of re-use between missions
  - Has led to increased cost of
    - Development
    - Deployment
    - and operator training
- Whilst over the long lifetimes of space missions the inability to
  - Update technologies and infrastructure
  - Replace systems without major system redesign
    - Including vendor lock in
  - Has led to increased operational costs
- With the expected growth in mission complexity it is clear that future budget pressures are going to rise

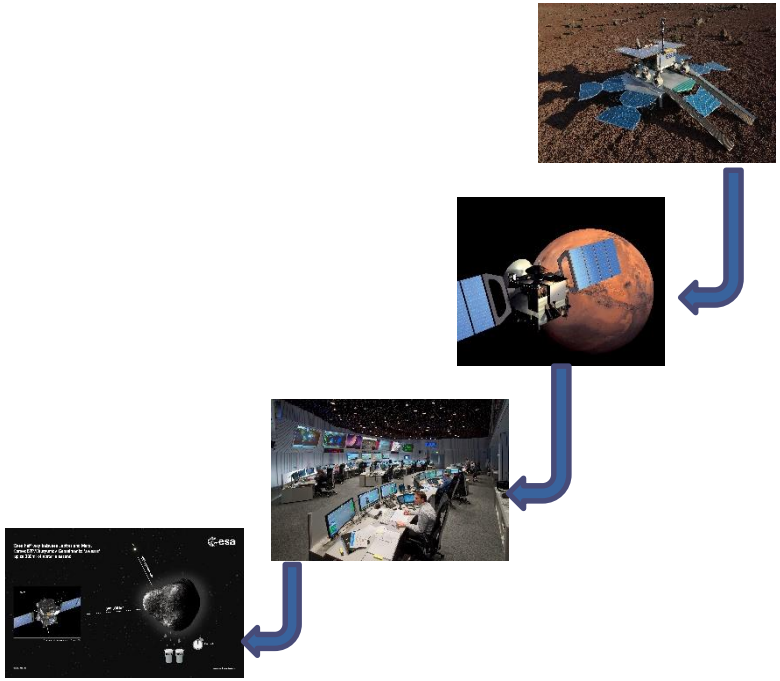
Mission Operations

CCSDS Mission Operations Services

- To alleviate this, CCSDS is standardising a set of services for Mission Operations
  - These services define a single specification for the exchange of similar information
- To support these standardised services CCSDS has also defined an open architecture and framework that is:
  - Independent from technology
  - Able to integrate new and legacy systems of different organisations
  - Designed to support the long lifetimes of space missions
  - Based on a Service Orientated Architecture (SOA)
- The CCSDS Mission Operations services provide the capability to hide the unavoidable difficulties that characterises the space environment whilst supporting future complexity needs

CCSDS Mission Operations Services

- For instance, whether Parameters are being accessed by:
  - A rover on Mars,
  - A satellite orbiting Mars,
  - Or centres on Earth
- A standard service for Parameters means:
  - A relay satellite can receive the rovers parameters and forward it to Earth
  - A Mission Control Centre can receive and analyse the Parameters of the satellite or rover
  - An external institution can receive and analyse the Parameters



- Basically, the same service can be used by completely different systems in completely different situations

Images © ESA

MO Services benefits

- Another way of putting this is that
  - By speaking a standard language
    different organisations around the world
    can interact more easily
- This leads to
  - Minimisation of the number of different interfaces for the same functionality
  - Increased cooperation between organisations
  - Increased reuse from one mission to another
  - A reduction in cost
- This improved flexibility will allow
  - Increased distribution of functions between different organisations
    - Including the migration of functions between ground and space
  - Better able to cope with technology obsolescence/longevity
- Increased interoperability and reuse means
  - Faster times for deployment
  - More innovation
  - and always more competitive costs

MO Services benefits

- CCSDS has defined an open architecture and framework that is:
  - Able to integrate new and legacy systems of different organisations
  - Service Orientated
  - Independent from technology
  - Designed to support the long lifetimes of space missions
- The framework allows different systems to communicate and interact together, independently from their
  - Location
  - Programming language
  - Hardware platform
  - Or communications technology
- This independence allows an organisation to choose
  - The technologies that are appropriate for them
  - Whilst still allowing bridging between these different choices where required
- It also increases long-term maintainability over the mission lifetime through replacement of both
  - Systems
  - And infrastructure

Mission Operations

Part 2

» MO Framework and Services

Service Orientated Architecture (SOA)

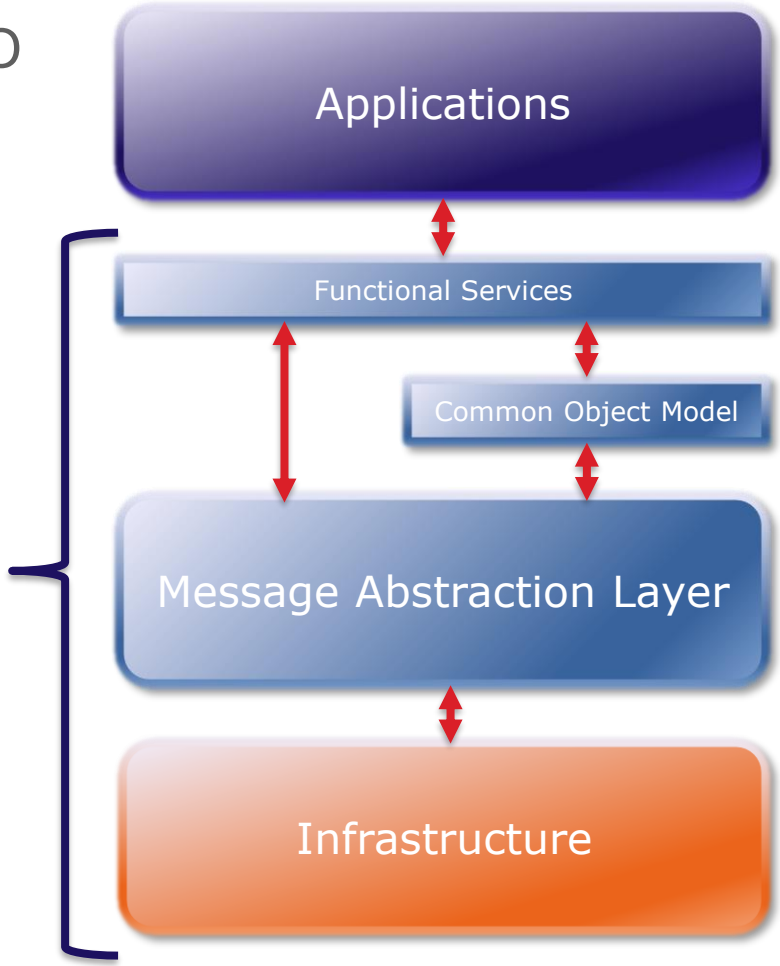
- Service orientated architecture (SOA) is an architectural pattern in which:
  - Components provide services *(Provider)*
    that are consumed by other components *(Consumer)*
    via a communications protocol over a network

- A service is a self-contained unit of functionality
  - For example, retrieving an online bank statement
- Services can be combined to provide more complex functionality

The MO Framework

- Central to the MO concept is the MO framework
  - Defines structure of an MO application
  - Provides generic model for data
    - Supports generic facilities such as archiving
  - Provides separation from technology

- It is defined by three specifications
  - A reference model
    - (CCSDS 520.1-M-1)
  - A Message Abstraction Layer (MAL)
    - (CCSDS 521.0-B-2)
  - A Common Object Model for data (COM)
    - (CCSDS 521.1-B-1)



The MO Framework and bespoke services

- MO framework also supports bespoke services
  - These are the services that you define for your:
    - Infrastructure
    - Missions
    - Agency
    - Anything else you identify

- If you define your bespoke services use the MO framework
  - In the same way the standard services provide future proof technology separation
  - The services specific to your environment can also benefit too

Message Abstraction Layer

Message Abstraction Layer (MAL)

- The Message Abstraction Layer is the building block for all MO services
- Has both a design time aspect and a runtime aspect
  - The following are design time
- Defines a formal service and data type language in XML
  - Service behaviour
  - Available operations
  - Message structures
  - A standard message header
- Defines the required abilities of communication technologies

Services

- What is a Service in the MO world?
- A service defines a functional context
  - e.g. Mission Planning is a functional context
- A Service is provided by one entity and used (consumed) by another
- How a service is accessed is defined by its service contract
  - Includes both functional and non-functional aspects of interacting with that service
  - The MO Service Specification is the service contract
- An MO Service Specification
  - Defines the expected behaviour of the service
  - Lists the contained Operations

Operations

- What is an Operation in the MO world?
  - An Operation is a functional unit that one entity implements and another invokes
  - It is triggered by the exchange of Messages
- Specified in terms of predefined Interaction Patterns
  - Standardised patterns of interaction
    - Provides a template for Service designers
    - Can be considered Operation abstraction
- An Operation Specification
  - Each operation is defined using one Interaction Pattern
  - Defines the Operations messages as required by the Interaction Pattern
  - Defines the failure conditions and failure responses as required by the Interaction Pattern
  - Details all functional aspects of that Operation

Messages and Message Abstraction

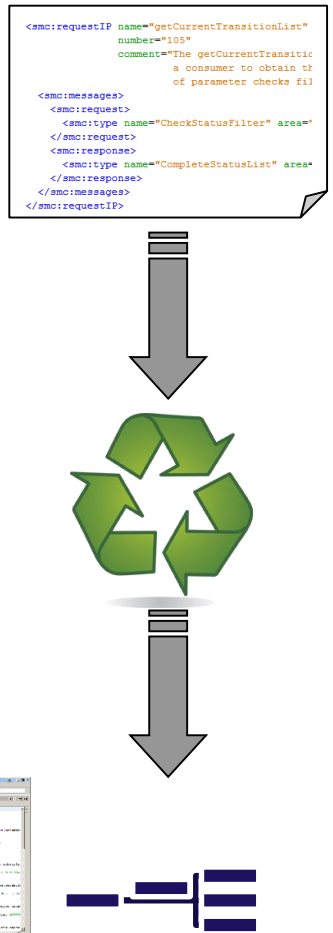
- What is a Message in the MO world?
  - A message is a unit of information passed from one entity to another
- We are concerned with Information Exchange
  - The logical content (or meaning) of the message is more important than the physical representation.
  - We can change the representation but not change the meaning
- Message Abstraction is a way of representing the message without tying it to a physical representation
  - Abstracts the representation but preserves the content
  - It could be considered pseudo code for data structures

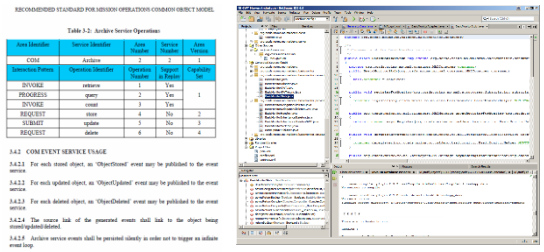
Interaction Patterns

- Standardised patterns of interaction
  - Provides a template for Service designers
  - Can be considered Operation abstraction
- Standardises
  - The sequence of messages
  - Some patterns fix certain messages (acknowledgements)
  - Failure points
  - Message header values
- Provides
  - Simplified operation specification for Service designers
  - Simplified operation implementation for system architects
  - Simplified operation behaviour for system operators

MAL mappings and transformations

- The MAL defines a standard XML notation for service specification
- Mappings define transformations from the XML to:
  - Language mappings for specific programming languages
  - Technology mappings for 'on-the-wire' transports/encodings
  - Bespoke 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
- So, from this we can automatically generate:
  - Documentation
  - Programming language APIs
  - System databases
  - …





MAL runtime aspects

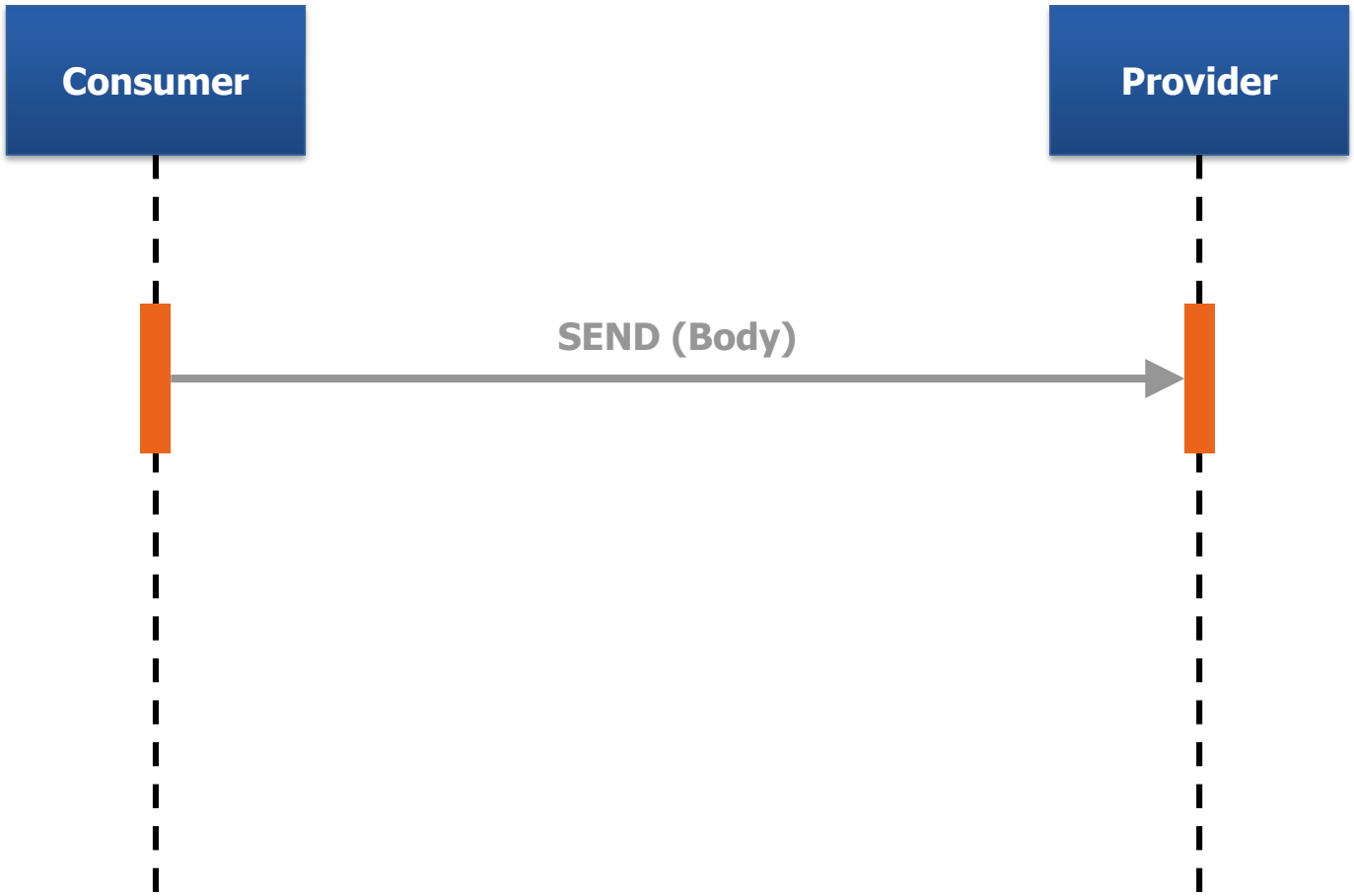
- Specifies the general requirements for all services
  - Expected behaviour of service providers
  - Expected behaviour of users of a service
- Provides the runtime messaging abstraction
  - This capability is optional
- Provides transactional behaviour
  - Matches the correct requests to their response
- Provides interaction state machine
  - Ensure that only the correct responses can be sent at the correct time

Interaction Patterns: In detail

- Six interaction patterns are defined by the MAL
  - SEND
  - SUBMIT
  - REQUEST
  - INVOKE
  - PROGRESS
  - PUBLISH/SUBSCRIBE

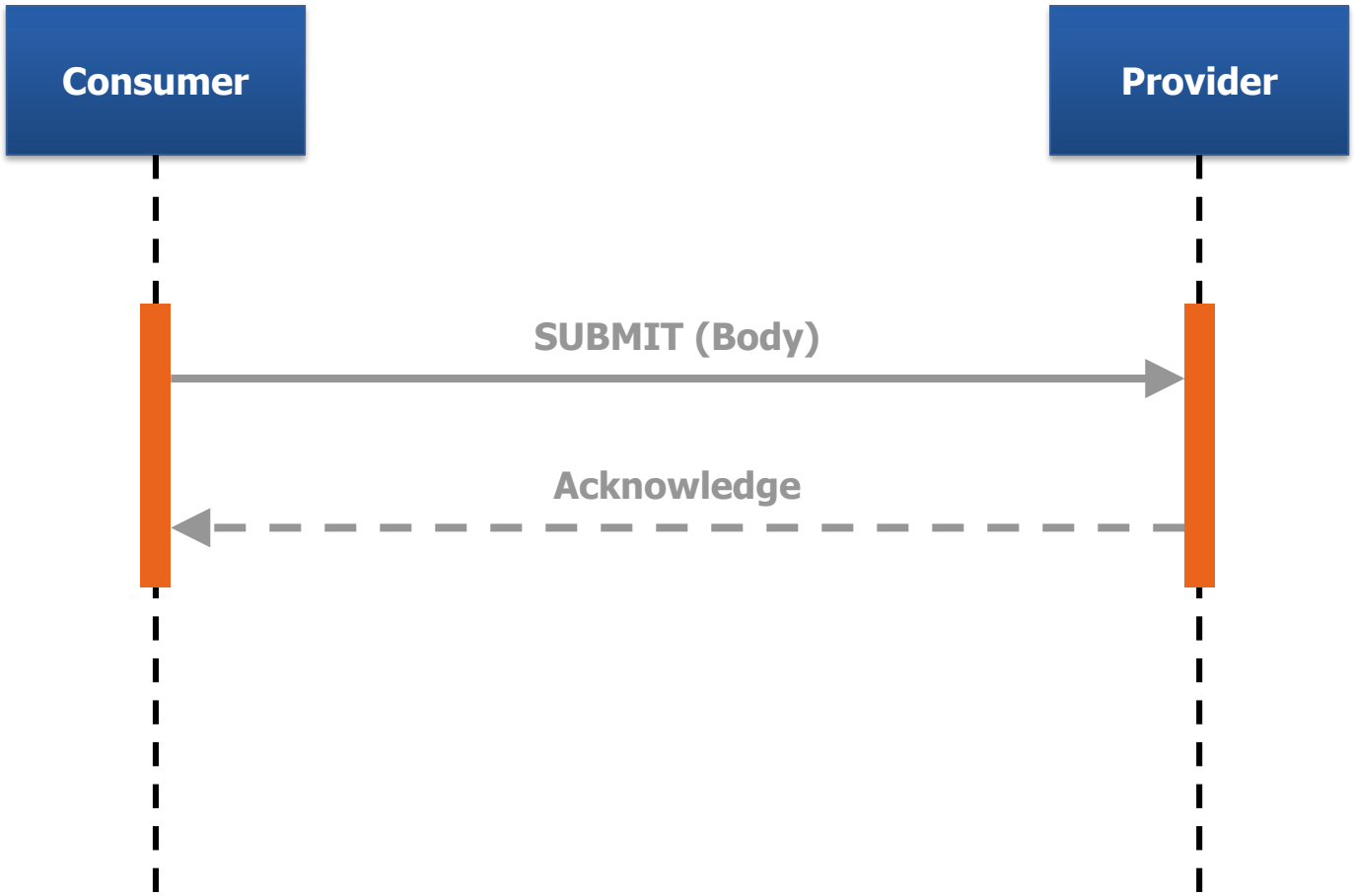
Interaction Patterns: SEND

- Simple unacknowledged operation



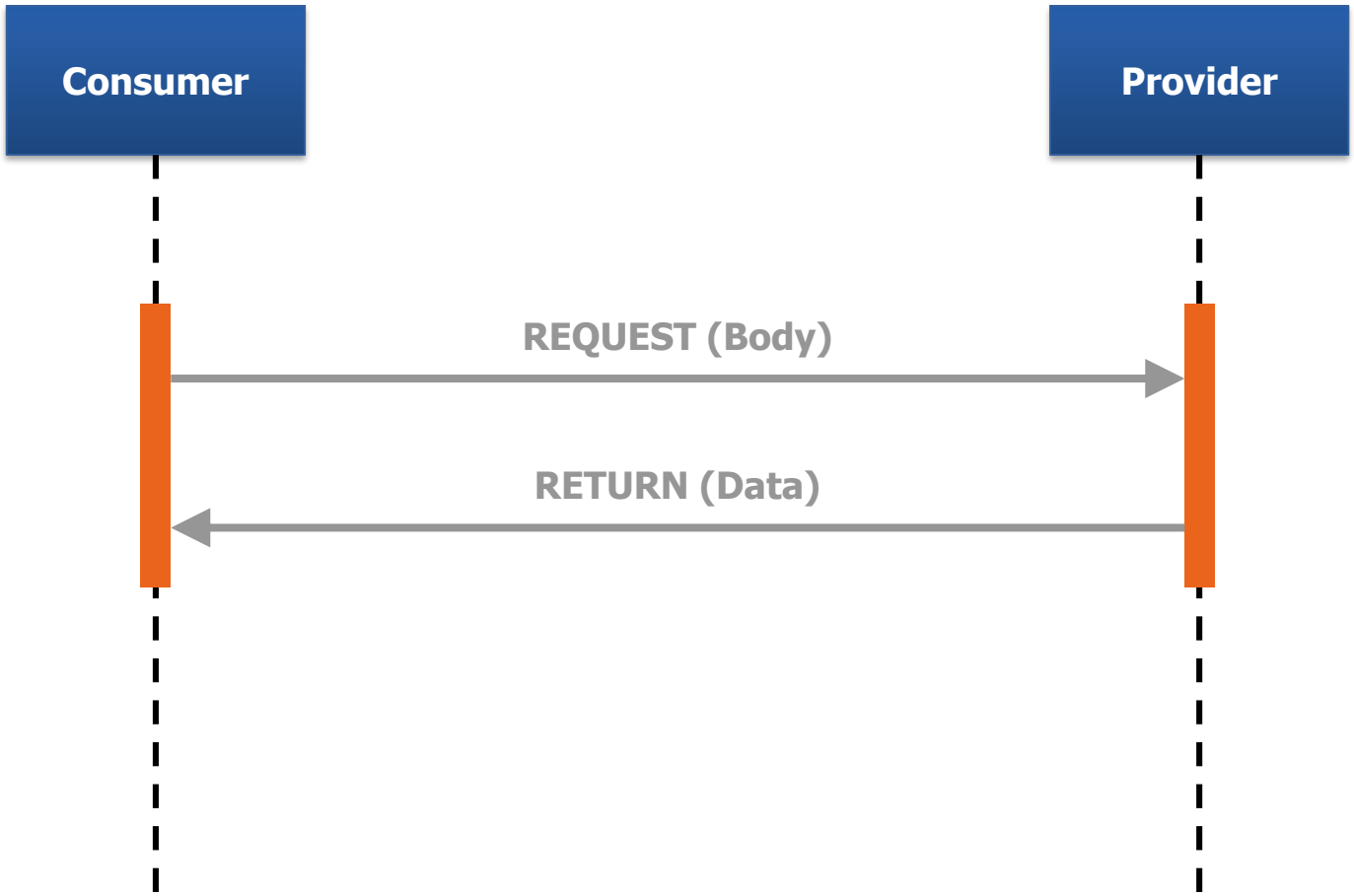
Interaction Patterns: SUBMIT

- Simple acknowledged operation



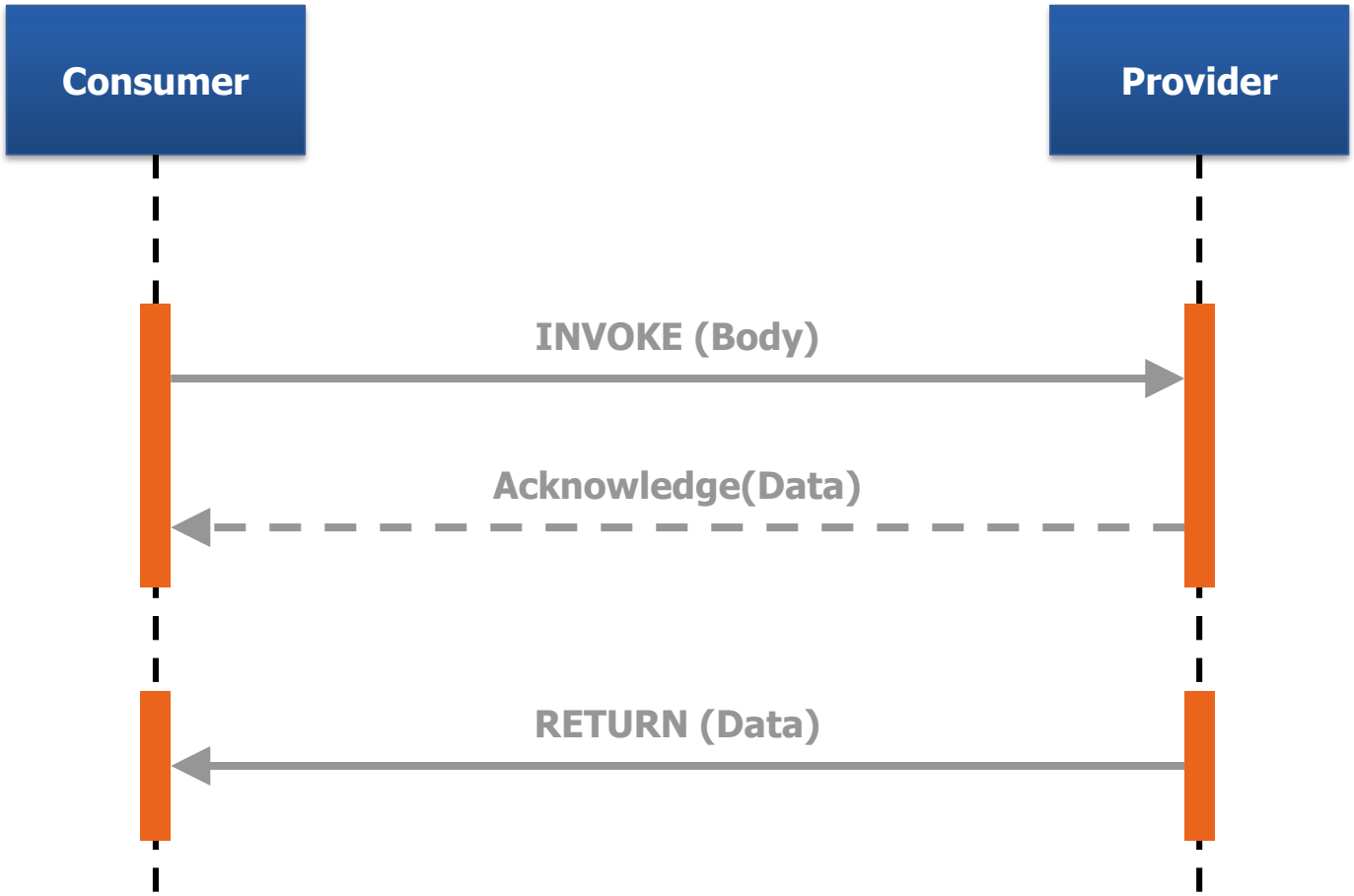
Interaction Patterns: REQUEST

- Simple request and response operation



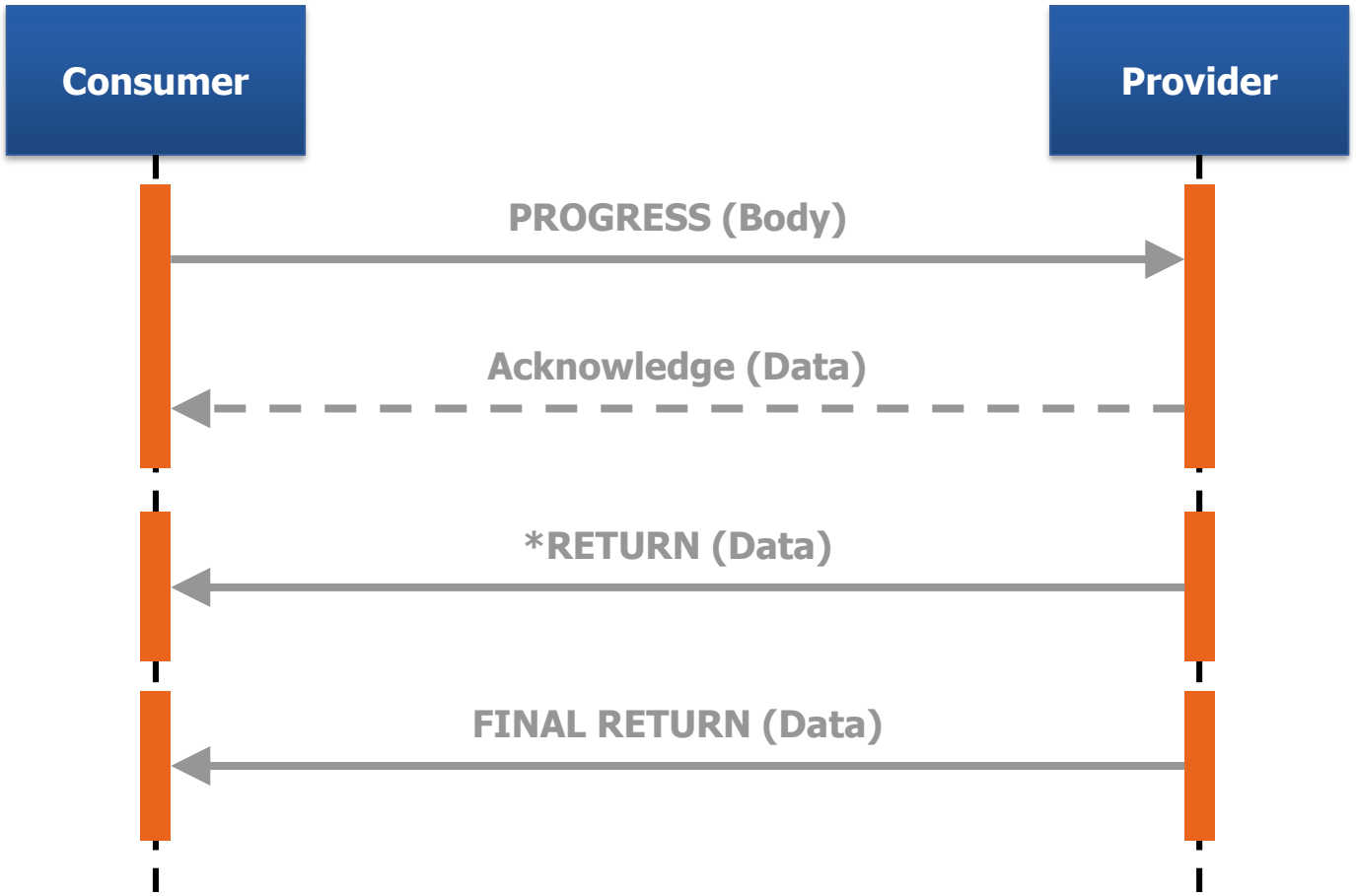
Interaction Patterns: INVOKE

- Acknowledged data request operation



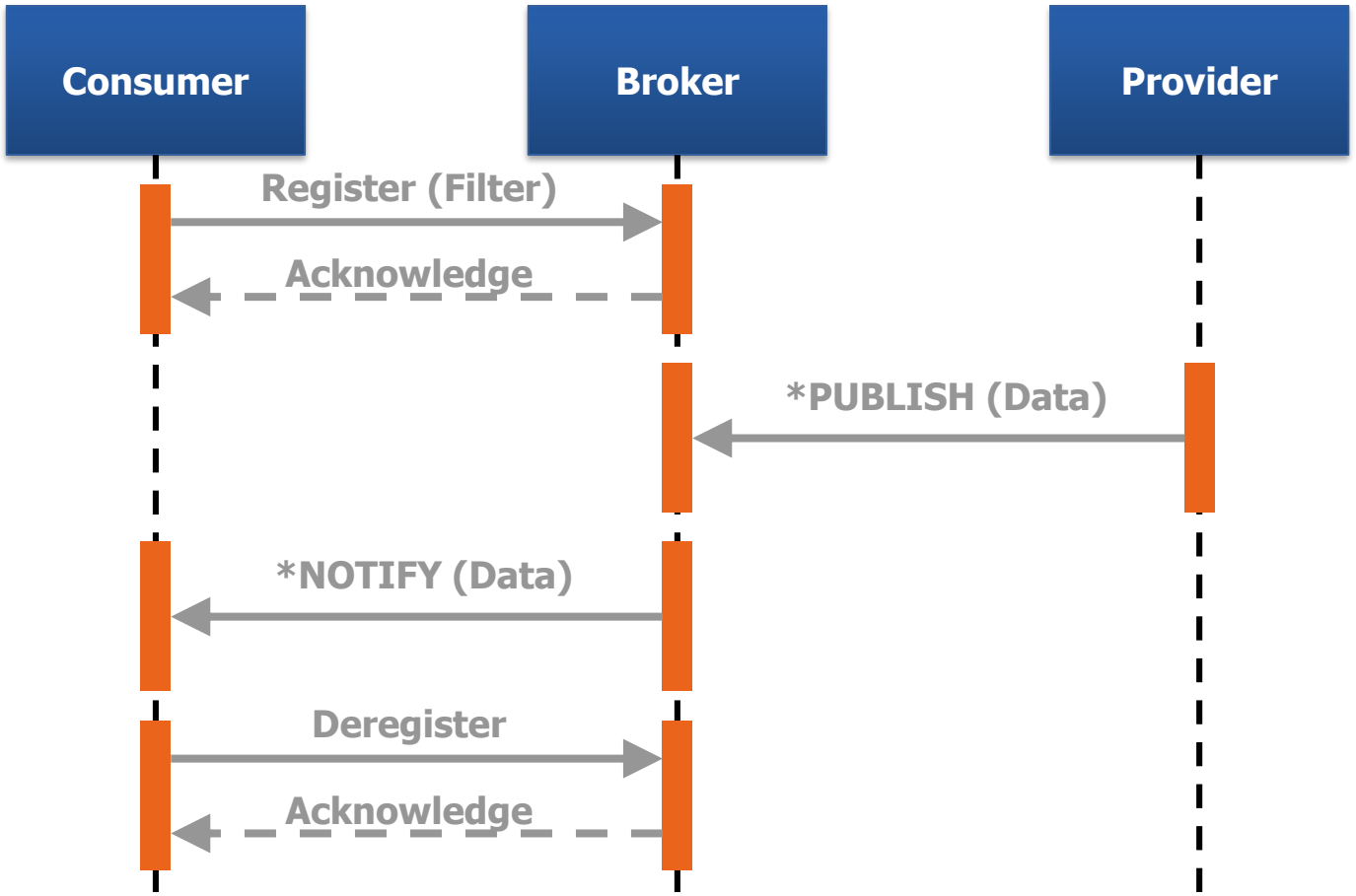
Interaction Patterns: PROGRESS

- Acknowledged multi-return data request operation



Interaction Patterns: PUB/SUB

- Multi-provider, multi-consumer data broadcast



Understanding the books: Services

- The following table is used to provide a summary of Service:

Area Identifier	Service Identifier	Area Number	Service Number	Area Version
COM	Archive	2	2	1
Interaction Pattern	**Operation Identifier**	**Operation Number**	**Support in Replay**	**Capability Set**
INVOKE	retrieve	1	Yes	1
PROGRESS	query	2	Yes	
INVOKE	count	3	Yes	
REQUEST	store	4	No	2
SUBMIT	update	5	No	3
REQUEST	delete	6	No	4

Understanding the books: Operations

- The following table is used to provide a summary of an Operation:

Operation Identifier	retrieve	
Interaction Pattern	INVOKE	
Pattern Sequence	Message	Body Type
IN	INVOKE	objType : (ObjectType) domain : (List<MAL::Identifier>) objInstIds : (List<MAL::Long>)
OUT	ACK	
OUT	RESPONSE	objDetails : (List<ArchiveDetails>) objBodies : (List<MAL::Element>)

Understanding the books: Composites

- The following table is used to provide the definition of a Composite MAL data type:

Name	ArchiveDetails		
Extends	MAL::Composite		
Short Form Part	1		
Field	Type	Nullable	Comment
instId	MAL::Long	No	The object instance identifier of the archived object.
details	ObjectDetails	No	The details of the Object.
network	MAL::Identifier	Yes	The network zone of the object.
timestamp	MAL::FineTime	Yes	The time the object was created.
provider	MAL::URI	Yes	The component that created the object (a component may be anything from an onboard equipment to a software process on the ground).

Understanding the books: Enumerations

- The following table is used to provide the definition of an Enumeration MAL data type:

Name	ExpressionOperator	
Short Form Part	5	
Enumeration Value	Numerical Value	Comment
EQUAL	1	Checks for equality.
DIFFER	2	Checks for difference (not equal).
GREATER	3	Checks for greater than.
GREATER_OR_EQUAL	4	Checks for greater than or equal to.
LESS	5	Checks for less than.
LESS_OR_EQUAL	6	Checks for less than or equal to.
CONTAINS	7	Case sensitive containment test (String types only)
ICONTAINS	8	Case insensitive containment test (String types only).

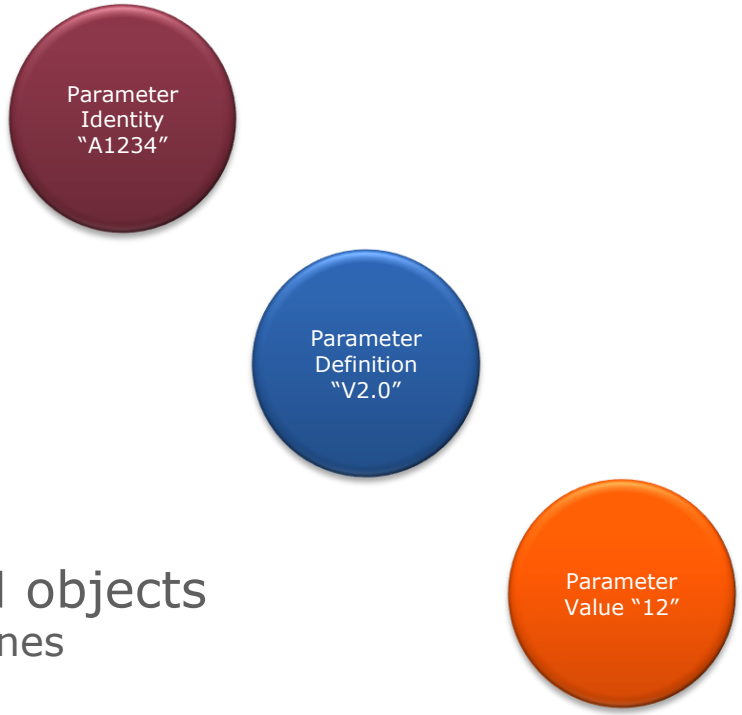
COM Services

Common Object Model (COM)

- The COM provides:
  - Archiving
  - Tracking of remote activities and operations
  - Asynchronous event reporting
- The object model:
  - Provides a standard way to identify data
  - Allows the development of standard capabilities such as
    - Archiving
    - Activity tracking
    - Event reporting
- Defined using the MAL notation
  - Technology agnostic

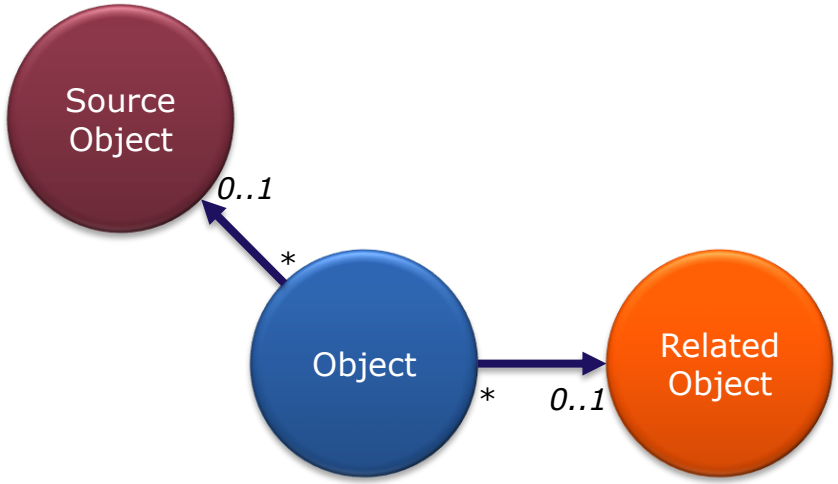
COM object model

- An object is defined as a thing that is:
  - Capable of independent existence
  - Can be uniquely identified
- An object may be a
  - Physical object such as a
    - Spacecraft
    - Ground station
    - An event such as an eclipse
  - A concept such as
    - Telemetry parameter
    - Automated procedure instance
- Each service must specify its COM objects
  - For example, the Parameter service defines
    - A Parameter Identity COM Object
    - A Parameter Definition COM Object
    - A Parameter Value COM Object



COM object model

- Each object can link to up to two other objects



- The two links have different roles:
  - The related link is expected to be used to link to a related object
  - The source link is expected to be used to link to an unrelated object
- It is service specific how these links are to be used
- For example, for a Parameter service, a value update may:
  - Use the related link to link to the Parameter definition COM object
  - Use the source link to link to the operator who requested the value change

COM Services: Event Service

- An event is a specific object representing
  'something that happens in the system at a given point in time'
- Examples of an Event are:
  - Violation of a parameter limit check
  - Successful completion of an on-board activity
  - Acquisition of the spacecraft carrier signal
- Each service can define the possible Events that it may raise
  - This information is held in the service XML
- An event is a COM object
  - The body of an event object may contain extra information
    - Specified by the service that declares the event
- The event service defines a single publish/subscribe operation
  - For publishing events
  - But also monitoring of events generated by other components

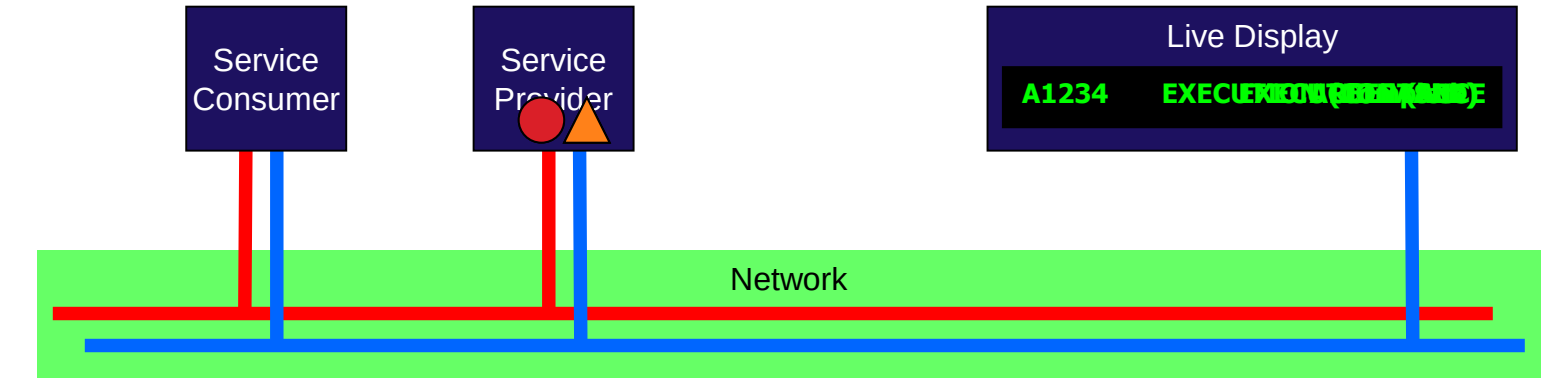
COM Services: Activity Tracking Service

- An activity is anything that has a measurable period of time
  - A command
  - A remote automated procedure
  - A schedule
  - …
- In distributed or complex system being able to monitor those activities is vital
  - The Activity Tracking service provides the mechanism for reporting the status of activities
- Defines an event pattern for reporting of the progress of activities from
  - The initial consumer request to start the activity
  - Reporting the progress of that request across any transport link
    - Including if being relayed
  - To reception by the provider
  - And execution of that activity in the provider.
- The service uses the COM Event service to report the progress of activities

COM Services: Activity Tracking Events

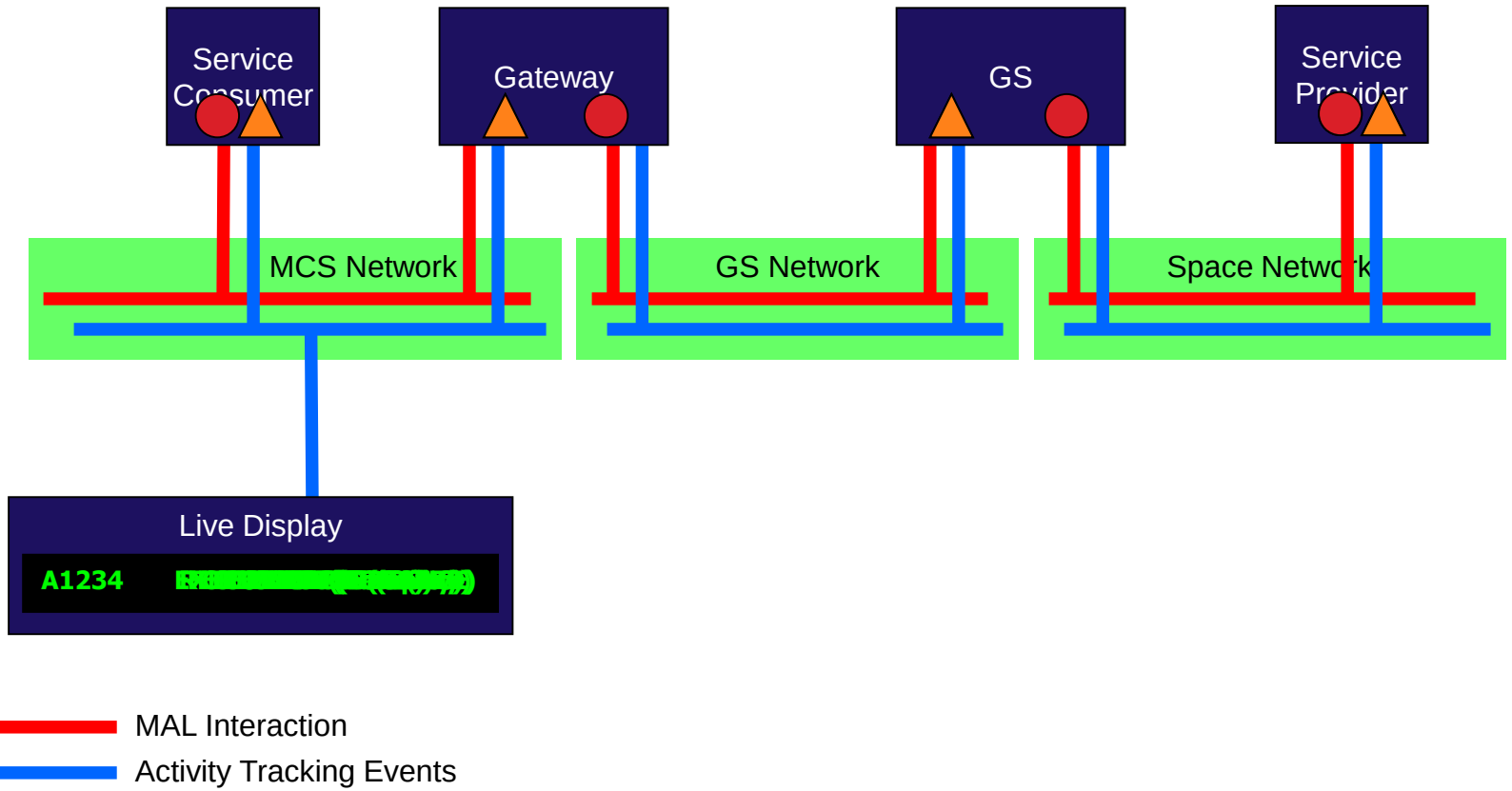
- Four transport notification events are defined
  - RELEASE — Release from source node
  - RECEPTION — Reception by an intermediate node
  - FORWARD — Forward is release from an intermediate node
  - ACCEPTANCE — Acceptance is reception by destination node
- A single event for execution progress, but may be sent multiple times
  - EXECUTION — Execution used to monitor execution progress
- The pattern is then
  (Release, (Reception/Forward)*, Acceptance, Execution*)

Single hop operation example



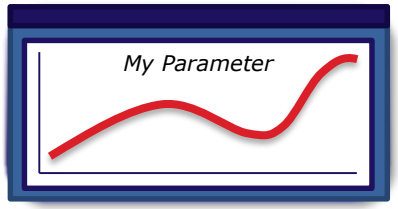
- Consumer sends two MAL messages
  - One for the INVOKE message as normal
  - But also an Activity Tracking event of type RELEASE
- Provider receives the INVOKE message as normal
  - Immediately publishes an Activity Tracking event of type ACCEPTANCE
- Provider returns
  - The INVOKE ACK message to the Consumer
  - Also an Activity Tracking event of type EXECUTION
- Provider returns
  - The INVOKE RESPONSE message to the Consumer
  - And an Activity Tracking event of type EXECUTION

Multi hop operation example



COM Services: Archive Service

- The Archive service provides
  - An archiving function for services that use COM
  - The ability to store all COM objects
  - The ability to store all updates to COM objects
  - Simple querying of the archive
    - More complex queries are supported but the specifications of these are outside this standard
- For example, the Parameter service
  - Stores Parameter definitions as COM objects
  - Stores updates to Parameter values as COM objects
- By storing these updates and objects in an archive any historical replay/retrieval function
  - Can correctly reflect the history of objects
  - And also show their evolution over time

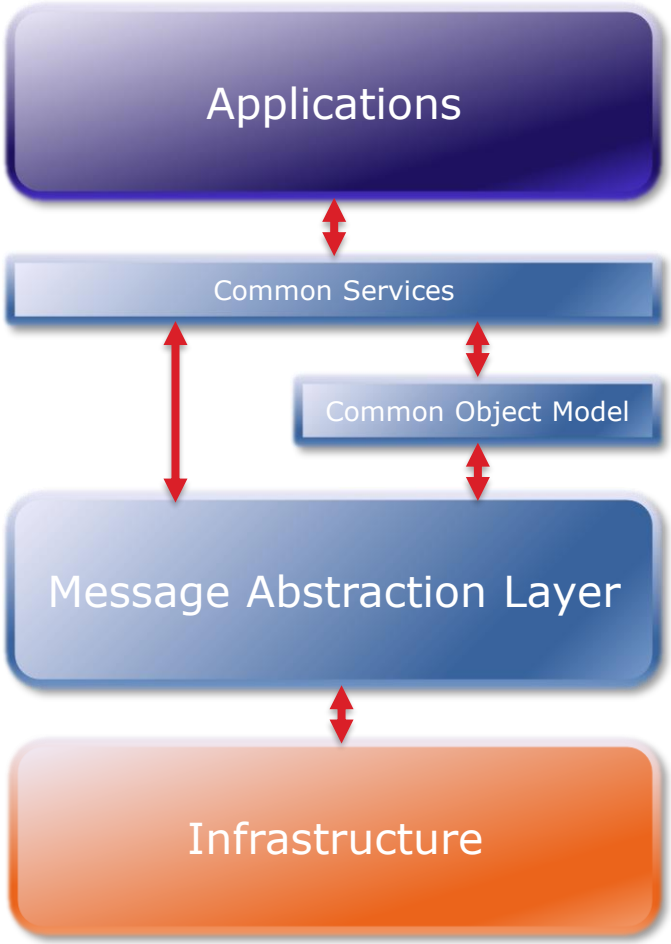


In-development Services

WIP: Common Services

WIP: Common Services

- The Common services:
  - Directory
    - Service lookup
  - Login
    - Operator log in
  - Configuration
    - Service provider configuration management



WIP: Directory Service

- The Directory service allows:
  - Service providers to publish information about which services they provide
  - Consumers to look up service provider address and capability information
- Provider information is made available using the publishProvider operation
- Consumers query the Directory using the lookupProvider operation
- Provider information is removed using the withdrawProvider operation
- A final operation allows a consumer to retrieve the service XML for the provider

Area Identifier	Service Identifier	Area Number	Service Number	Area Version
Common	Directory	3	1	1
Interaction Pattern	**Operation Identifier**	**Operation Number**	**Support in Replay**	**Capability Set**
REQUEST	lookupProvider	1	Yes	1
REQUEST	publishProvider	2	No	2
SUBMIT	withdrawProvider	3	No	
REQUEST	getServiceXML	4	Yes	3

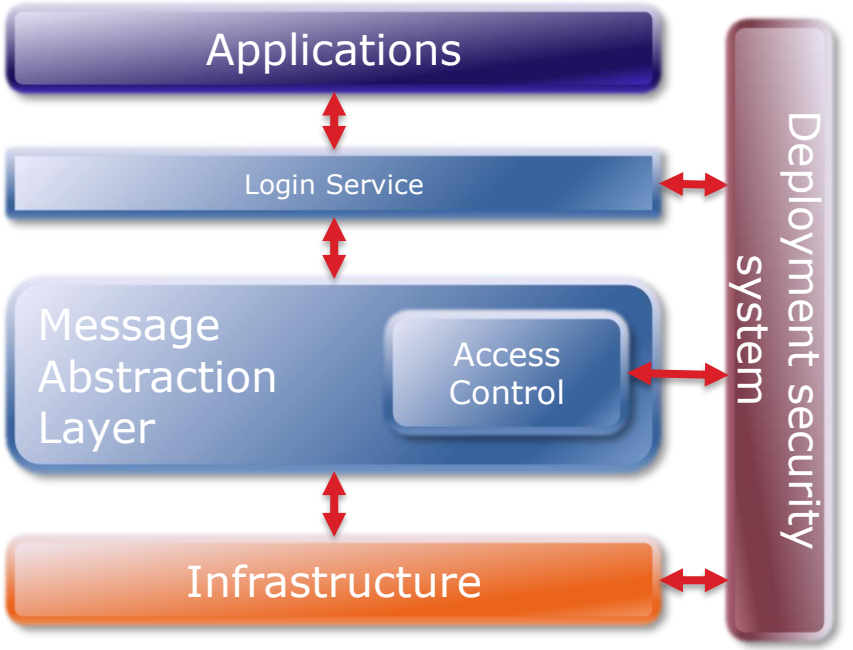
WIP: Directory Service XML

- The getServiceXML operation returns the service XML
  - Allows a consumer to request the MO XML specification of a service
  - Supports both standard and bespoke services
  - XML returned in a file
  - XML supplied by provider when registering with Directory service

- This allows development of generic service consumers
  - Applications use the service XML to define what they can do at runtime
  - Read the XML to specify the:
    - Supported operations
    - Returned messages
    - Supported COM objects
    - Supported COM events

WIP: Login Service

- The Login service defines the primary mechanism for
  - Submission of authentication details to a deployment specific security system
  - Closely tied to the Access Control aspect of the MAL
  - Also tied to secure transports
- It supports operations to allow a user to:
  - Login
  - Logout
  - Report available roles
  - Handover to another user



Area Identifier	Service Identifier	Area Number	Service Number	Area Version
Common	Login	3	2	1
Interaction Pattern	**Operation Identifier**	**Operation Number**	**Support in Replay**	**Capability Set**
REQUEST	login	1	No	1
SUBMIT	logout	2	No	
REQUEST	reportRoles	3	No	2
REQUEST	handover	4	No	3

WIP: Configuration service

- The Configuration service defines the concept of a Configuration
- Can be any type of configuration:
  - Hard-coded configurations (such as camera preset modes)
  - Bespoke configurations (such as application settings)
  - COM service configurations (such as supported Parameters of a service provider)
- The service allows a service consumer to
  - Activate pre-defined configurations of a service provider
  - List, get, add new, remove, and store current configurations
- It also defines a standardised XML representation for configurations
  - Export/import XML configurations

Area Identifier	Service Identifier	Area Number	Service Number	Area Version
Common	Configuration	3	5	1
Interaction Pattern	**Operation Identifier**	**Operation Number**	**Support in Replay**	**Capability Set**
INVOKE	activate	1	No	1
REQUEST	list	2	Yes	2
REQUEST	getCurrentConfiguration	3	Yes	
REQUEST	getXMLConfiguration	4	Yes	3
REQUEST	add	5	No	4
SUBMIT	remove	6	No	
REQUEST	storeCurrentConfiguration	7	No	5
REQUEST	storeXMLconfiguration	8	No	6

Mission Operations

WIP: Monitor and Control Services

WIP: Monitor and Control Services

- The M&C services are a set of services that enables a mission to perform basic monitoring and control of a remote entity
  - These services are defined in terms of the COM and the MAL

- It provides three basic classes of information:
  - Actions
    - Allow control directives to be invoked and their evolving status to be monitored
    - Spacecraft telecommands are an example of an action.
  - Parameters
    - Provides capability to monitor parameter values
    - Provides the capability to set parameter values
  - Alerts
    - Provide a mechanism for asynchronous notification of operationally significant events or anomalies

WIP: Monitor and Control Services

- The M&C services also include extra services to extend the basic three services:
  - Check service
    - Provides online checking of Parameter values
  - Statistic Service
    - Provides online statistical evaluation of Parameter values
  - Aggregation Service
    - Provides aggregation of separate Parameter values in to coherent sets
  - Conversion service
    - Provides the specification of calibrations and conversions for Parameter values
  - Group service
    - Provides the ability to define groupings for simplifying the operations of the other services

WIP: Action Service

- Actions are the operations of your remote entity
  - They can be invoked and their evolving status can be monitored
    - Spacecraft telecommands are an example of an action
- Actions have their definition held in a database
  - The COM Archive is an example of an entity that can hold these Action Definitions
  - This is different to the definition of Service Operations
    - These are held in the service XML specification
- The Action Service operations include:
  - Pre checking of action invocation requests
  - Issuing of action execution requests
  - List, get, add, remove of action definitions

WIP: Action Verification

- For a remote system it is normal to pre-check Actions before submission
  - This includes checks such as link status, spacecraft state, and argument checks
- The Action Service supports two patterns of submission
  - Check and then send
    - An action is checked locally to determine if it would be executed if subsequently sent
  - Just send
    - An action is sent for execution and the returned execution status is checked to see if the action succeeds
- The Action service uses the COM Activity Tracking Service
  - For reporting the transmission progress of Actions
    - The Action service uses the standard COM Activity Tracking events
  - For reporting the execution progress of Actions
    - The Action service extends the COM Activity tracking execution events

WIP: Parameter Service

- Parameters have an evolving value represented by a chronological sequence of value updates over an unbounded lifetime
- Parameters have a defined type
  - This is the data type of the value update, such as string or integer
  - Composition of parameters into complex structures supported by defining an Aggregation
    - See Aggregation service
- Value updates may be:
  - Periodic
  - Change-based
  - Or a mixture of the two
- It is also possible, when supported, to set the value of a parameter
- Parameter definitions are not defined in the service specification
  - Delegated to the runtime configuration of the provider
- Parameter definitions are managed using the operations of the Parameter service

WIP: Alert Service

- Alerts are operationally significant events or anomalies
  - Characterised by an identifier and a set of arguments
- The alert service differs from the COM Event service
  - It allows the definition of a set of alerts to be defined dynamically
  - COM Events are defined statically in the relevant service specification
- An alert is defined as a specialised COM event
  - The alert service uses the COM Event service to publish its alerts
- Alert definitions are managed using the operations of the Alert service

WIP: Check Service

- The Check service allows the definition a set of checks to be applied to Parameters
  - The check provider periodically samples the values of those parameters and checks them
- If a Check is violated a specific COM event is generated
- The following is the list of checks that must be supported:
  - Limit check
    - The value lies within a specified range;
  - Constant check
    - The value is checked against a specified value or value of another parameter;
  - Delta check
    - The change in value is checked against a pair of thresholds.
- However, service implementations may support additional custom types
- The Check service specifies operations to:
  - Define the checks
  - Associate parameters to those checks

WIP: Statistic Service

- The Statistics service allows the association of Parameters to defined statistical evaluations
  - e.g., min, max, mean, standard deviation
  - The service provider periodically samples the values of those parameters and evaluate them against the statistical function
- Service implementations may support additional custom types
- The resultant statistics evaluations are provided to registered consumers
- The Statistic service specifies operations to:
  - Associate parameters to statistical functions
  - It is not possible to define new statistics at runtime as that would require a software modification

WIP: Aggregation Service

- A logical extension to basic Parameter service is data aggregation
- The Aggregation service provides the capability to acquire several parameter values in a single request
- The Aggregation might be one of the following:
  - Predefined by the service provider, e.g., housekeeping parameters
  - Defined at runtime by the consumer, e.g., a diagnostic report
- The Aggregation service provides operations to
  - Define which parameters to aggregate
  - Report the current values of those parameters

WIP: Conversion Service

- Provides engineering unit conversion capability
  - A functional extension used by the other services
    - For example: Conversion of raw sensor Parameter to calibrated unit such as Volts
- The Conversion service defines a basic set of conversion types
  - Discrete
    - For example: '0' = 'offline'
  - Line
    - Holds a set of line segments that represent an arbitrary curve
  - Polynomial
    - For example: $2 + 6x + 4x^2 + 7x^3$
  - Range
    - For example: [0-10] = 'Low'
- The service does not perform the conversion
  - Conversion definitions are held in the COM archive
  - Other services then reference these Conversions as appropriate to those services
    - For example, a Parameter would reference a specific Conversion definition

WIP: Group Service

- The Group service simplifies the administration of the services:
  - It allows a group of objects to be referenced rather than each object individually
  - Reduces the bandwidth required for certain operations
- The Group service defines the concept of a group
  - A group is a simple object which holds a list of links to other objects
- Services reference Groups in their operations rather than objects
  - Groups are defined and maintained using the COM Archive
  - Other services reference Groups as appropriate to those services
- For example, for the Parameter service:
  - You can enable generation of a set of parameters using the operation:
    - enableGeneration (ParamA, ParamB, ParamC, ParamD, ParamE, ParamF, …)
  - Or you can use:
    - enableGeneration (GroupX)
    - Assuming an appropriate Group is already defined

Forthcoming Services

FC: Navigation Services

- The Navigation Services support the provision of spacecraft positioning information such as:
  - Position reports (e.g., from on-board GPS)
  - Spacecraft ranging and range-rate measurements
  - Antenna tracking azimuth and elevation
  - Orbit vectors
  - Attitude vectors
  - Trajectory requests
  - Predicted orbital events (including ground station visibilities)
- The provider is either:
  - A spacecraft
  - A flight dynamics application
  - Or a ground station facility
- The services use the data types and file formats defined by the CCSDS Navigation working group

FC: Planning Services

- The Planning services are concerned with the planning aspect of a mission:
  - Planning request
    - Allows the request for an operational task or goal to be included in a plan
  - Plan provision
    - Proving the output of a plan to a consumer
- Potential consumers include:
  - Operations Team
  - External Users
    - Principal Investigators
    - Mission Exploitation System
    - End Users
  - Flight Dynamics
  - Software Management
- The specification of the services does not prescribe the planning process or algorithms used by the Planning application

FC: Scheduling Service

- Scheduling is concerned with the distribution, monitoring and control of scheduled timelines of mission operations intended for automated execution
- The service provider is an application capable of executing the schedule
  - Whether ground-based or on-board the spacecraft
- The service object is a schedule which is a container for individual items that appear on the schedule timeline, including:
  - Planned Contacts (periods of connectivity between space and ground)
  - Predicted Events
    - Predicted Events correlate to those notified via Flight Dynamics services
  - Scheduled Tasks, potentially containing multiple Activities
    - Scheduled Tasks correlate to a source Planning Request
    - Activities correlate to individual Procedures or Functions that can be initiated via the Automation service
- The service also supports insertion, deletion and modification of individual scheduled items.

FC: Automation Service

- The Automation service provides support for the automation of mission operations
- The service provider is an application capable of executing pre-defined Procedures (or autonomous Functions)
  - Whether ground-based or on-board the spacecraft
  - Consumer functions include Schedule Execution and Manual Operations displays
- Schedule Execution or Manual Operations may invoke and execute a new instance of a Procedure via the service
- Subsequent control over the procedure may be exercised to
  - Suspend
  - Resume
  - Stop its execution
  - Perform manual control over its execution
- The execution status of the procedure can also be observed either at the level of the procedure object itself, or potentially at a lower level of detail in terms of the procedure model

FC: File Services

- The File services are split into two aspects:
  - Management of a remote file store
  - Request the transfer of file between those file stores
- The service supports operations to:
  - Obtain directory tree listings of the content of the remote store
  - Perform store management operations such as
    - Delete, move, rename, etc.
  - Provide information about changes to the remote store
  - Request the transfer files in both directions
    - A service consumer can both get files from and put files into the remote store
- It is anticipated that this service would utilise an appropriate file transfer protocol for the transfer of files:
  - Most likely CCSDS CFDP across the spacelink
  - FTP on ground

FC: Software Management Service

- The Software Management service supports the management of software loaded into the remote system
- The operations supported include:
  - Load Software Image
  - Dump Software Image
  - Check Software Image
- The service definition reflects the fact that most on-board software is managed in terms of binary images that are stored at known memory addresses
- It also supports systems that use an on-board file store
  - Where images are held as files

FC: Time Service

- Time correlation between on-board clocks and the system reference time is required many aspects of a mission:
  - For basic M&C purposes
  - On-board scheduling
  - Flight dynamics
- The time service includes the following operations:
  - Report Time
  - Correlate Time
  - Set Time
  - Configure rate of Time Report generation.
- The service does not define *how* time is correlated
  - That is specific to a mission
  - But does define how time is reported and correlation is requested

FC: Mission Data Product Distribution Service

- Mission Data Product Distribution Service is used for the distribution of historical archived data and on-line 'live' data
  - It provides two delivery modes, batch mode and stream mode
- For the historical archived data
  - It is delivered using the batch mode
  - The consumer specifies a time-range of the data to retrieve
- For the on-line ‘live’ data
  - It is delivered in a stream
  - The consumer subscribes for product updates
  - The consumer receives then once they are available
- The two request types can be set up as either
  - A online request
  - A predefined standard order style request

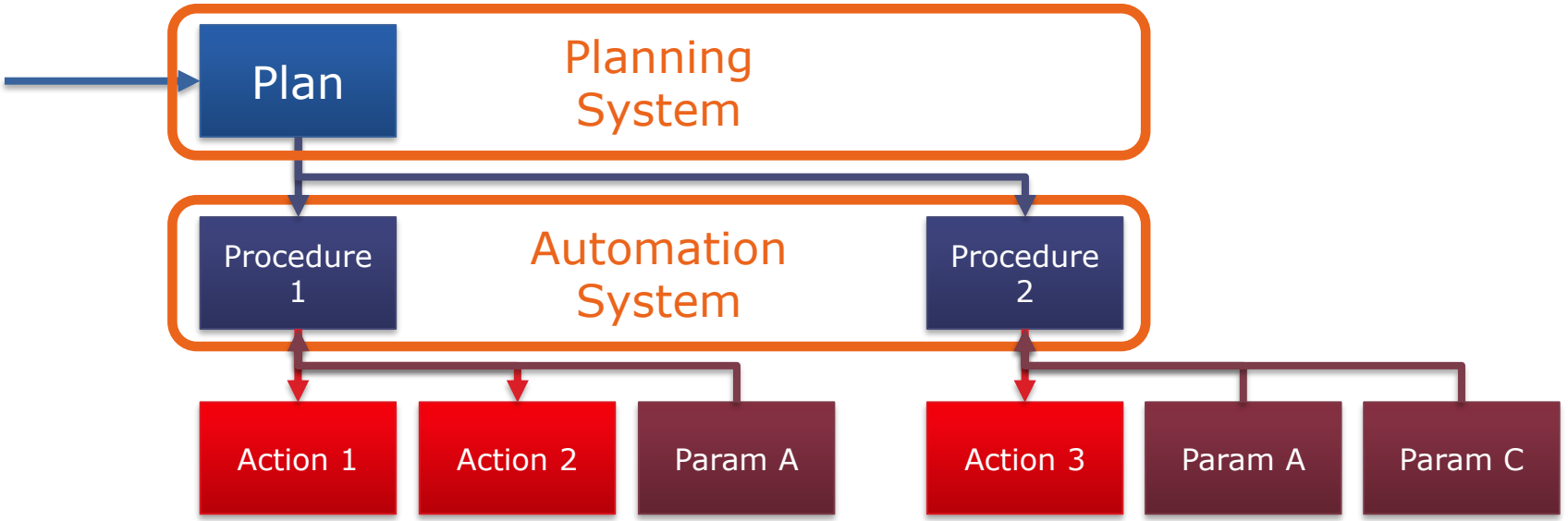
Conceptual deployment

Conceptual deployment

- In most modern systems the MO services will be used in conjunction with each other
  - It is unlikely that a single service will cover all needs
- Whilst the MO services have been designed to work together and complement each other
  - It is not required to use all the services if you have no need for
- Low level services such as COM Activity Tracking will be used in conjunction with higher level services such as Automation and Planning
- How you use and deploy the MO service is up to you

Conceptual deployment

- For example
  - A planning system may take in requests to perform activities
    - This would be used to form a plan
  - This plan may get decomposed into automated procedures
  - Which are executed by an automation system
    - Which in turn may monitor and control the remote entity using the M&C services

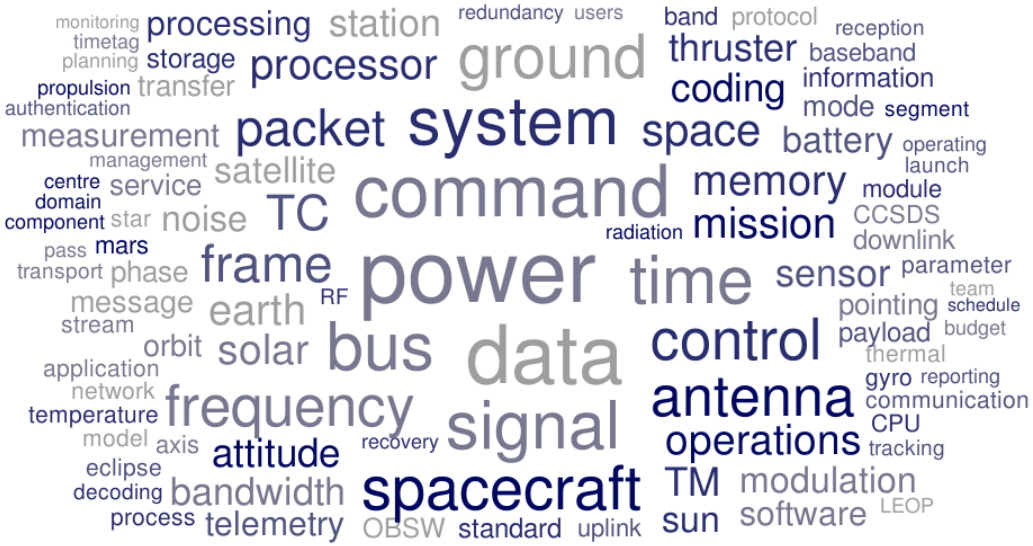


Part 3

- A Worked Example

A Simple Mission

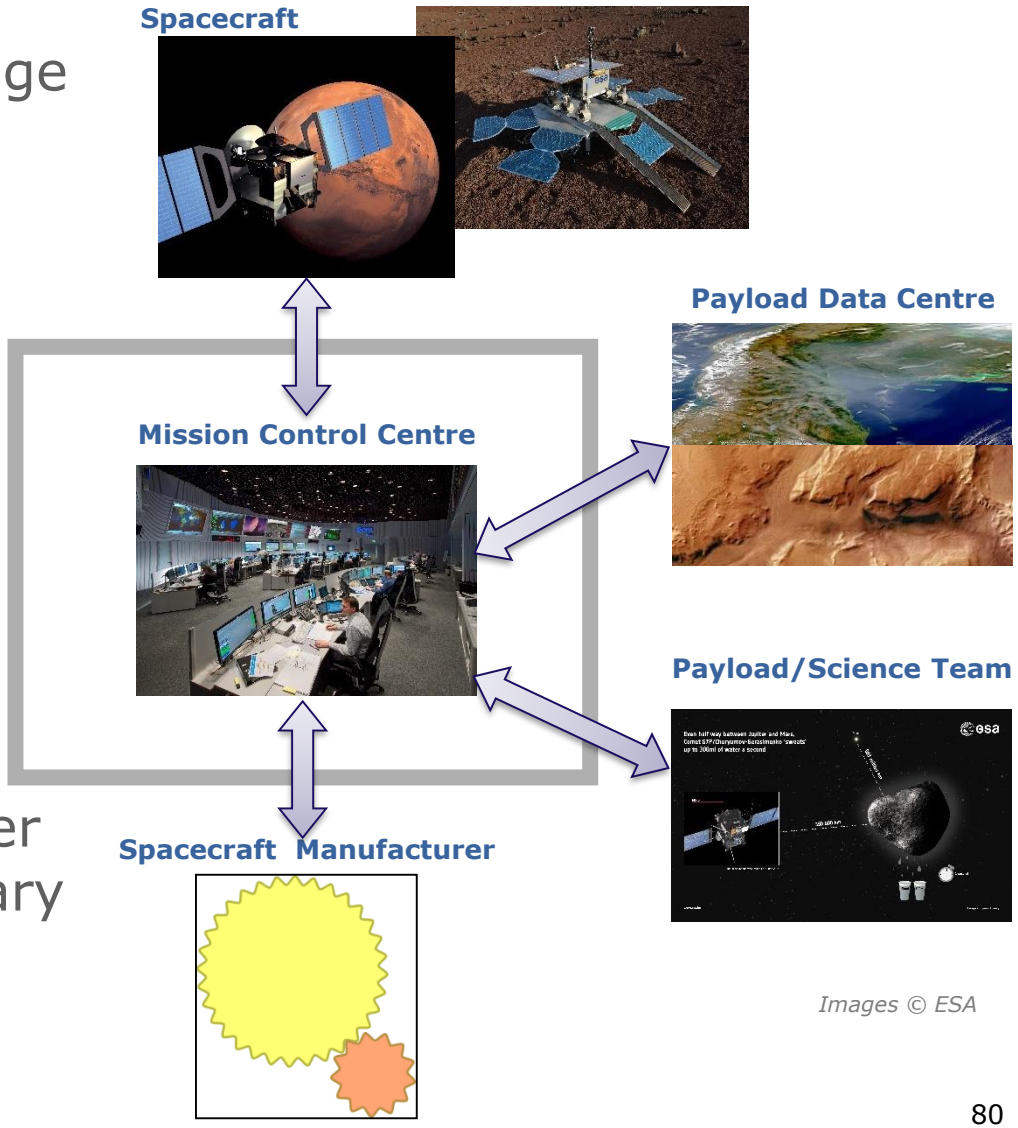
- Is there such a thing?!



- Assume a single:
  - Spacecraft
  - Ground station
  - Control centre

Service Boundaries

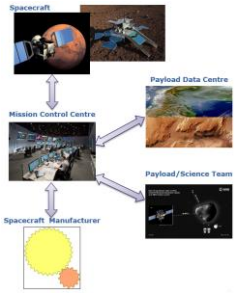
- Service boundaries form edge between one aspect of a mission to another
- They can exist
  - Between cooperative agencies
  - Between system elements
  - Inside an element
- Where information crosses from one element to another may form a Service Boundary



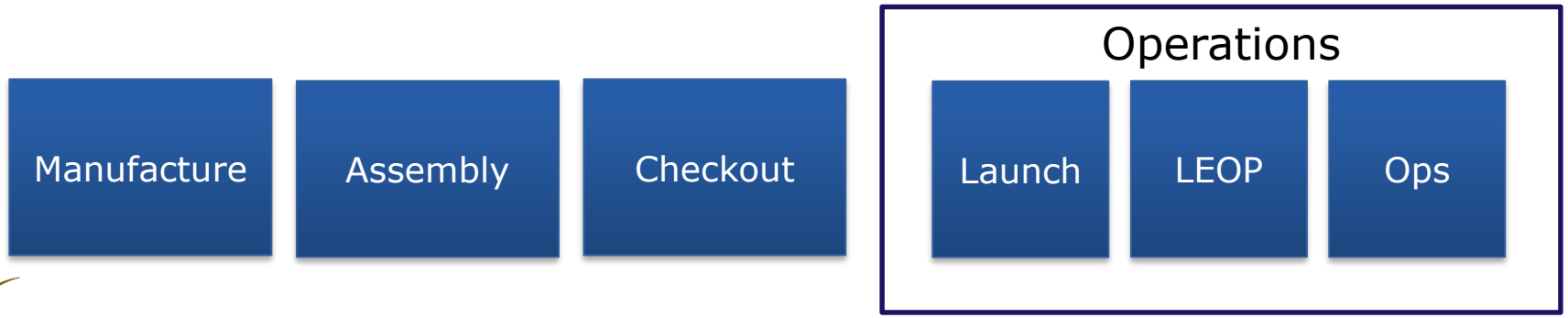
Images © ESA

Service Boundaries

- Service boundaries not only exist at physical boundaries such as:
  - Between Space and Ground
  - Between platform operations and payload operations
  - Between cooperative Agencies

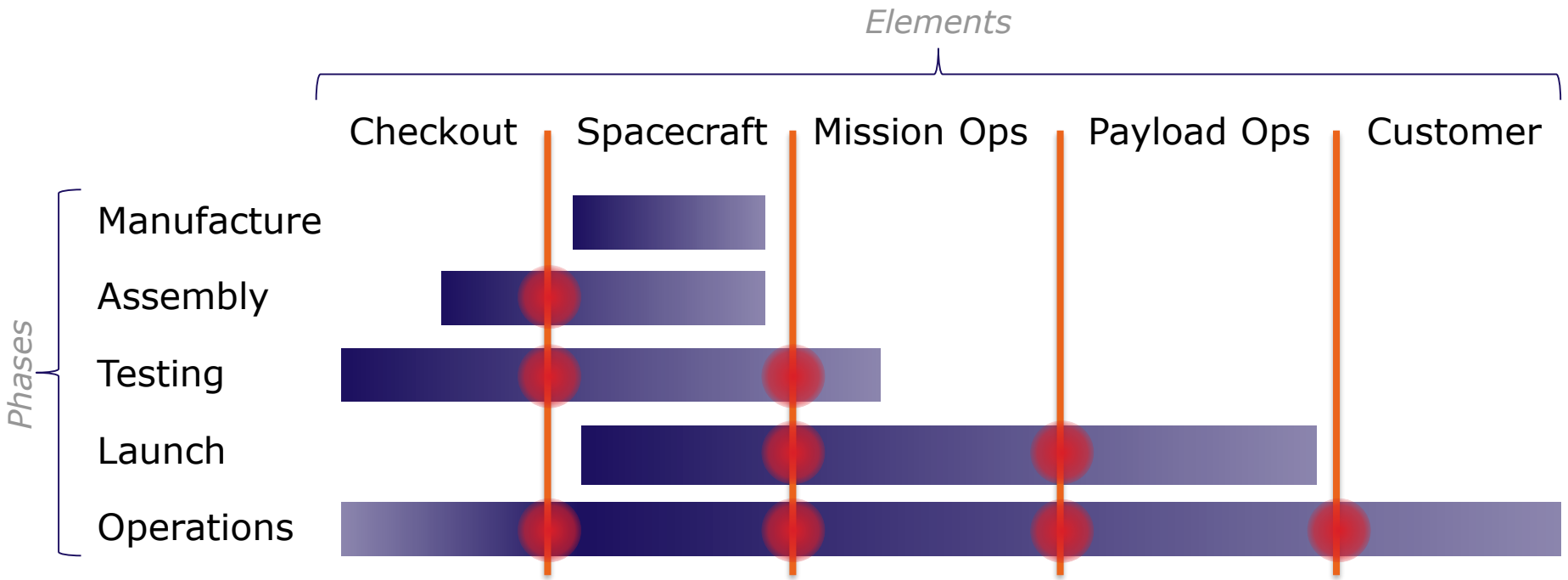


- But also exist at temporal boundaries such as:
  - Between spacecraft manufacture and AIT
  - Between checkout and operations
    - And inside the different phases of operations



Service Boundaries

- So where are the boundaries in our mission?



- These are the high level ones, there are many more
- This is repeated for each system component
  - Spacecraft, MCS, etc

Service Boundaries and our example

- For this example we will focus on a single crossing point
- The connection between Platform and Payload operations during the main operations phase
- There are many interactions at this point between these two entities:
  - Commands up
  - Live housekeeping telemetry down
  - Non-realtime platform TM
  - Maybe some navigation data
  - Maybe some high level activities that need to be planned
- For this example we will look at telemetry delivery

Example MO Service Deployment

- Two scenarios exist
  - The first is the transmission of live housekeeping TM during contact
  - The second is the transmission of off-line data after contact
- In our example we will use two services:
  - The Aggregation service for live housekeeping telemetry during contact
    - Focussed and fast with low latency
  - The MDPD service for delivery of other data after contact
    - Allows telemetry consolidation to happen before sending unlike "live" Aggregation
    - Better suited to larger payload data size
    - Can include non-TM items such as images, plans, etc
    - A COM service so uses the specification of an Aggregation from that service
- Our service provider is the platform operation centre
- Our service consumer is the payload operations centre

Aggregation service sequence

- For this service, and example, the consumer only needs to connect during contact

- The sequence is as follows:
  - Determine the connection details of the service provider
    - Either via a fixed address
    - The use of an MO Directory service
    - Or some out of band agreement/mechanism (email of IP address?)
  - Connect to the Aggregation service provider
    - May have to provide some credential information either
      - For the link itself (may be using something like SSL sockets)
      - Or for the MO interaction using the Common Login service
  - Subscribe for the required Aggregations
  - … receive the Aggregation as they are received by the platform centre
  - At the end of the contact unsubscribe
  - Disconnect from the provider
    - Whether this is actually required depends on the transport used

Aggregation service subscription

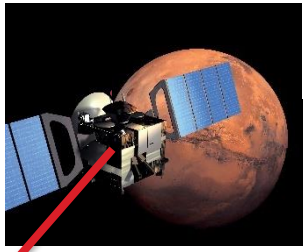
- So lets look at the data that is passed during the connection
  - Assuming we already have the connection details
- The subscription request needs to know:
  - The domain of the subscription and whether to include all sub-domains
  - The service being subscribed to
  - The items required
  - Finally an identifier for the subscription
    - This is used when receiving the response so that you can match the subscription and the returned data
- For our example we will subscribe for all data in a single domain

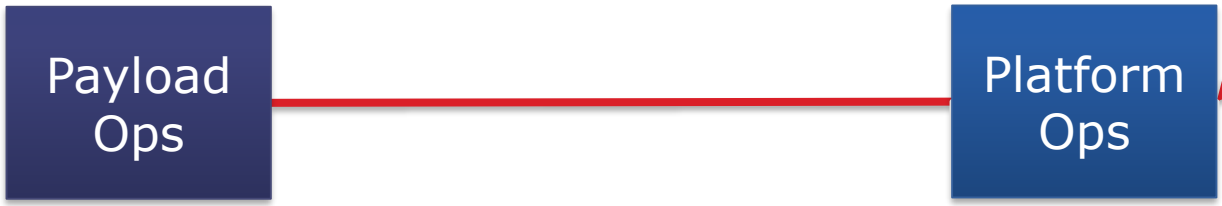
Aggregation service data

- We receive the Aggregation updates in a NOTIFY message from the broker, which contains
  - The subscription identifier
  - A list of update headers
  - A list of matching updates
- Each update header contains items such as:
  - Timestamp of the update
  - The identifiers of the item being updated
- An Aggregation update contains item such as:
  - Generation mode
  - A list of parameter values
    - This also includes optional delta times for when parameter values do not all have the same sampling time

Aggregation service data

- So, this complete sequence would look like:
  - Lookup
  - Connect
  - Register
  - Receive
  - Deregister
  - Disconnect





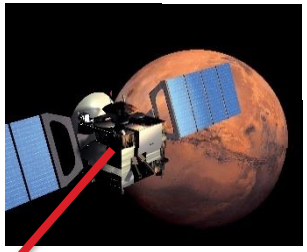
MDPD Service sequence

- For the MDPD service, the connection approach depends on how the data is requested
  - It offers two modes:
    - Request/Response
    - Subscription with later delivery (standing order)
- The standard order is like a normal PUB-SUB subscription
  - It holds a list of required data
- For the MDPD standing order
  - This can be either be set up dynamically
    - Where the MDPD service is used to set up the standing order
  - Or using some out of band agreement

MDPD Service subscription

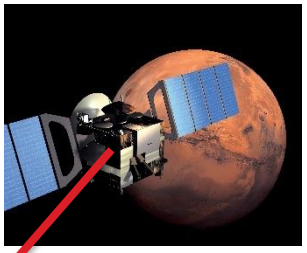
- For our example, we'll assume that a standing order for a specific set of data has already been setup
  - In this case there is nothing we need to do to set up the connection as it is always open
  - The opening and closing of the connection is something that is done at the start and end of the phase

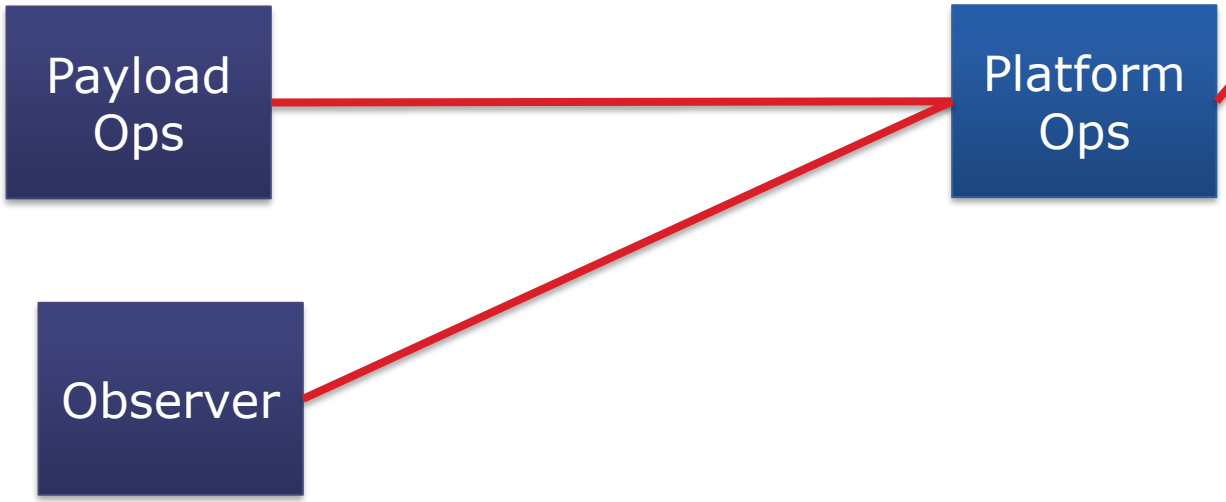
- So the sequence would be:
  - Receive the data



So why use MO?

- *Standard services define a single specification for the exchange of information*
- *An open architecture and framework*
  - *Independent from technology*
  - *Able to integrate new and legacy systems of different organisations*
  - *Designed to support the long lifetimes of space missions*





MO Example summary

- *Standard services define a single specification for the exchange of information*
- *An open architecture and framework that is*
  - *Independent from technology*
  - *Able to integrate new and legacy systems of different organisations*
  - *Designed to support the long lifetimes of space missions*
- *This leads to*
  - *Minimisation of the number of different interfaces*
  - *Increased cooperation between organisations*
  - *A reduction in cost*

- *Can easily exchange information with new organisations as specification exists*
- *The MO open architecture and framework would enable us to*
  - *Easily change our technology when needed*
  - *Support integration of new and legacy systems of different organisations*
  - *Cope easily with the long lifetimes of space missions*
- *Direct benefits to us are*
  - *Simplification of the development effort through minimisation of the number of interfaces*
  - *Increased cooperation between organisations*
  - *A reduction in cost*

Summary

Summary

- Two main trends are becoming apparent in current and future missions:
  - An increase in mission complexity
  - An increasing pressure to reduce costs
- In today's systems a lack of
  - Standardisation between and inside organisations
  - Combined with lack of re-use between missions
  - Has led to increased cost of development, deployment and operator training
- To alleviate this CCSDS is defining and standardising:
  - A set of services for Mission Operations
  - A open framework, independent from technology
- Whilst over the long lifetimes of space missions the inability to
  - Update technologies and infrastructure
  - Replace systems without major system redesign
    - Including vendor lock in
  - Has led to increased operational costs
- These provide:
  - Increased interoperability and reuse
  - Improved flexibility
  - Independence from location, programming language, hardware platform and communications technology

The CCSDS Mission Operations services provide the capability to hide the unavoidable difficulties that characterises the space environment whilst supporting your future complexity needs

Mission Operations

Links

General

CCSDS Website

http://www.ccsds.org/

MO in Wikipedia

http://en.wikipedia.org/wiki/CCSDS_Mission_Operations

Open Source Software

ESA on GitHub

https://github.com/esa

ESA MO OSS Wiki

https://github.com/esa/CCSDS_MO/wiki