



# CCSDS Mission Operations (MO) Services Standards Training

Module 2 - Medium

Training at ESOC  
29 Jan 2019



The subject training material by CNES and ESA is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Public License ([https://github.com/esa/CCSDS\\_MO/raw/master/training/MO\\_Service\\_Training\\_Licence\\_v1.pdf](https://github.com/esa/CCSDS_MO/raw/master/training/MO_Service_Training_Licence_v1.pdf))

- ▶ Introduction
- ▶ Part 1 - MO Framework and Services
  - The MAL and COM
  - Common services
  - Monitor and Control services
  - Forthcoming services
- ▶ Part 2 – A worked example
  - A simple mission
  - Service boundaries
  - MO and our mission
- ▶ Summary

# Introduction

- »» About the presenter
- Aims of the training module

# About the presenter



## MEHRAN SARKARATI



- ▶ Education:
  - MSc in Computer Sciences
  - MSc and PhD. in Aerospace Engineering
  - Berlin University of technology/ Arizona State University
- ▶ Current position (Since 2004):
  - European Space Agency (ESA), Darmstadt, Germany
  - Head of the Applications and Robotic Data Systems Section
- ▶ Role in CCSSD:
  - Chairman of Mission Planning and Scheduling WG
  - Member of SM&C WG and Telerobotics WG

# Aims of the training module



- ▶ The module will focus more on:
  - Overview of the MAL and COM layers
  - Services defined by MO
  - Intended usage of MO Services and how they can be used together to facilitate operational needs
- ▶ In Development MO Services:
  - Mission Data Product Distribution Service
  - Mission Planning and Scheduling Services
  - File Management Service
- ▶ Envisaged MO Services
  - Automation Service
  - Navigation Services
  - ...
- ▶ 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

» 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

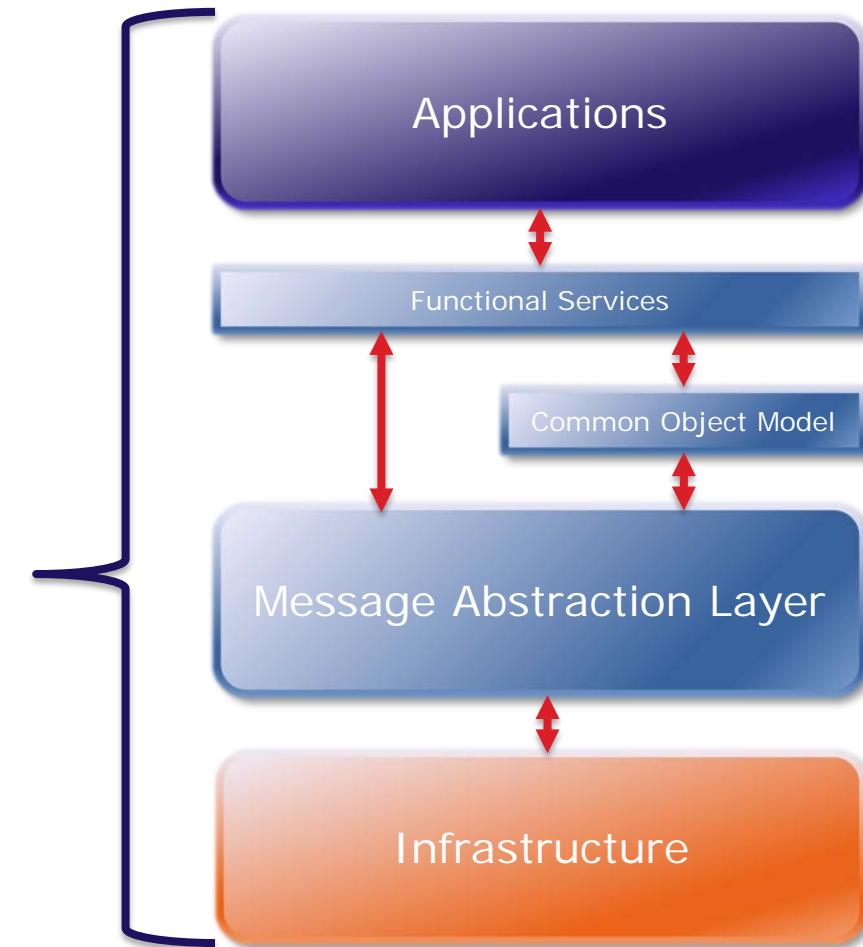


- ▶ A service is a self-contained unit of reusable functionality
  - For example, retrieving an online bank statement
- ▶ Services can be combined to provide more complex functionality

# The MO Framework



- ▶ MO Framework: Core of MO concept
  - 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
  - Future proof technology separation
  - Reuse of other MO services (standard or bespoke)
  - The services specific to your environment can also benefit too

# Message Abstraction Layer



- ▶ The MAL 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

# 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
  - ...

```
<smc:requestIP name="getCurrentTransitionList" number="105" comment="This getCurrentTransitionList operation is a consumer to obtain all of parameters checks fL.">
```



MANAGED RESOURCE FOR ACTIVATION SERVICE OPERATIONS						
Op Identifier	Op Name	Op Type	Op Status	Op Priority	Op Desc	Op Version
105	getcurrenttransitionlist	query	1	1		
106	getcurrenttransitionlist	query	2	1		
107	getcurrenttransitionlist	query	3	1		
108	getcurrenttransitionlist	query	4	1		
109	getcurrenttransitionlist	query	5	1		
110	getcurrenttransitionlist	query	6	1		
111	getcurrenttransitionlist	query	7	1		
112	getcurrenttransitionlist	query	8	1		

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

# MAL runtime aspects



Message Header																	
URI From	Auth Id	URI To	Timestamp	QoS	Priority	Domain	Zone	Session	Interaction	Stage	Trans Id	Area	Service	Operation	Version	Is Error	
Provider	SC X	Broker		BEST	1	A.B.C	GROUND	LIVE	PUBSUB	3		Example	Example	testPubSub	1	FALSE	

List<UpdateHeader>												List<TestNotify>							
List count	UpdateHeader						UpdateHeader						List count	Test Notify		Test Notify			
	Time stamp	Source URI	Update Type	EntityKey			Time stamp	Source URI	Update Type	EntityKey				Time	Value	Time	Value		
				1st	2nd	3rd	4th			1st	2nd	3rd	4th						
2		SC X	Update	A	2	1	2		SC X	Update	A	4	1	2	2	Today, 09:30	1234	Today, 09:30	8888

- ▶ What is a Service in the MO world?
  - A service defines a functional context
    - e.g. Mission Planning is a functional context
  - It specifies the capabilities related to that functional context
  - The service is formally accessible through its service contract
    - Which is in our case the MO service specification in terms of MAL
    - This includes both functional and non-functional aspects of engaging in an interaction with that service
  - A Service is provided by one entity and used (consumed) by another.
- ▶ A Service Specification
  - Lists the contained Operations
    - Support grouping of operations into Capability sets
  - Defines the expected behaviour of the service

- ▶ 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 an instance of 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

- ▶ 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

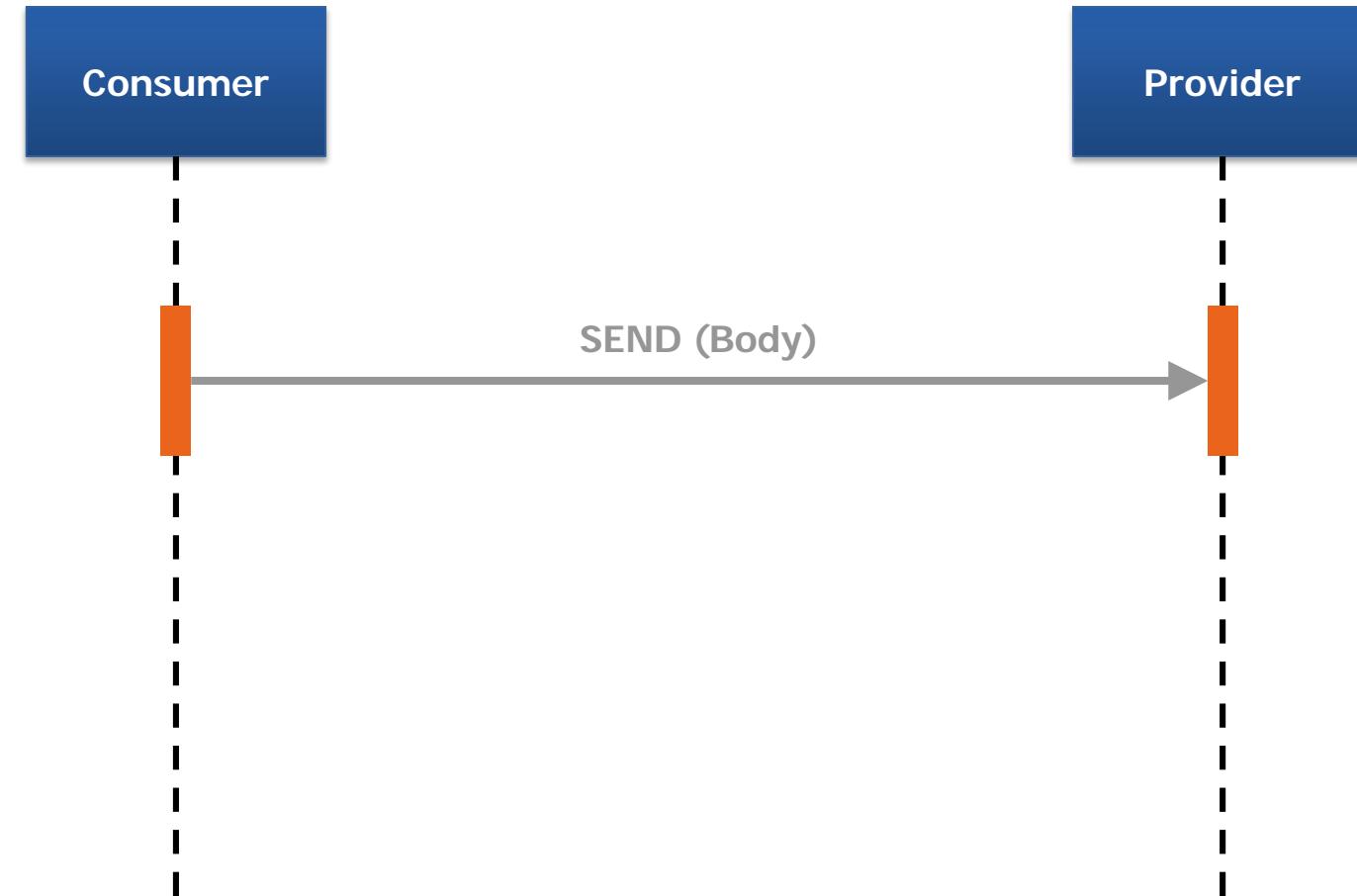
- ▶ Standardised patterns of interaction
  - Provides a template for Service designers
  - Can be considered Operation abstraction
- ▶ Standardises
  - The sequence of messages
  - Some patterns fix certain messages (acknowledgements)
  - Failure points
  - Message header values
- ▶ Provides
  - Simplified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

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

# Interaction Patterns: SEND



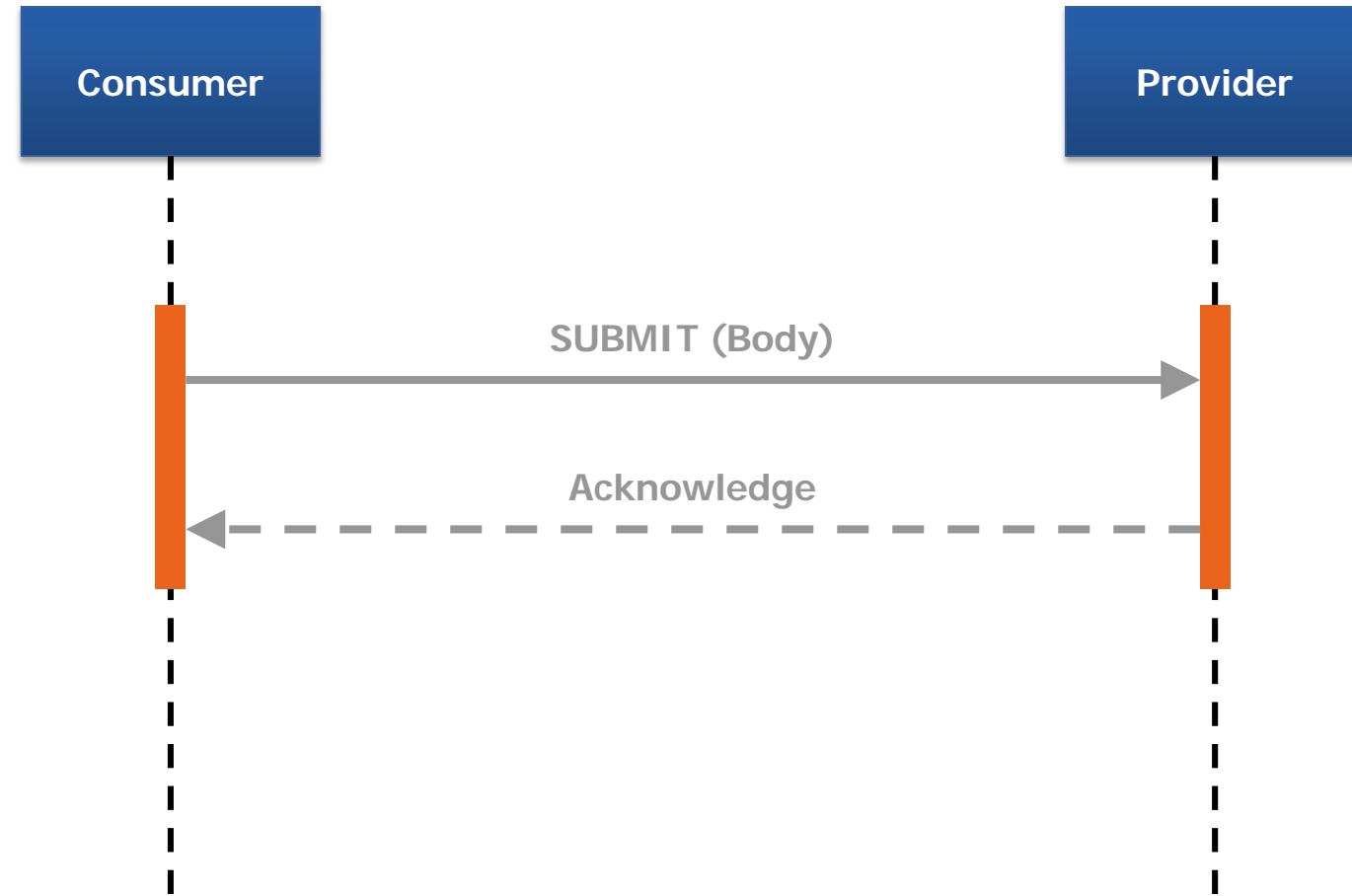
- ▶ Simple unacknowledged operation



# Interaction Patterns: SUBMIT



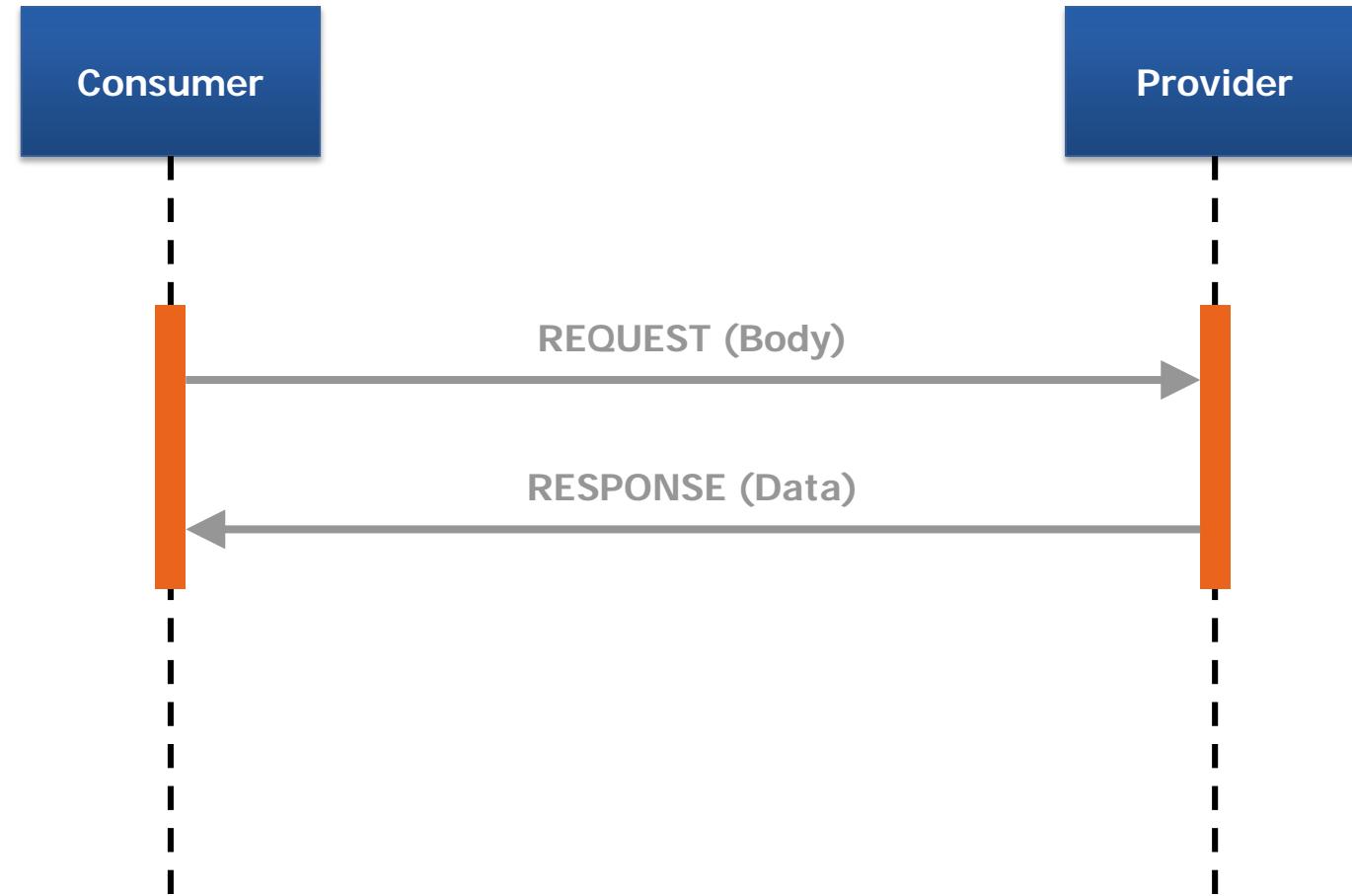
- ▶ Simple acknowledged operation



# Interaction Patterns: REQUEST



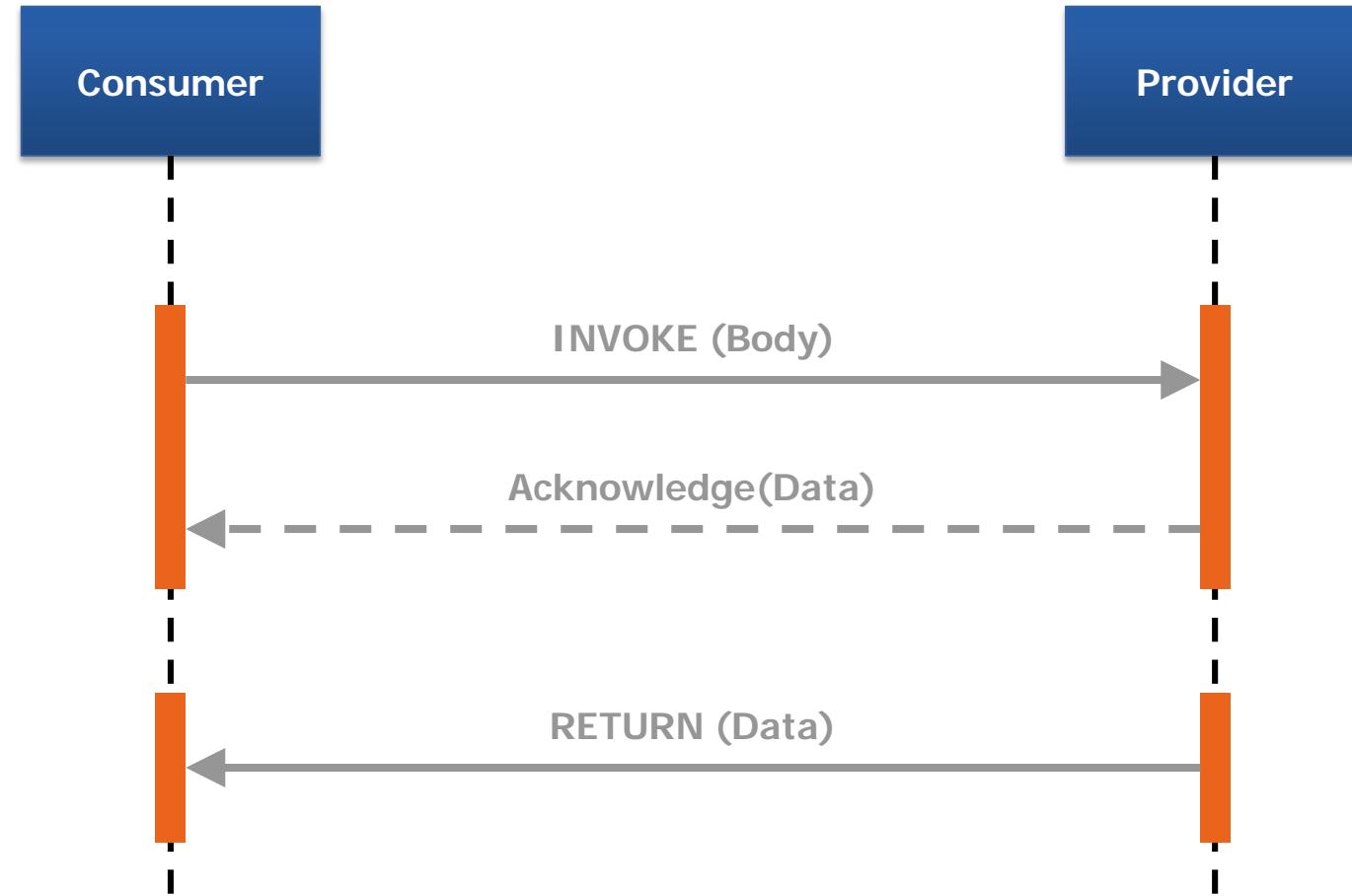
- ▶ Simple request and response operation



# Interaction Patterns: INVOKE



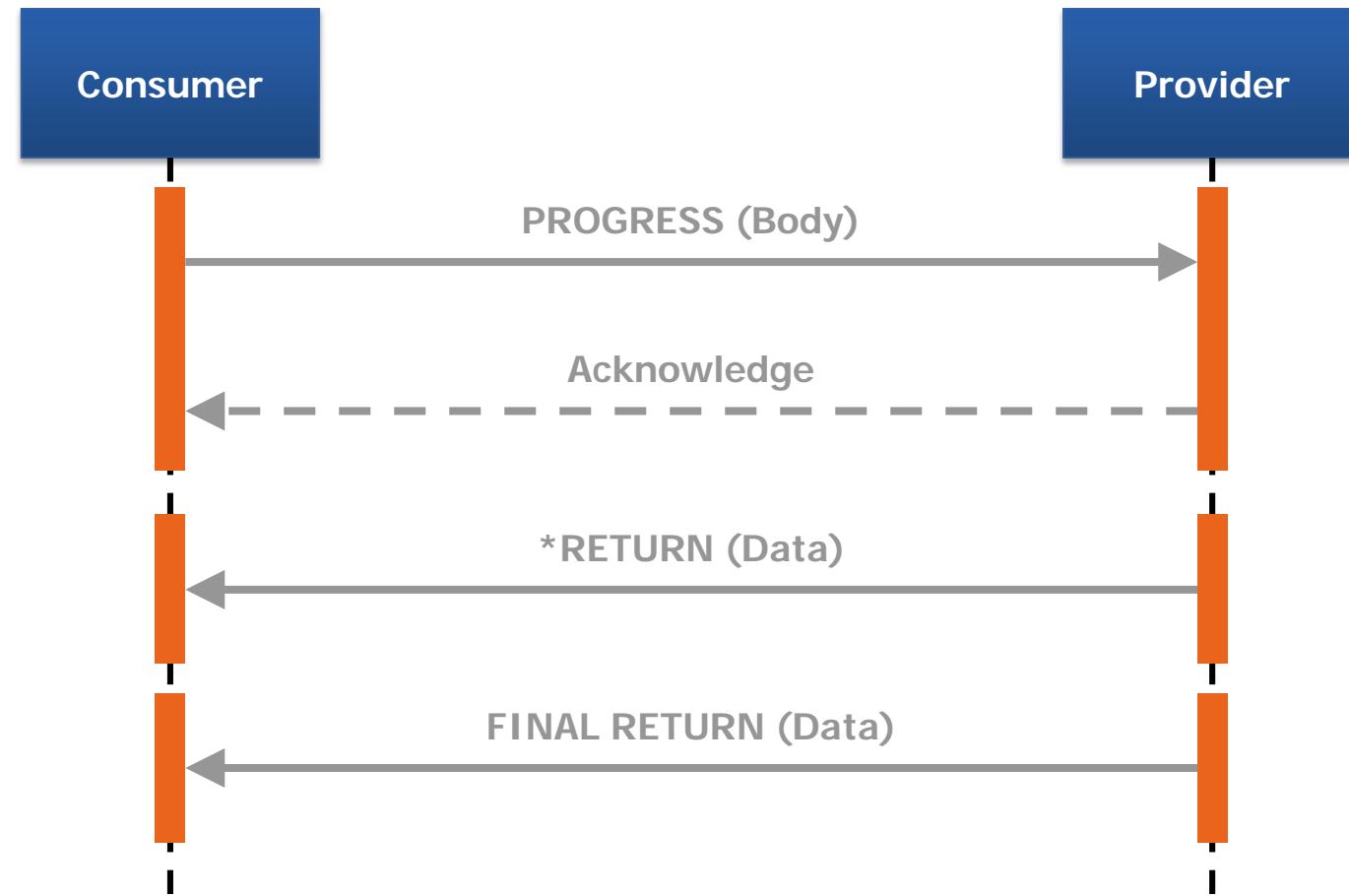
- ▶ Acknowledged data request operation



# Interaction Patterns: PROGRESS



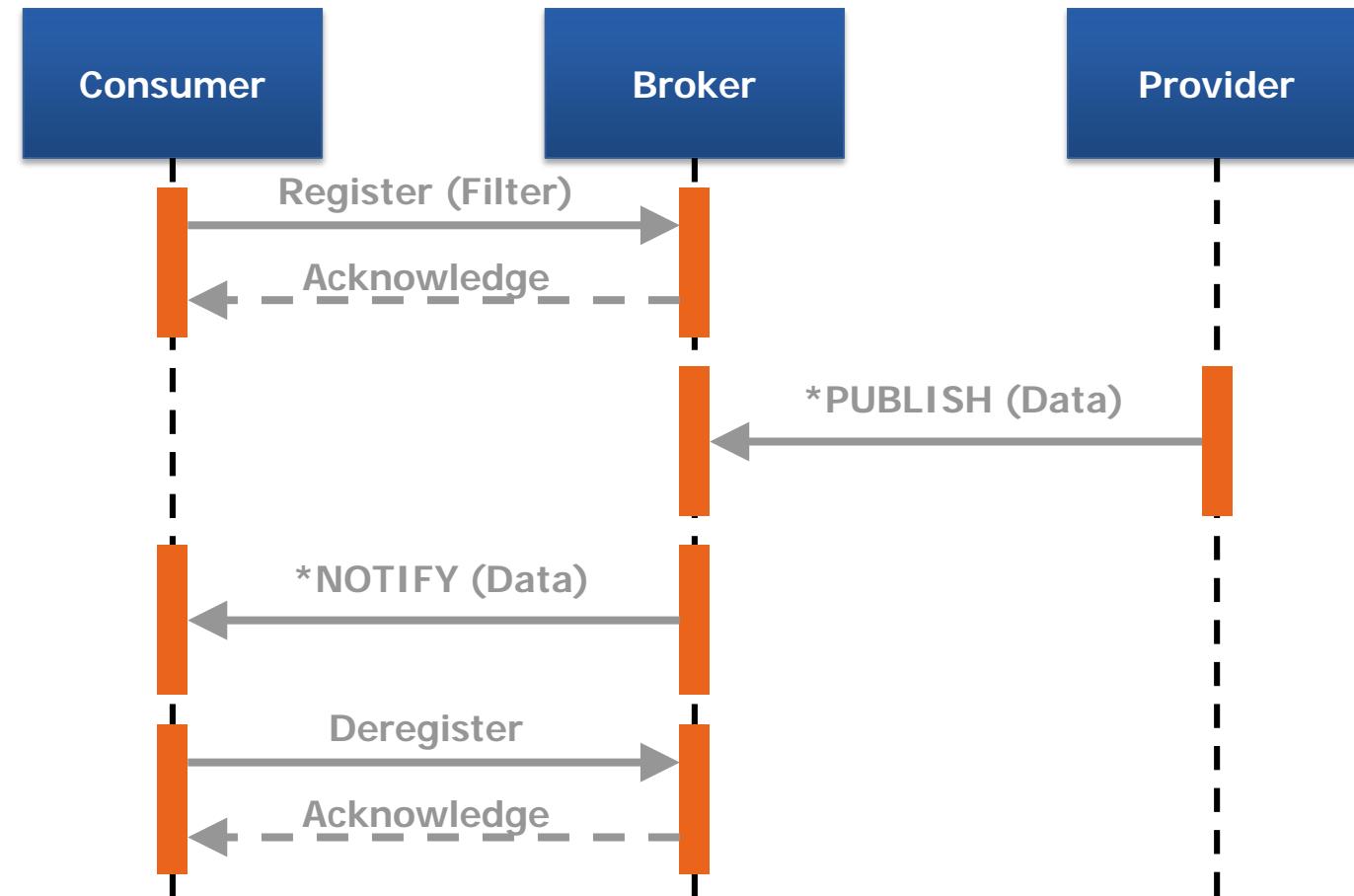
- ▶ Acknowledged multi-return data request operation



# 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	<a href="#"><u>retrieve</u></a>	1	Yes	
PROGRESS	<a href="#"><u>query</u></a>	2	Yes	1
INVOKE	<a href="#"><u>count</u></a>	3	Yes	
REQUEST	<a href="#"><u>store</u></a>	4	No	2
SUBMIT	<a href="#"><u>update</u></a>	5	No	3
REQUEST	<a href="#"><u>delete</u></a>	6	No	4

# Understanding the books: Operations



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

Operation Identifier	retrieve	
Interaction Pattern	INVOKER	
Pattern Sequence	Message	Body Type
IN	INVOKER	objType : ( <a href="#">ObjectType</a> ) domain : (List<MAL::Identifier>) objInstIds : (List<MAL::Long>)
OUT	ACK	
OUT	RESPONSE	objDetails : (List< <a href="#">ArchiveDetails</a> >) 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	<a href="#">ObjectDetails</a>	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	Expression Operator	
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).

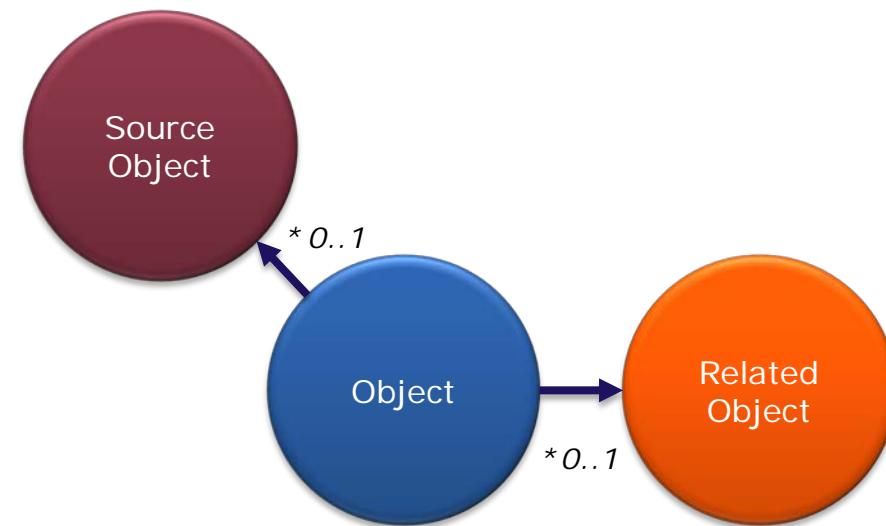


- ▶ The COM specification specifies:
  - A standard object model for services to use
  - Three services that support that model
- ▶ The object model:
  - Provides a standard data object model for MO Services to utilise
  - Provides the building blocks for the specification of the data objects of a service
  - This builds upon the MAL to define a standard data model for an MO service
- ▶ The support services are:
  - Activity Tracking
  - Event
  - Archive

- ▶ 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 that utilises the COM must define the types of object that form the data model of the service
- ▶ An instance of a COM Object is uniquely identified by:
  - Its domain
  - Its COM Object Type
  - Its object instance identifier, unique to the domain and object type

- ▶ 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

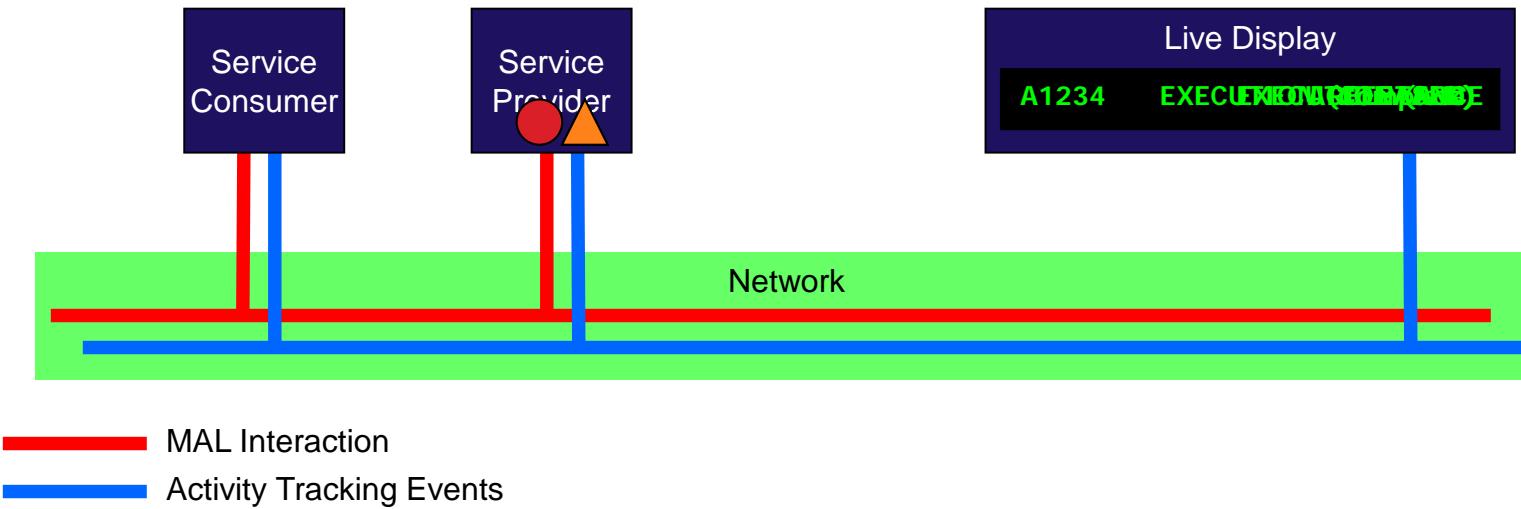


- ▶ An event is a specific object representing  
*'something that happens in the system at a given point in time'*
- ▶ Each service can define the possible Events that it may raise
  - This information is held in the service XML
- ▶ An event is a COM object
  - The body of an event object may contain extra information
    - Specified by the service that declares the event
- ▶ The object links of the event object are:
  - The source links to the object that caused the event to be generated
  - The related link is service and event-type specific
  - Both may be changed by the service defining the event
- ▶ The event service defines a single publish/subscribe operation
  - For publishing events
  - But also monitoring of events generated by other components

- ▶ An activity is anything that has a measurable period of time
  - A command
  - A remote automated procedure
  - A schedule
  - ...
- ▶ The Activity Tracking service provides the ability to track the progress of these activities
- ▶ The basic service provides the ability to track the progress of MAL operations
  - It is further expected to be used for other activities where appropriate
- ▶ It defines an event pattern that supports the reporting of the progress of activities from
  - The initial consumer request
  - Tracking its progress across a transport link
    - Including if being relayed
  - To reception by the provider
  - And execution in that provider.
- ▶ The service uses the COM Event service to report the progress of activities

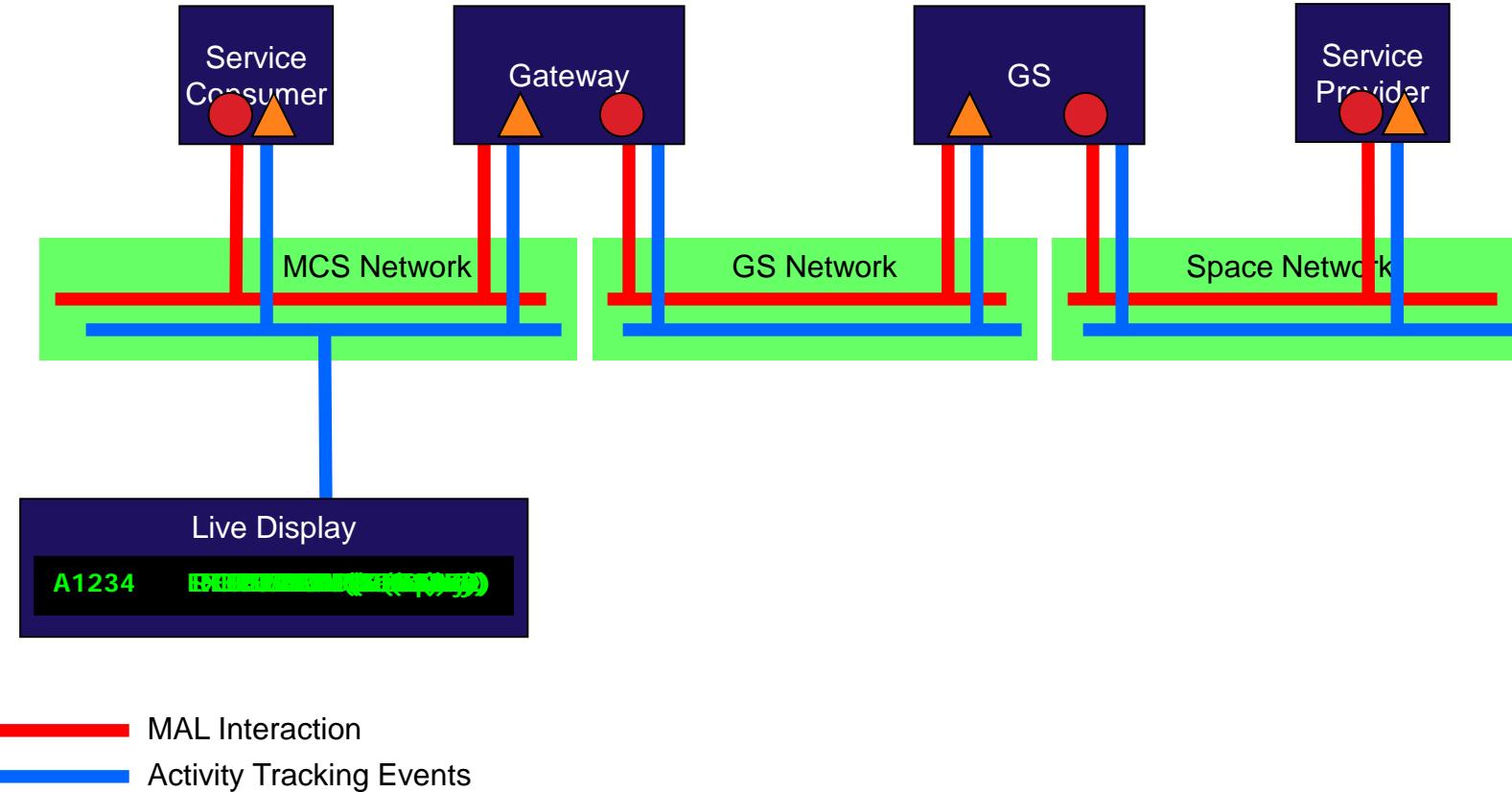
- ▶ Four transport notification stages are defined
  - RELEASE              Release from source node
  - RECEPTION          Reception by an intermediate node
  - FORWARD             Forward is release from an intermediate node
  - ACCEPTANCE         Acceptance is reception by destination node
- ▶ A single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

# 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



- ▶ The Activity Tracking Events are standard COM Events
  - Applications receive these through a subscription to the COM Event service
- ▶ If a failure Event is being generated by one of the nodes
  - That node is also required to fail the originating MAL interaction
- ▶ The consumer does not see the Activity Tracking events through the originating MAL interaction
  - As far as the MAL is concerned they are different transactions

- ▶ The Archive service provides a basic archiving function for COM objects
- ▶ It follows the Create Retrieve Update Delete (CRUD) principles
  - It supports simple querying of the archive
    - More complex queries are supported but the specifications of these are outside this standard
- ▶ It uses the COM Object Key as the archive primary key
  - These are Domain, Object Type, and Object instance identifier
- ▶ As changes are made during the lifetime of an object
  - This information is distributed to consumers using the service defined operations
  - As long as these updates follow the COM standard for object identification they can also be stored in a COM archive
- ▶ By storing these updates in an archive
  - Any historical replay/retrieval functions can correctly reflect the history of the objects.

# 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

- ▶ 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

- ▶ The interface of a provider can be specified using either
  - A bespoke service specification
  - Or by defining it in the Action service
- ▶ Actions are defined dynamically via the Action service
  - As opposed to being defined in a service specification
- ▶ The Action Service operations include:
  - Pre checking of action invocation requests
  - Issuing of action execution requests
  - List, get, add, remove of action definitions
- ▶ The advantage of the Action service approach is that
  - It allows the consumer and provider implementation to be independent of the specifications of the service
- ▶ The disadvantage of the Action service approach is that
  - The richness of the MAL interaction patterns and the service specifications cannot be defined as rigorously via the Action definition

- ▶ 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

- ▶ 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
  - They may also be managed via the Common Configuration 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
  - They may also be managed via the Common Configuration service

- ▶ The Check service allows the definition of 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

- ▶ 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

- ▶ 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
  - Predefined 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

- ▶ A functional extension of the other services is to add the engineering unit conversion capability
- ▶ The Conversion service defines
  - A basic set of conversion patterns
  - Instances of these conversions are defined using the COM archive
  - Other services then reference these Conversions as appropriate to those services
- ▶ The Conversion service does not provide any operations but specifies:
  - The structures used to define the conversions
  - How they use the COM structures
  - How it should be used with the COM archive

- ▶ 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

# Common Services



- ▶ The Common services are the following:

- Directory
  - Service lookup
- Login
  - Operator log in
- Replay
  - Replay session management and control
- Interaction
  - Application and Operator interaction
- Configuration
  - Service provider configuration management

- ▶ 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	<a href="#">lookupProvider</a>	1	Yes	1
REQUEST	<a href="#">publishProvider</a>	2	No	2
SUBMIT	<a href="#">withdrawProvider</a>	3	No	
REQUEST	<a href="#">getServiceXML</a>	4	No	

- ▶ The getServiceXML operation allows a consumer to request the MO XML specification of all service published by a provider
  - Support standard as well as proprietary services
  - Returns a list of XML files
- ▶ XML needs to be supplied by the service provider when adding its details to the Directory service
  - If it is not supplied then nothing is returned
- ▶ This allows the building of generic service consumers that use the service XML to define what they can do at runtime

- ▶ The Login service defines the primary mechanism for the submission of authentication details to a deployment specific security system
- ▶ It supports operations to allow a user to:
  - Login
  - Logout
  - Report available roles
  - Handover to another user
- ▶ The service is closely tied to the Access Control aspect of the MAL

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	<a href="#">login</a>	1	No	1
SUBMIT	<a href="#">logout</a>	2	No	
REQUEST	<a href="#">listRoles</a>	3	No	2
REQUEST	<a href="#">handover</a>	4	No	3

- The Configuration 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

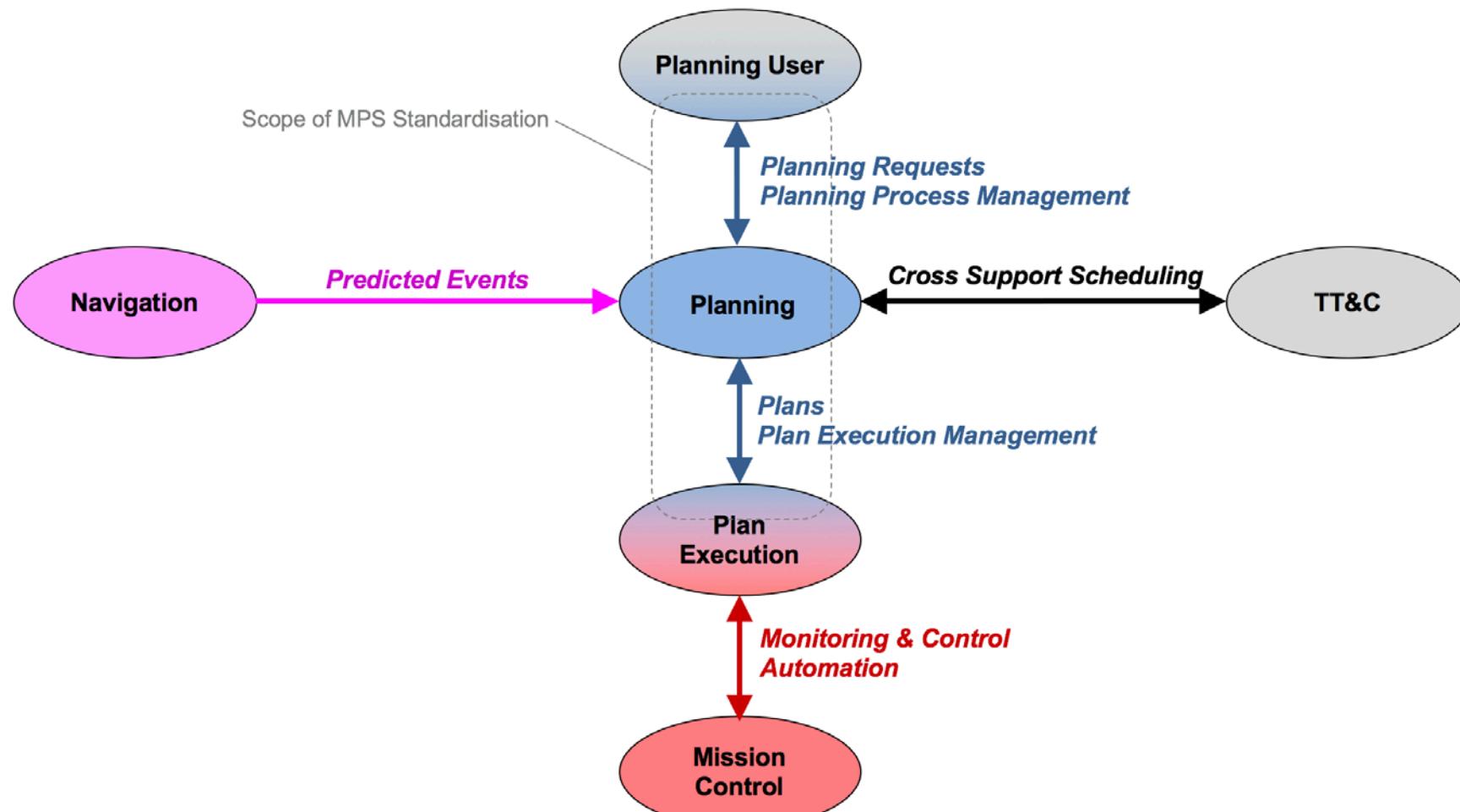
Area Identifier	Service Identifier	Area Number	Service Number	Area Version
Common	Configuration	3	5	1
Interaction Pattern	Operation Identifier	Operation Number	Support in Replay	Capability Set
SUBMIT	<a href="#">activate</a>	1	No	1
REQUEST	<a href="#">list</a>	2	No	2
REQUEST	<a href="#">getCurrent</a>	3	No	
REQUEST	<a href="#">exportXML</a>	4	No	3
SUBMIT	<a href="#">add</a>	5	No	4
SUBMIT	<a href="#">remove</a>	6	No	
INVOKE	<a href="#">storeCurrent</a>	7	No	5
REQUEST	<a href="#">importXML</a>	8	No	6

# In-development Services

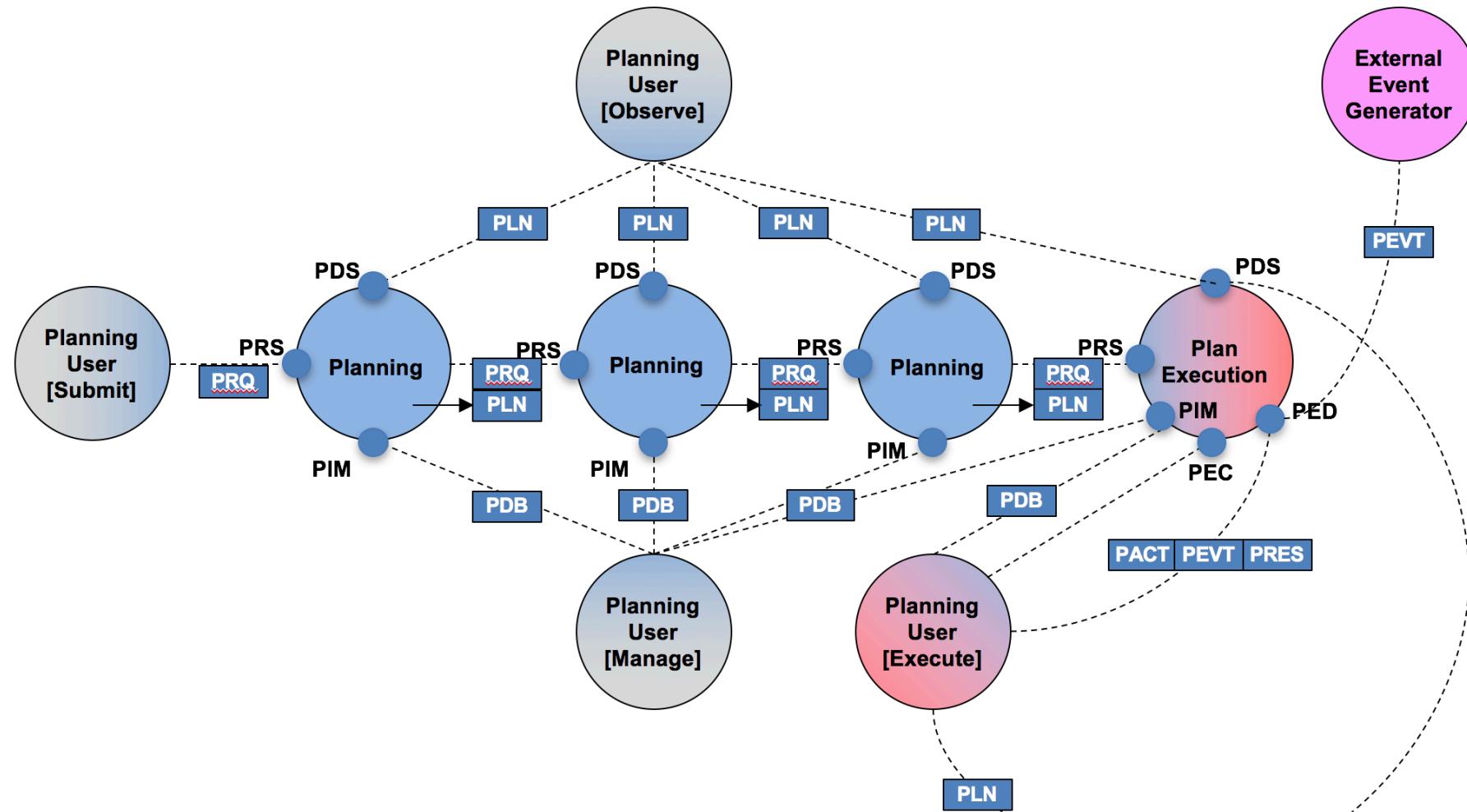


- ▶ The Planning services are concerned with the planning aspect of a mission:
  - Planning Request [**PRS**]
  - Plan Distribution [**PDS**]
  - Plan Execution Control [**PEC**]
  - Plan Information Management Service [**PIM**]
  - Plan Edit Service [**PES**]
- ▶ Potential consumers include:
  - Operations 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

# Mission Planning and Scheduling Services



# Mission Planning and Scheduling Services



Area Identifier	Service Identifier	Area Number	Service Number	Area Version
MP	<u>PlanningRequest</u>	5	1	1
Interaction Pattern	Operation Identifier	Operation Number	Support in Replay	Capability Set
REQUEST	<u>submitRequest</u>	1	No	1
REQUEST	<u>updateRequest</u>	2	No	
SUBMIT	<u>cancelRequest</u>	3	No	
REQUEST	<u>getRequestStatus</u>	4	No	2
PUBLISH-SUBSCRIBE	<u>monitorRequestStatus</u>	5	No	
PROGRESS	<u>getRequest</u>	6	No	3

Area Identifier	Service Identifier	Area Number	Service Number	Area Version
MP	<u>PlanDistribution</u>	5	2	1
Interaction Pattern	Operation Identifier	Operation Number	Support in Replay	Capability Set
REQUEST	<u>listPlans</u>	1	No	1
REQUEST	<u>getPlan</u>	2	No	2
REQUEST	<u>getPlanStatus</u>	3	No	3
PUBLISH-SUBSCRIBE	<u>monitorPlan</u>	4	No	4
PUBLISH-SUBSCRIBE	<u>monitorPlanStatus</u>	5	No	5
PROGRESS	<u>queryPlan</u>	6	No	6

- ▶ 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 them once they are available
- ▶ The two request types can be set up as either
  - Ad hoc request
  - A predefined standard order style request

# Mission Data Product Distribution Service



*"to provide controlled access to mission data for the community of users who do not have access to mission-control-system monitoring and control facilities."*

Service Identifier	Area Number
MissionDataProductDistribution	9
Operation Identifier	Operation Number
requestProduct	1
suspendRequest	2
resumeRequest	3
cancelRequest	4
getRequestStatus	5
queryProductRequest	6
monitorProduct	7
getProductsCatalogue	8
getProductSpecification	9

Service Identifier	Area Number
ProductsManagement	9
Operation Identifier	Operation Number
addProductType	1
removeProductType	2
addProductSource	3
removeProductSource	4
addProductFormat	5
removeProductFormat	6
addProductDetail	7
removeProductDetail	8
setProductSpecification	9
removeProductSpecification	10
addProductAttribute	11
removeProductAttribute	12
enableStreaming	13
disableStreaming	14

- ▶ What is a Mission Data Product

*"A set of the space mission data available for the user community."*

## How to specify a product?

- **type** - represents a category of mission data, which the product belongs to,
- **source** - is an indication of the origin of the data,
- **format** - is a standardized description detailing how to encode product.

a category of mission data:

- parameter value (or set),
- command history,
- packets,
- image files,...

Name	ProductTypeDetails
Field	Type
name	<a href="#">MAL:Identifier</a>
description	<a href="#">MAL:String</a>
productAttributeIds	<a href="#">List&lt;MAL:Long&gt;</a>

Each product type has a **list of attributes** associated with it.

Attributes define the fields (names, data types, and usage).

Some attributes can be selected to filter or sort products.

Examples:

- **RAW** and **JPEG** formats for images taken by OBSERVER2
- **Excel** file with a report containing reads from heating components
- **PDF** file with human-readable parameter representations of a list of commands in the onboard queue

Name	ProductFormatDetails
Field	Type
name	<b>MAL:Identifier</b>
description	<b>MAL:String</b>

Example

- *type* – **image**
- *source* – **esa.mission\_name.startracker-1**
- *format* – **RAW**

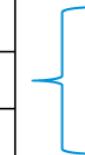


## Product [Provisioning] Details

A group of parameters of the related product. It contains:

- a list of formats (e.g. an image can be JPG or PNG),
- a provision mode,
- a list of compression methods,
- a list of available encryption methods

Name	ProductProvisioningDetails
Field	Type
productFormats	List<MAL:Long>
provisionMode	ProvisionMode
compression	List<MAL:UInteger>
encryption	List<MAL:UInteger>



Name	ProvisionMode
Enumeration Value	Numerical Value
BATCH	1
STREAM	2

## Product Specification 1/2

### XML File

- A set of *metadata attributes* of a product of a given type
- Independent of the format used to encode the product
- MAL data type description language (separate from any encoding used by the product formats)

e.g. ParameterDefinition composite (next slide)

# Mission Data Product Distribution Service



Name	ParameterDefinitionDetails
Field	Type
description	MAL:String
rawType	MAL:Octet
rawUnit	MAL:String
generationEnabled	MAL:Boolean
reportInterval	MAL:Duration
validityExpression	ParameterExpression
conversion	ParameterConversion

```
<mal:composite name="ParameterDefinitionDetails" shortFormPart="1">
  <mal:extends><mal:type name="Composite" area="MAL"/></mal:extends>
  <mal:field name="description" canBeNull="false">
    <mal:type name="String" area="MAL"/>
  </mal:field>
  <mal:field name="rawType" canBeNull="false">
    <mal:type name="Octet" area="MAL"/>
  </mal:field>
  <mal:field name="rawUnit">
    <mal:type name="String" area="MAL"/>
  </mal:field>
  <mal:field name="generationEnabled" canBeNull="false">
    <mal:type name="Boolean" area="MAL"/>
  </mal:field>
  <mal:field name="reportInterval" canBeNull="false">
    <mal:type name="Duration" area="MAL"/>
  </mal:field>
  <mal:field name="validityExpression">
    <mal:type name="ParameterExpression" area="MC"/>
  </mal:field>
  <mal:field name="conversion" comment="If present then parameter has a conversion">
    <mal:type name="ParameterConversion" service="Parameter" area="MC"/>
  </mal:field>
</mal:composite>
```

- ▶ The File services are split into two aspects:
  - Management of a remote file store
  - 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
  - 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

# Forthcoming 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

- ▶ 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

- ▶ 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 address systems that use on-board file-based systems

- ▶ 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 it does define how time is reported

- ▶ Concept Book – Green Book published
- ▶ WG is dormant
- ▶ Robotic Systems make use of other domain services (e.g. SM&C, P&S, NAV, ...)
- ▶ Domain Specific Services for Robotics such as
  - Motion Control
  - Localisation
- ▶ Prototypings performed over the Space Link from ISS in METERON Experiments

# Part 2

»» A Worked Example

# A Simple Mission



- ▶ Is there such a thing?!

A word cloud visualization centered around the term "packet system". The words are arranged in a circular pattern, with larger words in the center and smaller words radiating outwards. The words are color-coded in shades of blue, grey, and white. The central cluster includes "packet system", "command", "power", "bus", "data", "frequency", "signal", "spacecraft", "time", "control", "antenna", and "operations". Other prominent words include "processor", "ground", "satellite", "TC", "frame", "earth", "RF", "orbit", "solar", "application", "network", "temperature", "model", "axis", "eclipse", "decoding", "bandwidth", "process", "telemetry", "OBSW", "standard", "uplink", "redundancy", "users", "band", "protocol", "reception", "thruster", "baseband", "information", "coding", "mode", "segment", "space", "battery", "operating", "launch", "memory", "module", "CCSDS", "downlink", "radiation", "mission", "sensor", "parameter", "pointing", "team", "schedule", "payload", "budget", "thermal", "gyro", "reporting", "communication", "CPU", "tracking", "TM", "modulation", "LEOP", "sun", and "software".

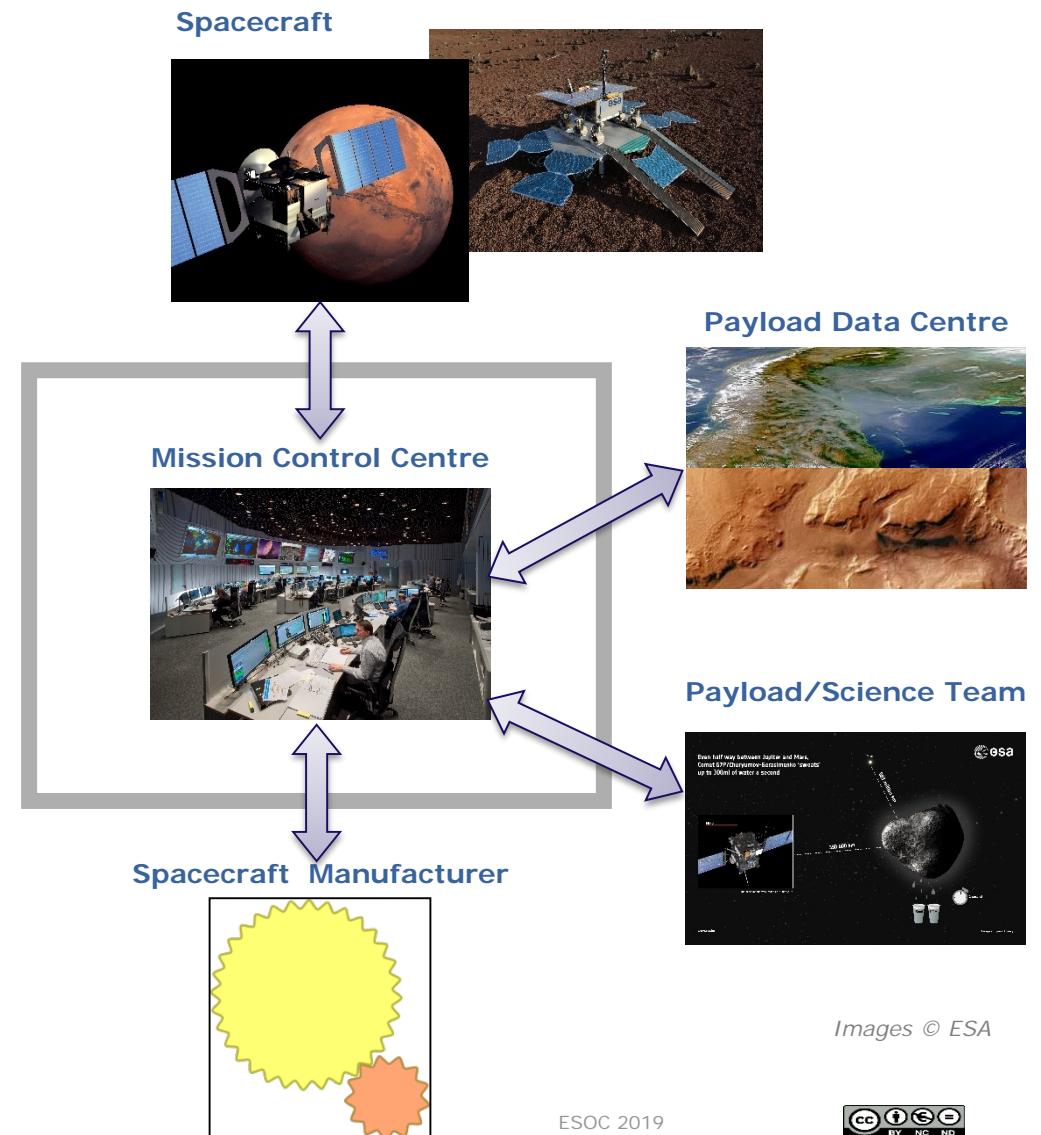
- ▶ Assume a single:
    - Spacecraft
    - Ground station
    - Control centre



# 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

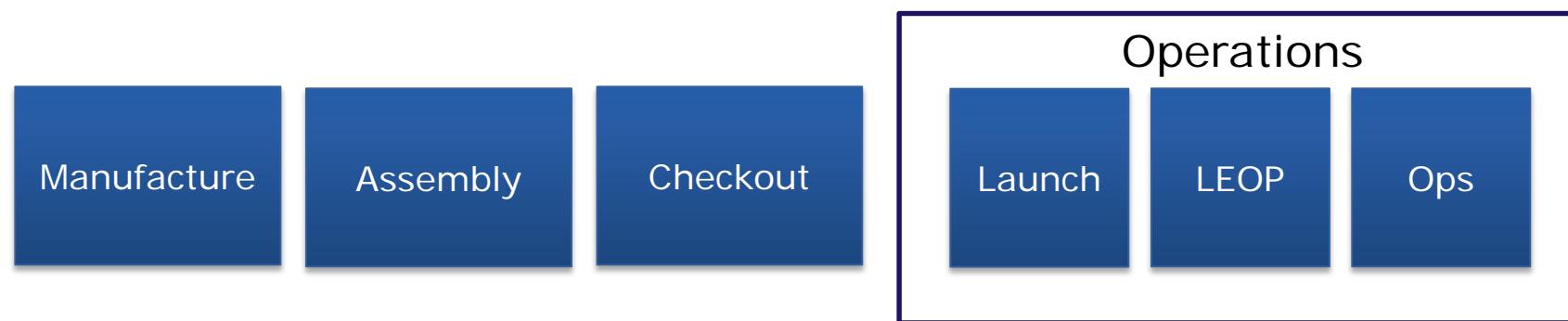
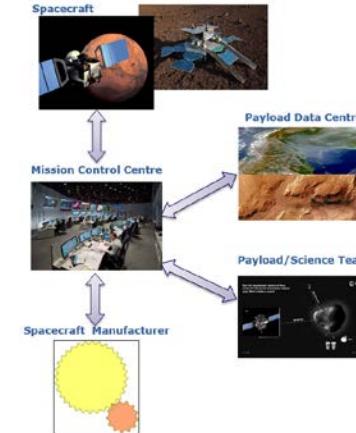


Images © ESA

# 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 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



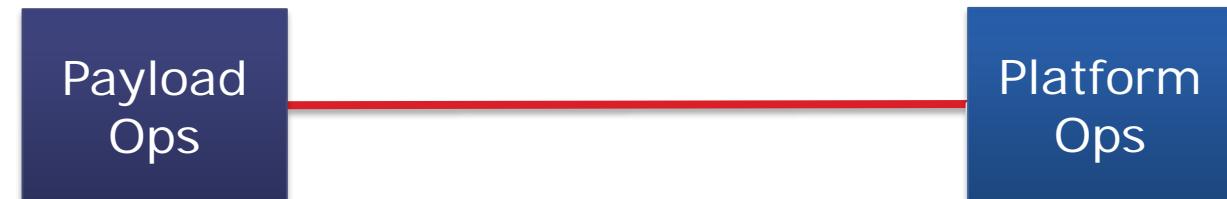
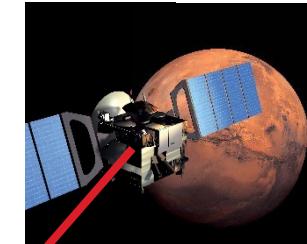
- ▶ In our example we will use two services:
  - The Aggregation service for live housekeeping telemetry
  - The MDPD service for off-line delivery of other telemetry
- ▶ Our service provider is the platform operation centre
- ▶ Our service consumer is the payload operations centre
- ▶ Two scenarios exist for our services
  - The first is the reception of live Aggregations during a contact
  - The second is the reception of off-line data via the MDPD service

- ▶ 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 or 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

- ▶ 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 entity identifiers 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

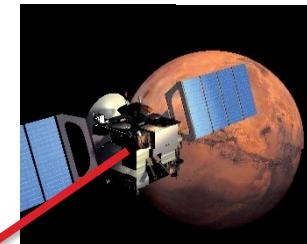
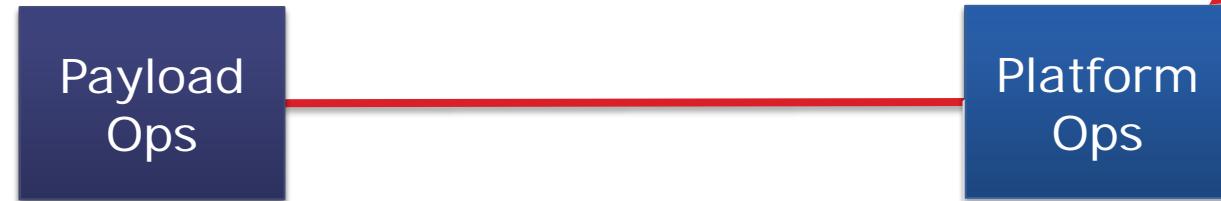
- ▶ We receive the Aggregation updates in a NOTIFY message from the broker, which contains
  - The subscription identifier
  - A list of update headers
  - A list of matching updates
- ▶ Each update header contains:
  - Timestamp of the update
    - This can be different to the timestamp in the NOTIFY message header
  - Source URI
    - This can be different to the source URI in the NOTIFY message header
  - Update type
    - Enumeration containing [CREATION, UPDATE, MODIFICATION, DELETION]
  - The entity key of the update

- ▶ For an Aggregation value update it contains:
  - Generation mode
  - Boolean on whether it was filtered or not
  - A list of parameter values
    - This also includes optional delta times for when parameter values do not all have the same sampling time
    - The actual parameters it includes is specified in the Aggregation definition
- ▶ So, this complete sequence would look like:
  - Lookup
  - Connect
  - Register
  - Receive
  - Deregister
  - Disconnect



- ▶ 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 standing order is like a normal PUB-SUB subscription
  - It holds a list of required data
- ▶ For the MDPD standing order, it is setup in an earlier phase
  - This can be either dynamically where the MDPD service is used to set up the standing order
  - Or using some out of band agreement

- ▶ 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



# 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*
- ▶ *An open architecture and framework provides*
  - *Independence 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*

# 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
- ▶ 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
- ▶ To alleviate this CCSDS is defining and standardising:
  - A set of services for Mission Operations
  - A open framework, independent from technology
- ▶ 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*

## Links

[CCSDS](#)

[Website](#)

[MO in Wikipedia](#)

[ESA OSS](#)

[ESA MO Wiki](#)

[ESA GitHub](#)

[ESA OSS License](#)