25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

A modeling pattern for layered system interfaces

Peter Shames
Marc Sarrel
Jet Propulsion Laboratory
California Institute of Technology

INCOSE International Symposium
July 15, 2015

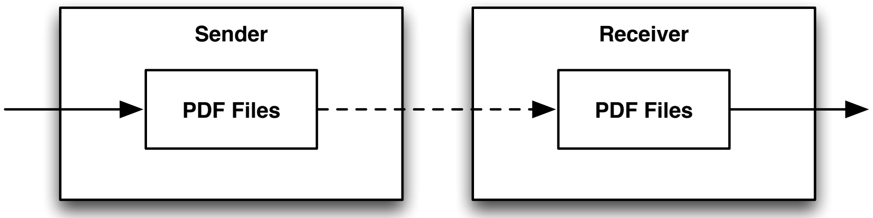
Overview

- Interfaces are the heart of systems engineering.
- In many kinds of systems we must specify and realize an interface at several levels of abstraction simultaneously.
- All levels must be correct for the interface to be correct as a whole.
- This model is well demonstrated by the success of computer network protocol stacks, e.g. the OSI model.
- In this presentation, we'll show one way that layered interfaces, using OSI as an example, can be represented in SysML.
- We hope to lay the foundation for the future application of this layering pattern to other forms of interfaces like electrical, mechanical, thermal, etc.

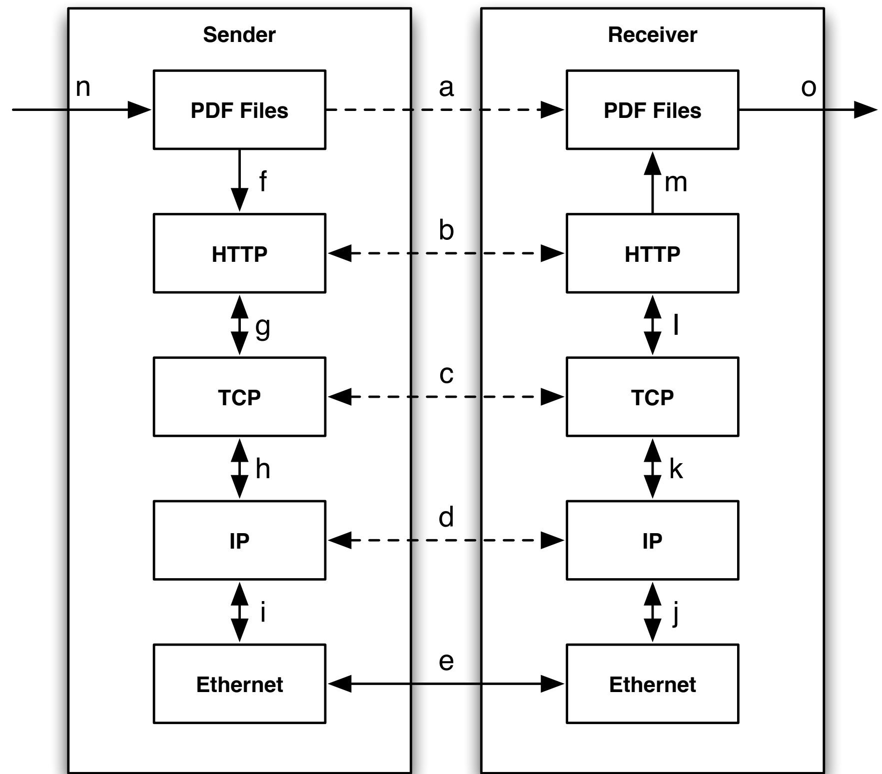
Motivation

- This work came out of a task to re-engineer the three space-communication networks run by NASA.
  - Deep Space Network
  - Near Earth Network
  - Space Network (TDRSS)
- Give space missions a unified interface to the capabilities and services of those three networks (planning, scheduling, uplink, downlink, etc).
- Share implementation and operations of those capabilities and services across the three networks.
- Needed to model multiple ways to implement a given data exchange.
- Same technique quickly found application in integrating the flight with the ground system on a human spaceflight mission (i.e. Exploration Flight Test-1).

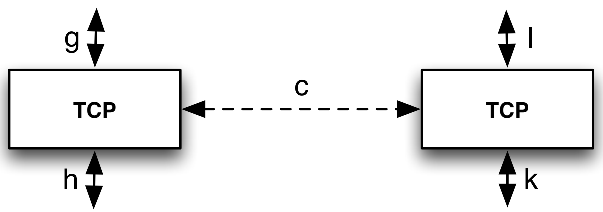
A Simple Example

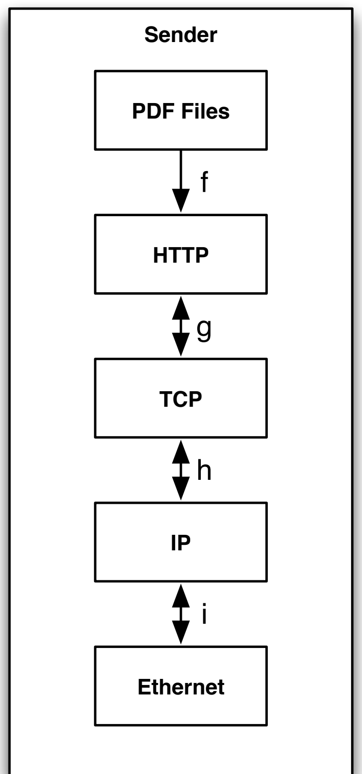


- Send a PDF file from A to B.
- This is the requirement, what the user sees.
- It’s implemented with HTTP, TCP, IP and Ethernet.
- Each component is connected both horizontally and vertically.



We Will Control the Horizontal We Will Control the Vertical





- We can slice this matrix in either direction.
- Separation of concerns.
- We can focus on just the TCP layer.
  - How it is connected (horizontally).
  - How it behaves (horizontally).
- We can focus on just the Sender.
  - How it is connected (vertically).
  - How it behaves (vertically).
- Structure and behavior work both vertically and horizontally.

Concerns

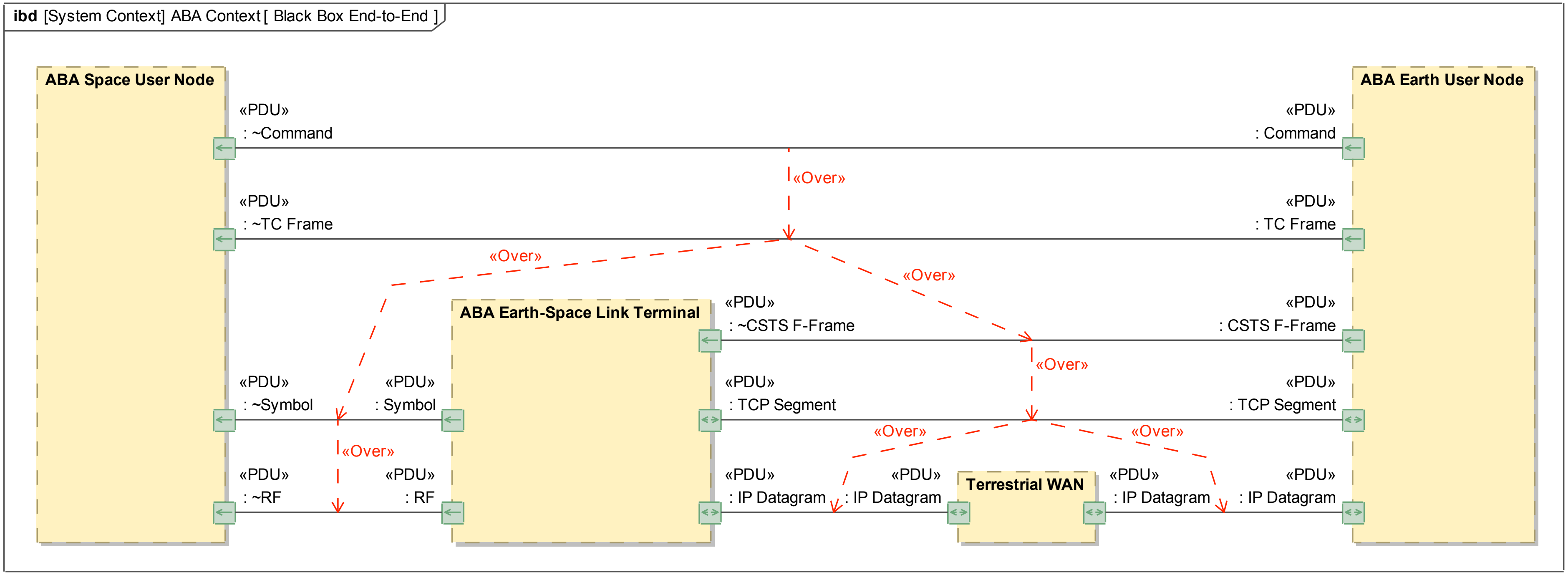
- Project model in ways that are relevant to stakeholders.
- Viewpoints that address concerns, per ISO 42010.

	Concern	View
1	What is the end-to-end construction of the system in terms of major elements?	End-to-End black box view
2	What is the specific stack of protocols needed in each element?	Protocol stack view
3	What is the behavior within a given protocol layer?	Protocol state machine view
4	What are the standards or specification that govern the behavior of each layer?	Interface binding view
5	How are the protocol stacks deployed, end-to-end in order to meet the system requirements?	End-to-End white box view
6	What is the end-to-end behavior or performance characteristics along a given connector as constrained by lower-level connectors?	End-to-end constraints and analysis

A Tale of Two Boxes – Black Box

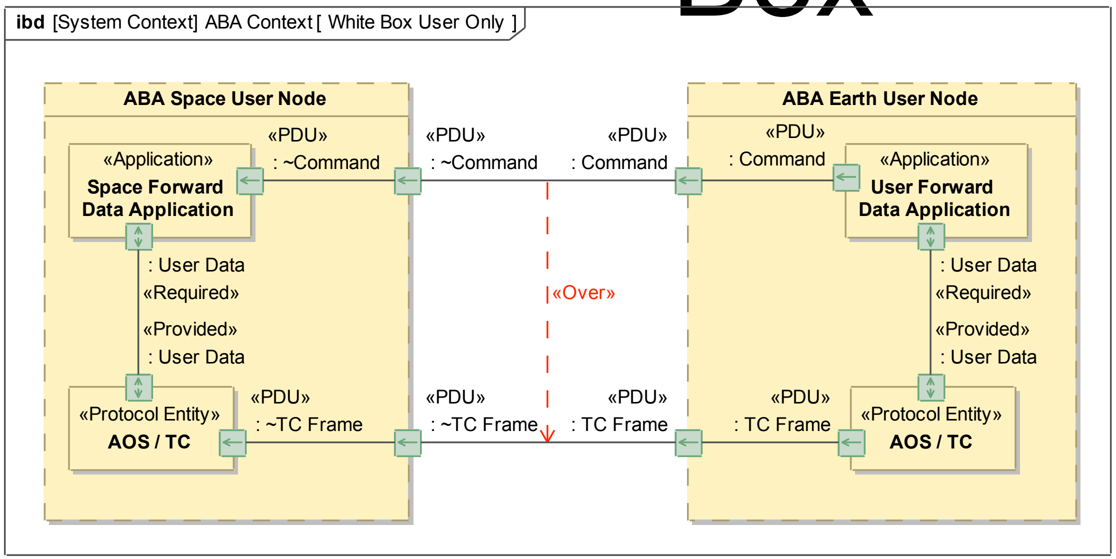
- Transition to a different example, in SysML notation.
- Send a command from the Earth node to the Space node (right to left).
- This is the logical connection.
- In this example, not only are there layers below, but intermediate systems that appear only at the lower layers.
- This is a Black Box view in the sense that we see no internals of the Space or Earth nodes.
- Intent is to describe connections between systems.

A Tale of Two Boxes – Black Box



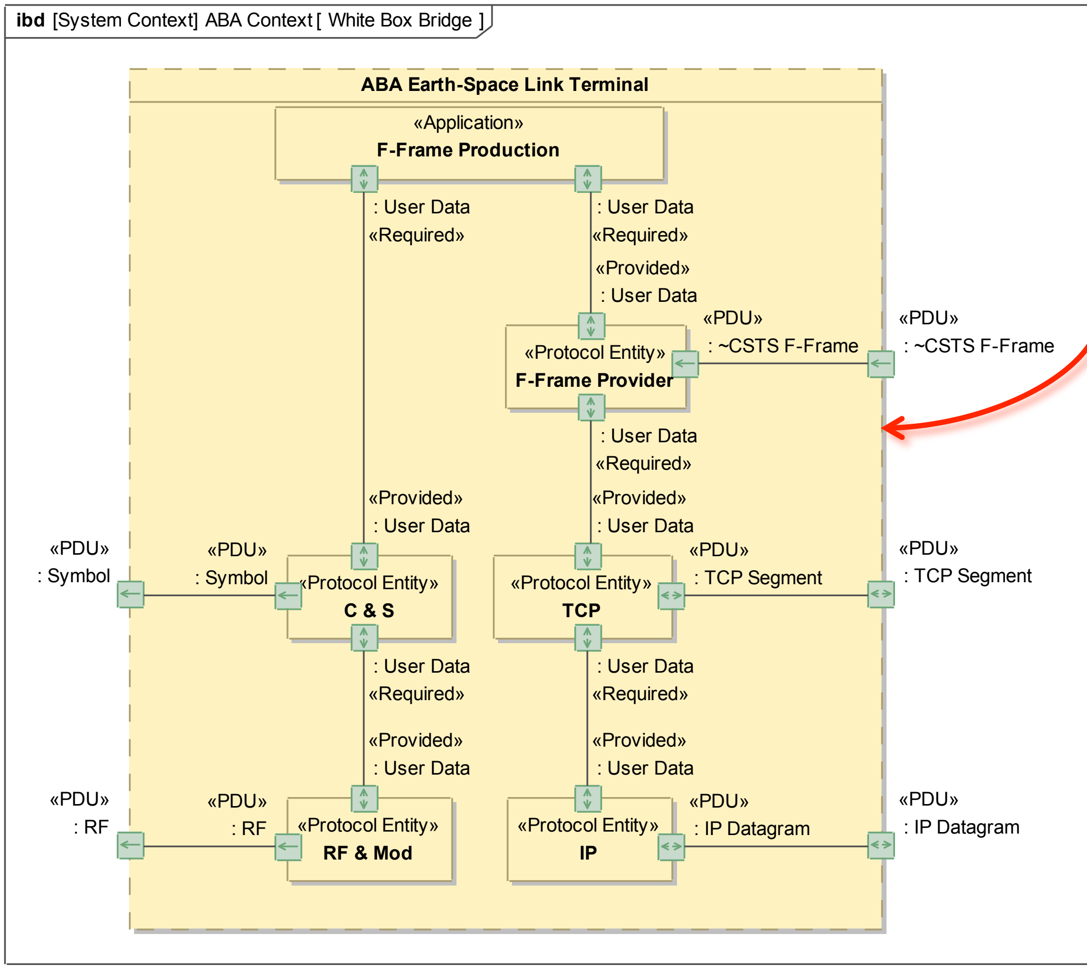
- Correspondence between upper and lower layers (red arrows).
- Layering changes, ground only vs. flight-ground.
  - More detail in flight-ground portion.
- A higher layer might have several possible sets of lower layers.
- A lower layer might carry several higher layers.
- Allows analysis of interaction between higher layers.
  - e.g. command and telemetry over the same TCP link.

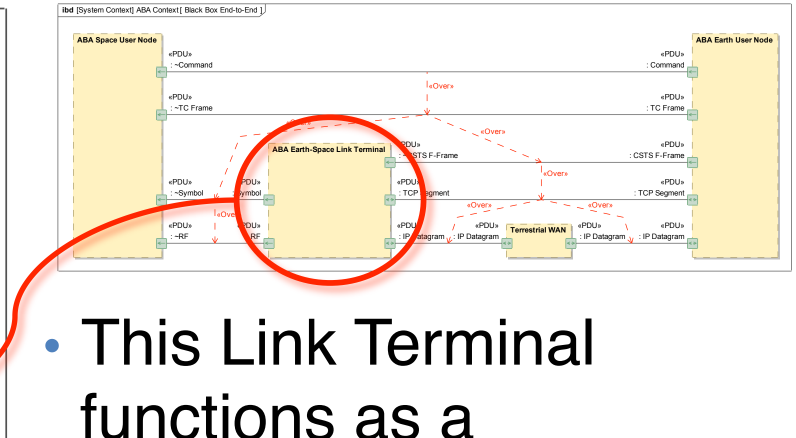
A Tale of Two Boxes – White Box



- Look inside Space and Earth nodes at top two layers.
- Sub-components that implement each layer.
- Vertical communication between sub-components within Earth and Space nodes.
- Correspondence of «Over» dependencies with vertical communication.
- Rules for how sub-components can be stacked vertically, i.e. which are compatible.

Another White Box

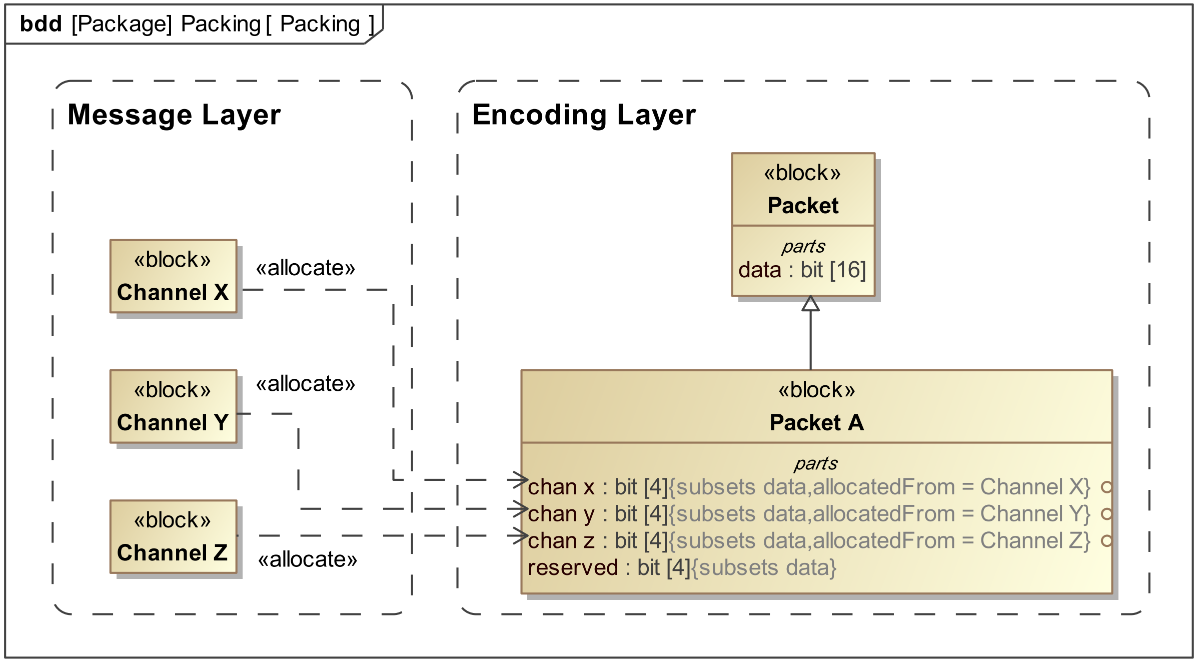




- This Link Terminal functions as a converter between two stacks.
- F-Frame Production component does the translation.
- It bridges the two stacks.

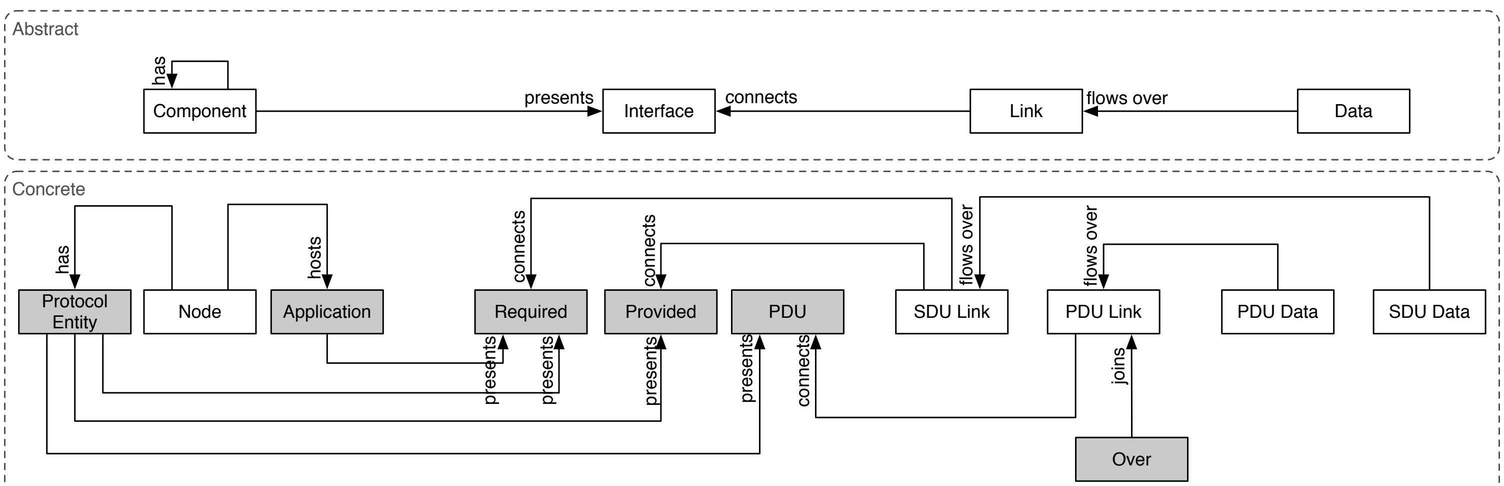
Data Encoding

- As data goes between two systems it is *exchanged.*
- As data goes between a lower layer and an upper layer within a system it is *transformed.*
- Useful to model that transformation.
- How physical measurements are placed in a packet.



Modeling Pattern

- For the modeling nerds among you.
- Defined an abstract and concrete set of terms and relationships.
  - Component, Interface, Link and Data.
  - SDU – Service Data Unit (vertical flow within component)
  - PDU – Protocol Data Unit (horizontal flow between components)
- Data can be interpreted to mean anything that is exchanged, including physical material.



Extension to Other Domains

- Can this model be applied to other kinds of interfaces, e.g. electrical, mechanical, thermal, etc?
- Perhaps there is potential.
- For example, an electrical interface might have a signal (voltage and current over time) at an upper layer, and a wire at a lower.
- Conversely, a thermal interface exists between two components if they exchange heat.
- But, the lower level that supports that exchange might be
  - Physical connection in case of *conduction*.
  - An intermediary substance in case of *convection*.
  - Simple line of sight in case of *radiative transfer.*

Four Layer Structure

- OSI definition of seven layers has worked in domain of computer networking.
- Other domains may need different choice and number of layers.
- Simple, four layers of abstraction for traditional engineering systems.
  - Message, Encoding, Signal and Physical
- Message is end-to-end.
- Lower layers may have different realizations along their paths.

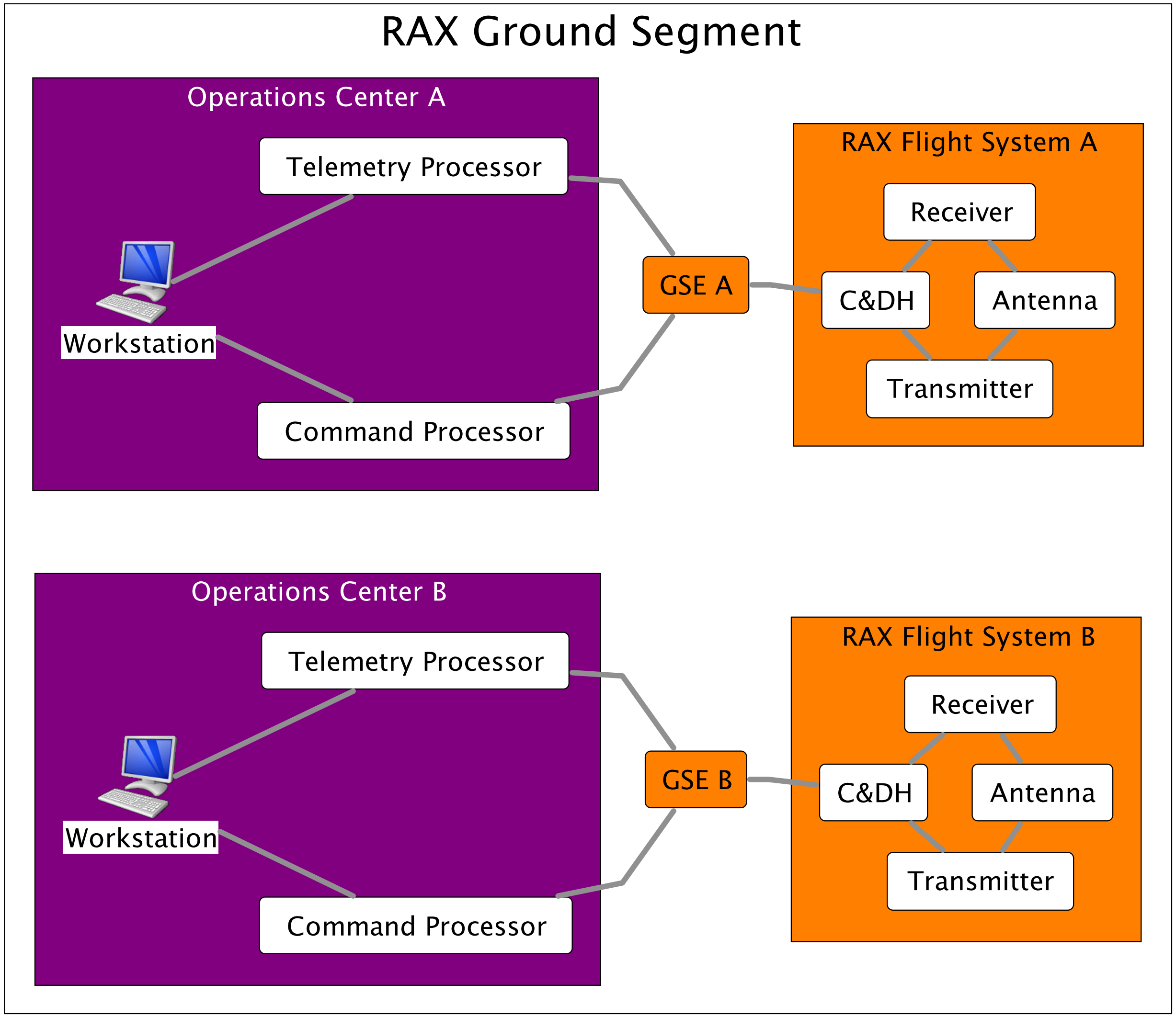
Example	Message	Encoding	Signal	Physical
Document Transfer	Document	PDF file	HTTP stack	Ethernet
Automobile	Stop car	Brake pedal pressure	Hydraulic pressure	Brake caliper pressure
Air Conditioner	Desire 68F	Thermostat setting	Electrical Signal	Compressor on

An Example With Graph Analysis

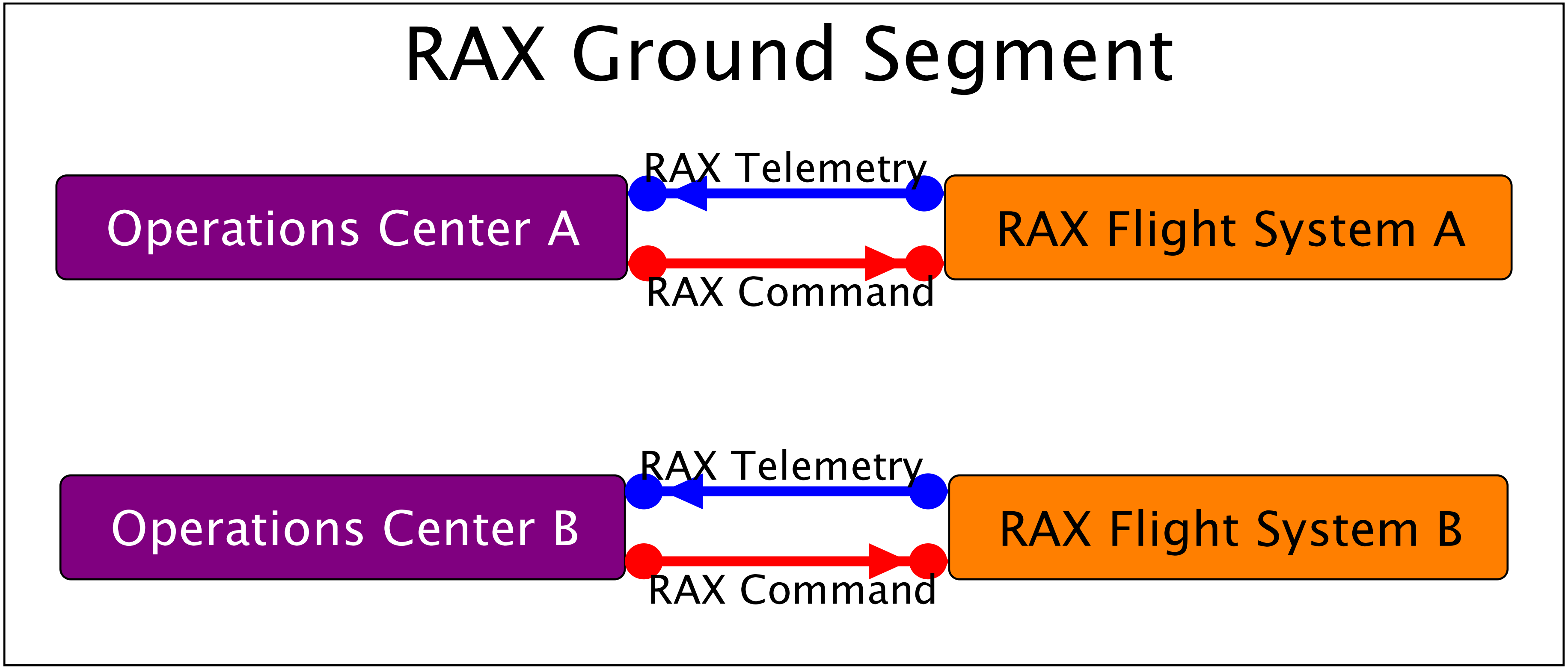
- Graph Analysis could be a fundamental tool for systems engineers in an MBSE environment.
- In these examples, we're using it to specify the physical path for a logical flow from among several possibilities.
- This example has just two layers (logical and physical), but concept is easily extended to many.
- In this hypothetical example, we suppose two operations centers and two spacecraft.
- We then show the data flows between the operations centers and the spacecraft in two different mission phases, ATLO pre-launch, and Flight post-launch.
- These routes are *derived entities* that do not *necessarily* exist in the model – might be stored as characterizations if needed.

Physical Layer Pre-Launch

- Operations Center A is connected to Flight System A, and B to B.
- GSE = Ground Support Equipment.
- C&DH = Command and Data Handling System.

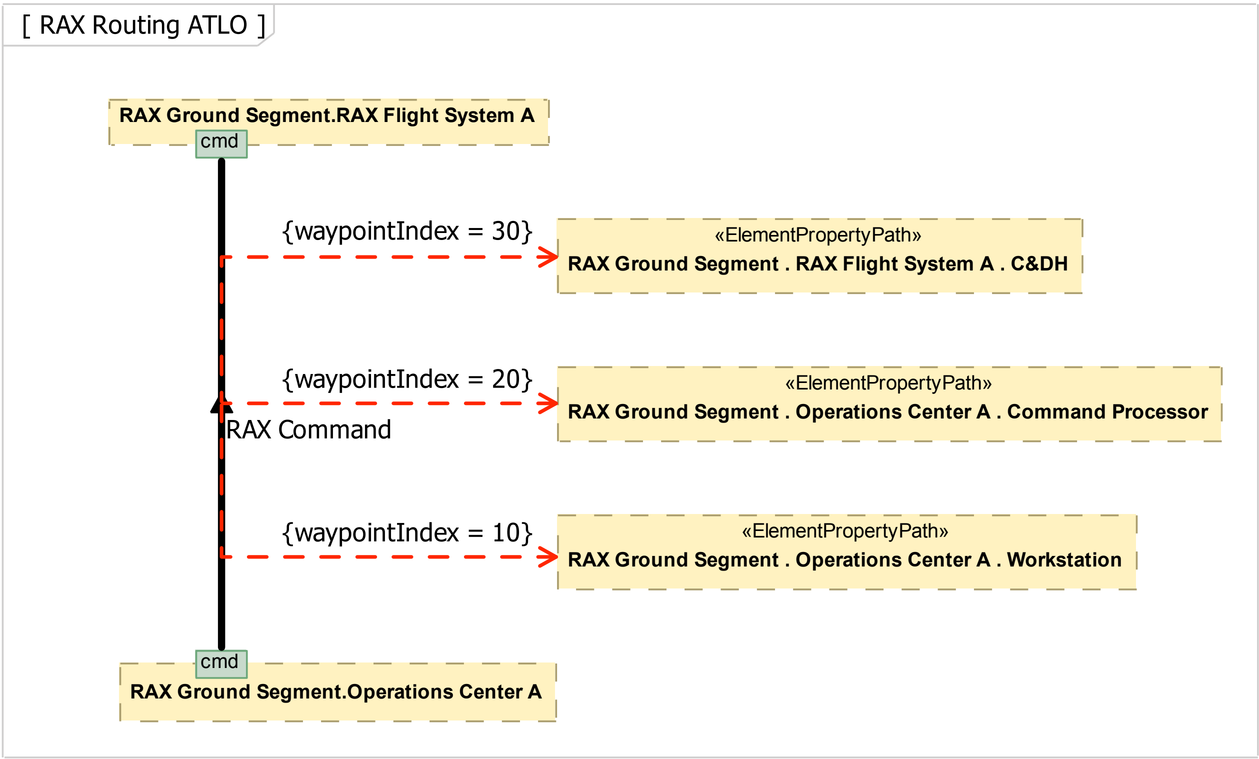


Logical Layer Pre-Launch

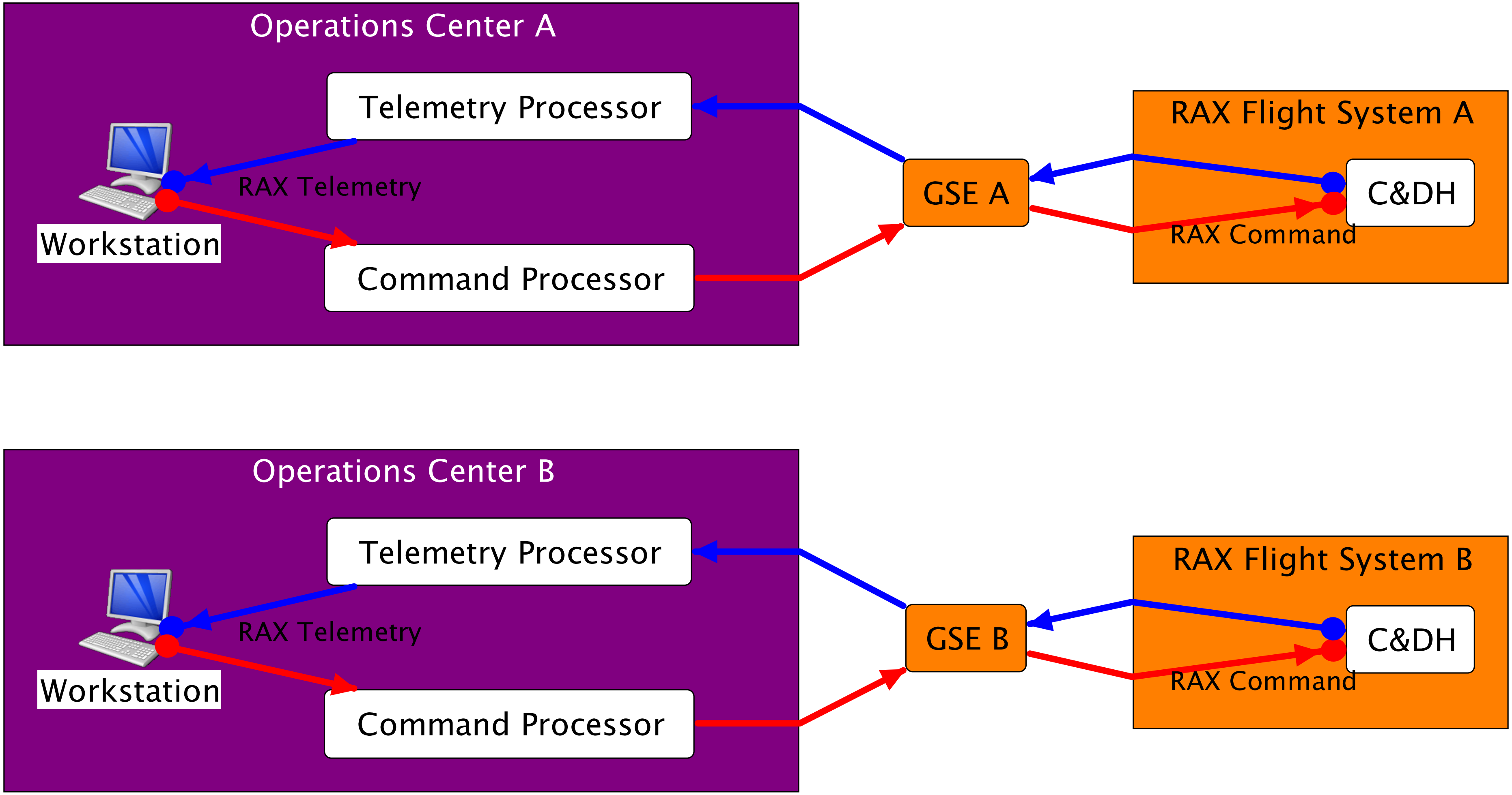


- Ops Center A sends commands to Flight Sys A, and receives telemetry.
- Same for B and B.

Routing Constraints Pre-Launch

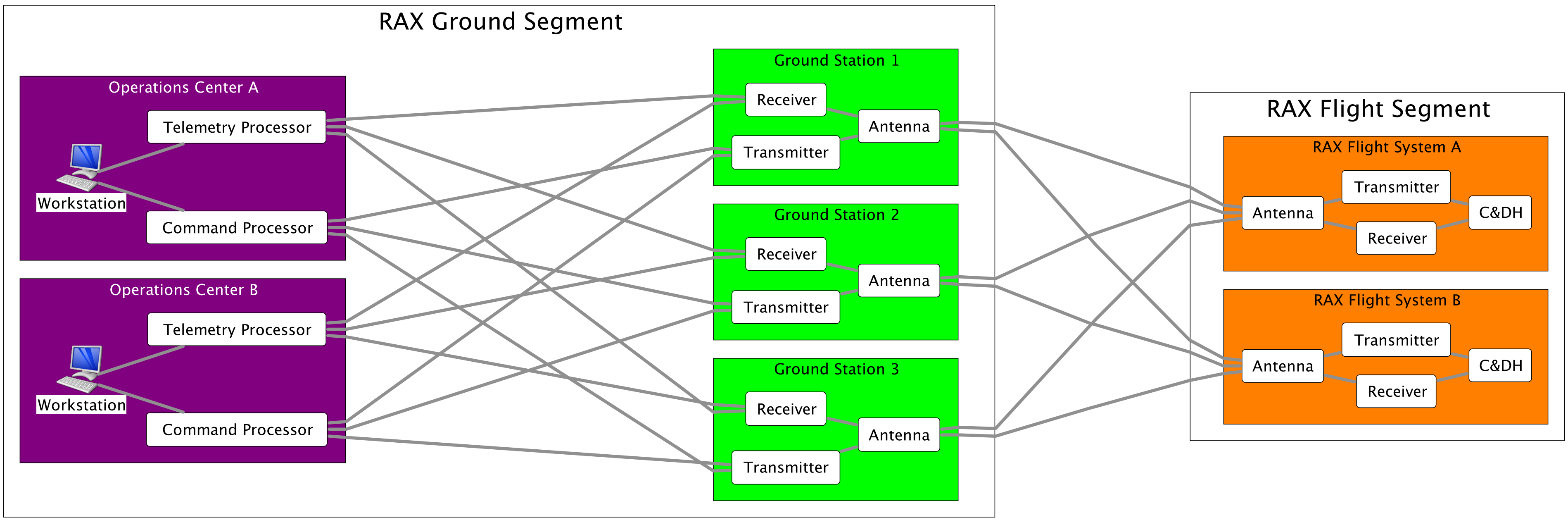


Routing Pre-Launch



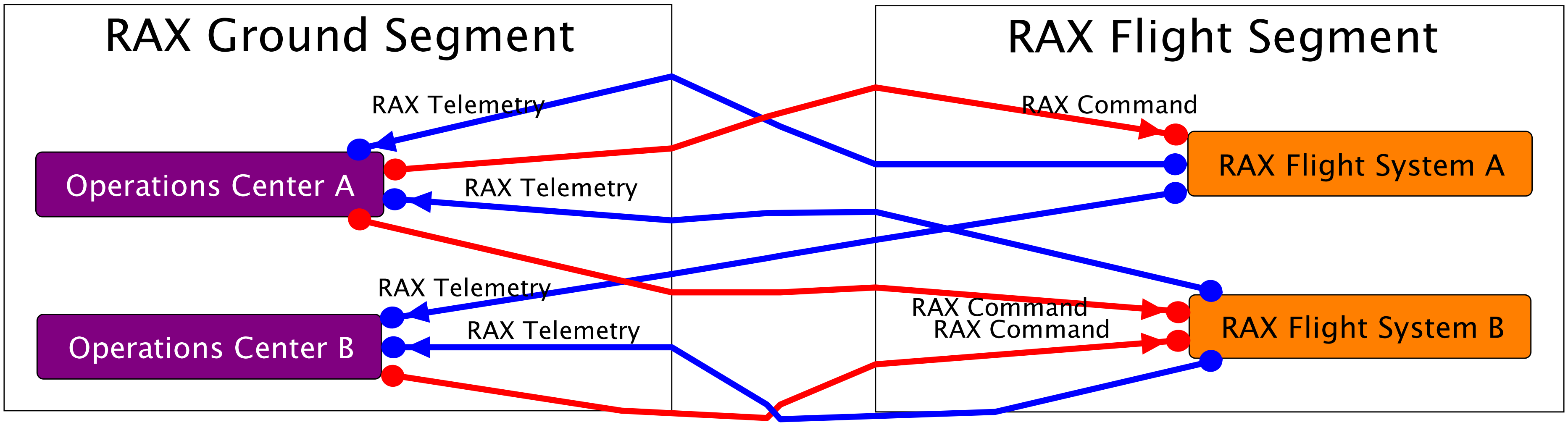
- Route specified with «xferdOver» dependencies.
- *Many* other mechanisms are conceivable based on properties and requirements of the connections.
  - Data type, data volume, bandwidth, latency, security etc.

Physical Layer Flight



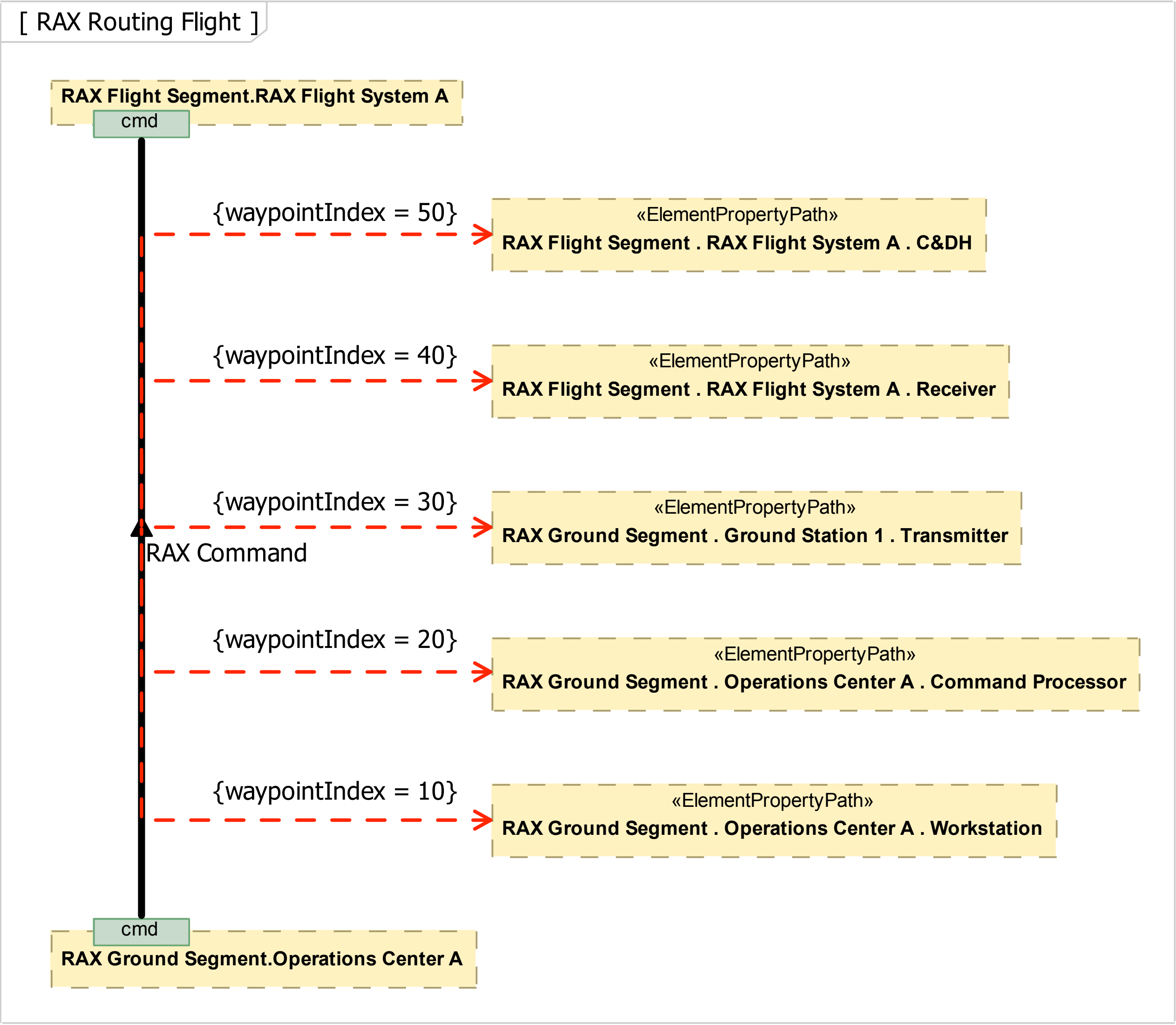
- Spacecraft are now in flight, no longer on ground.
- Three Ground Stations replace the GSE.

Logical Layer Flight

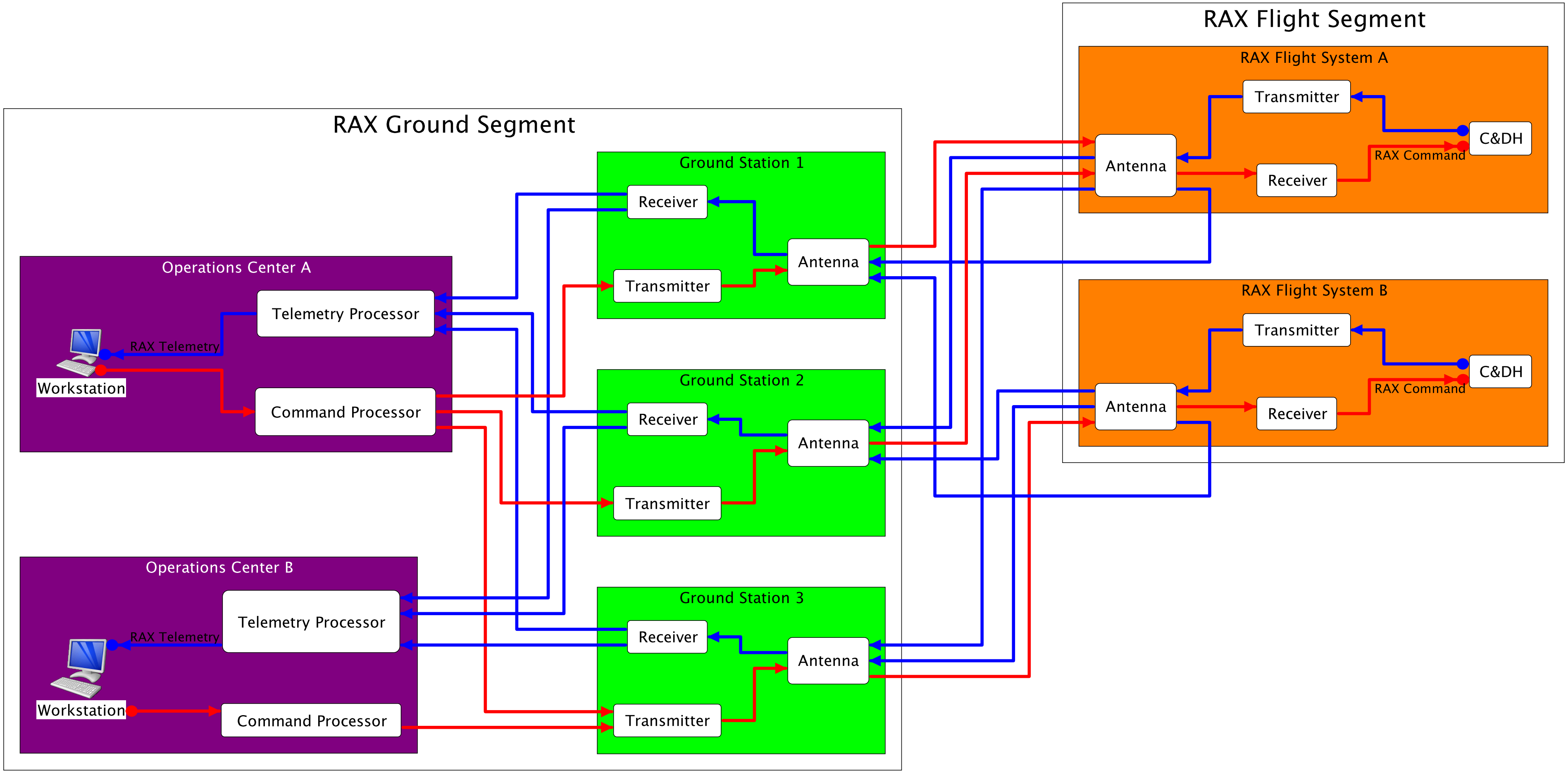


- Logical flows are different.
- Each Operations Center gets telemetry from each Spacecraft.
- Operations Center A can command either Spacecraft.
- Operations Center B can only command Spacecraft B.

Routing Constraints Flight



Routing Flight



- Further constraints on which Ground Stations are used in which combinations.
- Note routing over telemetry processors, command processors, receivers and transmitters.

Textual Representation of Routes

Pre-Lauch

	Physical Component	Exchanges Data With	Physical Component
1	RAX Ground Segment . Operations Center A	---> RAX Command	RAX Ground Segment . RAX Flight System A
	Physical Route RAX Ground Segment . Operations Center A . Workstation RAX Ground Segment . Operations Center A . Command Processor RAX Ground Segment . GSE A RAX Ground Segment . RAX Flight System A . C&DH ------- **Number of Routes = 1**		

Flight

	Physical Component	Exchanges Data With	Physical Component
1	RAX Ground Segment . Operations Center A	**A1A cmd** ---> RAX Command	RAX Flight Segment . RAX Flight System A
	Physical Route RAX Ground Segment . Operations Center A . Workstation RAX Ground Segment . Operations Center A . Command Processor RAX Ground Segment . Ground Station 1 . Transmitter RAX Ground Segment . Ground Station 1 . Antenna RAX Flight Segment . RAX Flight System A . Antenna RAX Flight Segment . RAX Flight System A . Receiver RAX Flight Segment . RAX Flight System A . C&DH ------- **Number of Routes = 1**		

Conclusions

- Conversation within Systems Engineering community.
- Multi-layer approach is useful and broadly applicable.
- Build frameworks for other engineering domains.
  - Electrical, mechanical, thermal, etc.
  - Define layers of abstraction.
  - Concerns, Viewpoints and Views.
  - Relate to traditional domain-specific CAD and analysis tools.
- What new analyses does this approach enable?
- How to project such models into useful views?
- What questions can we now answer better?

Acknowledgements

- Thanks to
  - Sandy Friedenthal
  - Kim Simpson