



Requirements**Experts**

Training and Services for Project Success

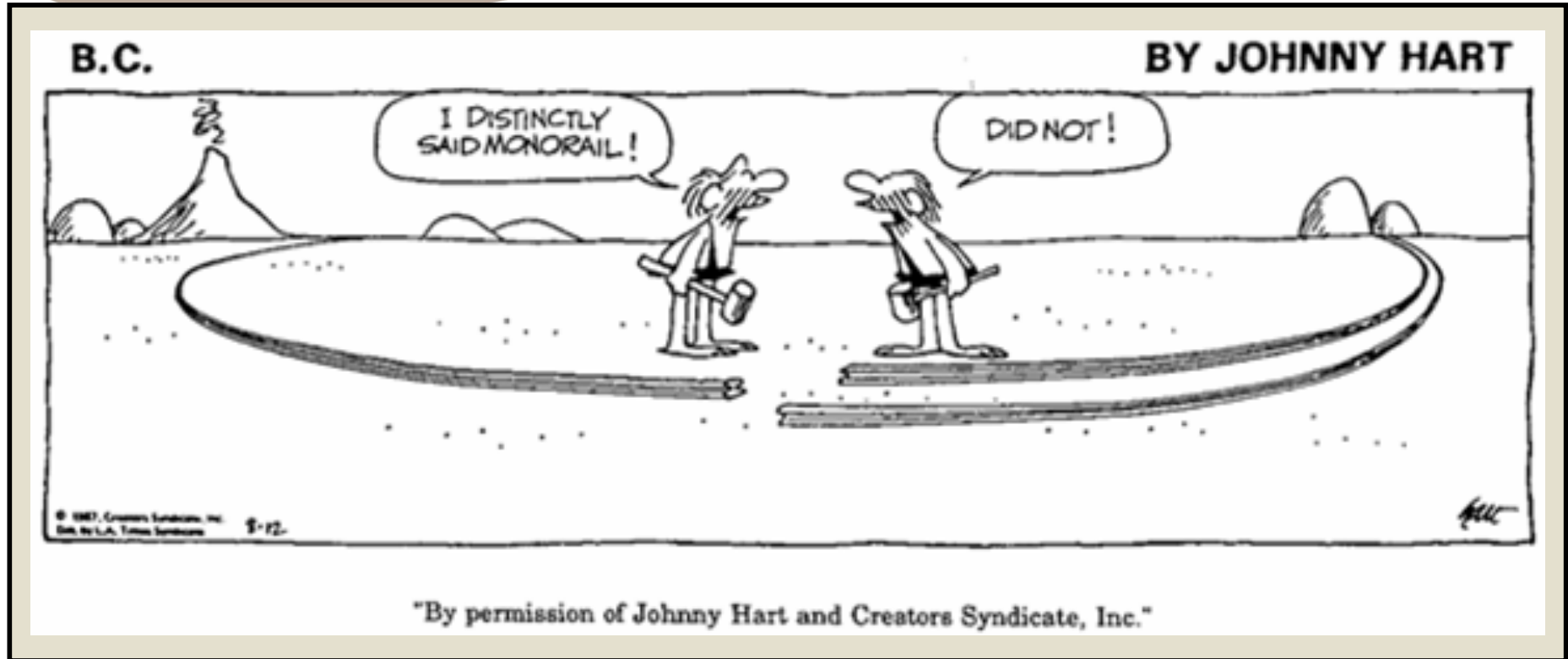
Advanced Writing and Managing *Interface Requirements*

Ivy Hooks

Objectives

- Unravel some of the mystery of interface requirements
- Present proven techniques for documenting and managing interface requirements
- Show how these techniques fit into the product life-cycle

Sound familiar?





Overview

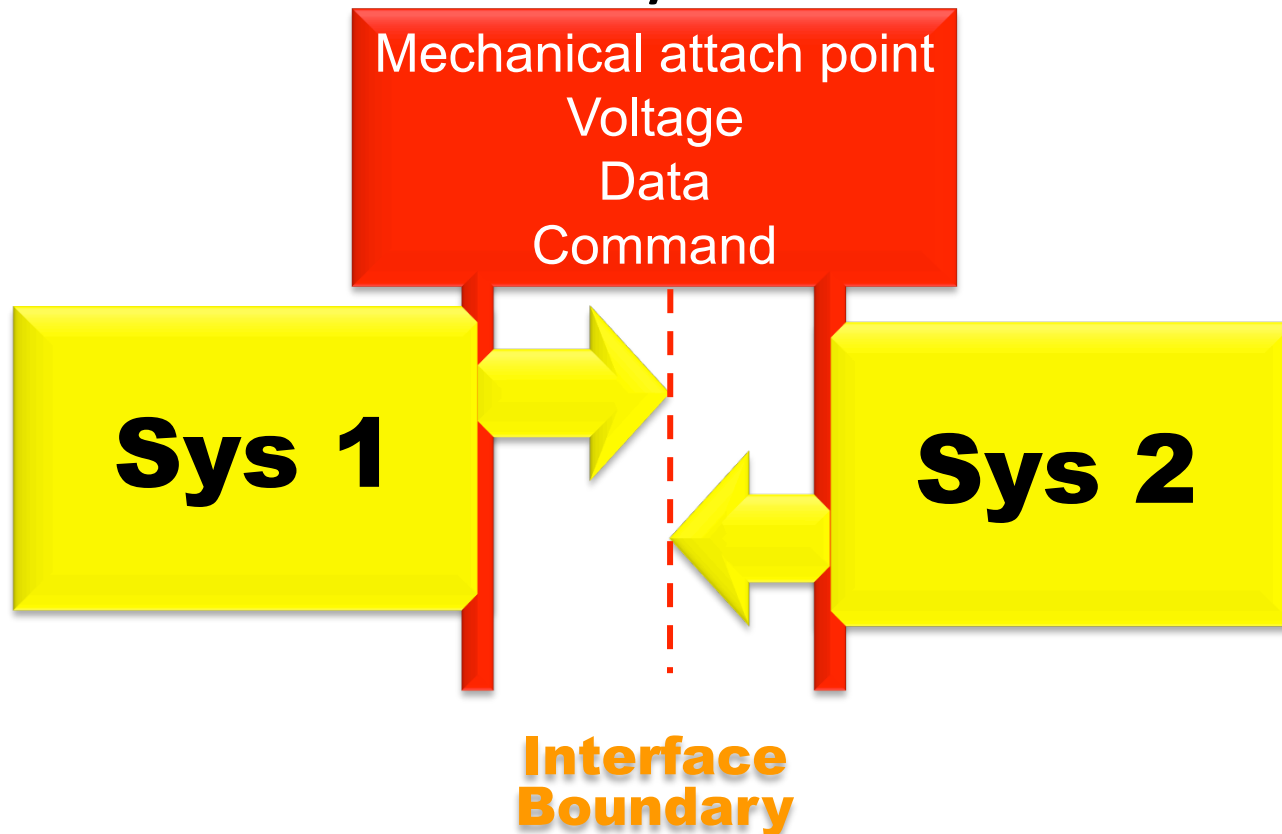
- Major types of interfaces
- What is an interface
- How do interfaces fit into the product life-cycle
- What do interface requirements look like

Major Types of Interfaces

- Physical interfaces
 - Mechanical
 - Electrical
 - Electronic
 - Power
 - Propulsion
- Functional interfaces
 - Information transfer
 - Computer-human
 - Maintenance
 - Installation
- Environment

What is an Interface?

A common functional or physical boundary where two systems interact.



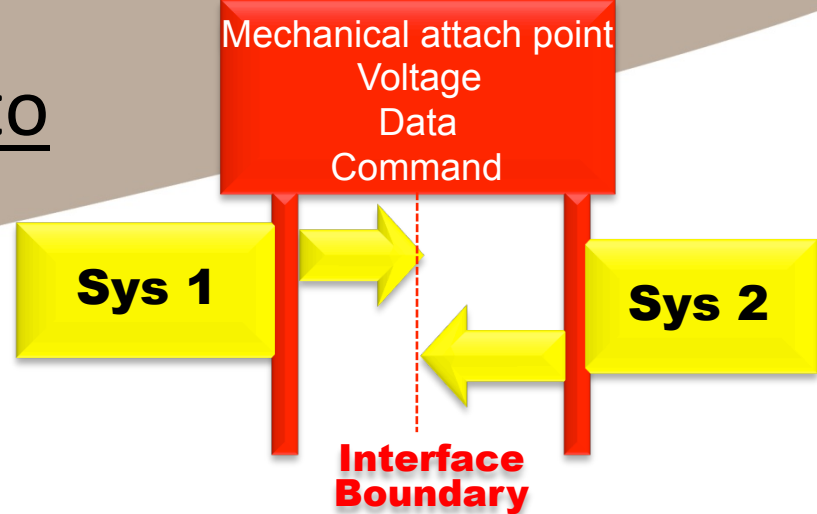
Boundary Definition

- Interface boundaries must be:
 - Documented
 - Maintained through design
 - Validated through design
 - Verified
- Interface boundaries are:
 - Defined by an existing system, or
 - Mandated from higher level, or
 - Worked out between two systems

Examples of Existing Systems with Defined Interface

- Your Spacecraft (S/C) is going to be carried by the launch vehicle
- Launch vehicle exists – has an ICD
- You want to put an instrument into the S/C
- S/C exists – has an ICD

Sys 1 is Built and Sys 2 wants to Interface to it



- Sys 1 has an ICD # 2345 that defines what another system has to do to interface to it.
- Sys 1 Specification has no interface requirements concerning working with Sys 2 though it may have generic interface requirements.
- Sys 2 Specification will have interface requirements concerning Sys 1

Boundary Definition is in the ICD

Spec contains requirements

- Sys 2 shall obtain power from Sys 1 per ICD 2345 Table 3-4.
- Sys 2 shall operate on power obtained from Sys 1 per ICD 2345 Table 3.6
- Sys 2 shall transmit data to Sys 1 per ICD 2345 Table 4-1.

ICD defines boundary

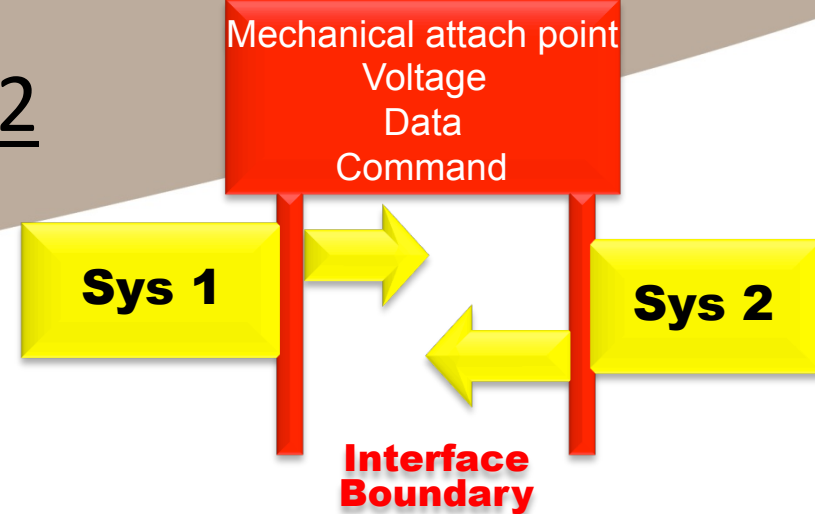
- Table 3-4 contains the connections and grounding information in order to obtain power
- Table 3-6 contains power characteristics such as noise, filtering, ...
- Table 4-1 data rates, types, formats, validity checks....

Boundary Definition

- Interface boundaries must be:
 - Documented
 - Maintained through design
 - Validated through design
 - Verified
- Interface boundaries are:
 - Defined by an existing system
 - Mandated from higher level, or
 - Worked out between two systems

Identification of Interfaces – 2

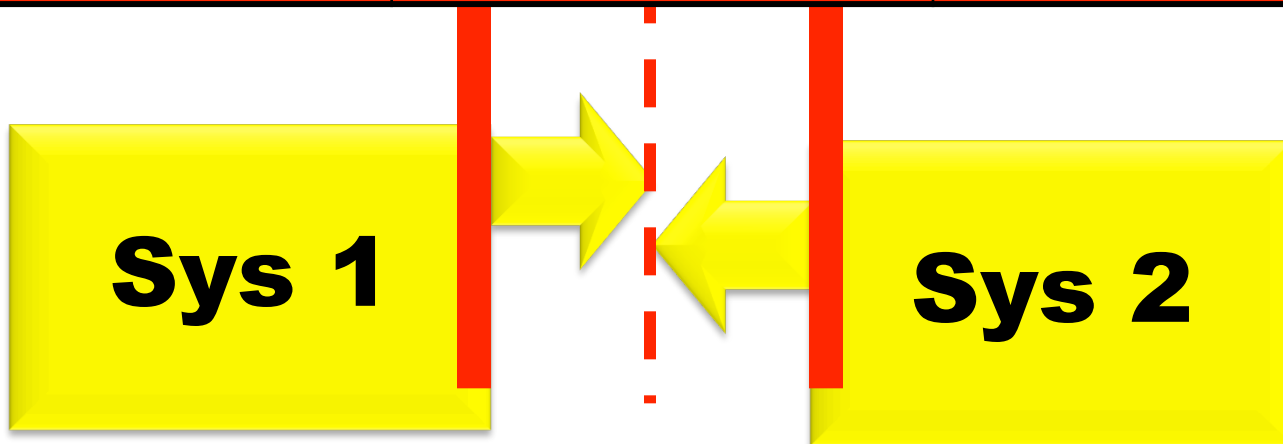
Systems in Development



- Sys 1 analyzes how it will operate and with whom it will interface
- Sys 2 is doing likewise
- Each is going to find that it must interface with the other
- Each will start determining what this means to them
- Agreeing to the identification of the interfaces between Sys1 and Sys 2 is the first step

Interface Requirements Between Two Developing Systems - *Identified*

Sys 1 Specification	<u>I/F Boundary Doc</u> #1234	Sys 2 Specification
<i>Sys 1 shall transmit xyz command defined in #1234 to Sys 2.</i>	<i>The xyz command will have the characteristics described in Table TBD.</i>	<i>Sys 2 shall execute the xyz command defined in #1234.</i>



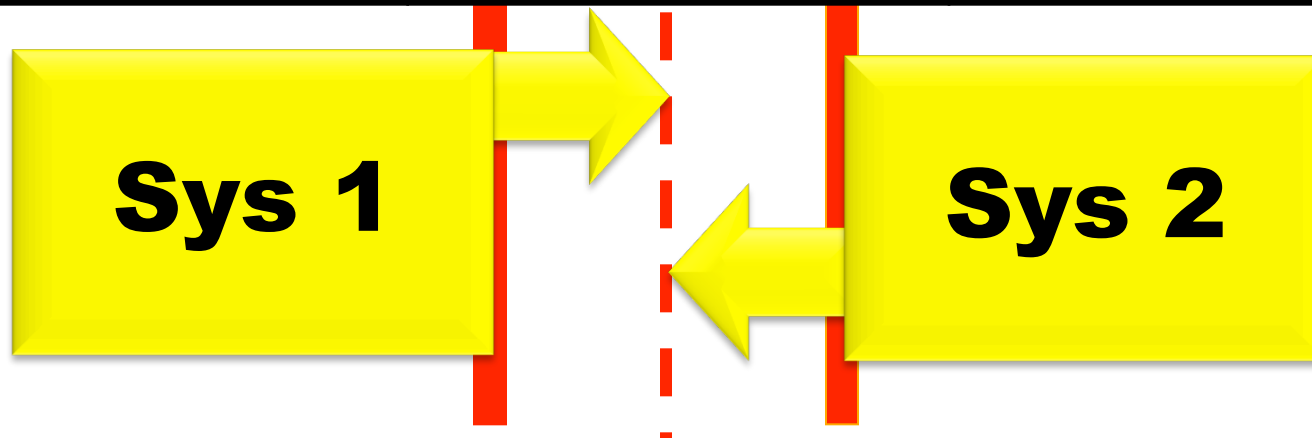
Interface Requirements Between Two Developing Systems - *Identified*

Sys 1 Specification	<u>I/F Boundary Doc</u> #1234	Sys 2 Specification
<i>Sys 1 shall transmit xyz command defined in #1234 to Sys 2.</i>	<i>The xyz command will have the characteristics described in Table TBD.</i>	<i>Sys 2 shall execute the xyz command defined in #1234.</i>

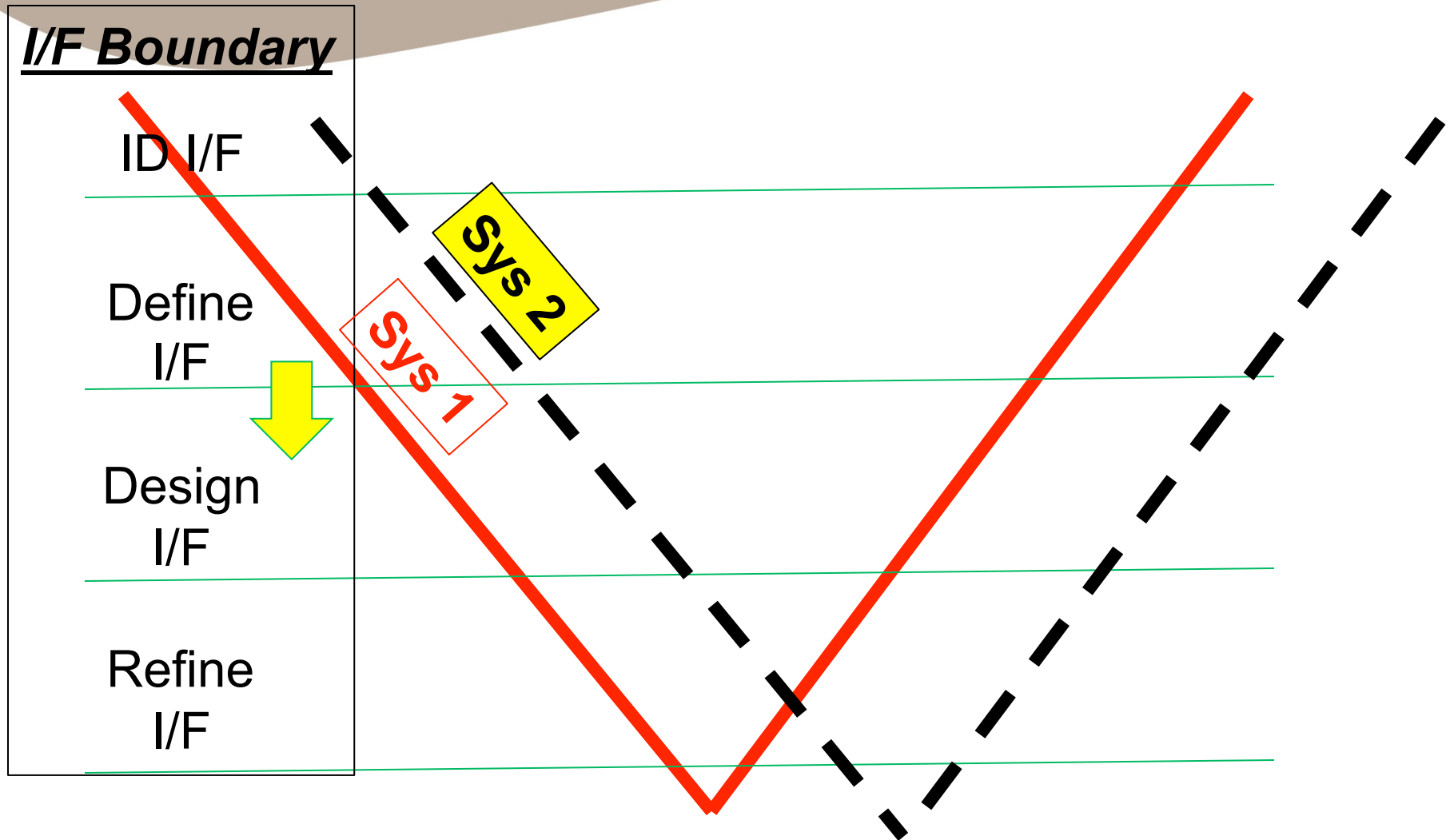


Interface Requirements Between Two Developing Systems - *Defined*

Sys 1 Specification	I/F Boundary Doc #1234	Sys 2 Specification
<i>Sys 1 <u>shall</u> transmit the xyz command defined in #1234 Table 2-2 within 5 ms of cmd input.</i>	<i>The xyz command <u>will</u> have the characteristics described in Table 2-2.</i>	<i>Sys 2 <u>shall</u> execute the xyz command defined in #1234 Table 2-2 within 4 ms of receipt.</i>



The Vee-model and Interfaces Across Products' Life-cycles



Validation of I/F

Sys 1 Requirements & Design

Interface Boundaries Doc

Sys 2 Requirements & Design

Requirement to I/F Identification of I/F Requirement to I/F
←→ (Scope Review) →

Expanded requirements Words and tables Expanded requirements
←→ (System Reqt Review) →

Preliminary design Figure/diagram / drawings Preliminary design
←→ (PDR) →

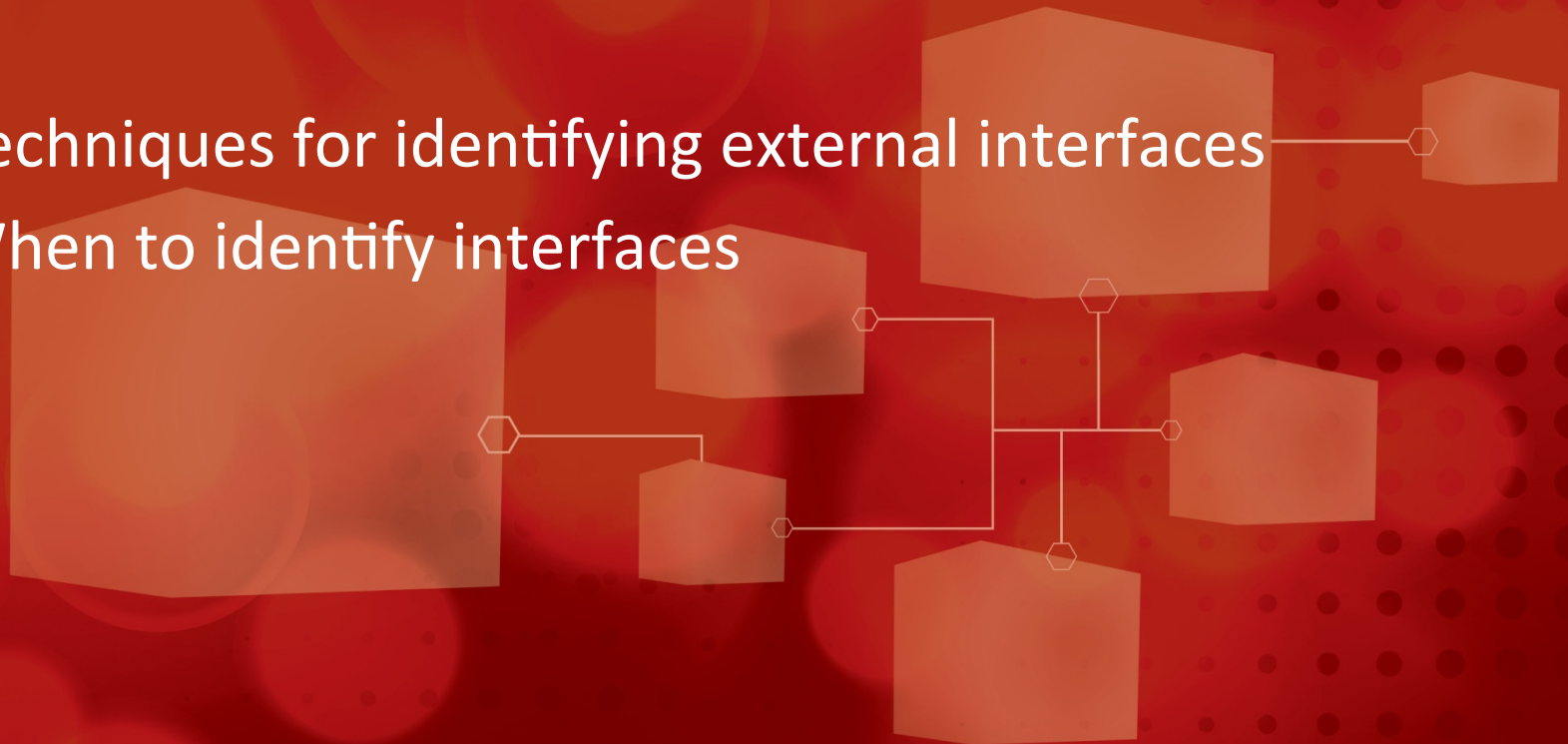
Detailed design Detailed schematics & drawings Detailed design
←→ (CDR) →



- Take a moment to stretch and make note of questions you now have.
- I'll take some questions after you stretch – some of which I may defer because I know I have the topic covered later.



- Techniques for identifying external interfaces
- When to identify interfaces



S/C External Interfaces

- Include those systems that you need to do your job
 - **Launch vehicle**
 - Power
 - Structural
 - Communication
 - **Deep Space Network**
 - **Test equipment** - not provided by your system
- Include those systems who need your system to do their job
 - Provide data to scientists
 - Provide status to Mission Ops

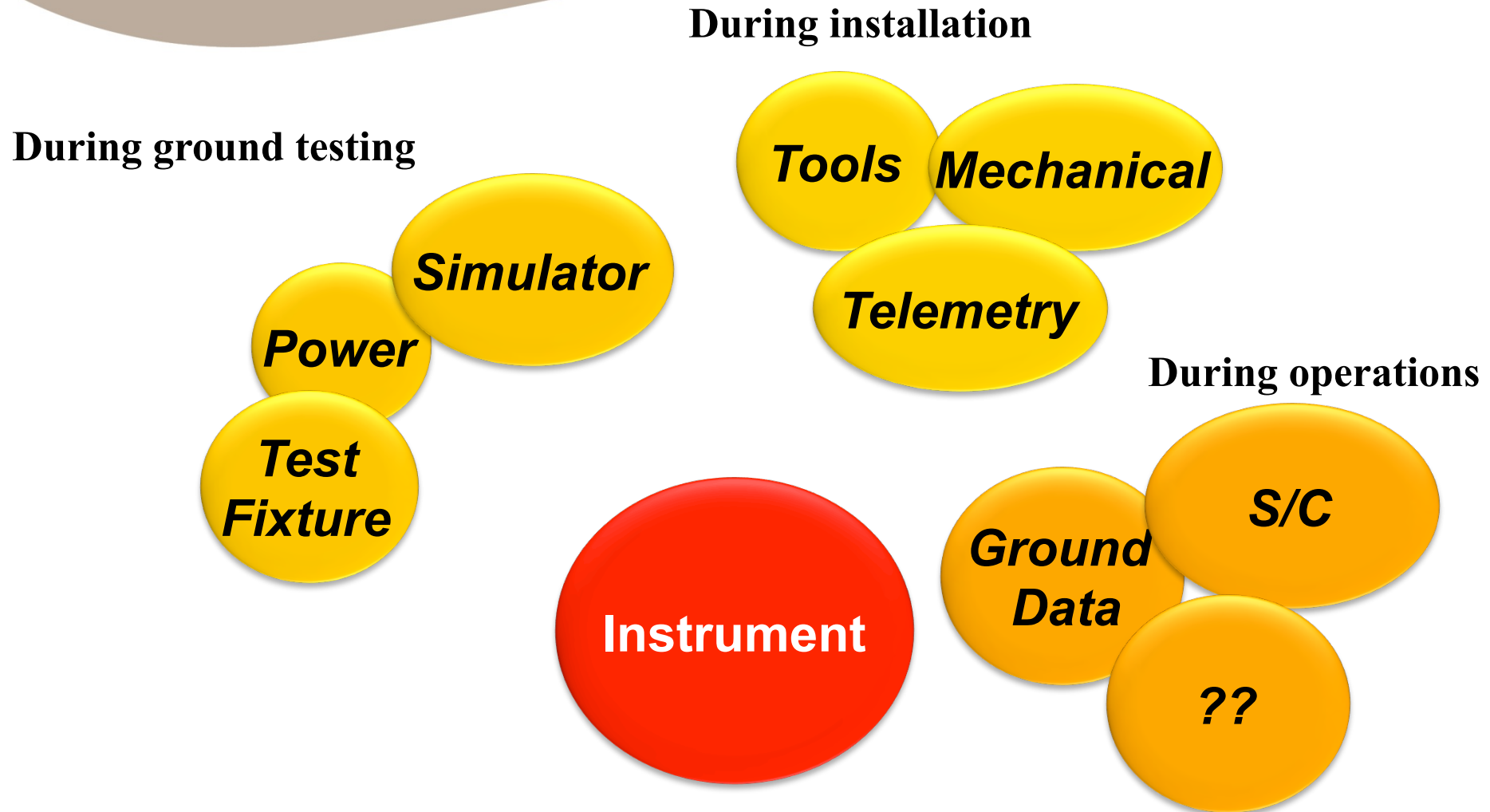
Define for all lifecycle stages

Identifying Interfaces – Think S/C Life-Cycle

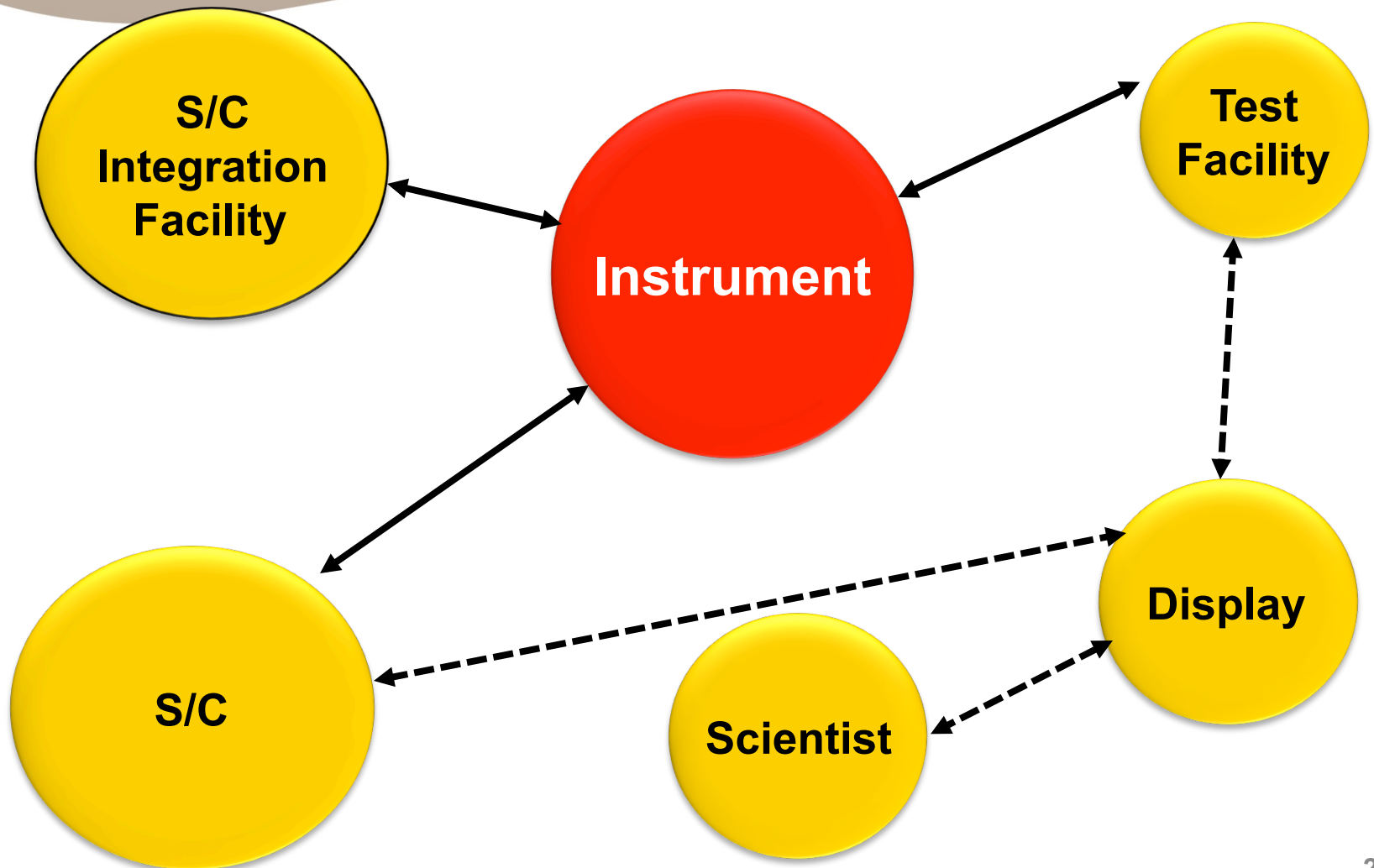
- Factory Integration and Test
- Factory Acceptance Testing
- Transportation on the ground
- Integrate with launch vehicle
- Integration test
- Ground Operations including maintenance
- Mission operations
- Disposal

Can you think of others?

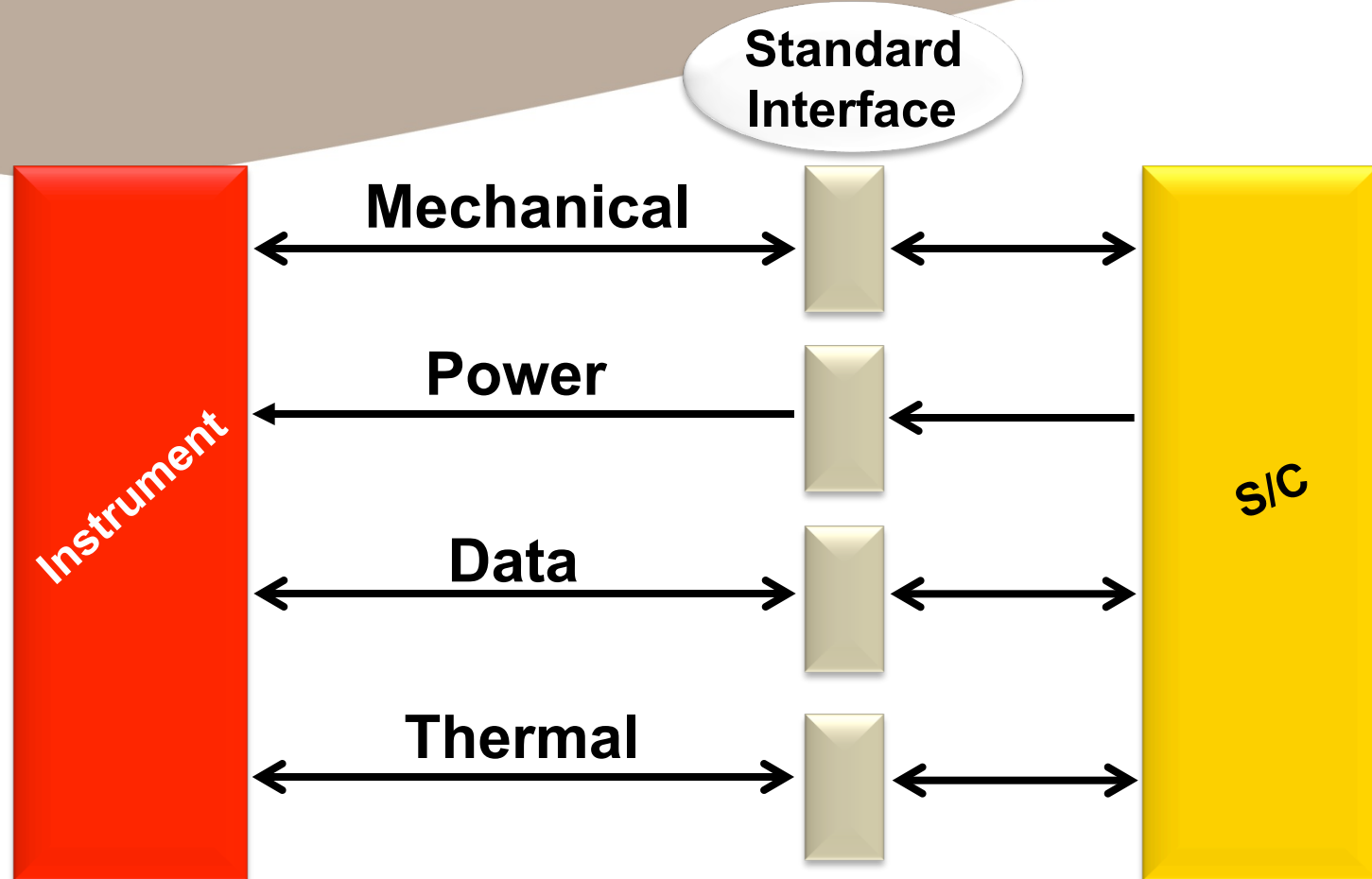
External Interfaces: Think Instrument Life-Cycle



Use Operational Concepts

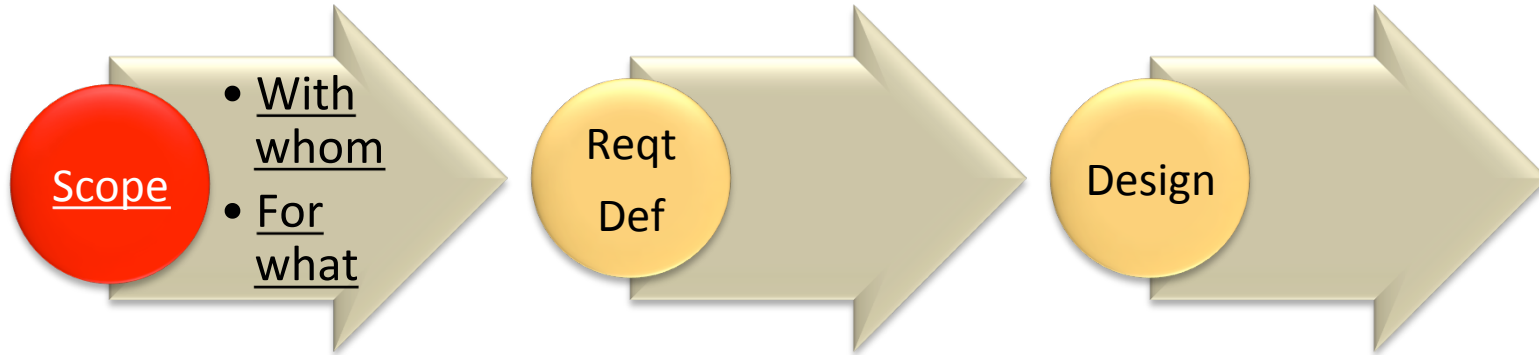


Interface Block Diagram



Notional Instrument-to-S/C Functional Interfaces

When to **Identify** Interfaces



Understanding Interfaces Early

For What

- Acquire power
- Thermal cooling
- Receive commands
- Transmit data
- Mechanical attachment

With Whom

- Rack during my testing
- S/C during integration testing
- S/C during operations

Scope Definition Phase

- Define Need, Goals, and Objectives
- Identify Stakeholders
- Develop operational concepts – for all life-cycle phases of the product
- Identify Drivers
- Identify Interfaces

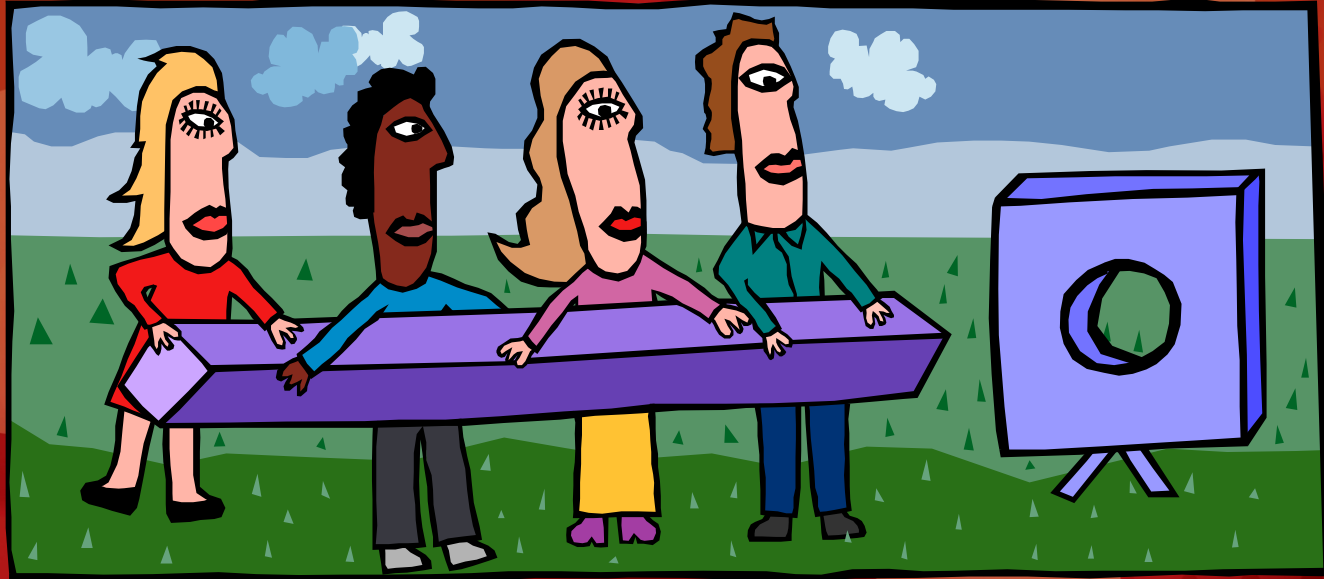
Define boundaries for your product and gain agreement before committing to requirement gathering and documentation



- How many of you do interface definition in the Scope phase – during formulation, concepts, or whatever you call the phase ?
- What techniques do you use? Ops Concepts? Context diagrams? Work from lists of previous interfaces? Other?



Details of I/F Types



Major Types of Interfaces

- Physical
- Electronic
- Electrical
- Hardware/Software
- Software
- Environment
- Human/Machine

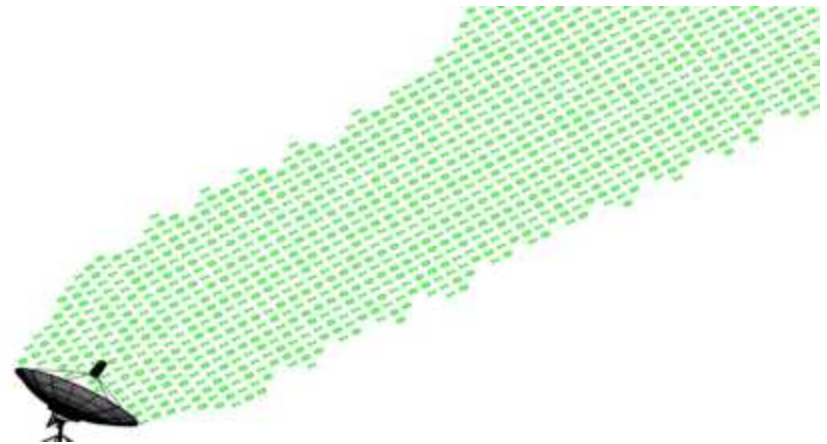
Physical

- Mechanical
 - Envelope
 - Attachment/Mounting
 - Obscuration
 - Alignment
- Tooling



Electronic Interface

- Command Signals
 - Format
 - Rates
 - Identification
- Data Signals
 - Frequency characteristics
 - Format
 - Rate
- Discrete Signals
 - Voltage levels
 - Impedance
 - Signal meaning
- Telemetry Signals
 - Format
 - Clock rate
 - Identification
 - Recording media/method
- Multi-spectral Communications



Electrical Interface

- Power
 - Type
 - Voltage/Current
 - Power profile
 - Protection
 - Grounding and bonding
- Connector type and pin assignments
- Electromagnetic
 - Compatibility (EMC)
 - Interference (EMI)
 - Pulse (EMP)

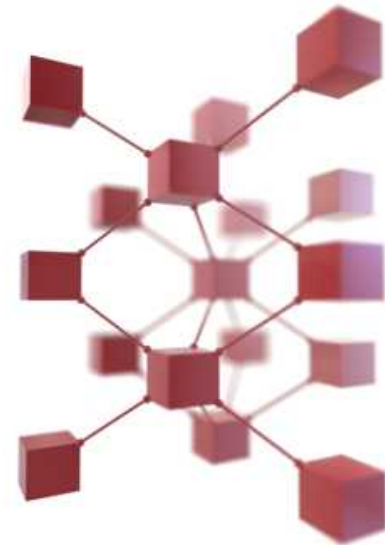
Hardware/Software

- Timing & Sequencing
- Analog-to-Digital conversion
- Host operating system
- Peripherals
- Protocols/Standards
 - **Open System Interconnection (OSI)**



Software

- Data
 - Content
 - Inputs and Outputs
 - Format
 - Rates
 - Accuracy
 - Latency
- Messages
 - Identification (name)
 - Format
 - Content
 - Storage
- Application Program Interfaces (APIs)



Environment

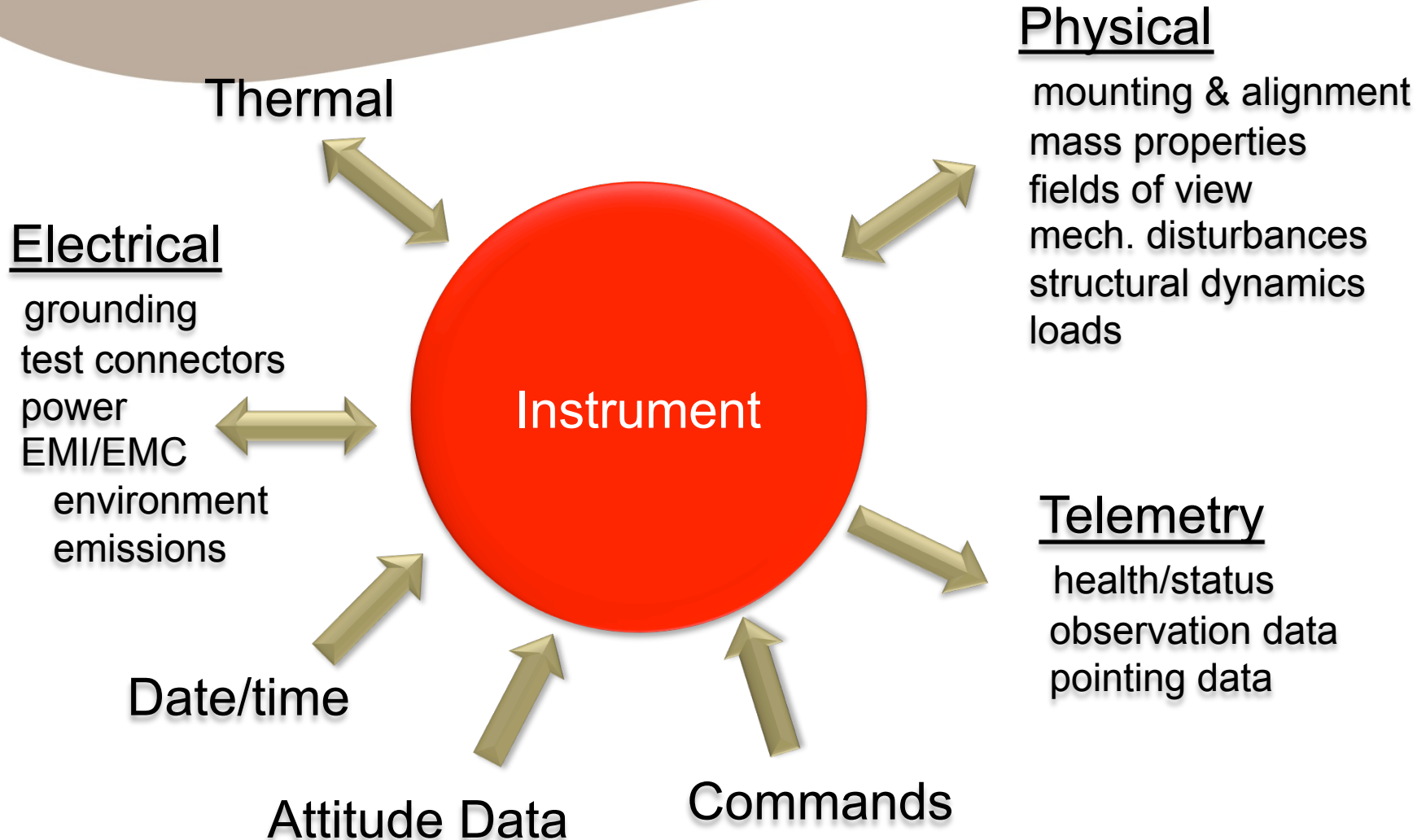
- Structural
 - Vibration
 - Shock
 - Acoustic
 - Loads
 - Dynamic mode shapes
- Thermal
 - Temperature range
 - Heating rates
 - Heat transfer surfaces
- Magnetic
 - Flux density
 - Rate-of-Change
- Radiation
 - Type
 - Flux Density
 - Total Dose
- Ambient
 - Pressure
 - Temperature
 - Humidity
 - Contaminants

Human/Machine

- Set of
 - inputs,
 - outputs,
 - special actions
- Computer-human interaction mechanism
 - dialogue procedures
- Interrelationship identified for these entities in the various functional areas.



Thinking Interfaces







- What did you learn or see more clearly in the pre-break session? This could be:
 - A new approach or something your process is missing
 - Clarification of a point that had confused you before
 - A way to explain a concept to others, because while you understand, you've had trouble explaining it.
- Will you change something in your process or approach to requirements because of something you learned ?
- Do you have more questions?



Documenting Interface Boundaries

Specification

ICD

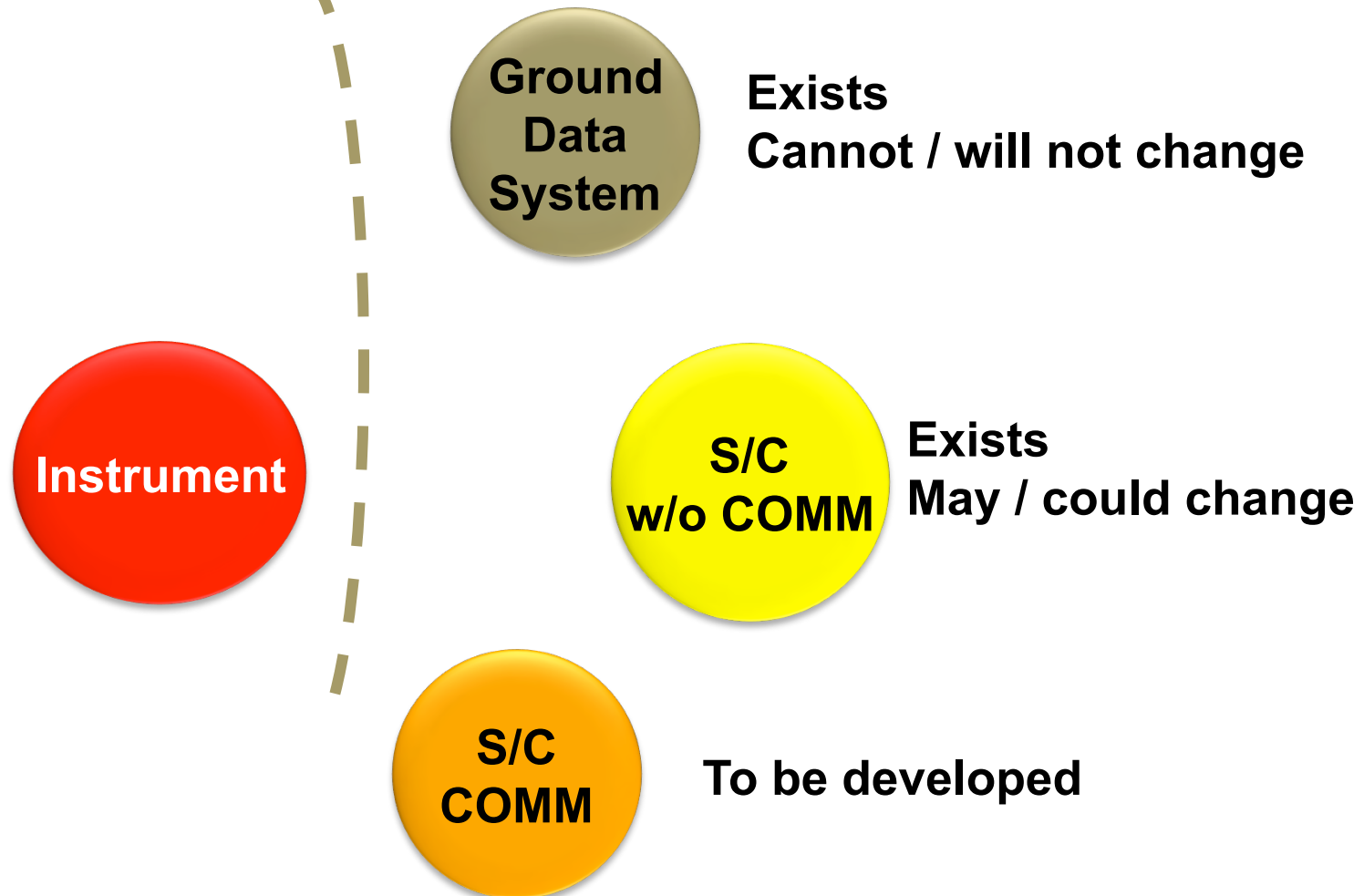
Standard

IRD



Maturity of Interfaces

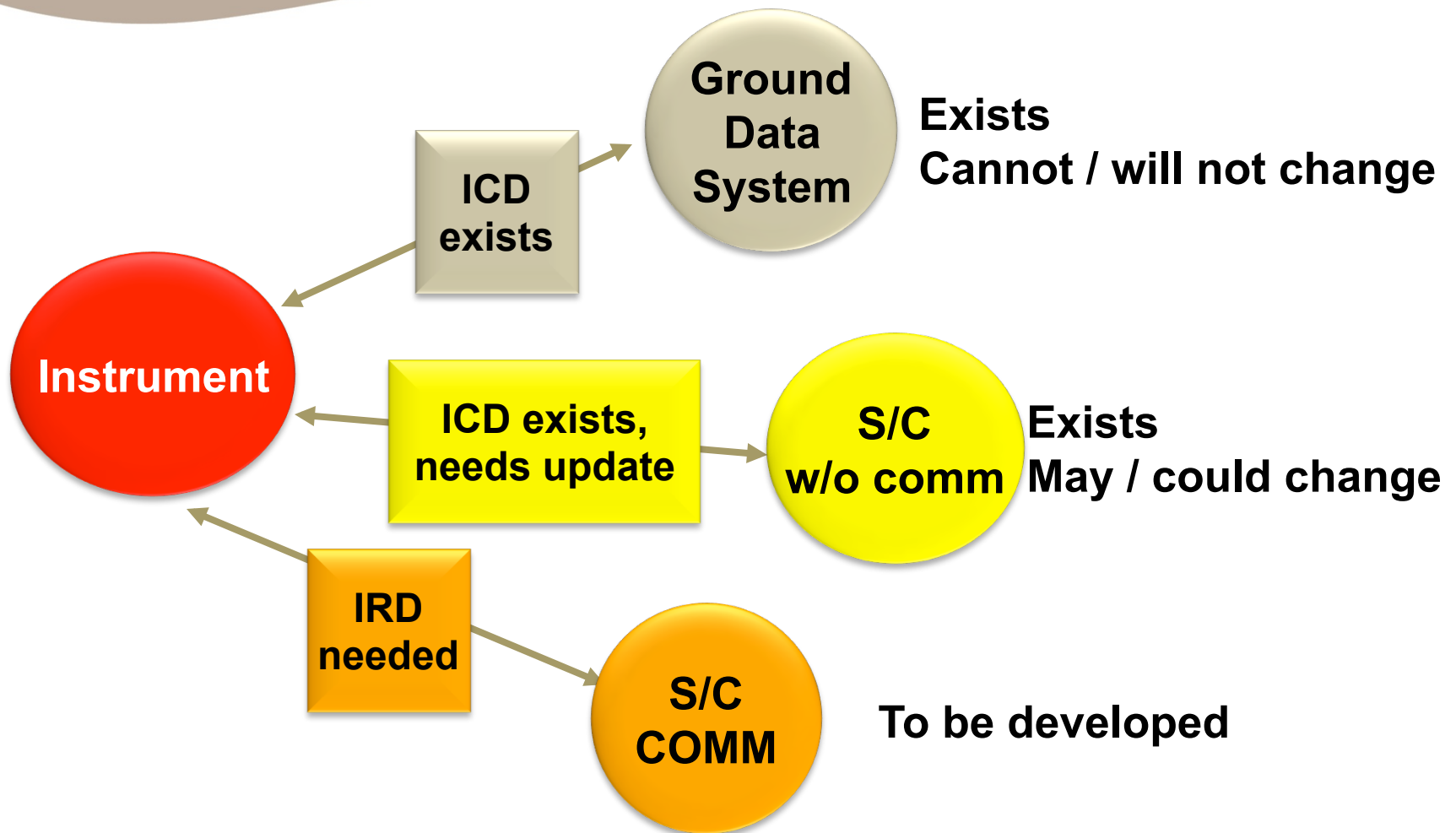
Some existing and some to be developed



Defining I/F Boundaries

- Answer these questions
 - Which external interfaces are known?
 - Which external interface definitions will require a cooperative effort?
 - What is my schedule for needing details?
- Develop a plan to resolve interface issues and unknowns
- Identify risks

Maturity of Interface Documents



What's the Difference

Interface control document (ICD):

Written by the person responsible for the existing system.

Description of existing system interfaces.

What you have to do in order to interface to this existing system.

Or, what is available to you to use, communicate with, etc.

Standard:

Written by some standards organization or group

Special case of an ICD.

Everyone works to the standard, e.g., 1553, and then all can work together.

Interface requirement document (IRD)

Jointly written by two systems that need to have an interface.

Description of the common point.

Description for each interface.

Terminology Clarification

- Some organizations use the term ICD for all interfaces, and this is fine
 - The ICD begins with information described above in the IRD
 - The ICD continues with all the details from each side of the interface included to the as-built configuration
- An organization should be consistent in terminology. The world is never going to be consistent in terminology.

Interface Definitions Control

- Existing system uses an ICD
 - belongs to the manager of the system with which you want to interface
 - probably is not going to change
- Two sides in development – options
 - Upper Level SRD – managed by SRD change board
 - IRD – managed and signed jointly by the managers of the two interfacing systems
 - Evolving ICD – managed and signed jointly as IRD

Interface Control Document (ICD)

- A formally controlled document that:
 - Represents the as-built configuration
 - Identifies, quantifies, and controls the design characteristics of the interface.
 - Ensures interface compatibility by documenting form, fit, and function.
 - Is prepared by and controlled by the owner of the system
- Some may be well-written and some may be awful

Using Higher Level Specifications

S/C Specification (SRD ABC)

The S/C shall provide 28 VDC power with the characteristics shown in table 3-4.

S/C owns both the Power and the Instrument and has a high level specification that controls both

Power organization writes and signs

Refers to

Power Specification

The generator shall supply 28 VDC as described in SRD ABC table 3-4

INST organization writes and signs

INST Specification

The INST shall use 28 VDC as described in SRD ABC table 3-4.

Refers to

Using an IRD

Power and Instrument organizations jointly write and sign

Power to Instrument IRD XYZ

The 28 VDC power will have the characteristics shown in table 3-4.

Power organization writes and signs

Refers to

Power Specification

The generator shall supply 28 VDC as described in IRD XYZ table 3-4

Instrument organization writes and signs

Refers to

INST Specification

Instrument shall use 28 VDC as described in IRD XYZ table 3-4.

Interface Requirements Document (IRD)

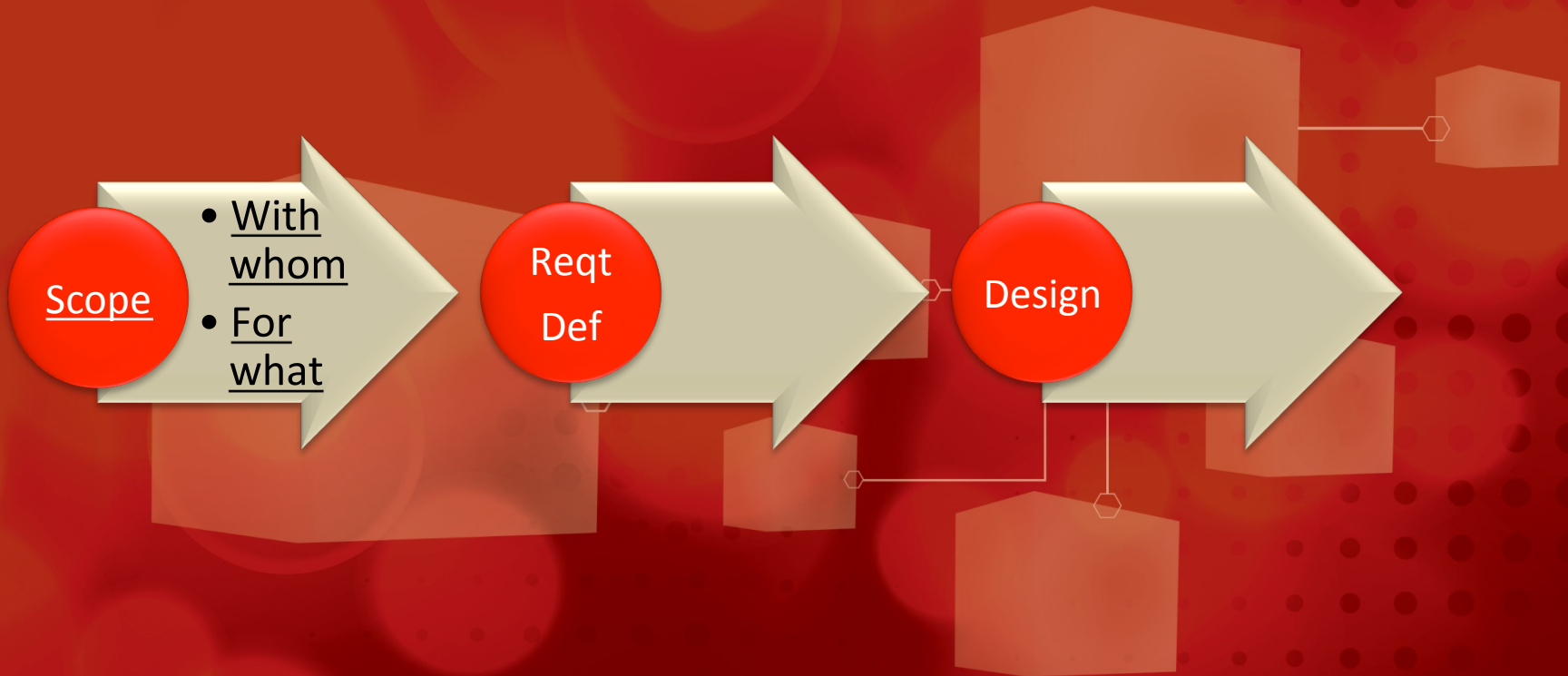
- A formal agreement that documents the interface between two systems
- Defines the “common” point(s) where the two systems interface
- Will describe in detail the data formats, data types, etc.
- Will describe physical interfaces to the drawing level
- Both systems have to work together to develop the document and both must sign and formally control

Benefits of a good ICD/IRD

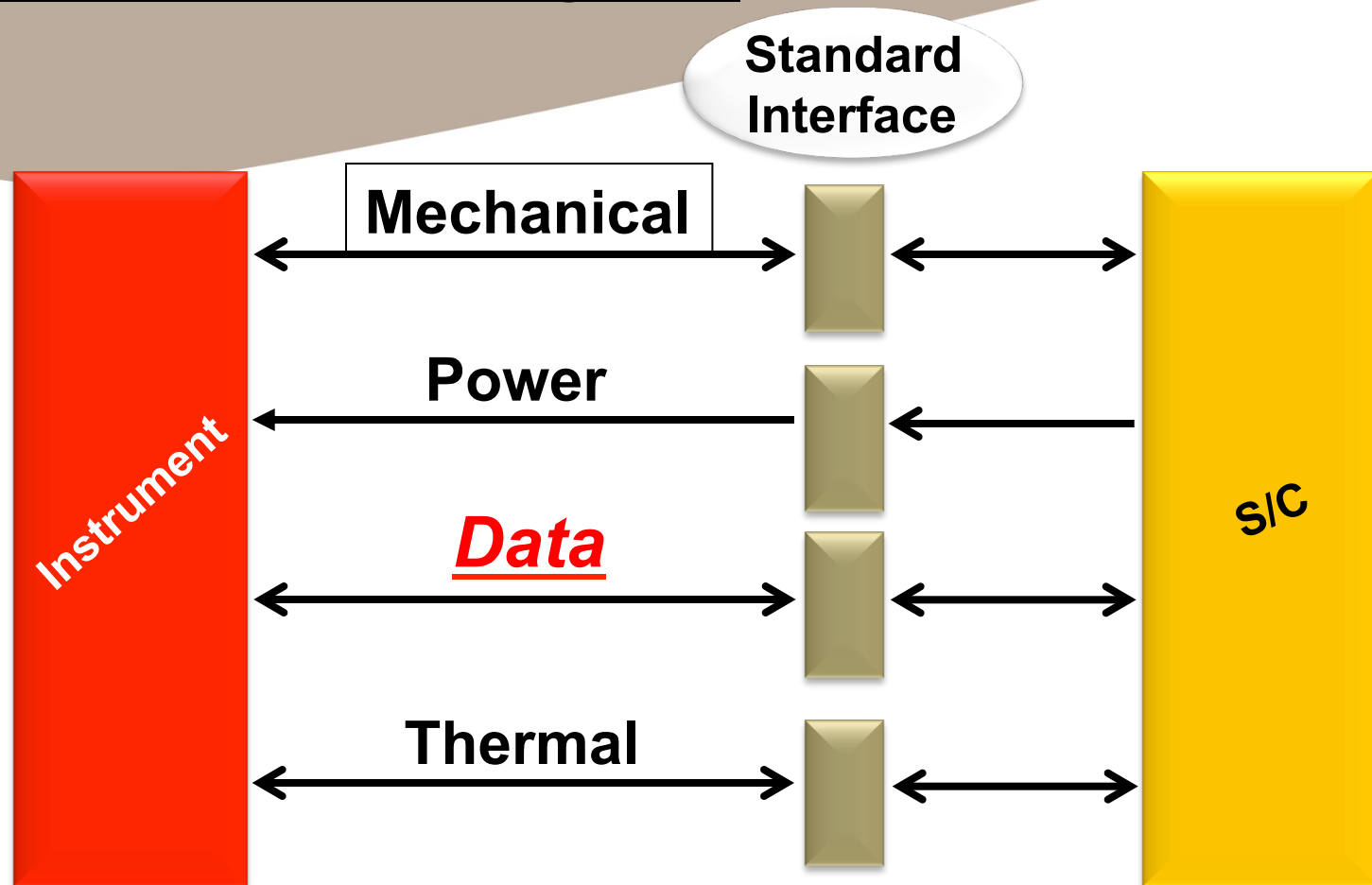
- Provide a mutual interface agreement agreed to by affected parties
- “Living” document exists for life of program so that changes can be accommodated
- Common format
- Easily understood and eliminates unnecessary verbiage
- Provide a means to control cost and schedule



Boundary Evolution



Interface Block Diagram



Notional Instrument-to-S/C Functional Interfaces

Pre-Design Data Interface

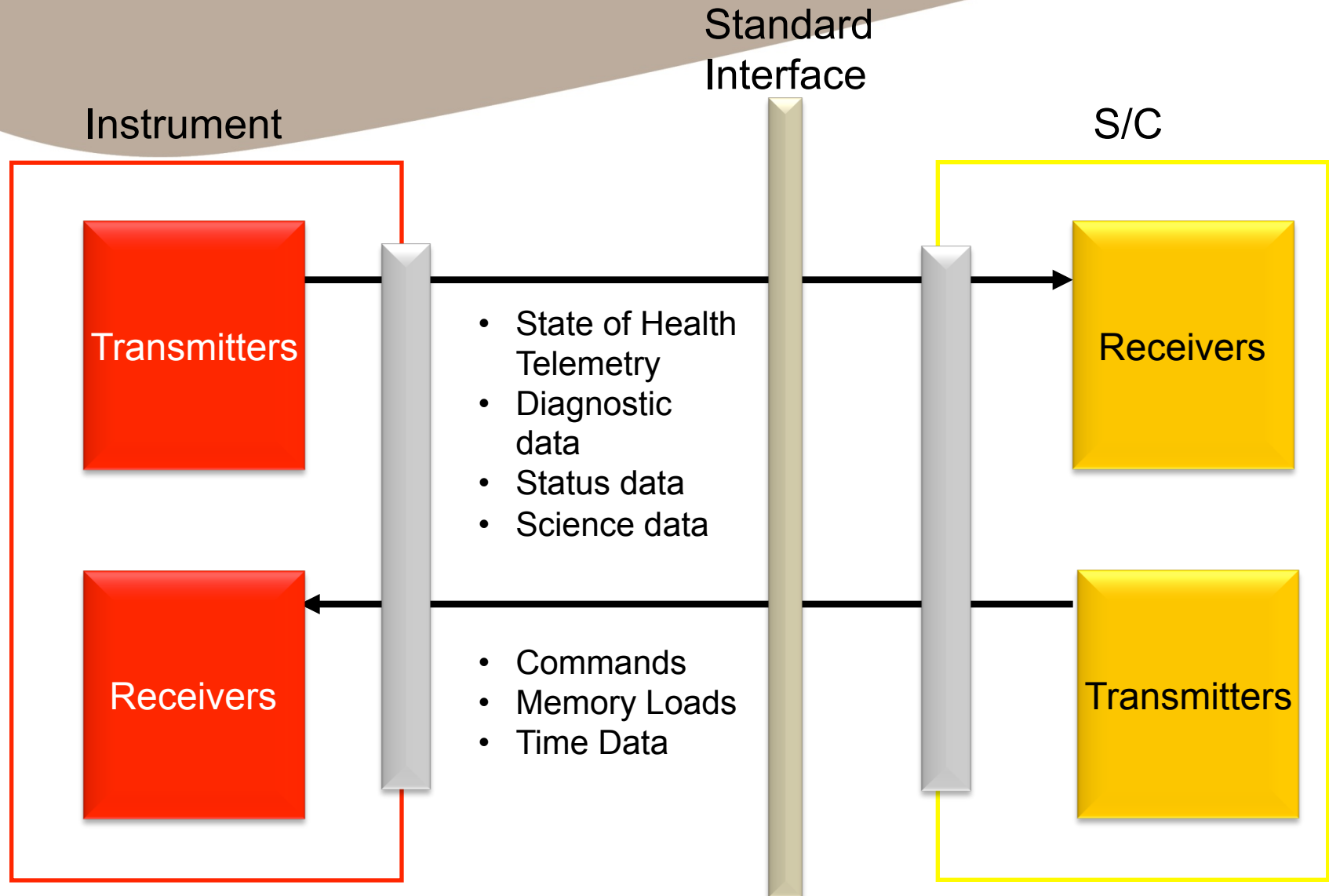


Figure 5: Data Transfer Interface

Post Design Interface

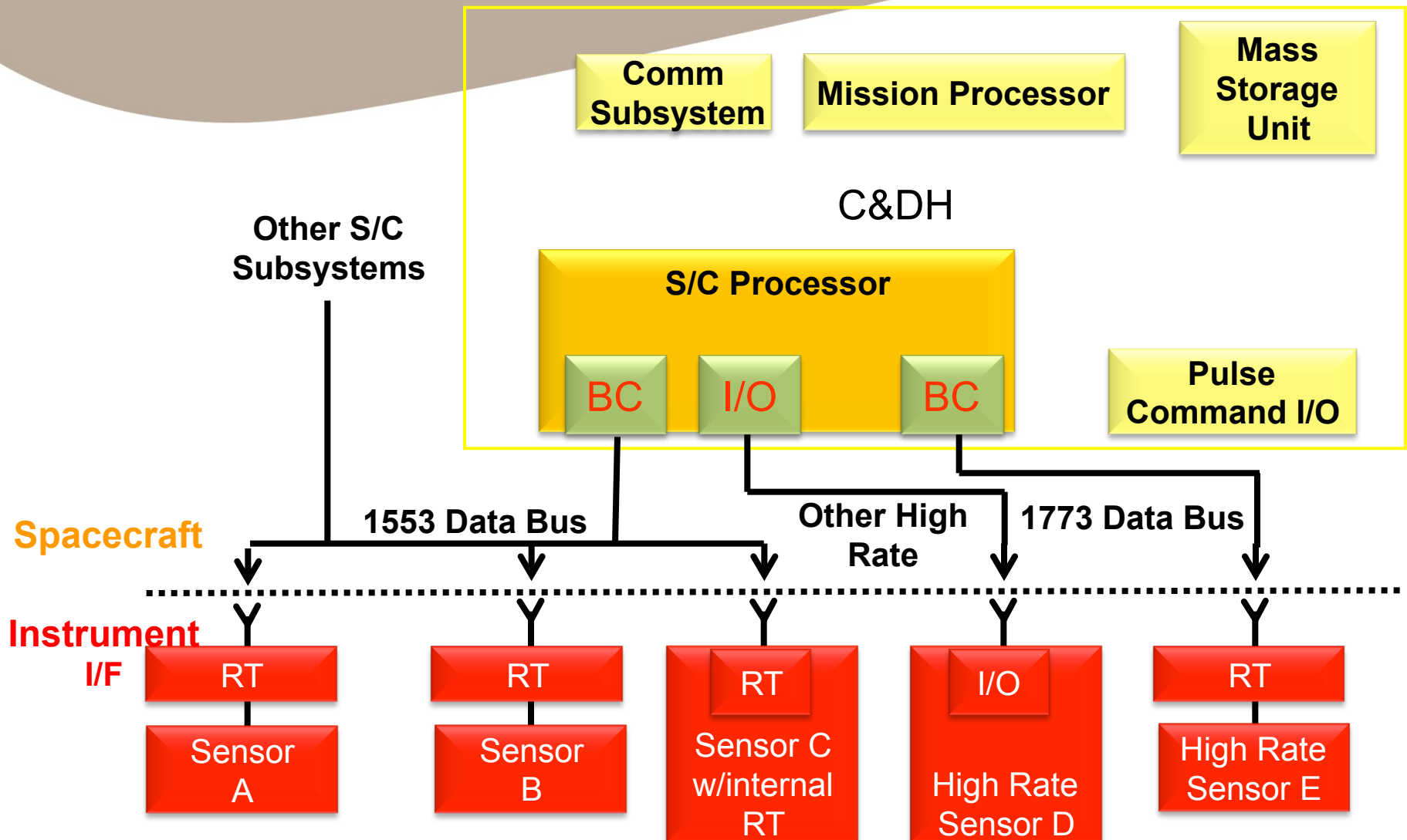
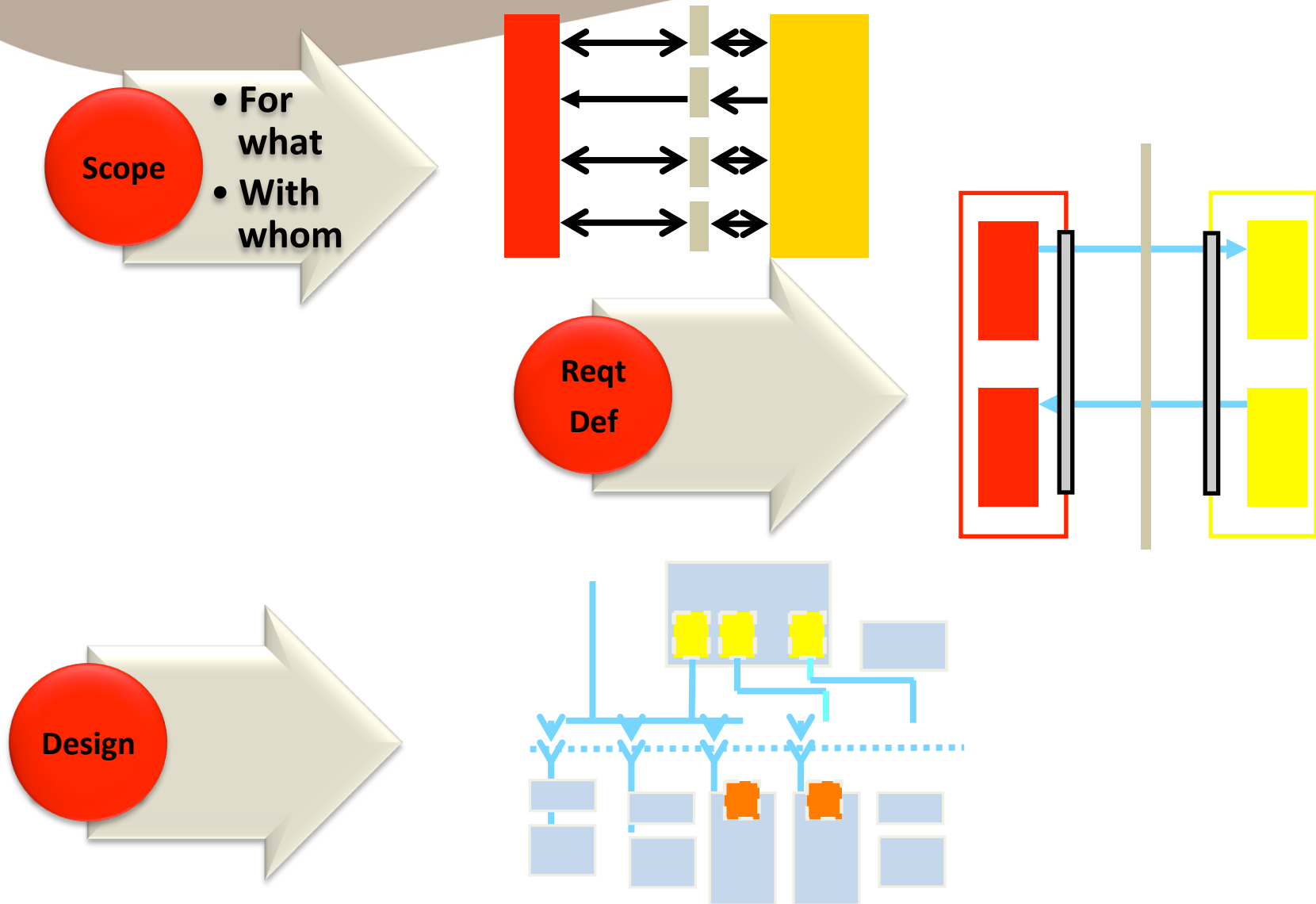


Figure 6: Notional Command and Data Handling Interface Topology

Data Interface Information

IRD contains all levels





Who uses an IRD, or equivalent document for interface definition, that is referenced from interfacing systems' specifications?

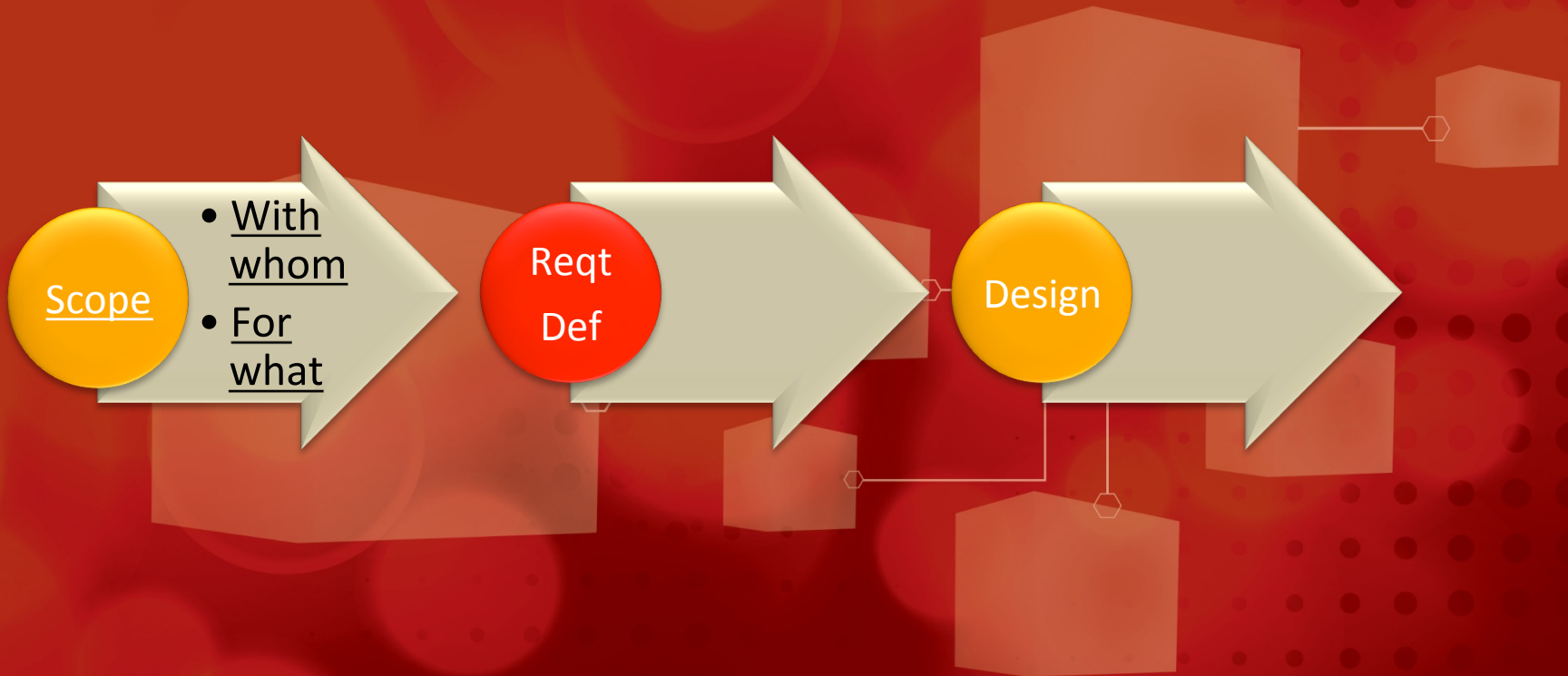
Who puts a *shall*-statement in that definition document?

Who has no standard way of doing interface requirements?

Does anyone put the system requirements in the IRD?



Writing Interface Requirements



Interface Documents

Specification



ICD

Standard

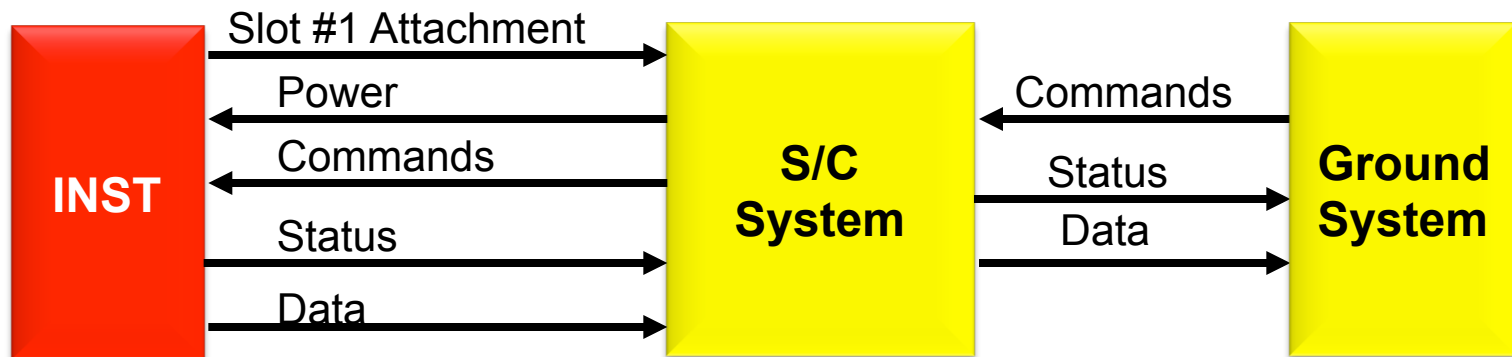
IRD

Document in your Specification

What	Where	Why	Caveats
Narrative description of external interfaces	3.1 System Description	Make boundary clear	There are no shall statements
Diagram of interfaces	Immediately after narrative	So everyone doesn't have to do this	Can be context diagram or picture or other graphical representation
Interface Requirements	3.2xx where needed	Impose on your system	Refer to ICD or IRD or higher level for info

Instrument - Interface Description

The Instrument will be installed in the S/C and will receive power from the S/C. The Instrument will receive ground commands via the S/C communications system and will supply its status and data to the ground through the S/C communications system.



Interface Requirement Instrument Specification

WRONG

- The S/C system shall provide power to the Instrument
- The Instrument shall interface to the S/C system.

RIGHT

- The Instrument shall obtain 28-volt power from the S/C system per ICD 1234 table 2.

Instrument Specification

Power Section

- **3.x.x.1 Power.**
- **3.x.x.1.1 The Instrument shall operate on 28 volts dc power**
 - Rationale: We will be using S/C power that is already 28 volts and we can obtain all our components to work at this voltage level
- **3.x.x.1.2 The Instrument shall receive ground power from a [certified test fixture].**
 - Rationale: Not sure how to handle this but want to make sure that anywhere we need to apply power on the ground we are only getting that power from a “good source” that cannot harm our system.
- **3.x.x.1.3 The Instrument shall receive power from the S/C per ICD ABCD table 4.2.**
 - Rationale: The ICD table provides all of the information needed to design for the S/C power interface.

Referencing Another Document

- In general, reference don't repeat information
- Use Section 2.0 Reference Documents for information about each document
- Reference a specific version – DO NOT put in “latest version” this is not acceptable.
- In requirement use short title or document number
- Refer to exact location, e.g. paragraph number

Reference Documents

Number	Title	Rev	Date	Sections
MIL-STD-464	Electromagnetic Environmental Effects Requirements for Systems	A	Dec 19 2002	3, 4, 6
MIL-STD-461	Requirements for the Control of Electromagnetic Interference (EMI) Characteristics of Subsystems and Equipment	E	Aug 20 1999	2 only

Citing a Reference Document



- **THIS:**

The System shall do something according to MIL-STD-464, Section 3.

The System shall do something as defined in MIL-STD-464, Section 3.

The System shall do something IAW MIL-STD-464, Section 3.

- **NOT THIS:**

The System shall do something according to MIL-STD-464, *Electromagnetic Environmental Effects Requirements for Systems, Rev A.*

Do's and Don'ts of Reference Documents

- Do not use “latest version” in the reference table or anywhere else.
- Do not update your document if you are not impacted by the change.
 - This is a contract change
 - You cannot just invoke a new Rev number
- Keeping track is simpler if you add a column to the table to show exact sections/ paragraphs called out in your requirement document.

Reference Change Impact

- You determine that you must change a reference – e.g. an ICD changes and you cannot interface successfully without invoking the change
- Prepare a change request
 - To update the Rev and date in the table
 - To change referenced paragraphs if needed
 - To elicit impact of the change from those affected.



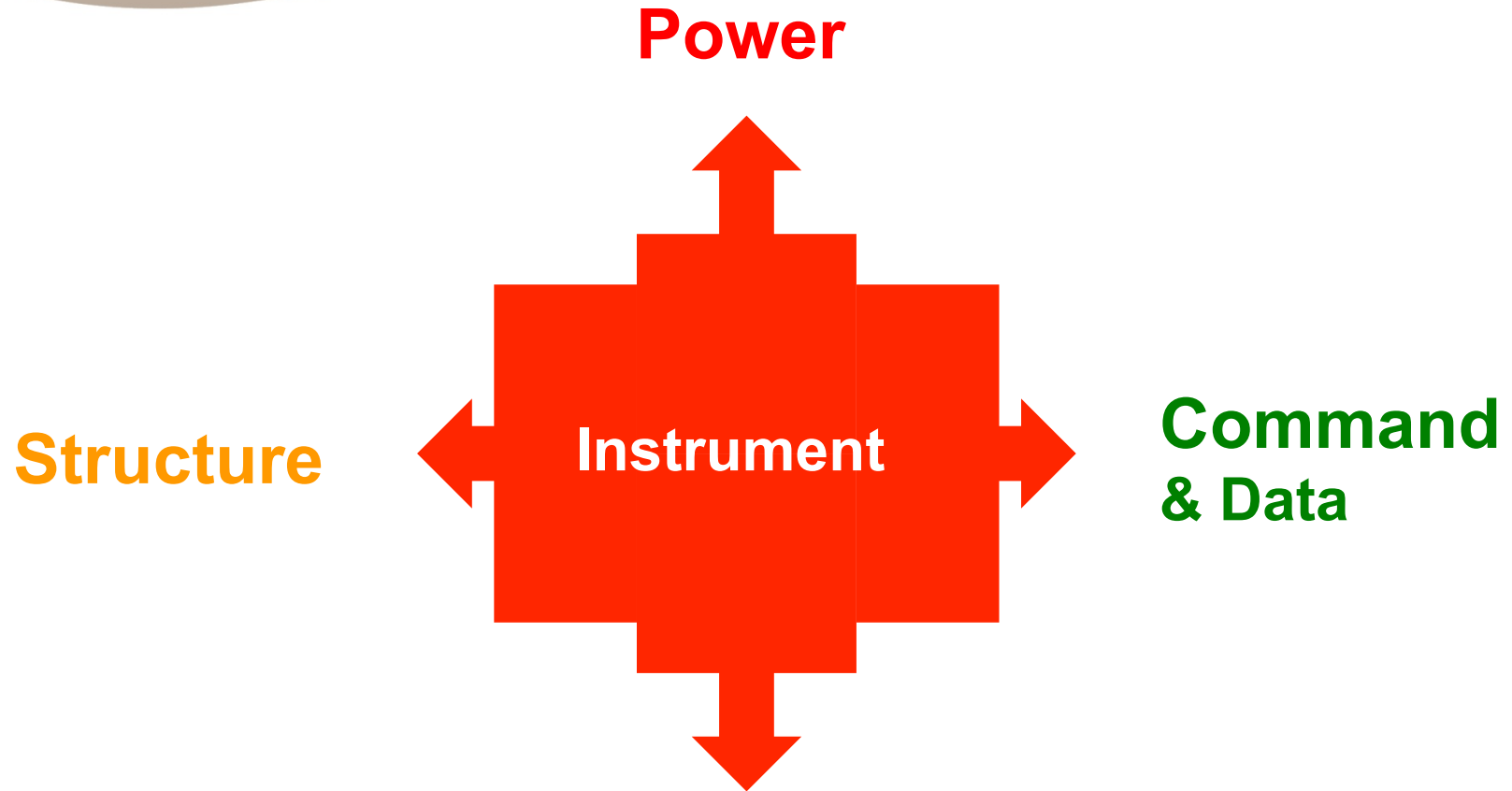
More on Interfaces

- Internal interfaces
- Managing interfaces
- Verifying interfaces

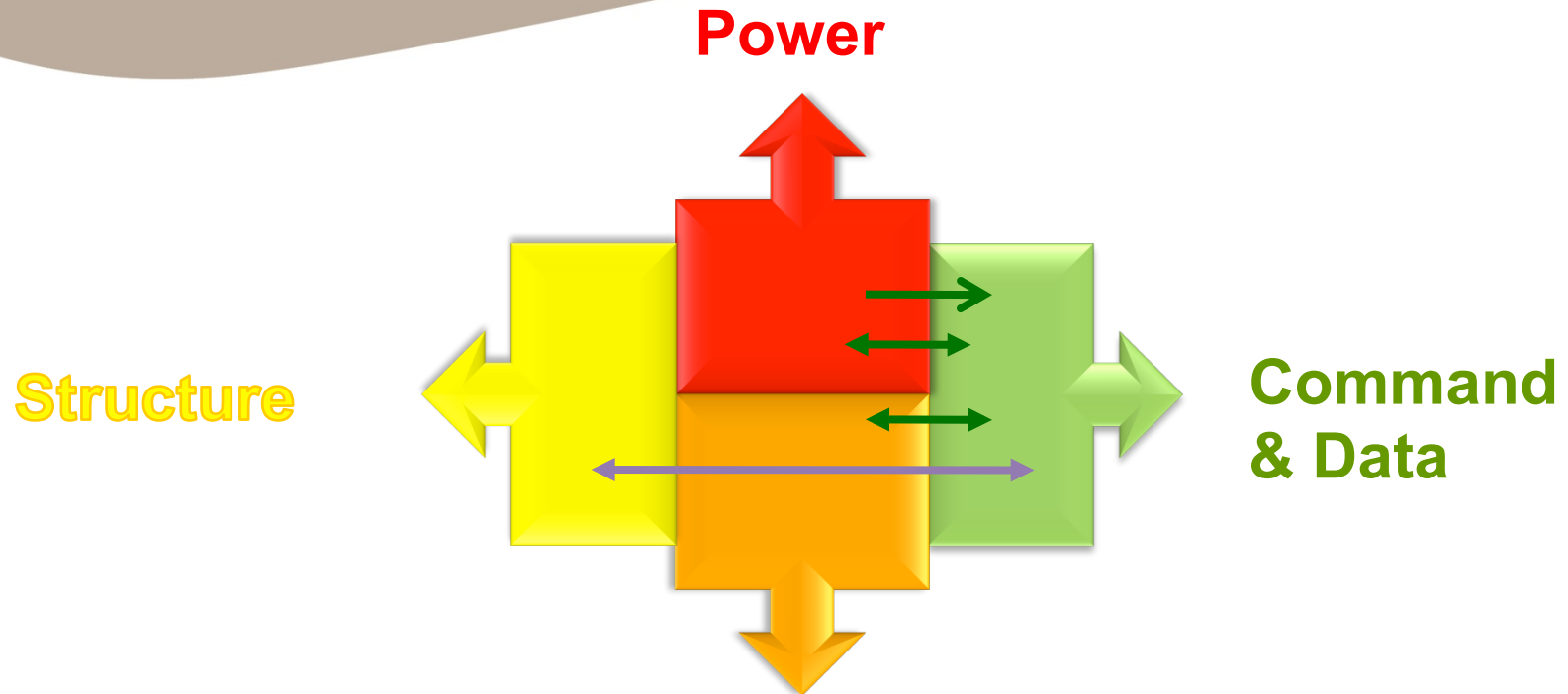


System External Interfaces

Instrument to S/C



Instrument Subsystems' Interfaces



- Subsystems will have external interfaces to the S/C
- Subsystems will have interfaces to each other – these are external to each subsystem and internal to the system

System Design

- Identifies subsystems
- Identifies where external interfaces meet the subsystems
- Identifies internal interfaces of the subsystems
- Allocates system requirements, including interface requirements, to the subsystems
- Adds a level of detail to the existing interface definition

System Internal Interfaces

- System design will drive interfaces between subsystems
- Each subsystem will have interfaces external to itself
 - Within the system
 - Requires process like external interfaces described above
 - Definition may be in System Spec or in separate I/F Def Doc
 - Outside the system
 - Continue with the I/F definition defined at the system level
 - Add more details

Consider throughout Interface Requirement Evolution

- How will each interface affect my system (subsystem)?
- How do I protect myself from their failures?
- How could my system affect the other side of the interface?
- How do I prevent my failures affecting them?
- Can the interfaces be quickly identified?
- Where is my system most sensitive?
- Where are the big risks?

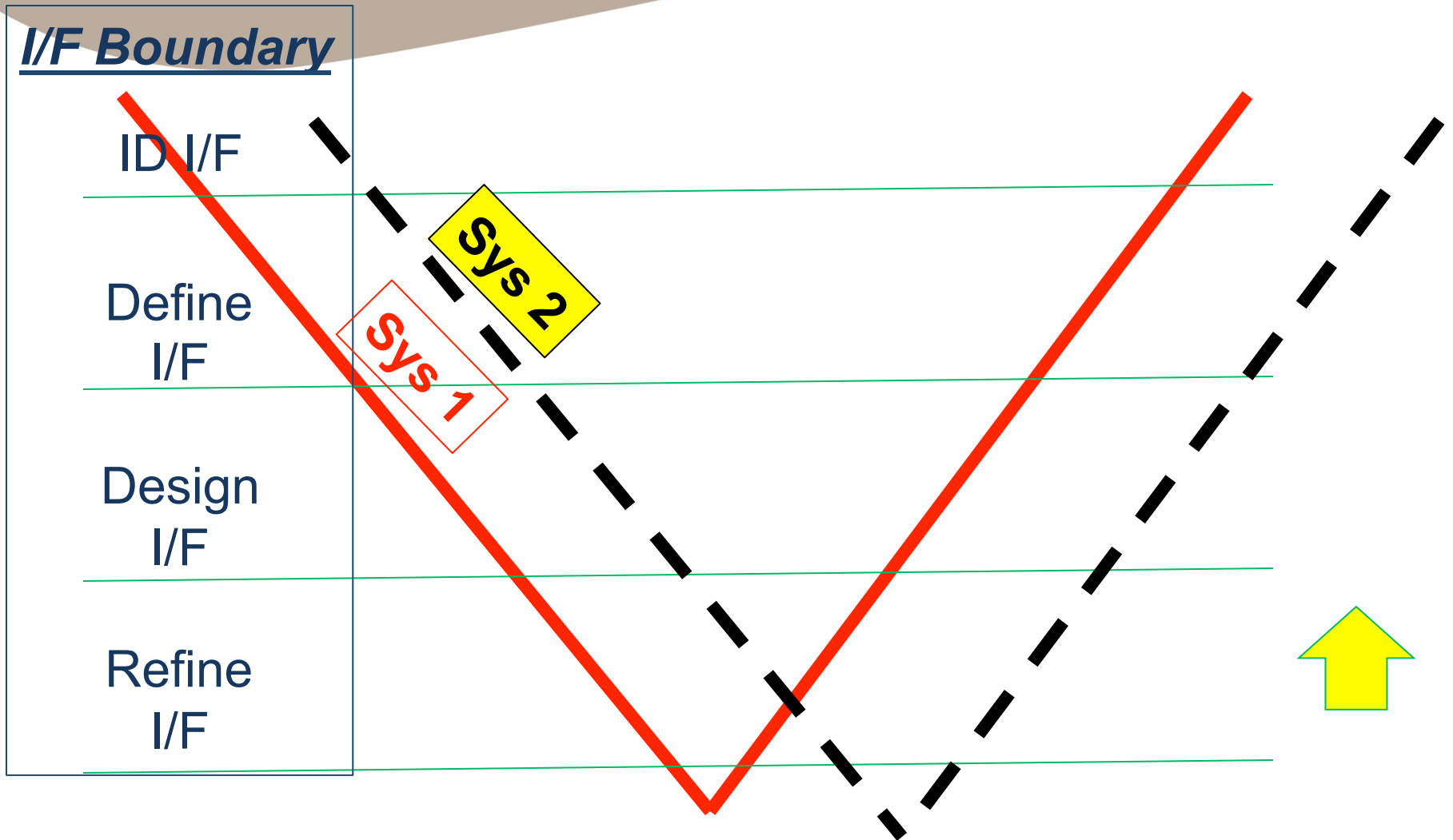
Managing Interfaces

- Identify interfaces at high level – what are the interfaces
 - Power
 - Mechanical
 - Data
- Determine how you will document each interface boundary (IRD)
- Each document needs a custodian
- Set up control board to review and approve IRD and subsequent changes

Managing Interfaces (cont)

- Assign subject matter experts (SME) from each side of the interface to follow throughout the life-cycle
- Provide traceability between the IRD and the specifications
- Schedule updates to your IRDs per your development plan – at SRR, PDR, CDR, TRR
- Assign a V&V or QA person to ensure the specifications, IRDs, and design documents stay in sync

The Vee and Interface Verification



Verification

- Verification
 - Each side: design meets the common definition
 - Each side: using models, simulators, actual hardware or s/w starting early
 - Both sides: but does it work
- IV&V especially useful
- Validation from beginning to end
 - Is I/F feasible
 - Are open items closing on schedule
 - Are design maturing toward the common definition



Requirements**Experts**

Training and Services for Project Success

Where do you go from here?

