Writing and Managing Interface Requirements

Ivy Hooks

Abstract

Caveat: If you have not had basic training or mentoring in writing requirements, this tutorial assumes you have those skills. This tutorial assumes you know how to write a good requirement and avoid the basic requirement defects. It also assumes you have seen some interface requirements, at some level in your work.

One of the more difficult requirement skills involves writing and managing interface requirements in an effective manner. This topic is covered at an introductory level in other seminars, but this seminar gets into significant details. These details are not available in the earlier classes because it takes some experience improving other requirement skills before tackling these details. This seminar will aid the student in improving processes in working with interface requirements and will provide further insight into why some options do or don't work.

Students will be involved in discussions about their own problems in writing, documenting and managing interface requirements. Class lecture and exercises will help to find answers to many of these problems such as:

- Different types of interfaces
    - With existing systems
    - Between two developing systems
- Techniques for identifying interfaces
    - Operational concepts
    - Context diagrams
    - Stakeholder related
- Major categories of interfaces
    - Physical
    - Electrical
    - Electronic
    - Mechanical
- Expansion of categories
- Where and how to document an interface
    - System Specification
    - ICDs and IRDs
- What goes in your specification
    - What doesn't go in your spec
    - Why you need info in your spec
- How interfaces are verified
    - From you viewpoint
    - From the system viewpoint

Writing and Managing Interface Requirements

Ivy Hooks

Biography

Ivy HOOKS, USA, is CEO of Compliance Automation Inc., a charter member of INCOSE, an internationally recognized expert in requirements, an author and speaker. Ivy has managed and owned a company since 1985, Ivy previous had a highly exciting 20-year career at the NASA Johnson Space Center where she served on the initial Space Shuttle design team and in a number of engineering and management jobs. Ivy holds a BS and MS in Mathematics from the University of Houston and is the recipient of a number of awards.

Advanced Writing and Managing *Interface Requirements*

INCOSE 2010

by

Ivy Hooks

of

Compliance Automation, Inc.

The Requirement Experts

Tutorial 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

What’s Coming Next

OVERVIEW

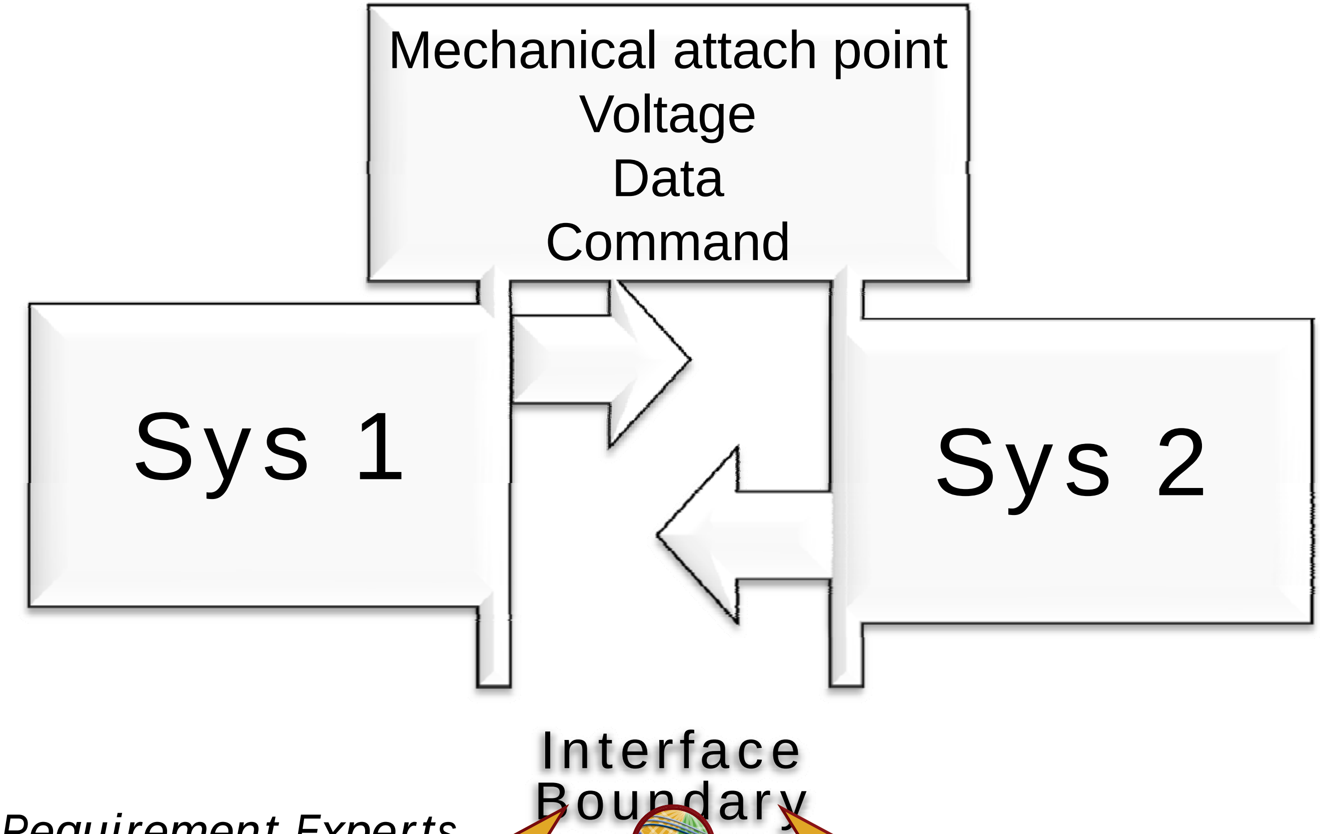
- Major types of interfaces
- What is an interface
- How do interfaces fit into the product life-cycle
- What do interface requirements look like

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

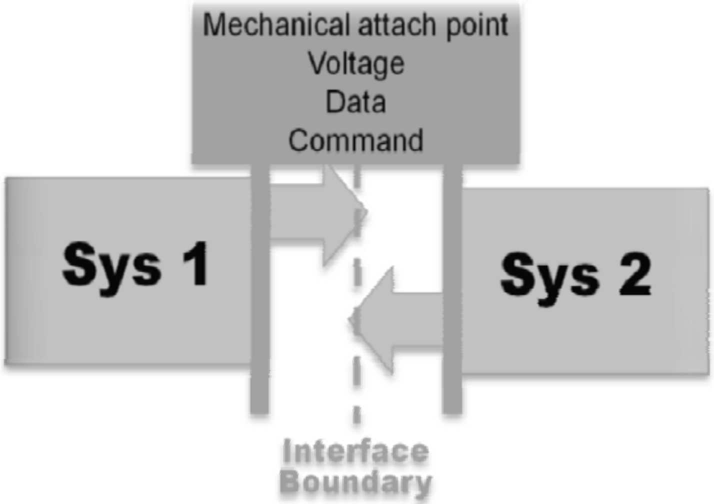
- Launch vehicle exists – has an ICD
- Spacecraft (S/C) exists – has an ICD

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

Can you think of others?

ICD: Interface Control Document

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.

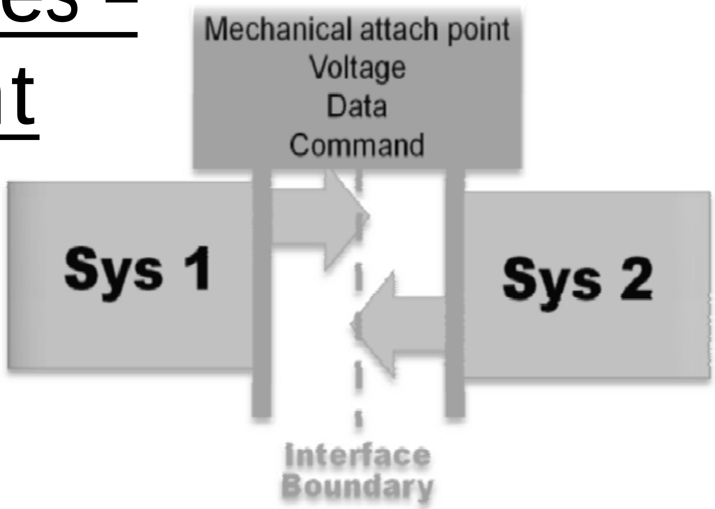
ICD defines boundary

- 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, or
  - Mandated from higher level, or
  - **Worked out between two systems**

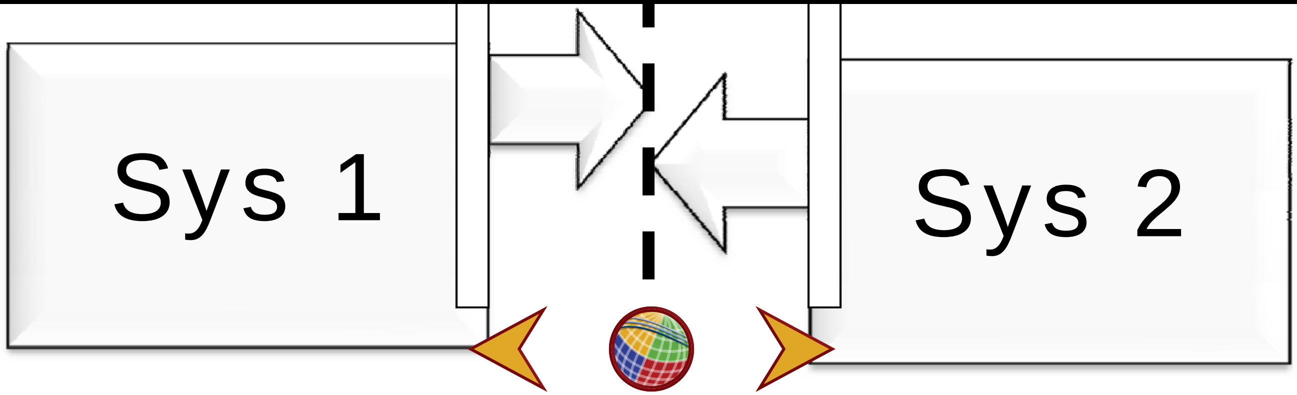
Identification of Interfaces – 2 systems in development



- 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

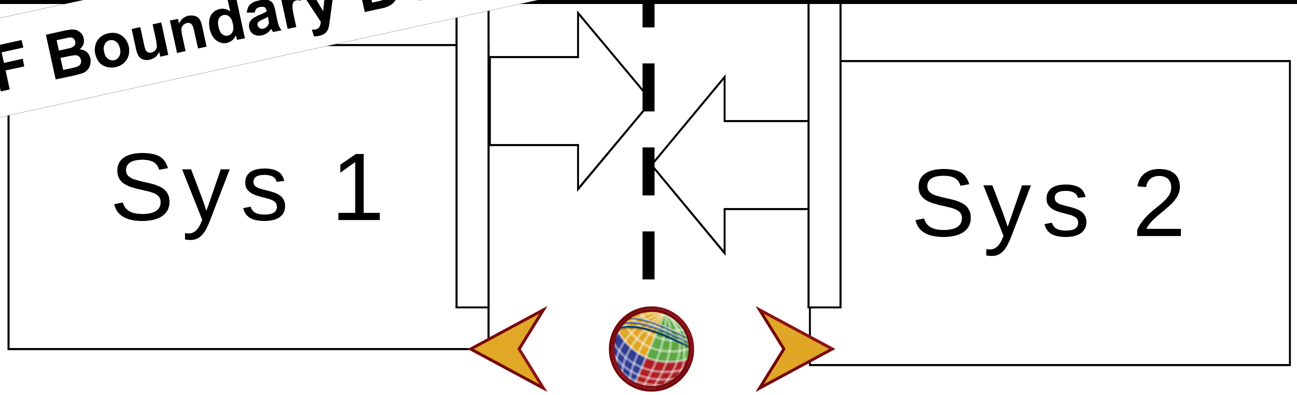
Sys 1 Specification	*I/F Boundary Doc* **#1234**	**Sys 2 Specification**
Sys 1 shall transmit xyz command defined in #1234 to Sys 2.	*The xyz command will have the characteristics described in Table TBD.*	*Sys 2 shall execute* the xyz command defined in #1234.



Interface Requirements Between Two Developing Systems - Identified

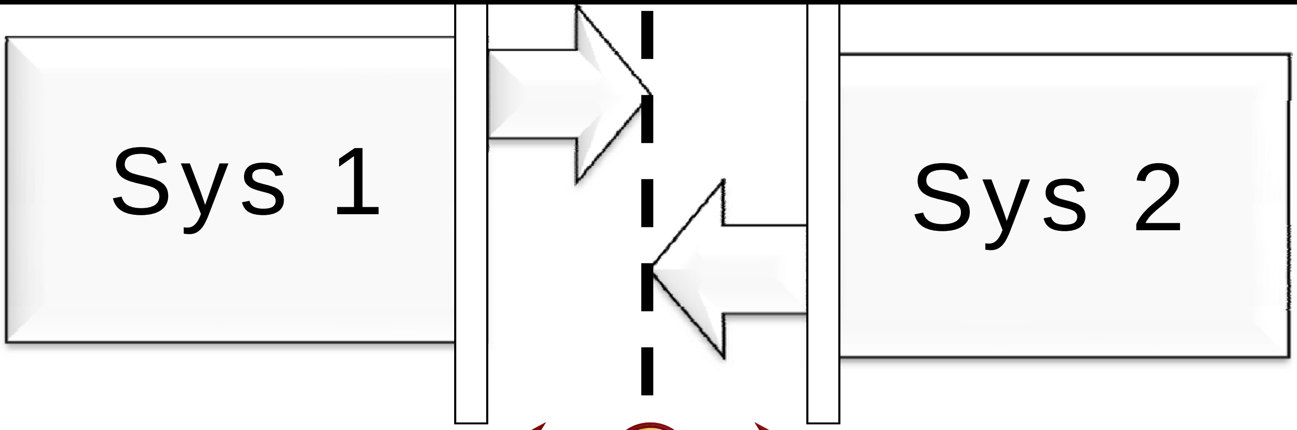
Sys 1 Specification	*I/F Boundary Doc* **#1234**	**Sys 2 Specification**
Sys 1 shall transmit xyz command defined in #1234 to Sys 2.	*The xyz command will have the characteristics described i[illegible] B.*	*Sys 2 shall execute* the xyz command defined in #12[illegible] Tabl[illegible]

I/F Boundary Doc DOES NOT contain a shall statement.

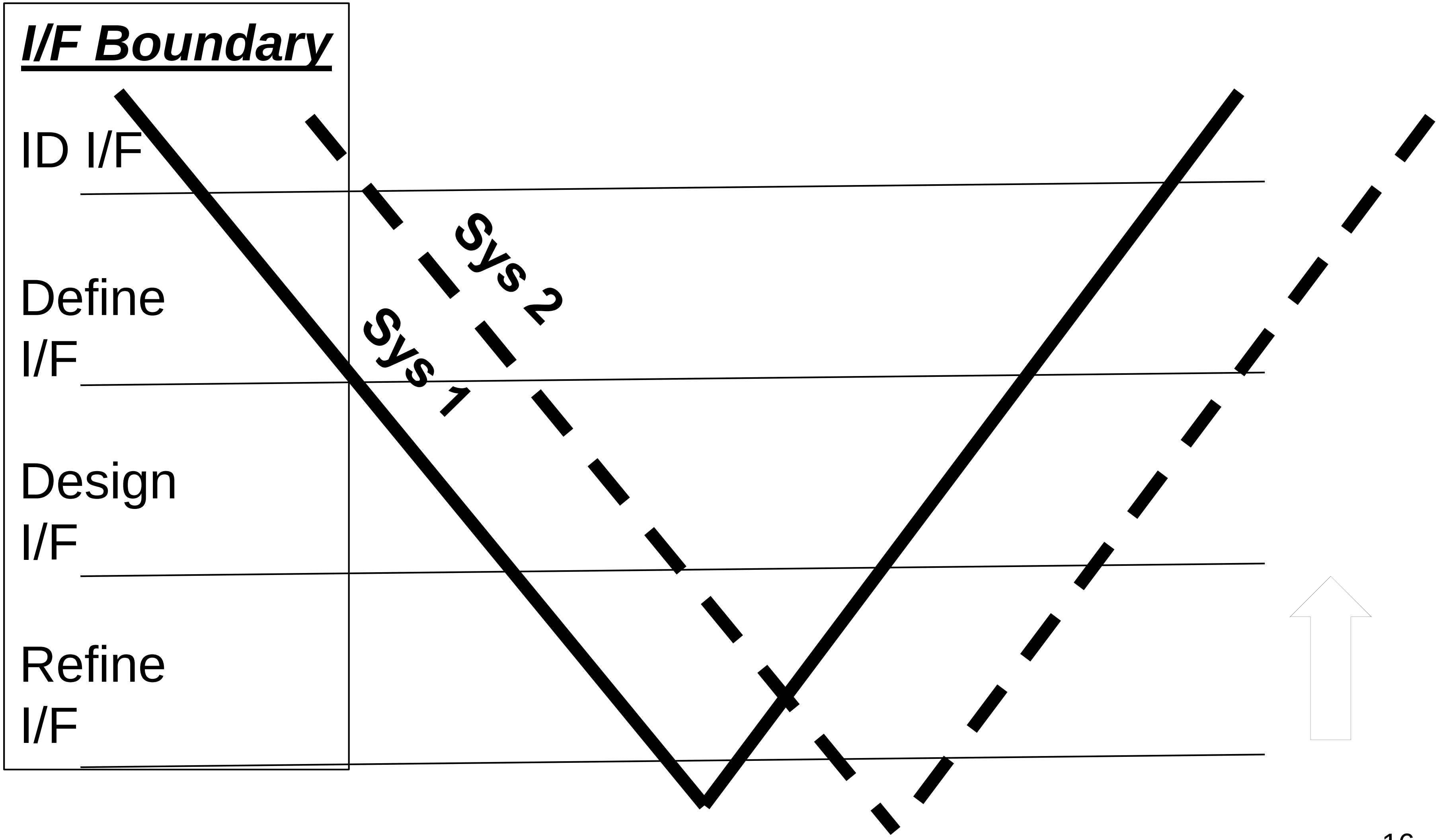


Interface Requirements Between Two Developing Systems - Defined

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



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



I/F Boundary

ID I/F

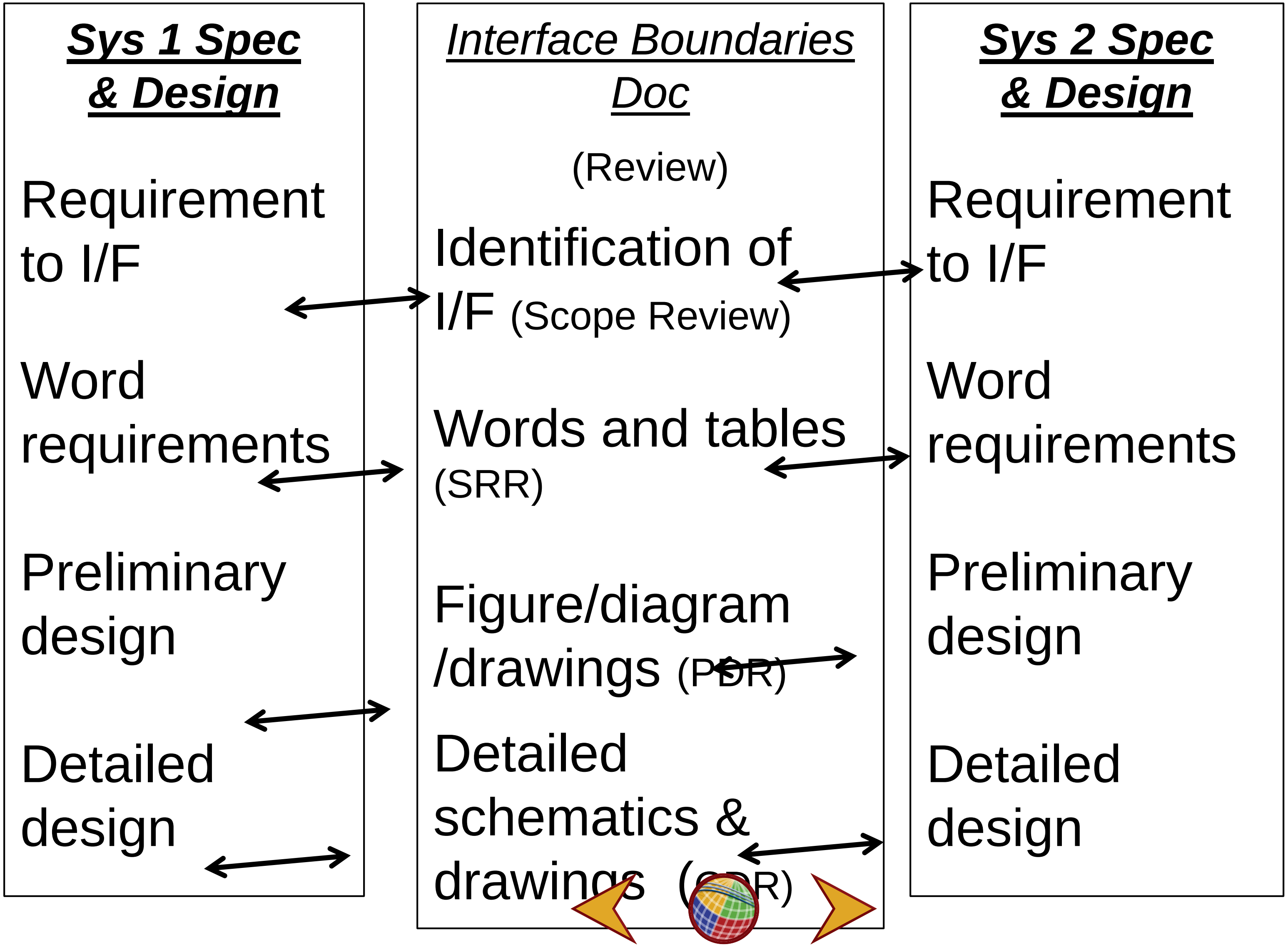
Define I/F

Design I/F

Refine I/F

Validation of Command and Data I/F

Often and at each review



Sys 1 Spec & Design

Requirement to I/F

Word requirements

Preliminary design

Detailed design

Interface Boundaries Doc

(Review)

Identification of I/F (Scope Review)

Words and tables (SRR)

Figure/diagram /drawings (PDR)

Detailed schematics & drawings (CDR)

Sys 2 Spec & Design

Requirement to I/F

Word requirements

Preliminary design

Detailed design

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.

What's Coming Next

- 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

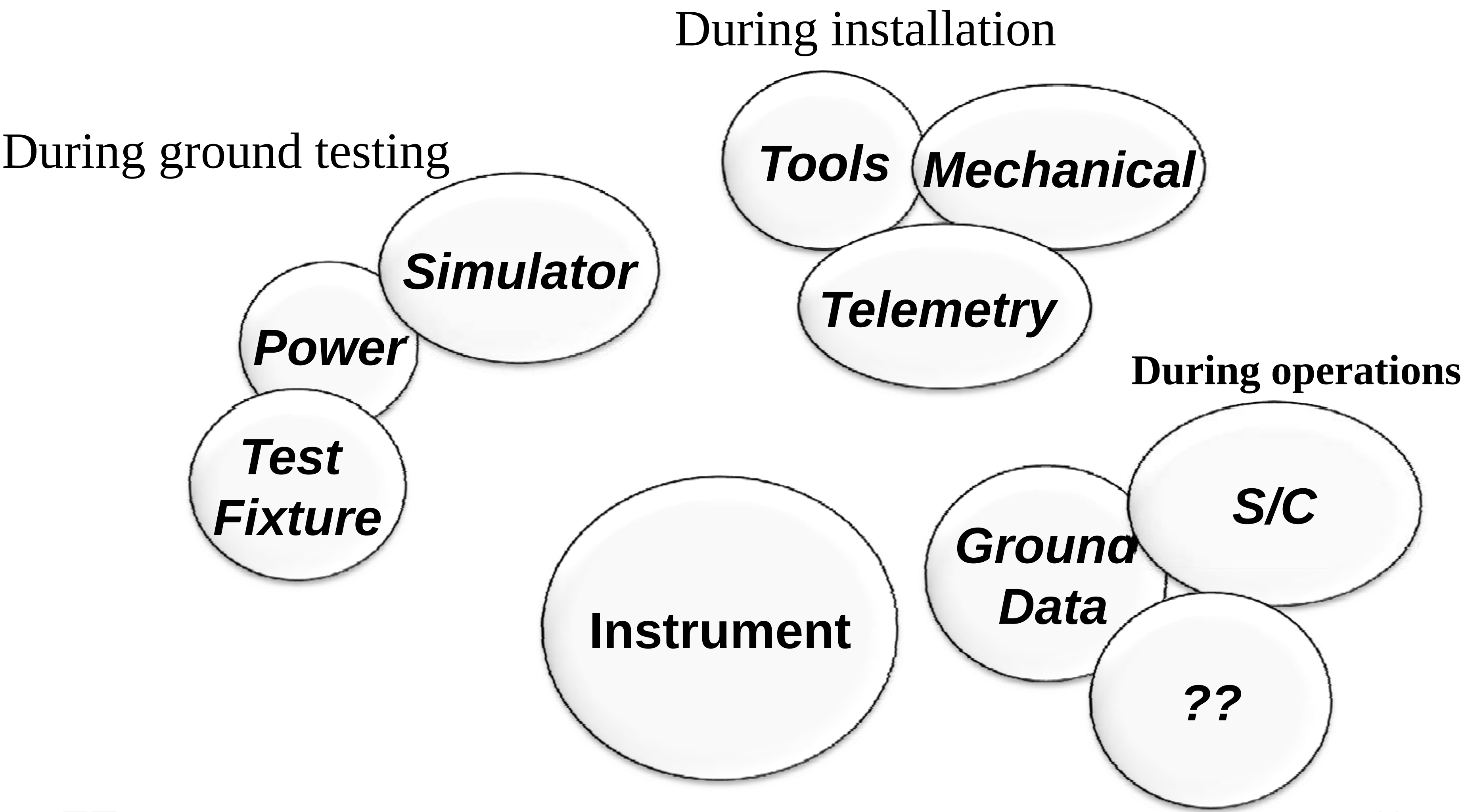
***Customer-Centered Products*, p. 9**

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



During installation

Tools

Mechanical

Telemetry

During ground testing

Simulator

Power

Test Fixture

Instrument

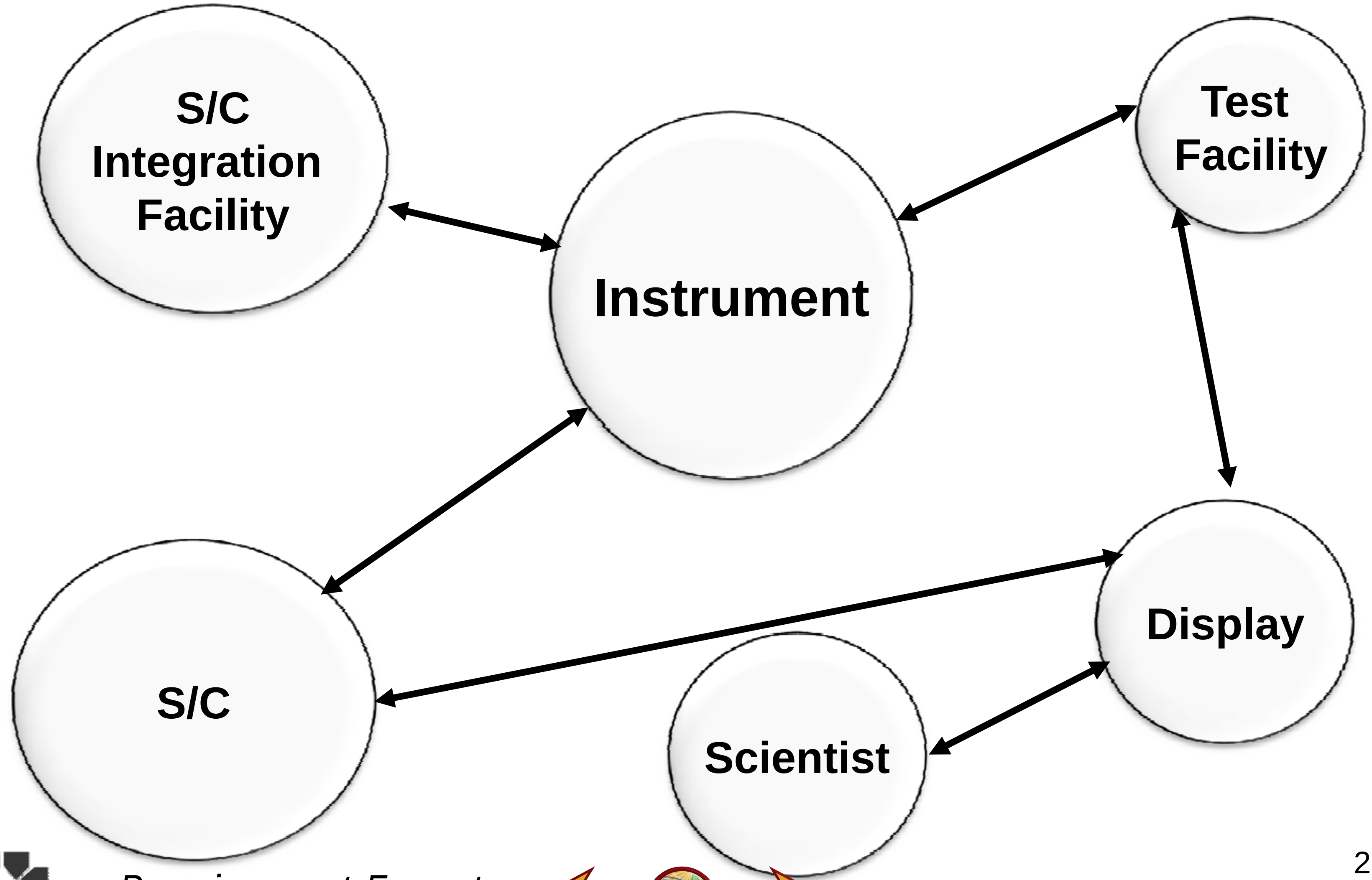
During operations

S/C

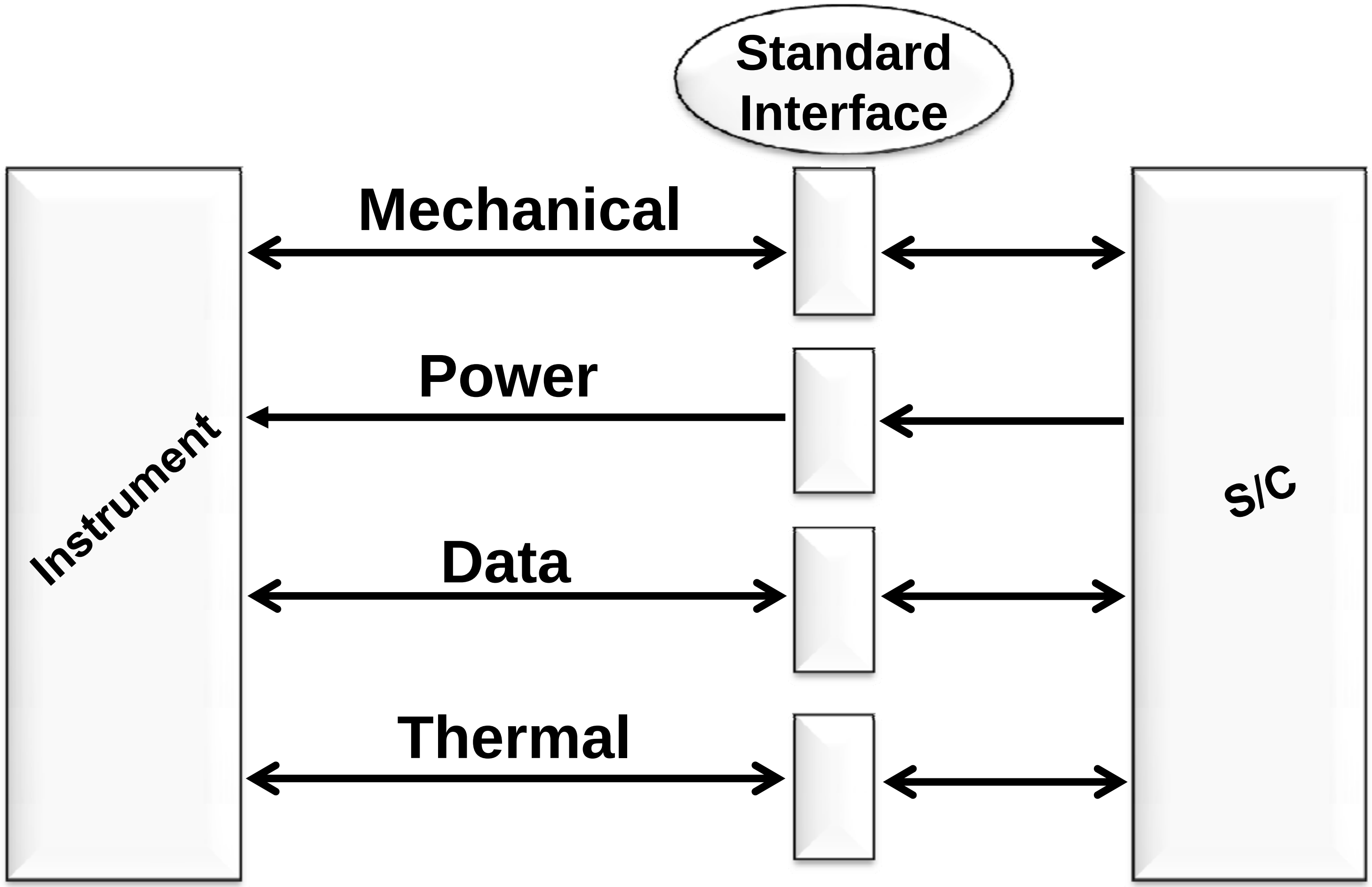
Ground Data

??

Use Operational Concepts

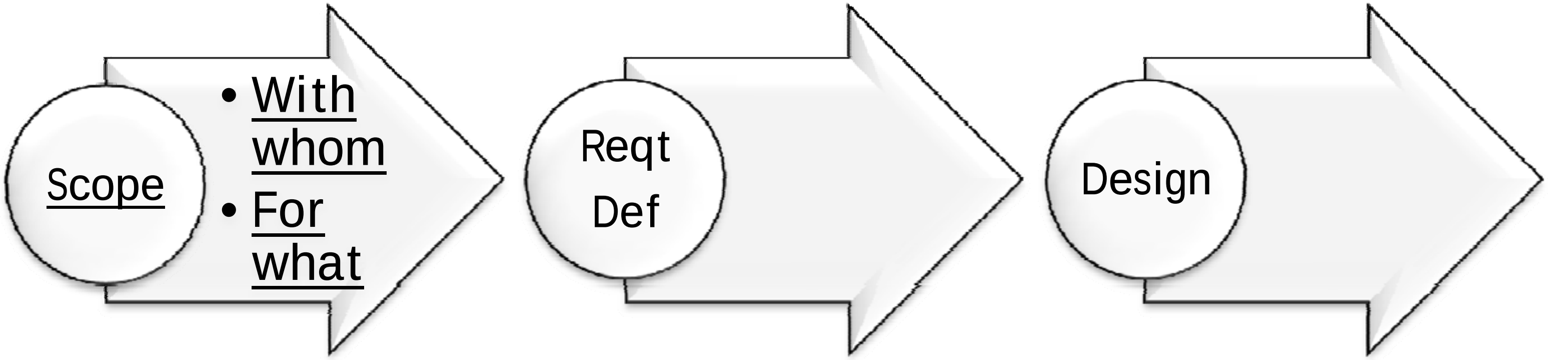


Interface Block Diagram



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

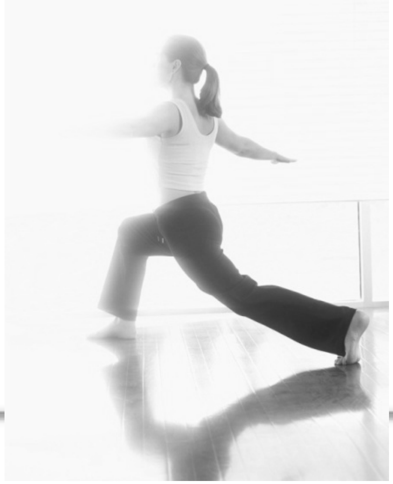
Scope Definition Phase

- 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

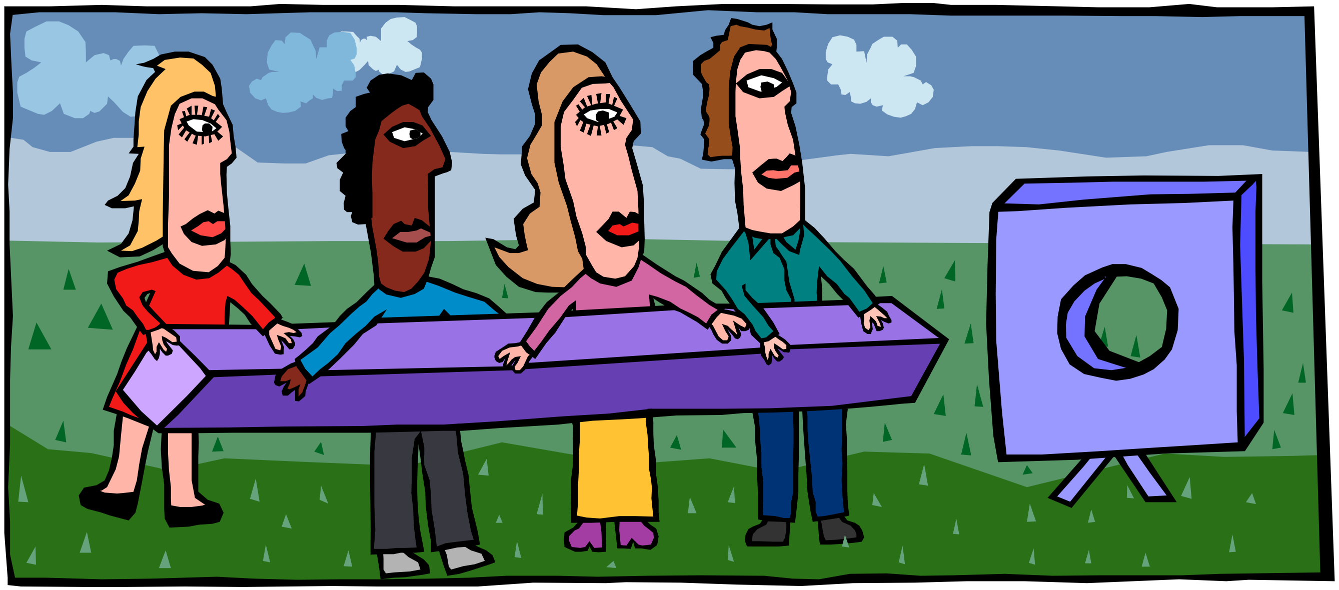
- How many of you do interface definition in the Scope phase – during formulation, concepts, or whatever you call the phase ?

- What techniques do you use? Ops Concepts? Context diagrams? Work from lists of previous interfaces? Other?



What's Coming Next

- Interfaces – More Details about Categories

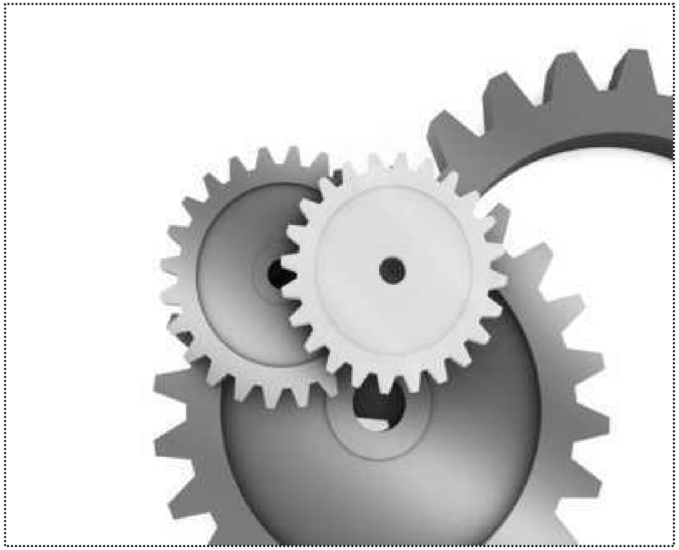


Major Categories of Interfaces

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

Physical

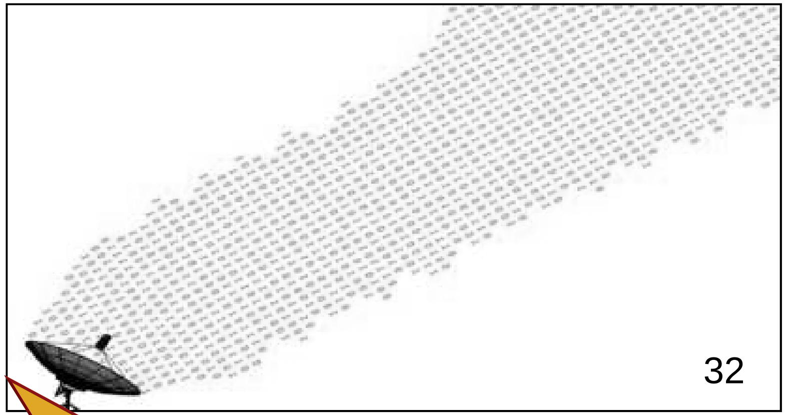
- Mechanical
  - Envelope
  - Attachment/Mounting
  - Obscuration
  - Alignment
- Tooling



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)

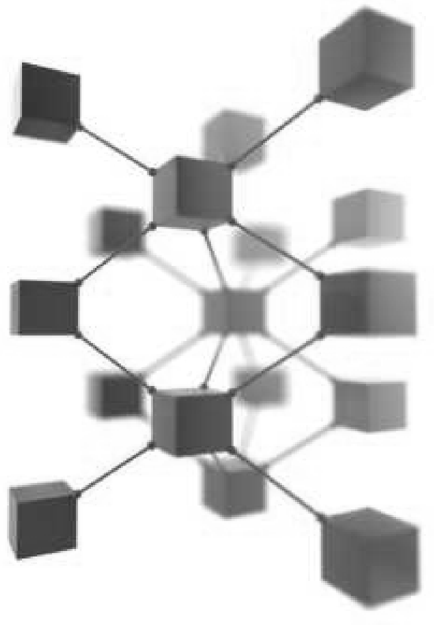
Hardware/Software

- 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

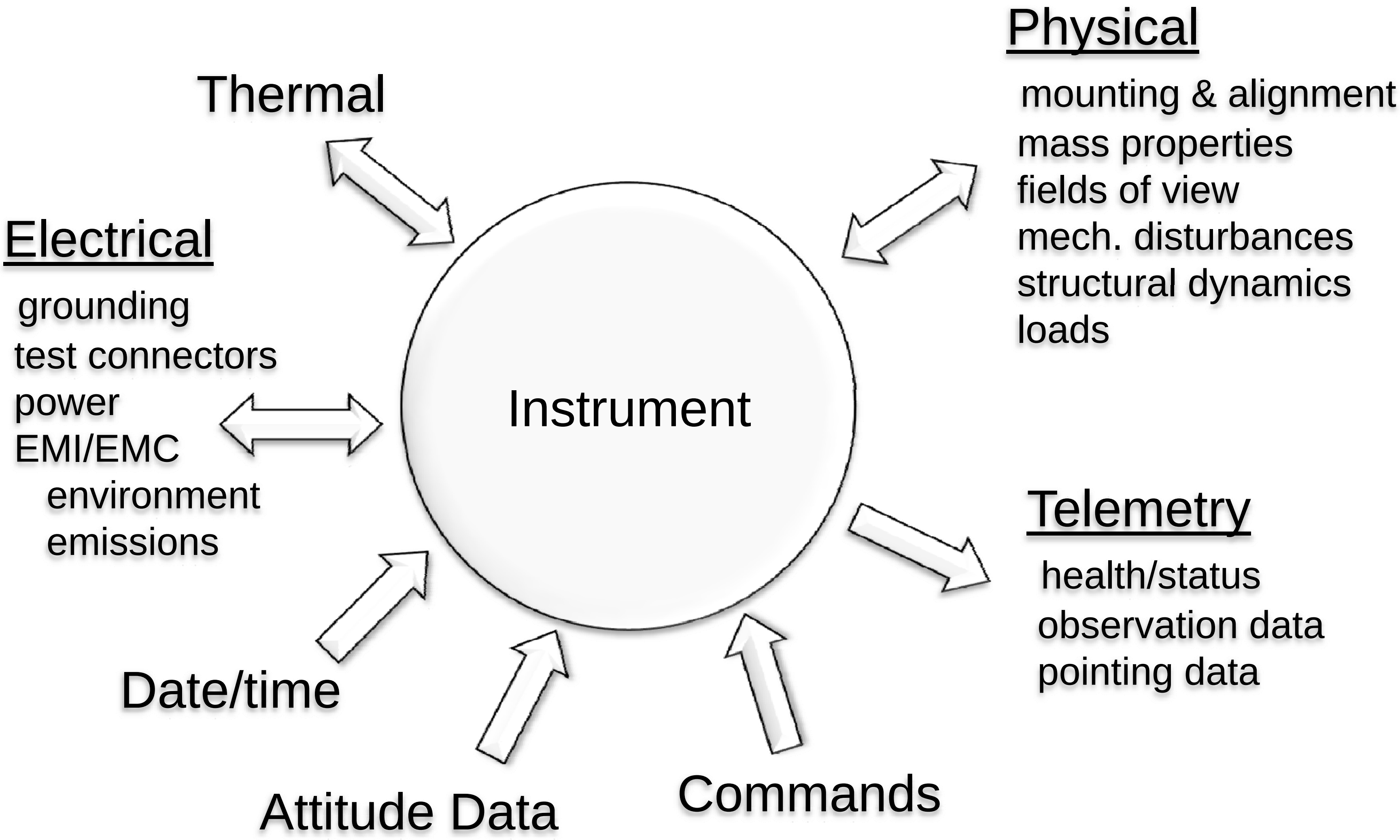
- Structural
  - Vibration
  - Shock
  - Acoustic
  - Loads
  - Dynamic mode shapes
- Thermal
  - Temperature range
  - Heating rates
  - Heat transfer surfaces
- Magnetic
  - Flux density
  - Rate-of-Change

- 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



Instrument

Thermal

Electrical

grounding

test connectors

power

EMI/EMC

environment

emissions

Physical

mounting & alignment

mass properties

fields of view

mech. disturbances

structural dynamics

loads

Telemetry

health/status

observation data

pointing data

Date/time

Attitude Data

Commands

- 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?

What's Coming Next

- Documenting interface boundaries

ICD

Specification

Standard

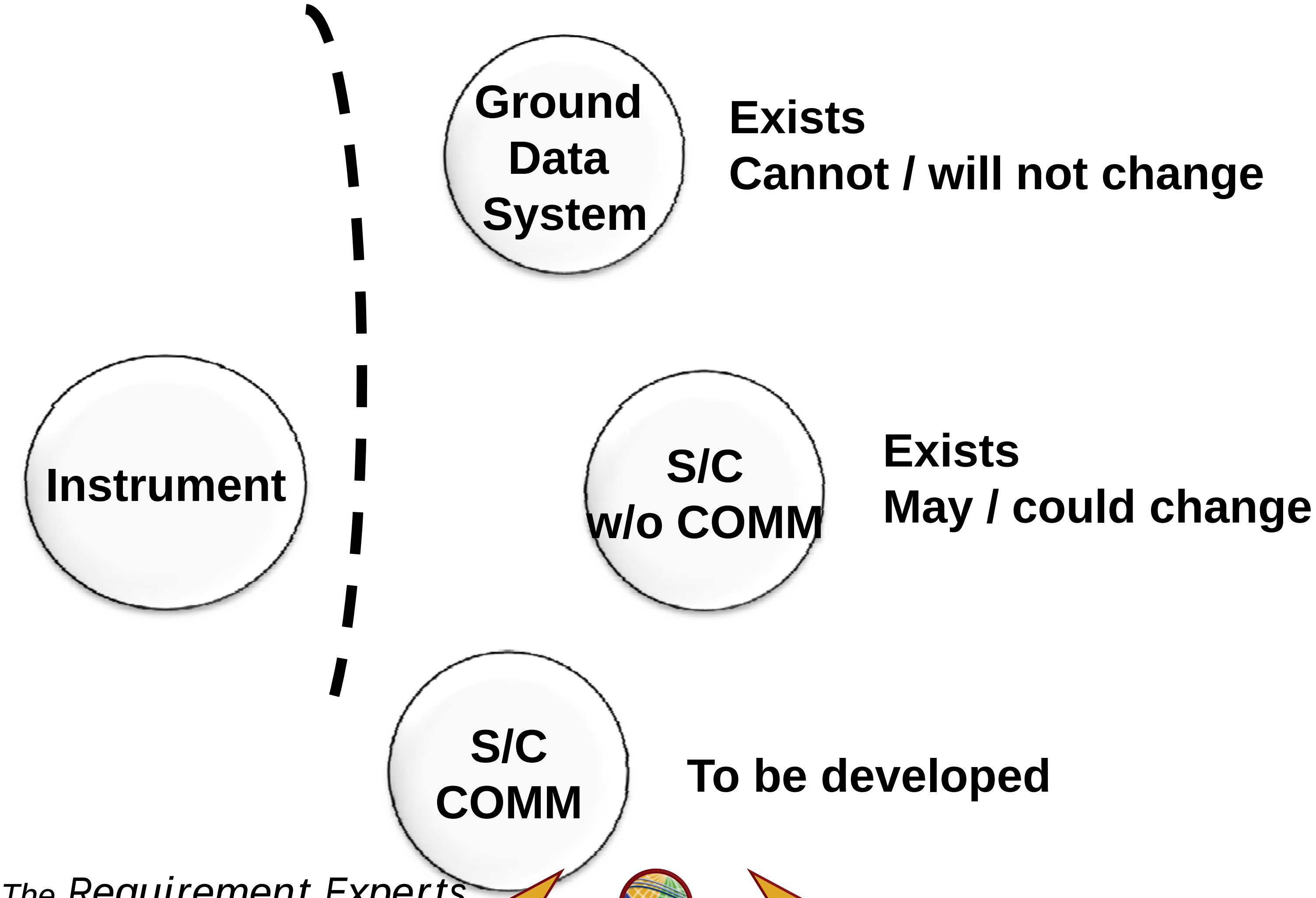
IRD

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 Interfaces

Some existing and some to be developed



Instrument

Ground Data System

Exists
Cannot / will not change

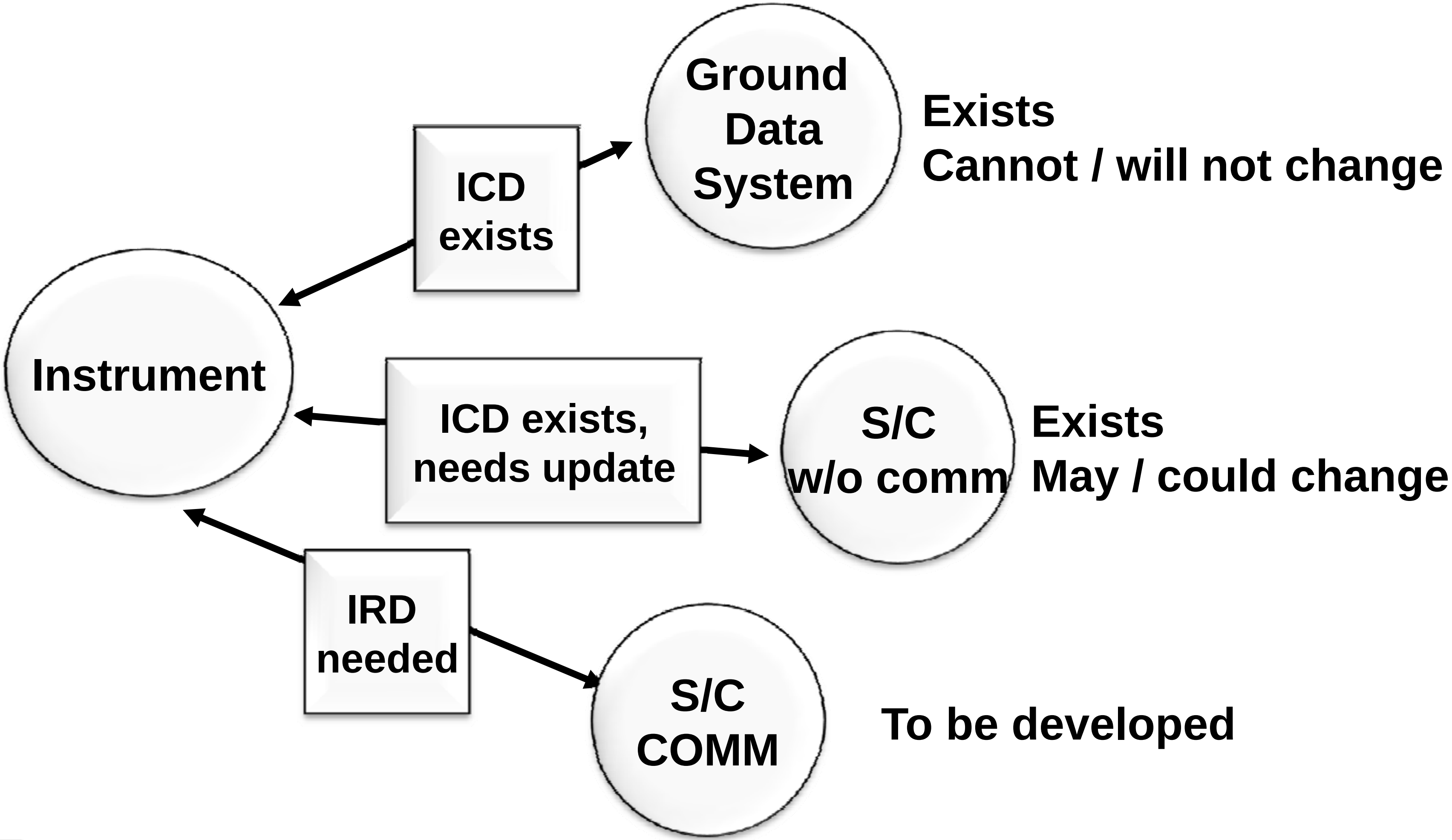
S/C w/o COMM

Exists
May / could change

S/C COMM

To be developed

Maturity of Interface Documents



Instrument

ICD exists

Ground Data System

Exists
Cannot / will not change

ICD exists, needs update

S/C w/o comm

Exists
May / could change

IRD needed

S/C COMM

To be developed

What's the Difference

Interface control document (IRD):

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

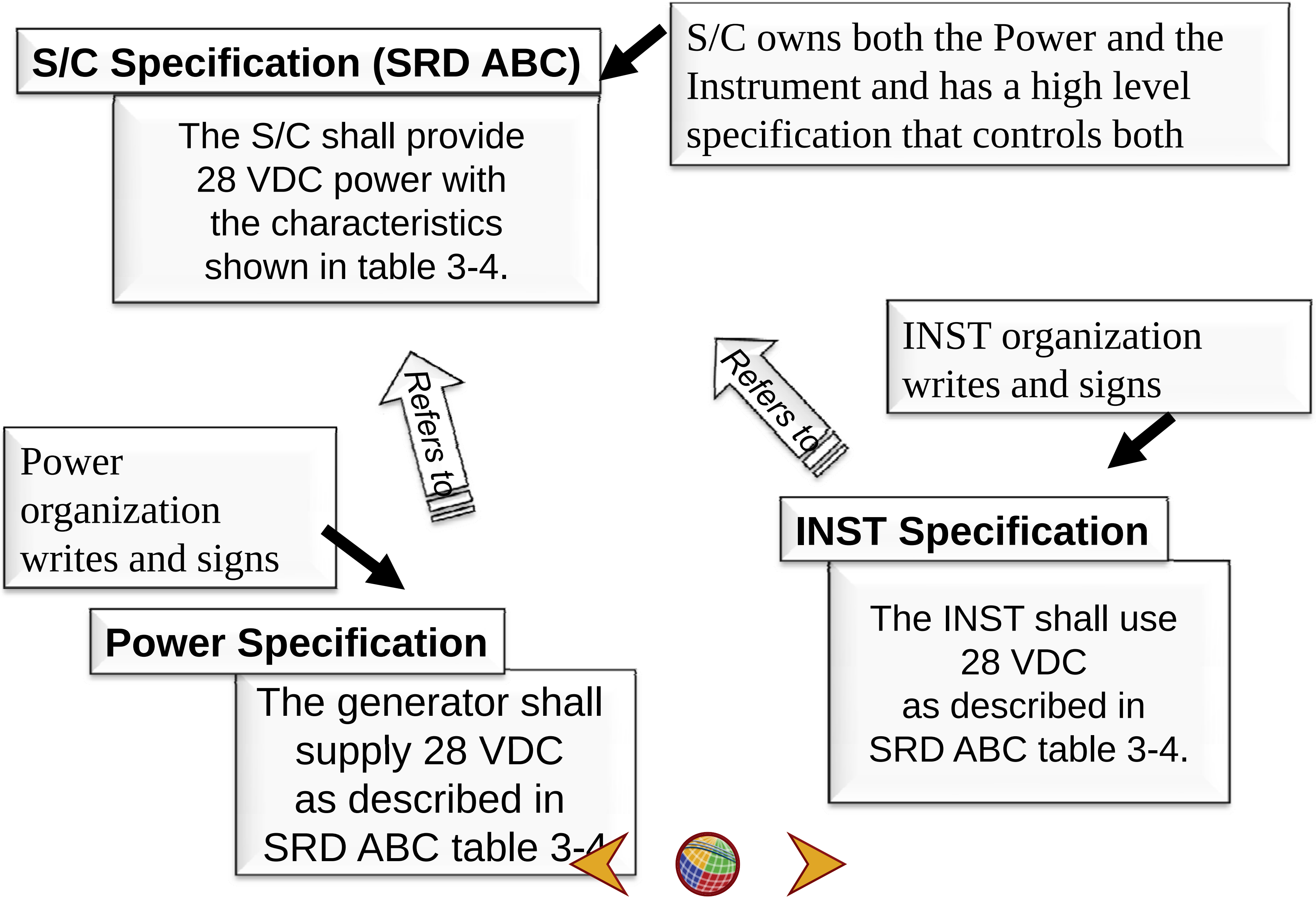
- Two sides in development – options
  - Upper Level SRD – managed by SRD change board
  - IRD – managed and signed jointly by the managers of the two interfacing systems
  - Evolving ICD – managed and signed jointly as IRD

Existing System 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

Refers to

Refers to

INST organization writes and signs

Power organization writes and signs

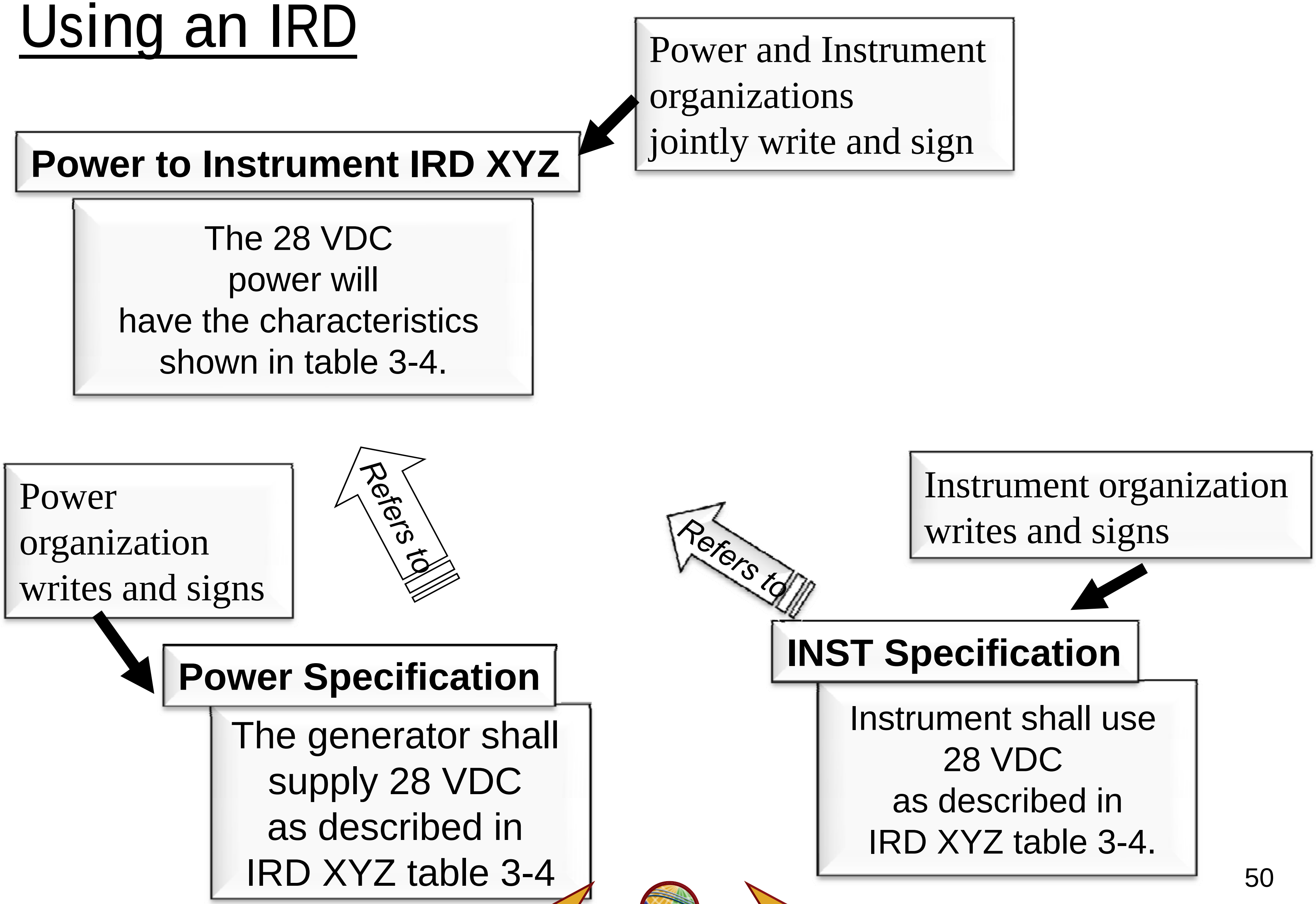
INST Specification

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

Power Specification

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

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

Power Specification

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

Instrument organization writes and signs

INST Specification

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

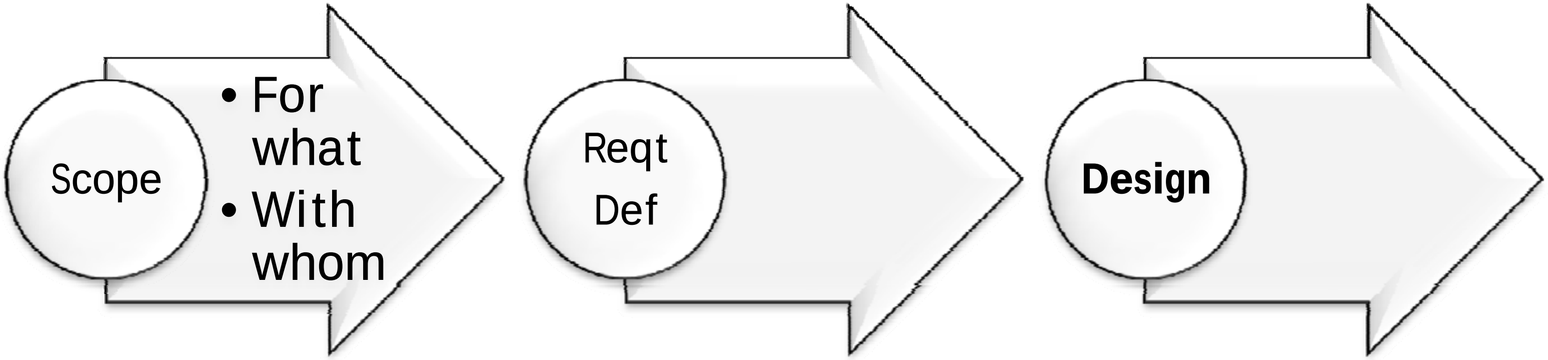
Interface Requirements Document (IRD)

- 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

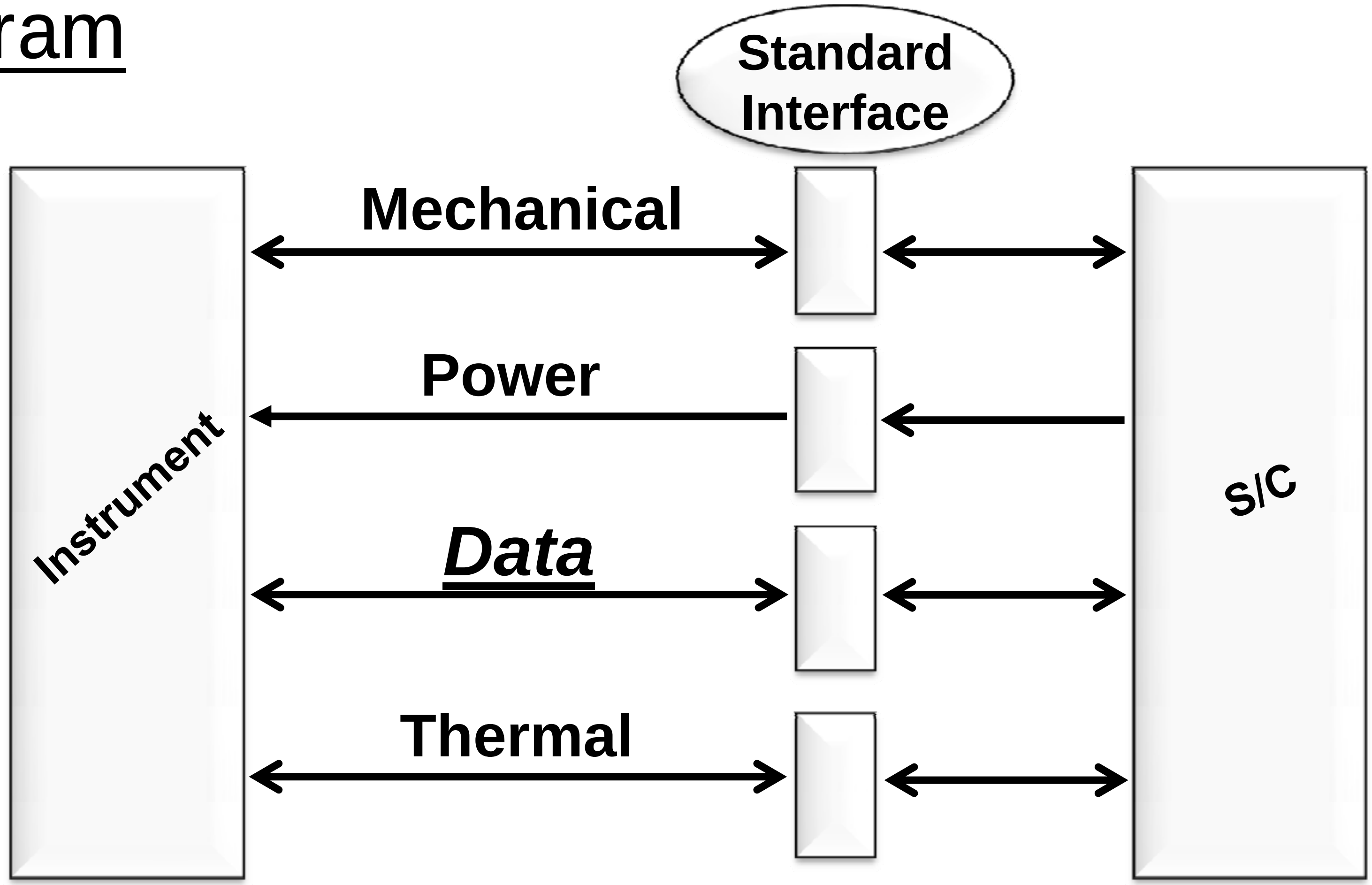
Benefits of a good ICD/IRD

- 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

Documenting interface boundary evolution

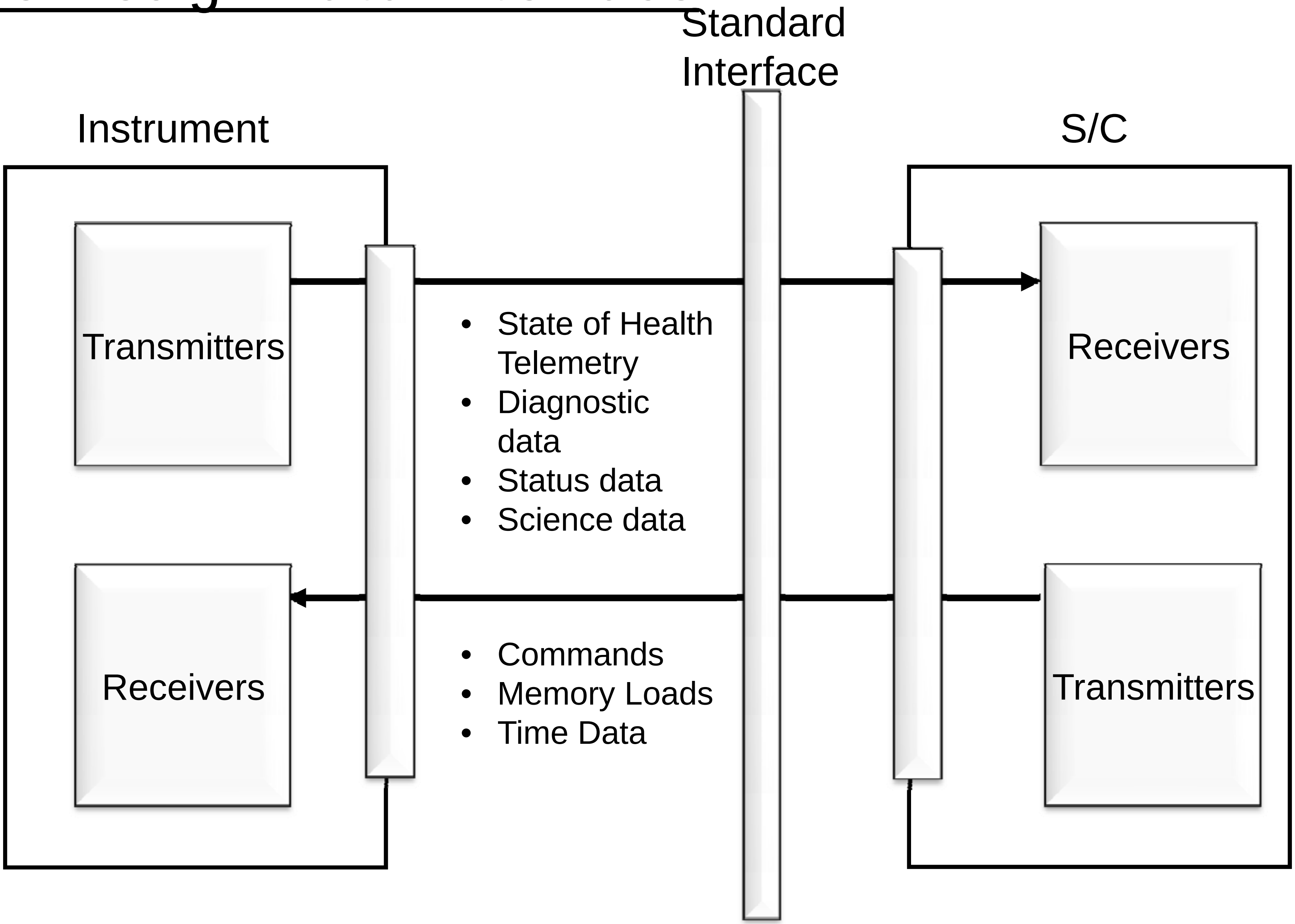


Interface Block Diagram



Notional Instrument-to-S/C Functional Interfaces

Pre-Design Data Interface



- State of Health Telemetry
- Diagnostic data
- Status data
- Science data

- Commands
- Memory Loads
- Time Data

Figure 5: Data Transfer Interface

Post Design Interface

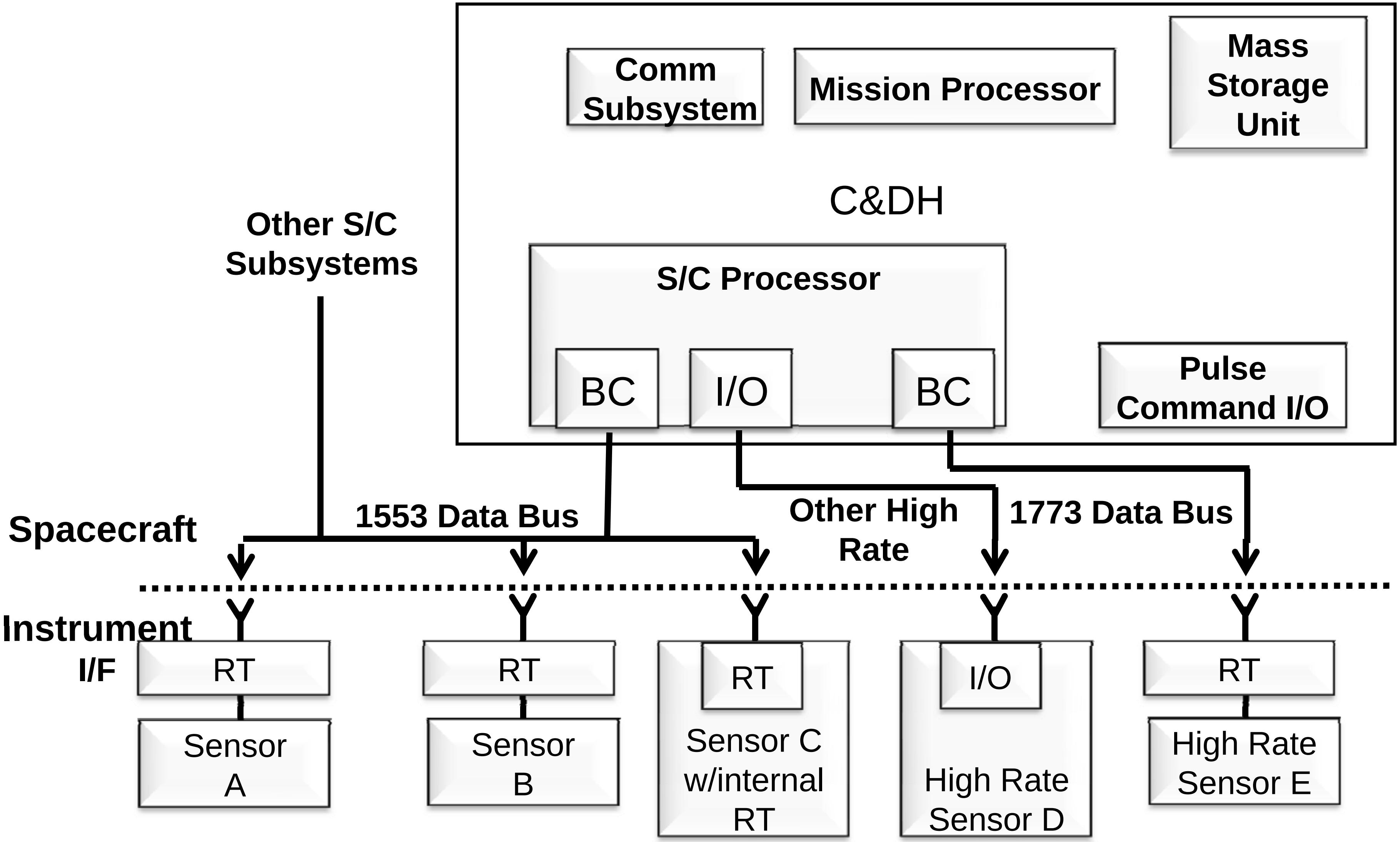
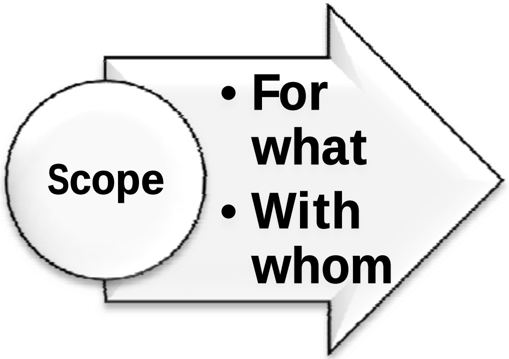


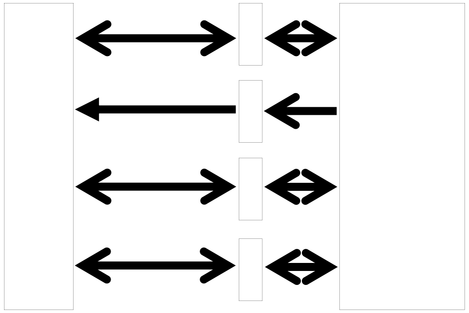
Figure 6: Notional Command and Data Handling Interface Topology

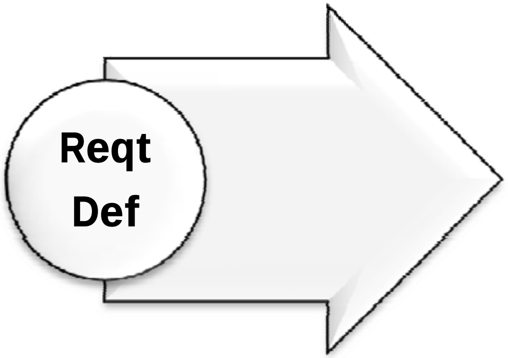
Data IRD contains all levels of data interface information



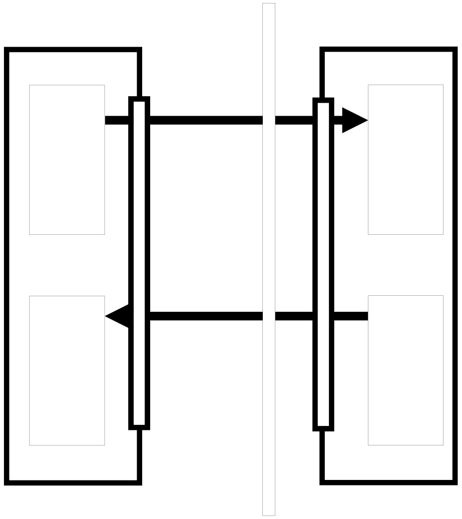
Scope

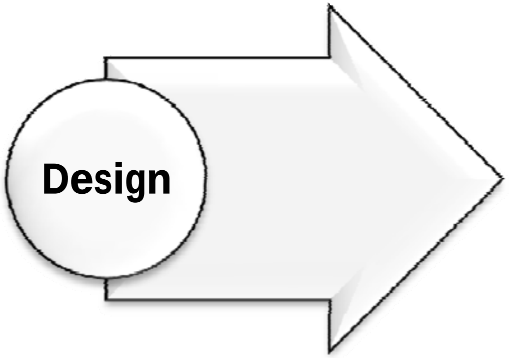
- For what
- With whom



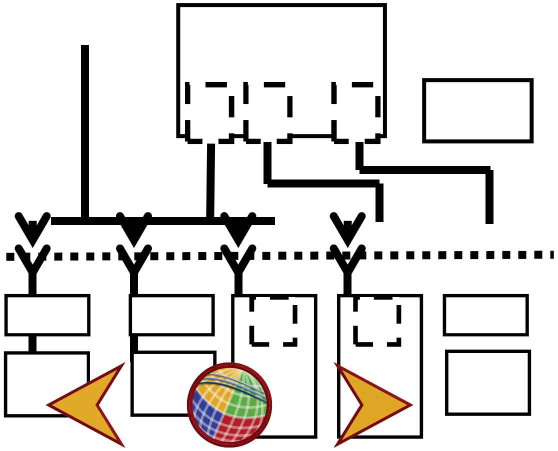


Reqt Def





Design



- 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?

What's Coming Next

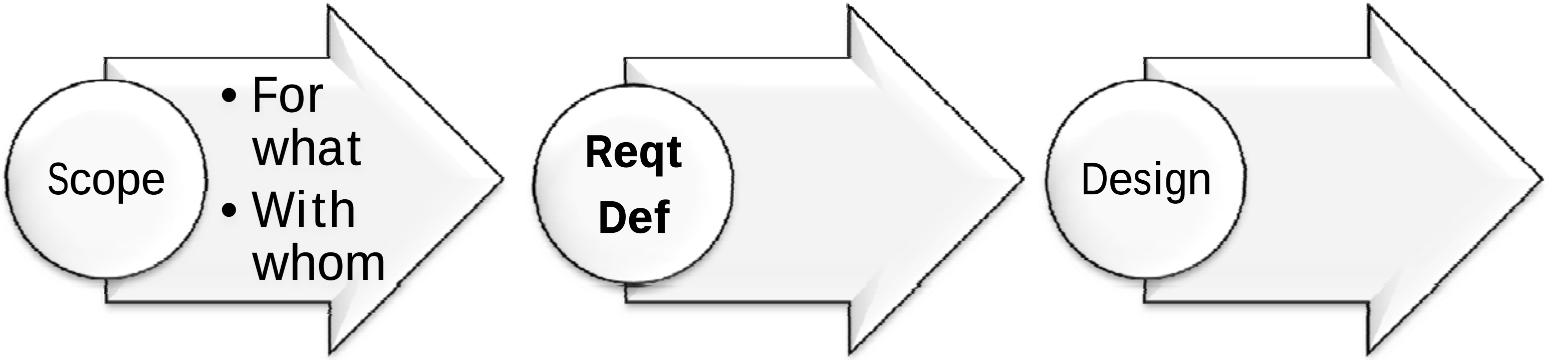
Specification

ICD

Standard

IRD

Writing Interface Requirements

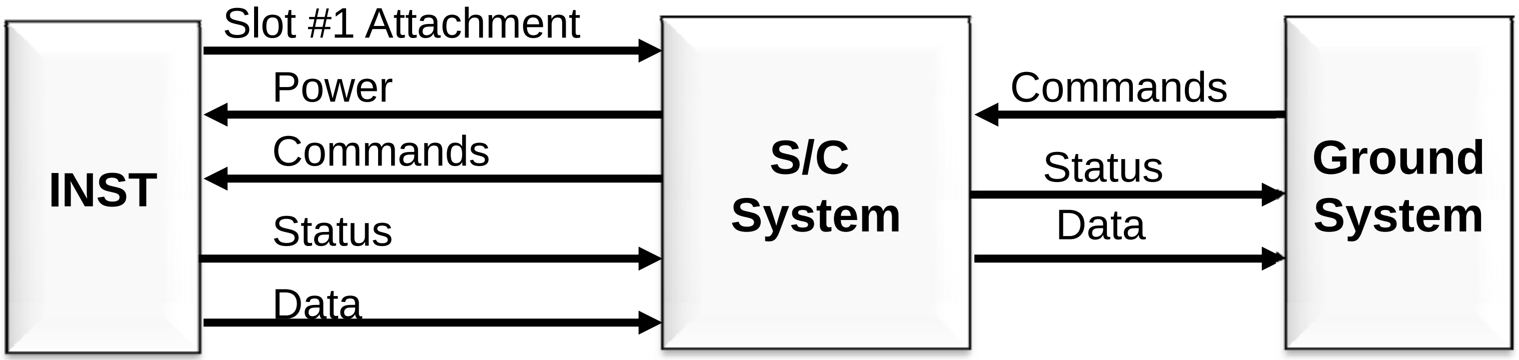


What to Document in your Specification

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
in 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 interfa

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

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

How to cite a reference document

Pick one

- 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.

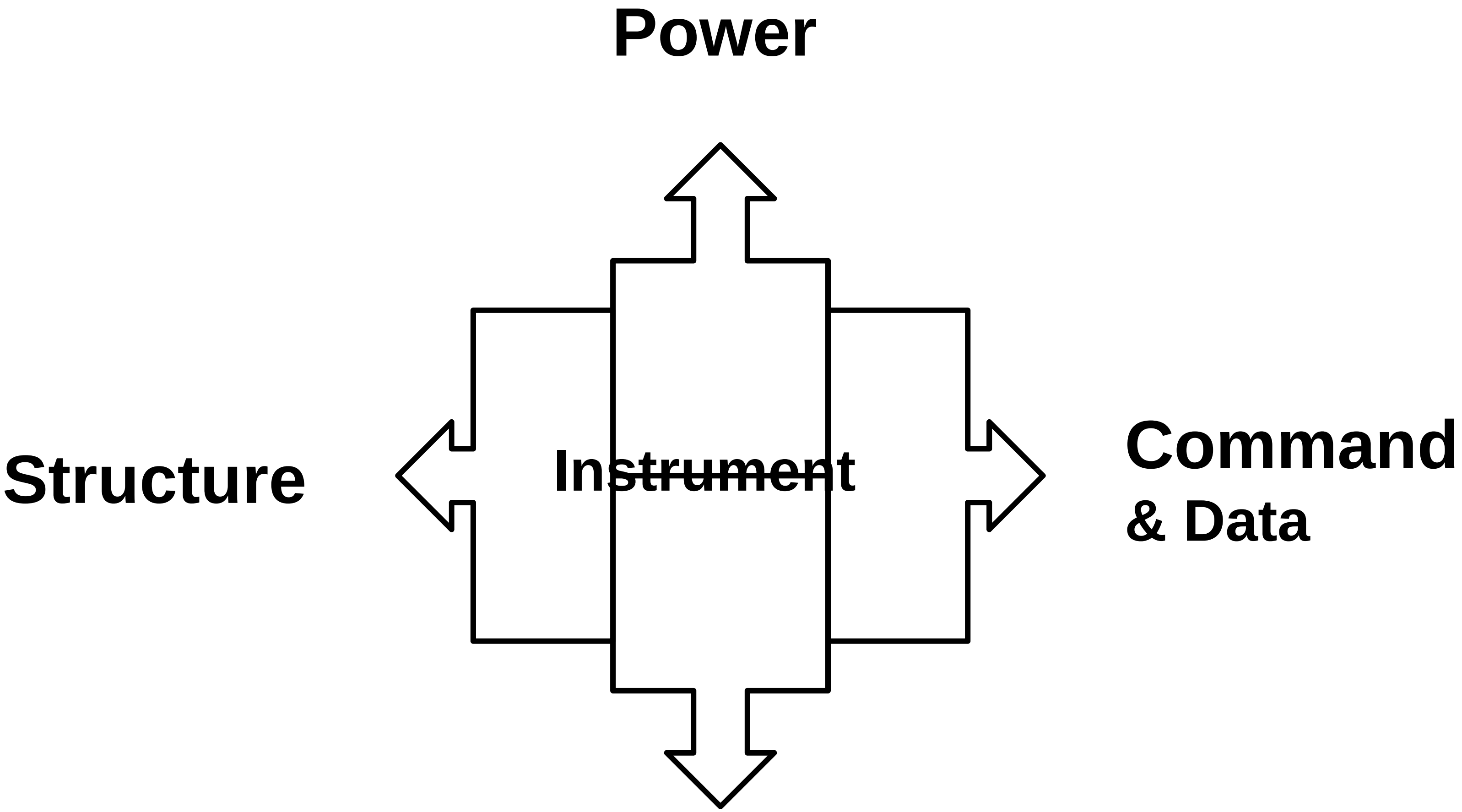
Reference change impact

- 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.

What's Coming Next

- Internal interfaces
- Managing interfaces
- Verifying interfaces

Instrument system external interfaces to the S/C



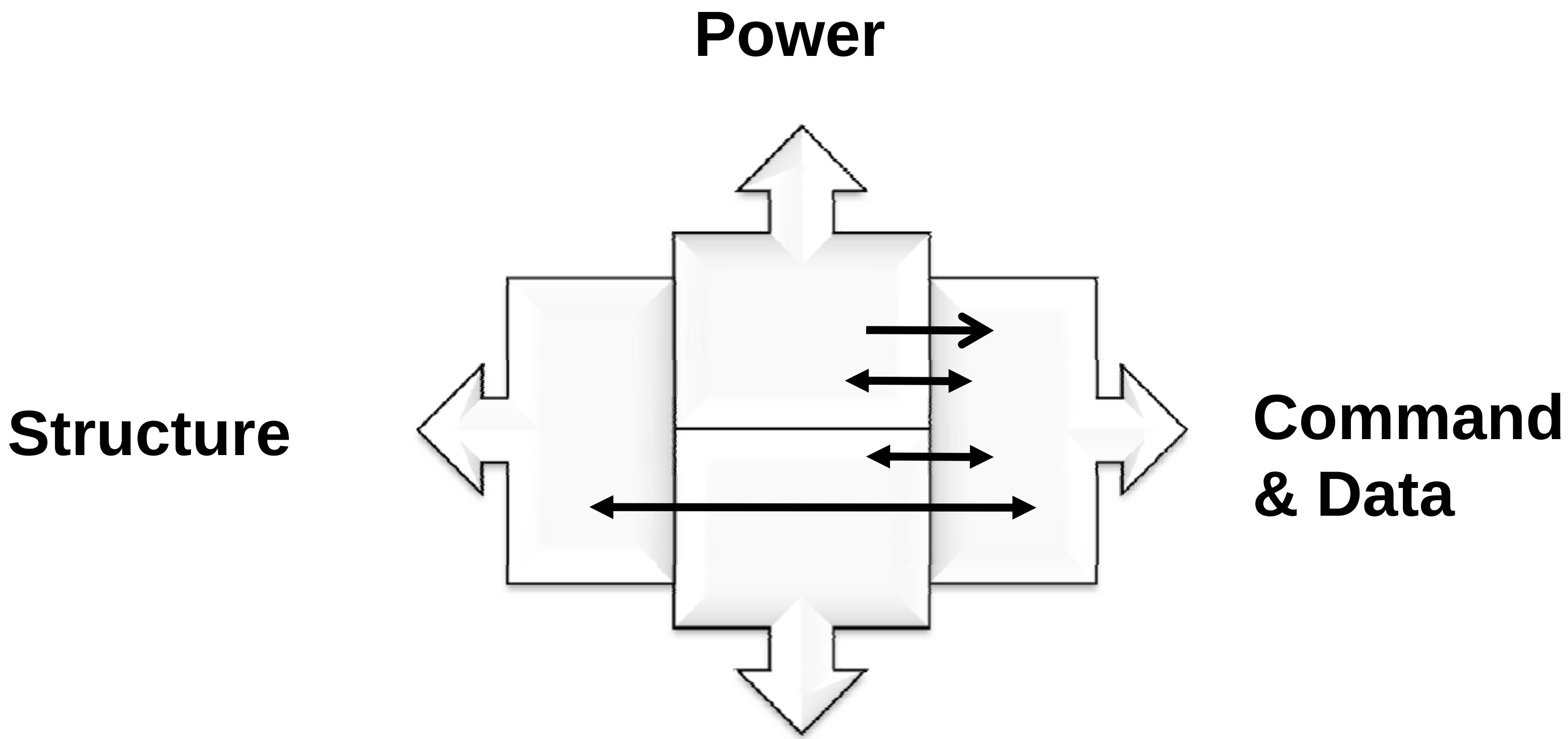
Power

Structure

Instrument

Command & Data

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

Things to 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?

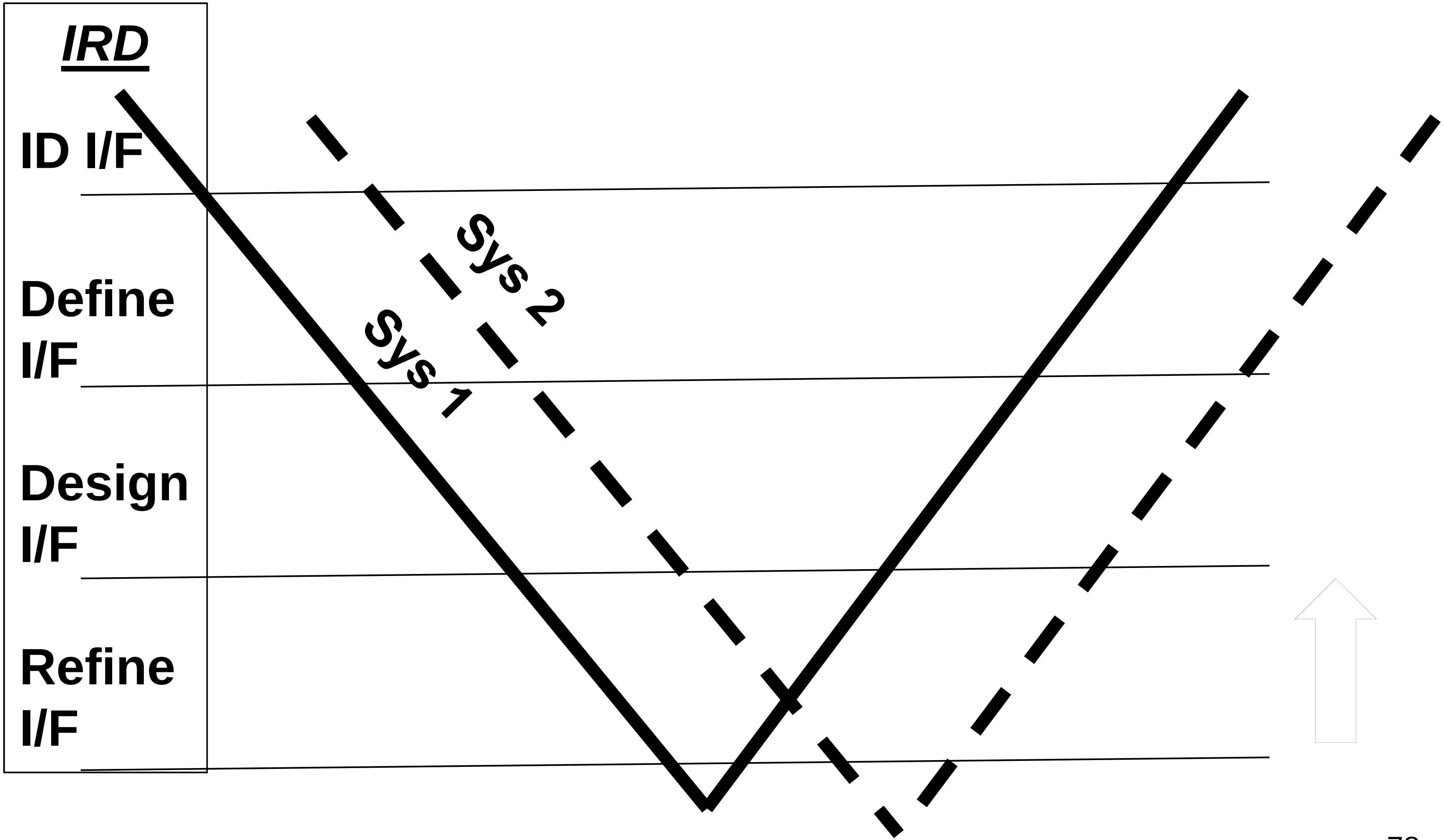
Managing Interfaces

- 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



IRD

ID I/F

Define I/F

Design I/F

Refine I/F

Verification

- 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

- 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

Where do you go from here?

Compliance Automation, Inc.
The Requirements Experts

The **Compliance Automation, Inc.** (CAI) team of recognized requirements experts takes pride in implementing quality requirement management processes for our clients. We have trained and supported government agencies, private companies and service organizations since 1990. As your company implements more complex systems, the need for good requirements will be essential to your success and CAI can help.

Products from Compliance Automation, Inc.

Requirements: Best Practices (Video/Audio Series)

We selected major topics from our training seminars to create this series as a refresher for your existing employees or as support for bringing new people up-to-speed. Choose the format that fits your needs: video (DVD), audio-only (CD) or web-based.

Customer-Centered Products

By Ivy F. Hooks and Kristin A. Farry

This highly practical book shows you how to elicit the right requirements at the start of a project, keep the development process on track, and make the right product the first time.

Guide for Managing and Writing Requirements

The Guide provides many of the best practices using examples from two small space programs conducted by the NASA Johnson Space Center. It has been extremely helpful to non-space related industries, such as oil companies and banks, and for hardware and software.

Online Resources

Visit complianceautomation.com for more resources collected and developed by our consultants:

- Whitepapers
- Templates
- Good Requirements Reference Card
- and more!

Consultation and Facilitation Services

Tap into the wealth of experience and knowledge of our senior consultants:

- Process and Requirement Audits
- Moderation of Requirements
- Requirements Facilitation

www.complianceautomation.com
Phone: (830) 249-0308 Fax: (830) 249-0309
training@complianceautomation.com
217 E. Bandera Road, Suite 200, Boerne, Texas 78006-2991