



# System Architecture Representation

**Dr. Dinesh Verma**  
**Associate Dean and Professor**  
**The SDOE Program**  
**Stevens Institute of Technology, USA**

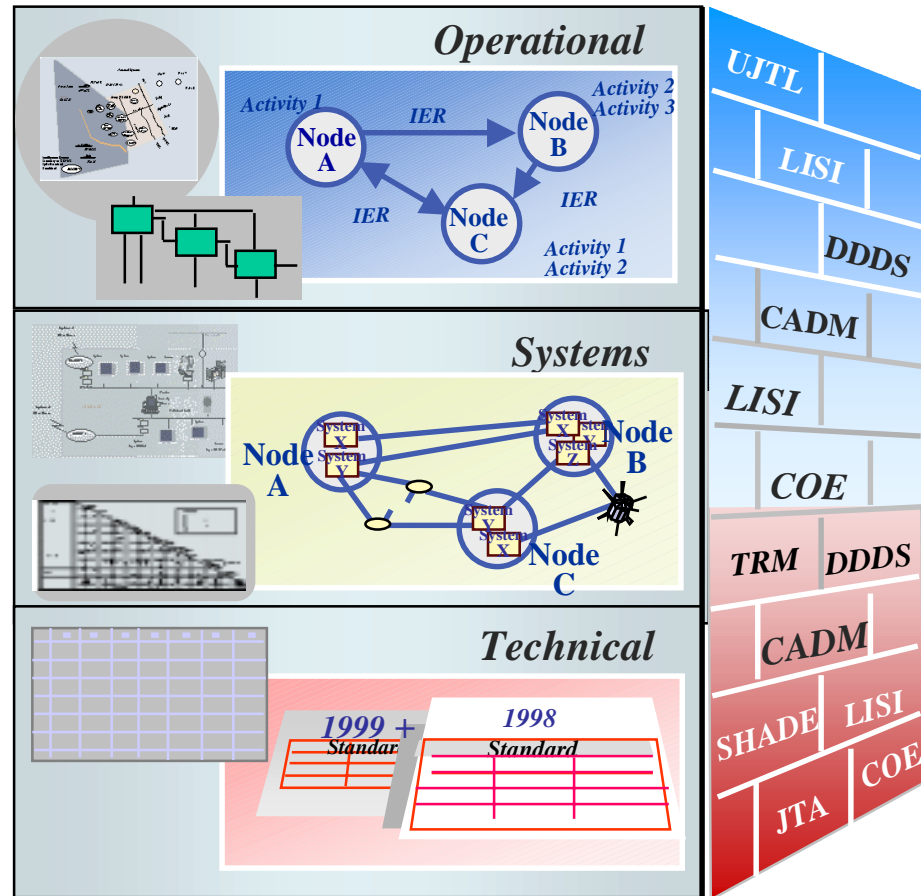
# Architecture Definition?

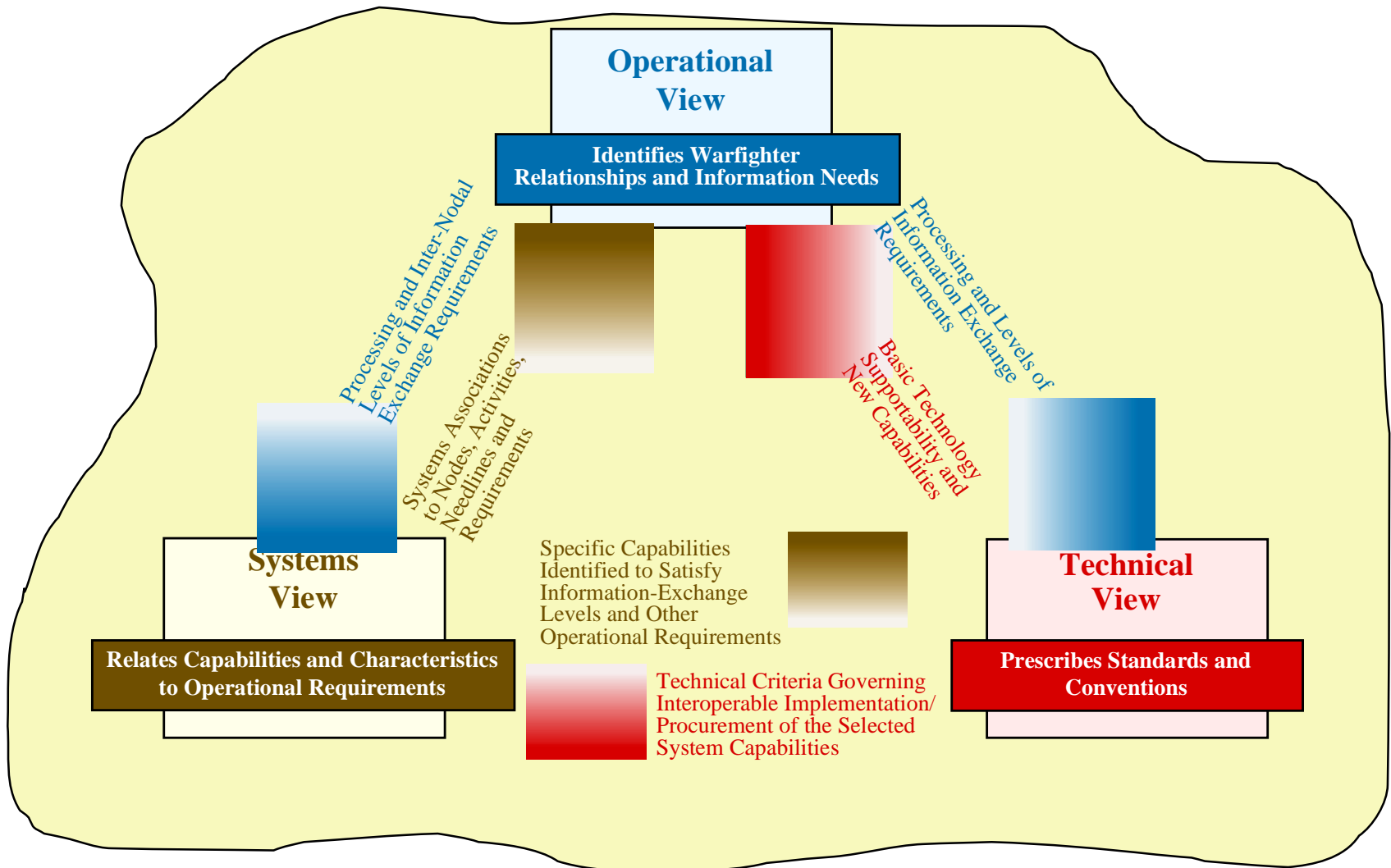
**“ The purpose of C4ISR architectures is to improve capabilities by enabling the quick synthesis of “go-to-war” requirements with sound investments leading to the rapid employment of improved operational capabilities, and enabling the efficient engineering of warrior systems. ”**

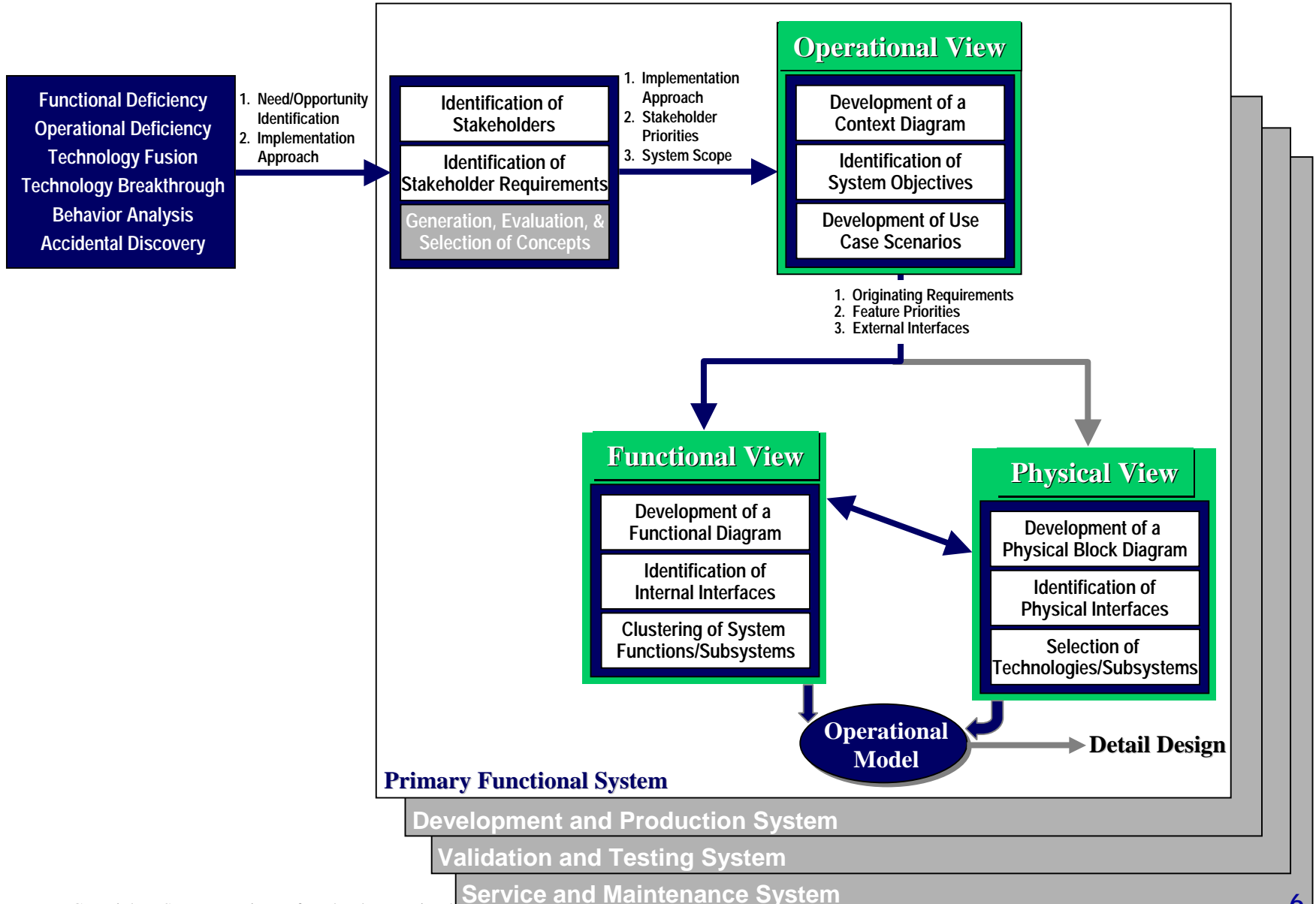
**- Adapted from: C4ISR Architecture Framework, Version 2.0**

# Components of the Framework

- ◆ **Common Definitions**
- ◆ **Common Products and Data**
- ◆ **Common References**







# Architecture Definition

## User's/Operator's View

- **Elements**

- People

- External Users
    - Internal Operators
    - Maintainers

- Human Machine Interface Devices

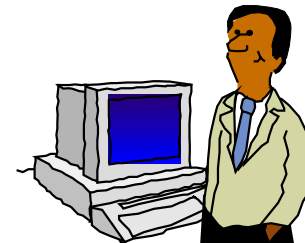
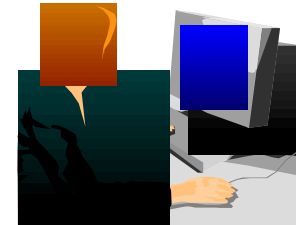
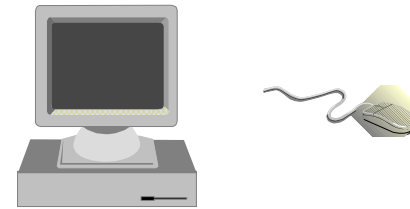
- Displays
    - Keyboards
    - Mice/Joysticks
    - Headsets
    - Printers

- Interconnectivity

- Vision
    - Audio
    - Touch/Feeling
    - User's/Operator's/Maintainer's Environment

- **System Impacts**

- Equipment Configuration and Location
  - Training Requirements - Level of Expertise
  - Cost of Operation
    - Number of People Required
    - Productivity



# Architecture Definition Application (Functional) View

- **Elements**

- Capabilities

- For Example:

- Interface Management
- Organic Tracker DB
- In-Organic Tracker DB
- Target Motion Analysis
- All Source Contact Management
- Launch Control

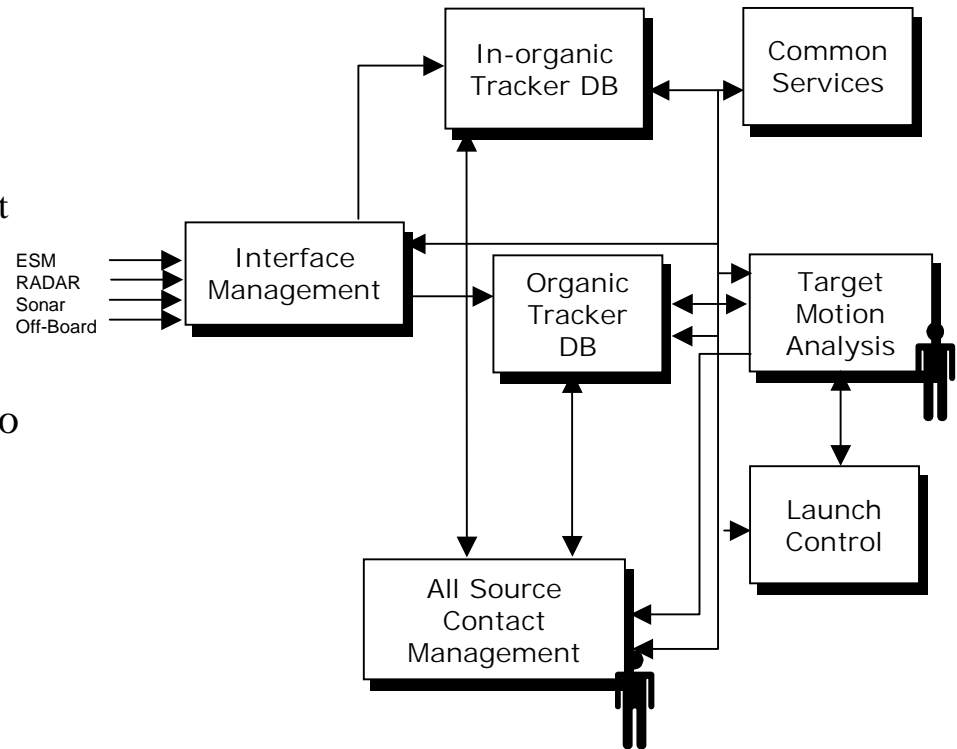
- Interconnectivity

- Messages Passed Between Capabilities to Perform an End to End Function

- **System Design Impacts**

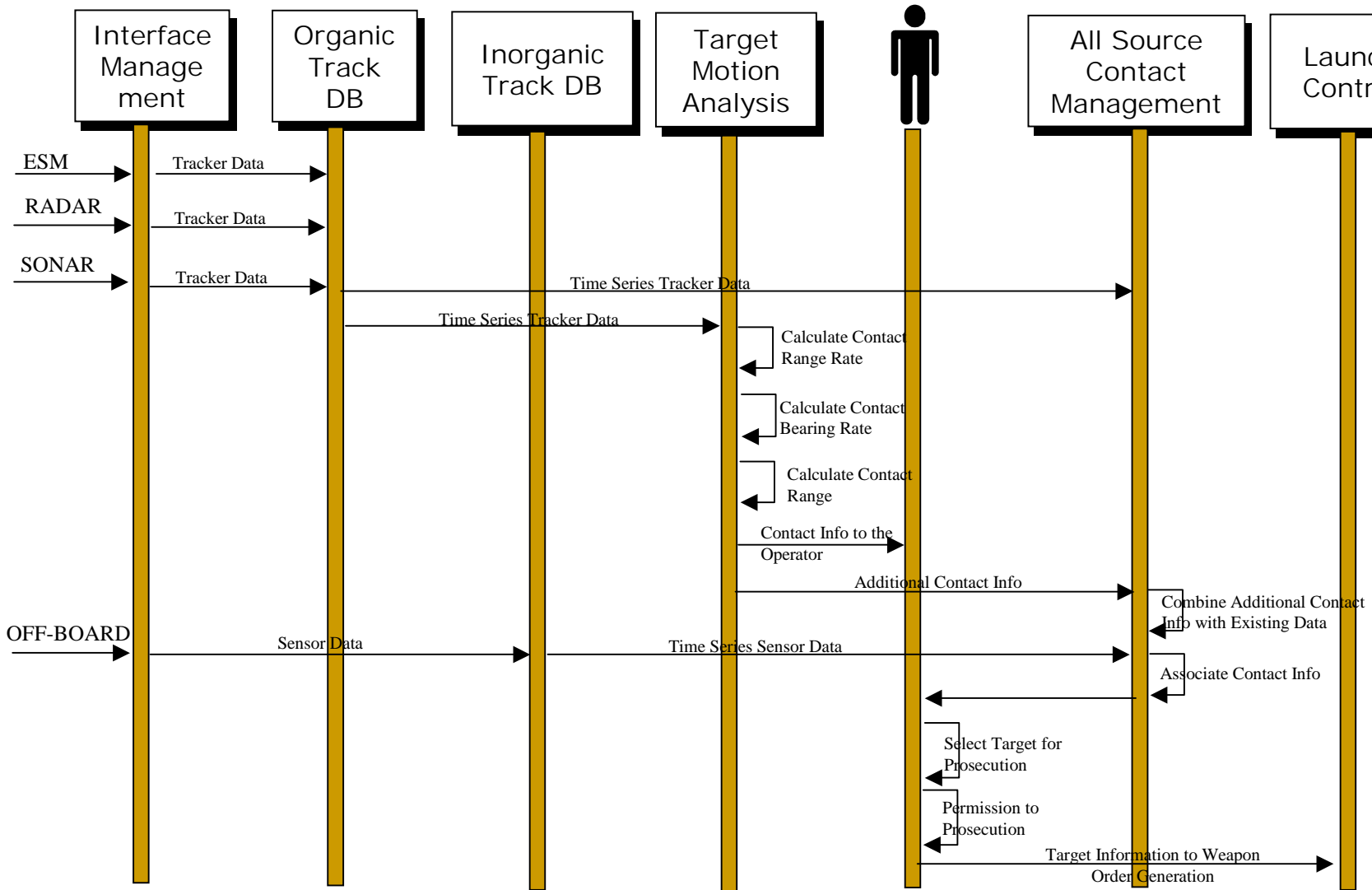
- Resource Requirements
  - Bandwidth
  - Processing Throughput
  - Storage
- System Complexity

## Static View

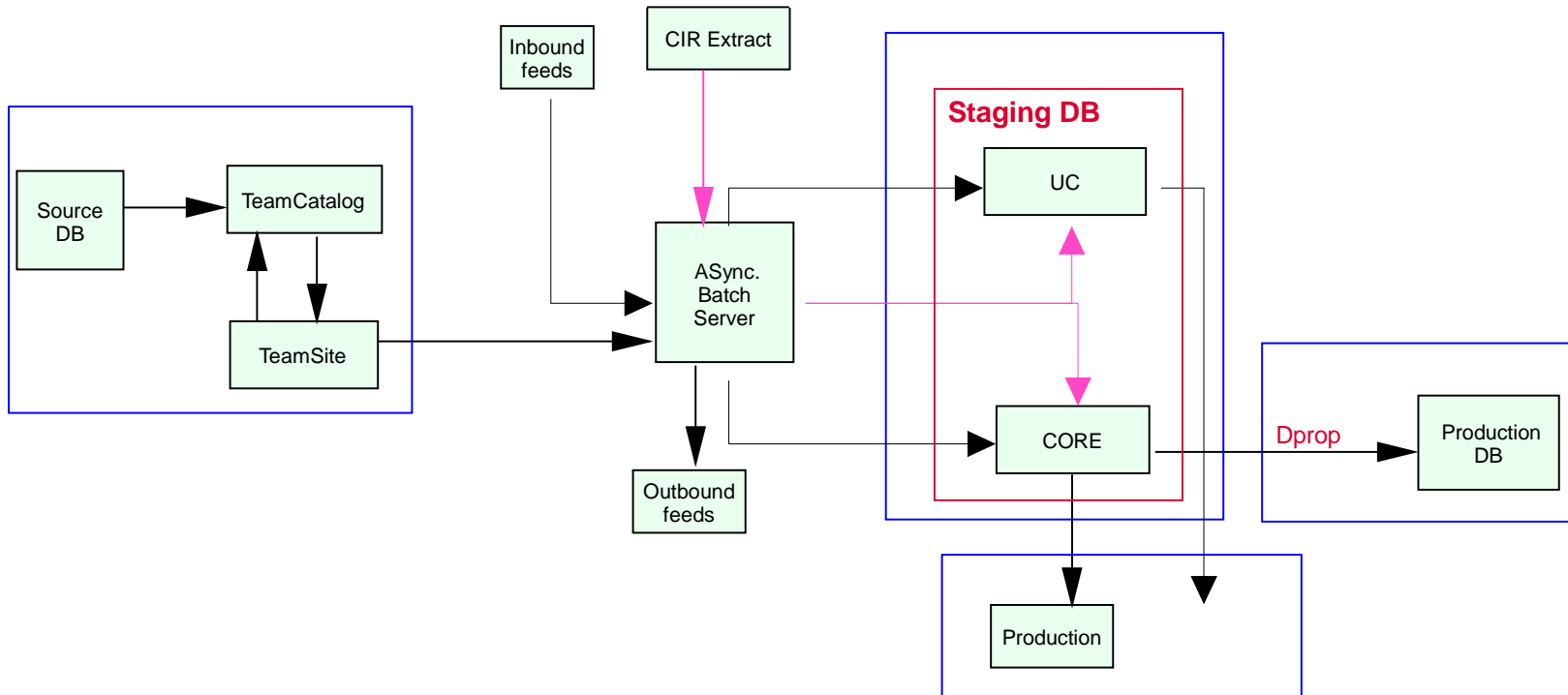


# Architecture Definition Application (Functional) View

## Dynamic

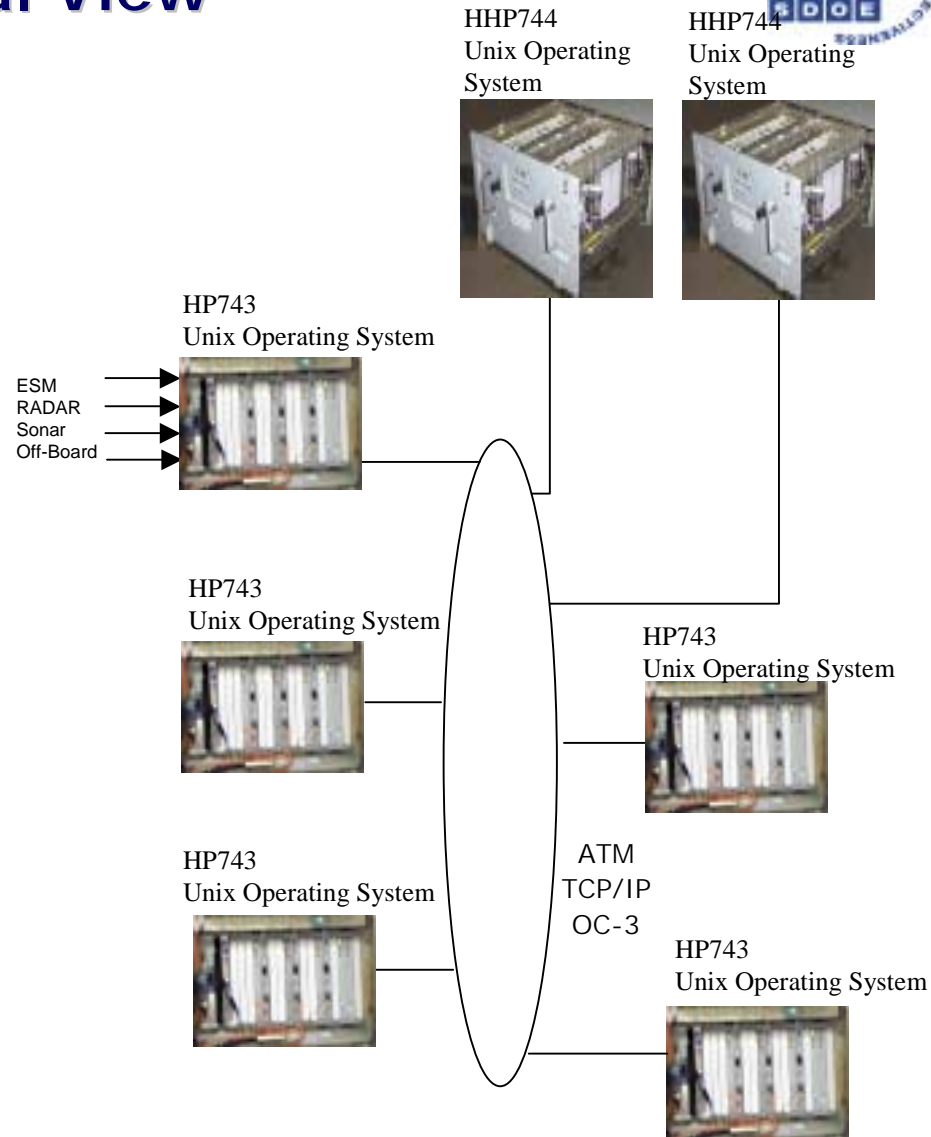


# Architecture Definition Data View



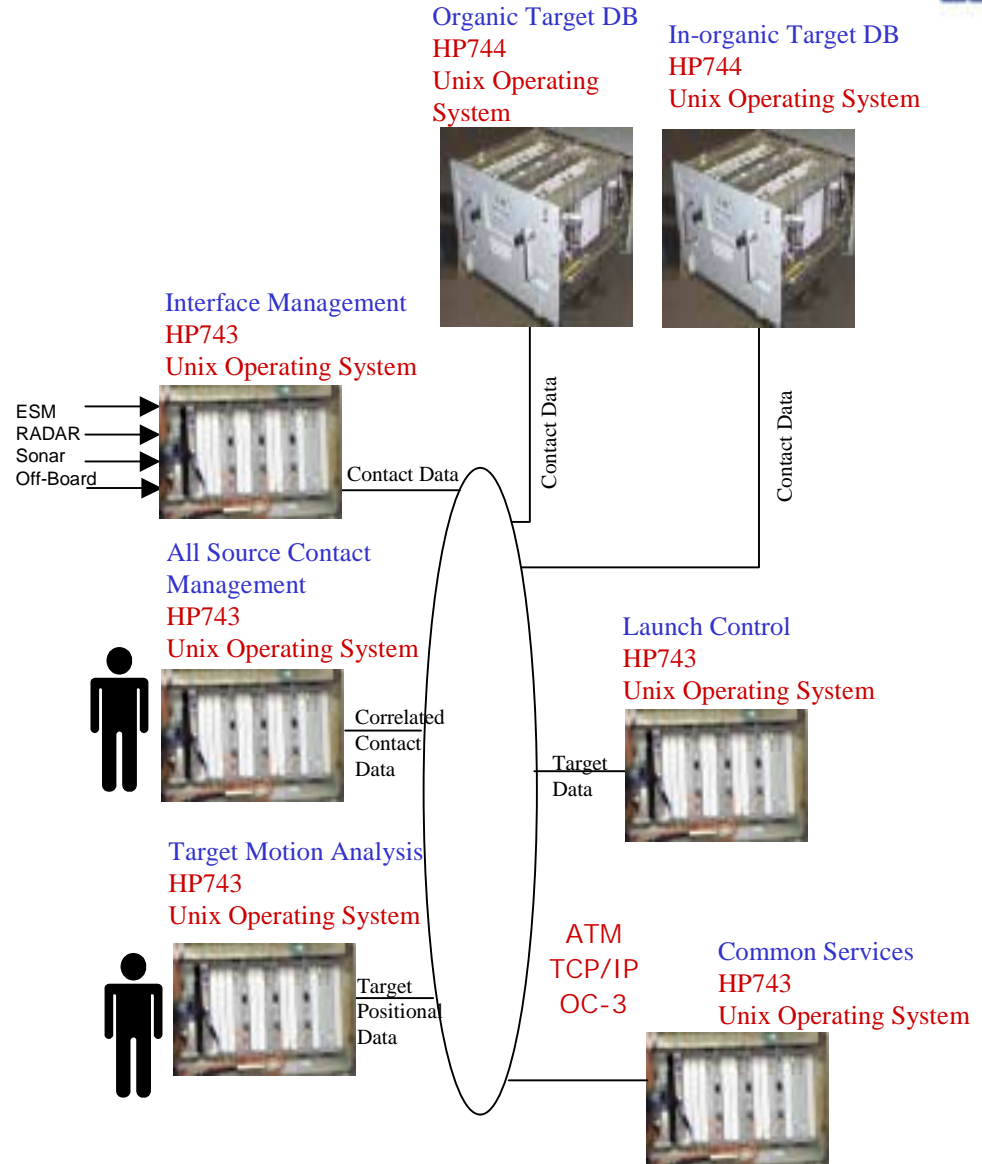
# Architecture Definition Physical View

- **Elements**
  - **Configuration Items**
    - **Subsystems**
    - **Units**
    - **HWCI's and CSCI's**
    - **Replaceable Assemblies**
  - **Interconnection**
    - **Physical Links**
      - **Networks**
      - **Point to Point**
- **System Design Impacts**
  - **Performance Limitations**
  - **Inability to Upgrade**



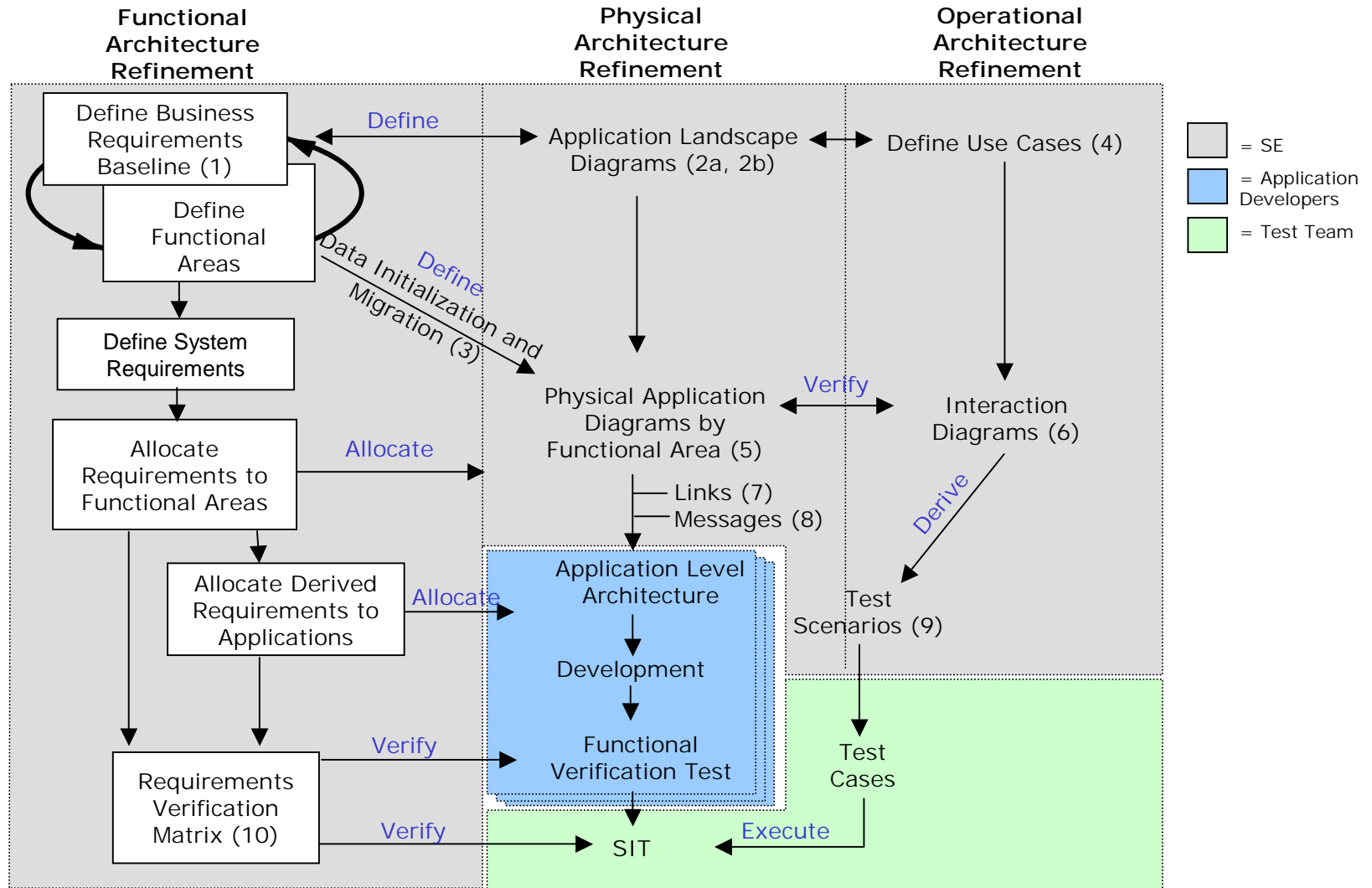
- **A Complete System Description Includes All Three Views of the Architecture**

- The People and Where They Interact with the System
- The **Capabilities** Within the System and the **Messages** that Flow Between Capabilities
- The **Hardware, Software, and Physical Links**



# Documenting a System Architecture

# Capturing an Architecture Baseline



- **Why Is Capturing the Baseline Important**
  - **Clear, Concise Problem Definition to the Development Team**
    - **Build What the Customer Wants**
  - **Test Planning**
    - **Definition of the Test Environment**
    - **Data Migration and Initialization**
  - **Reduce Learning Curve for Individuals Coming onto the Project**
    - **Future Upgrades**

# (1) Technical Scope Document



		Application S/W Impact			Data		Link			Schedule Information		
Reqm't	Affected Application	S	M	L	N	M	N	M	Requirement Description	Design/ Code/ Test Dates	SWIT/ Unit Test Dates	Delivery to System Test
1	Target motion Analysis		X			X			Localization to within 10% of actual in both bearing and range for contacts held by organic sensors	7/1/02 - 10/1/02	10/1/02 - 12/1/02	12/5/2002
2	All Source Contact Management	X							Multi-sensor correlation of contacts held on organic sensors	8/1/02 - 10/1/02	10/1/02 - 11/15/02	11/15/2002
3	All Source Contact Management			X	X		X		Contact association across 5000 contacts from in-organic sensors within 15 minutes of receipt of data	7/1/02 - 12/1/02	12/1/02 - 2/1/03	2/3/2003
4	All Source Contact Management		X			X			Contact association across all sensors	8/15/02 - 12/1/02	12/1/02 - 2/1/03	2/3/2003
5	All		X				X		Full operational capability with 4 or less operators	9/1/02 - 11/1/02	11/1/02 - 12/1/02	12/5/2002
5.1	Common Services	X							All Operators shall receive the same training	10/1/02 - 11/15/02	11/15/02 - 12/1/02	12/5/2002
6	All		X			X		X	0.995 Operational Availability for a 90 day mission	9/1/02 - 12/1/02	12/1/02 - 2/1/03	2/3/2003
7	Launch Control		X						Horizontal launch of ADCAP and Tomahawk weapons	9/1/02 - 12/1/02	12/1/02 - 2/1/03	2/3/2003
8	Launch Control			X	X		X		Vertical launch of Tomahawk weapons	7/1/02 - 11/15/02	11/15/02 - 2/1/03	2/3/2003

# (2b) Application Landscape



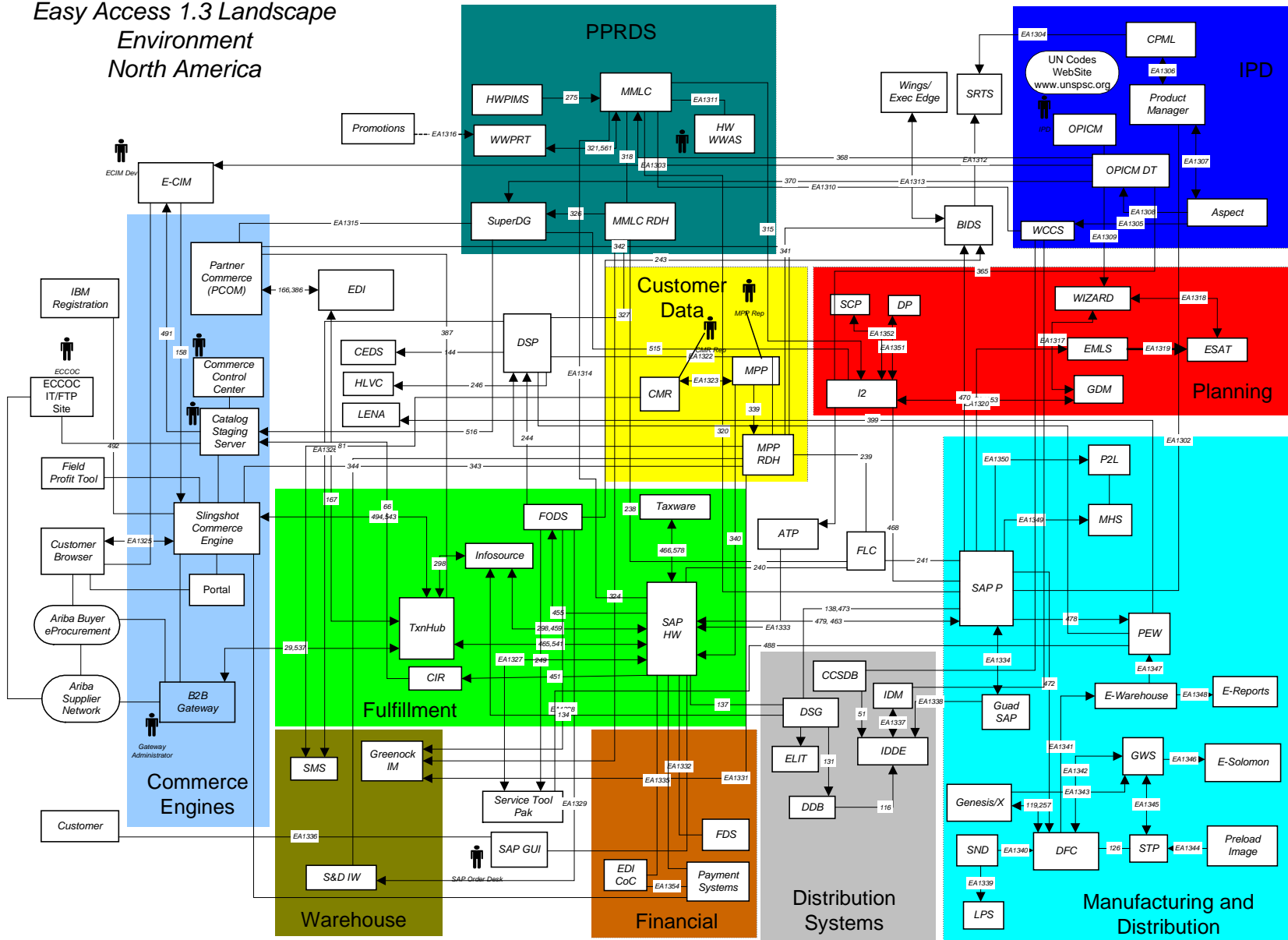
<b>Role or Application Acronym/Name</b>	<b>Capabilities</b>	<b>Version</b>	<b>Technical Infrastructure</b>
<b>Common Services (CS)</b>	<b>Display formatting Auto Plots Performance Monitoring Diagnostics</b>	<b>1.0a</b>	<b>Unix Operating System on HP743 Single Board Computer</b>
<b>Target Motion Analysis (TMA)</b>	<b>MATE, KAST, and Polar Coordinate Ranging Algorithms</b>	<b>1.0</b>	<b>Unix Operating System on HP 743 Single Board Computer</b>
<b>Organic Sensor Database</b>	<b>Storage and Retrieval of contact time series information (contact id, bearing, base frequency, D/E angle)</b>	<b>4.11a</b>	<b>Unix Operating System on HP777 Server</b>



# Enterprise System

## xxx Easy Access

Easy Access 1.3 Landscape  
Environment  
North America

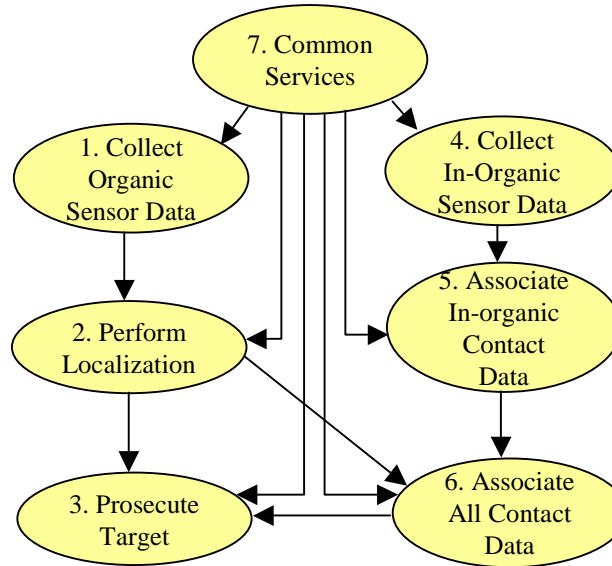


# (3) Data Initialization and Migration



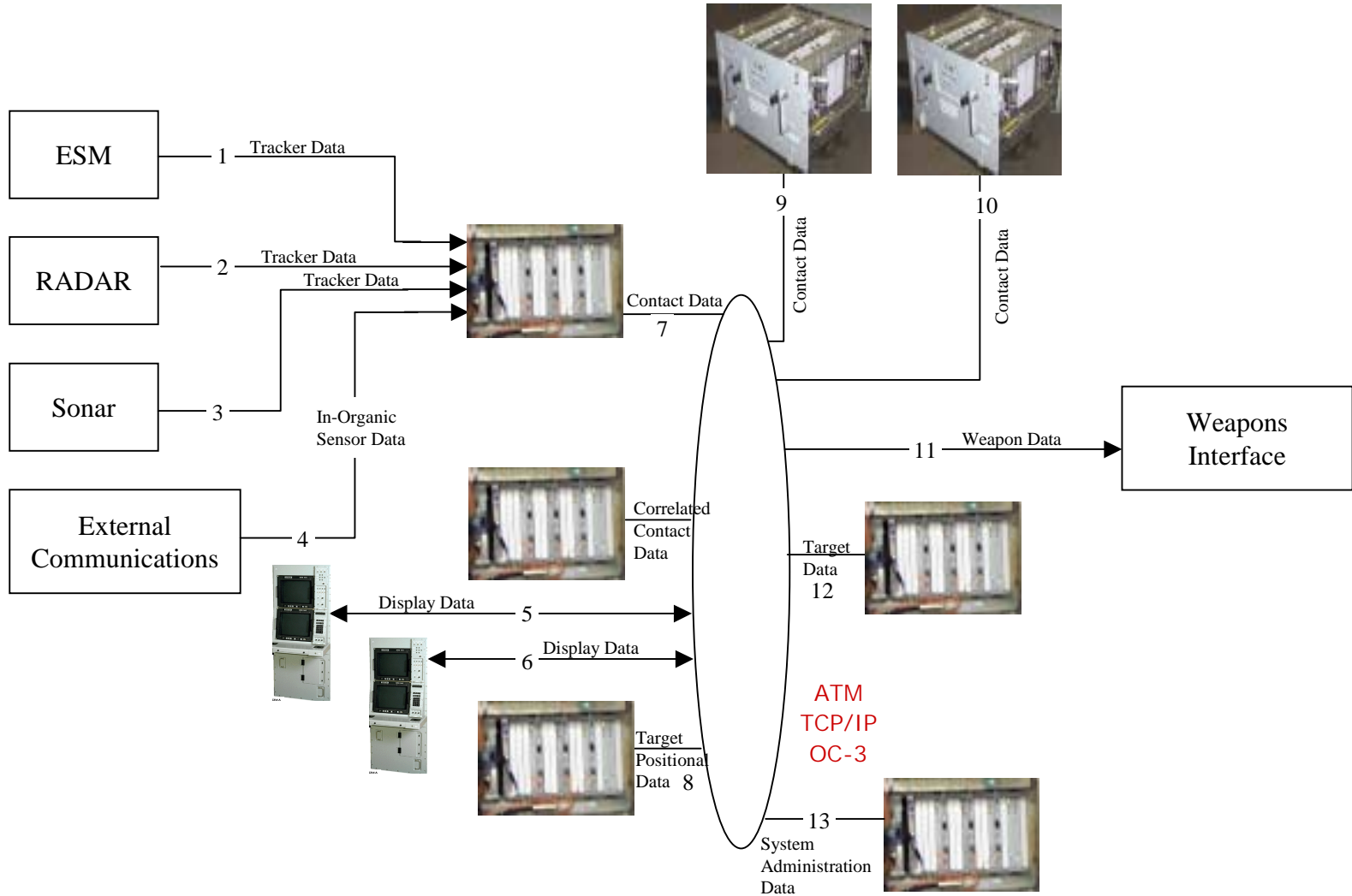
Requirement	Data Type	Data Source	Attribute Needed	Test Requires Attribute Toggle/Change?
1	Contact Position	TMA	Range	Yes
2	Contact Position	Off-Board Sensors via Interface Management	X,Y,Z position and time of	Yes

# (4) Use Cases



System Use Case ID	System Use Case Name	System Capability	System Test Scenario	Invoked by:	Predecessor(s)	Average Rate	Peak Rate	Hyper link
1	Collect Organic Sensor Data	Store Time Series Data from Organic Sensor Trackers	Test Organic DB Store/Schema	Dataflow	None	12 Hz per contact	12 Hz per contact	
2	Perform Localization	Calculate contact bearing rate, range, and range rate	Test Localization	Dataflow	Collect Organic Sensor Data			

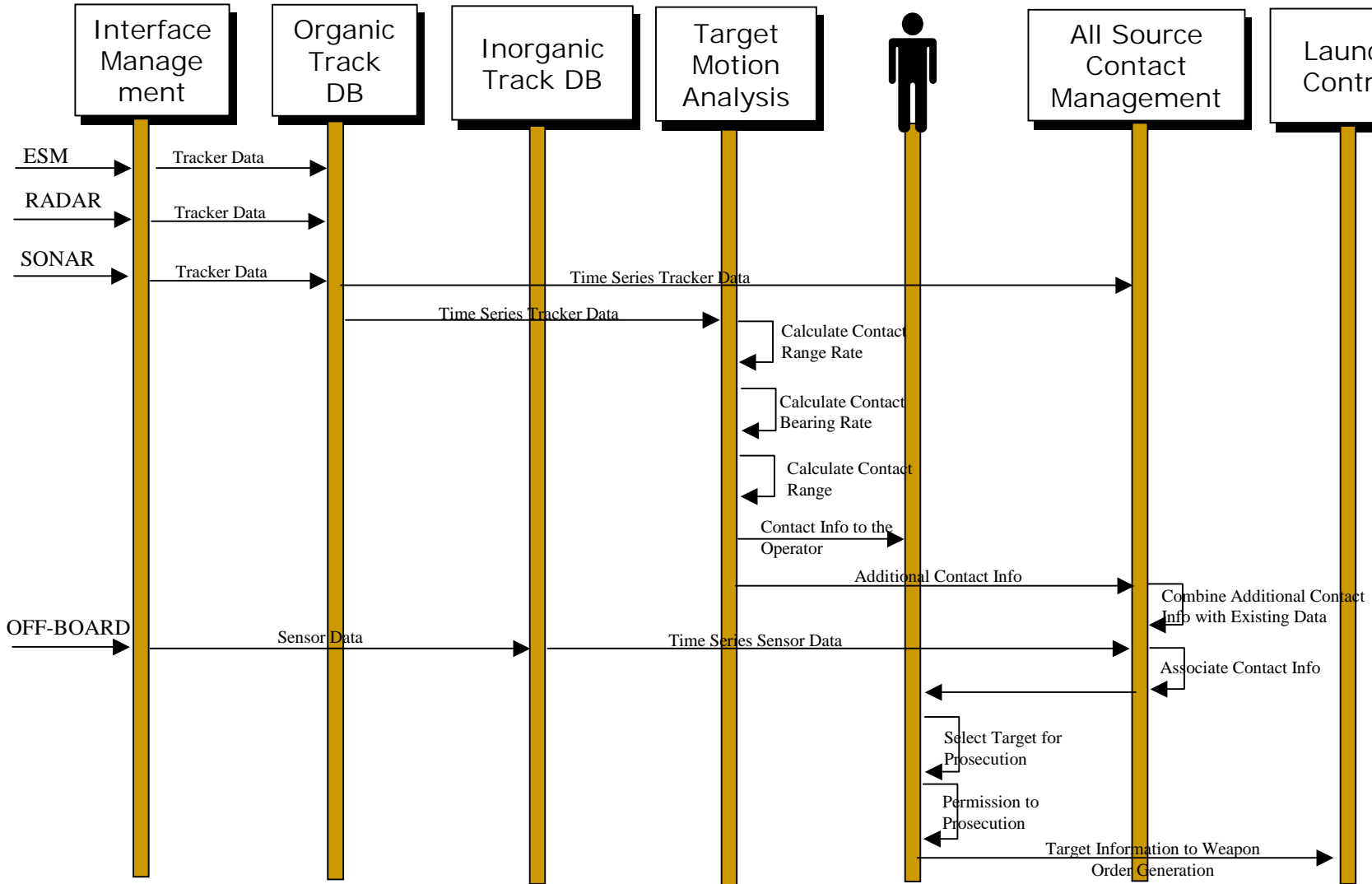
# (5) Functional Area Physical Diagrams



# Architecture Definition

## Logical (Functional) View

### Dynamic



# (7) Links



I/F No.	Source	Destination	N M E	Protocol	Physical	Capacity	Average Utilization	Peak Utilization	Owner
1	ESM	Interface Management	N	Request /Acknowledge	NTDS Fast	667 Kbytes/Sec	30%	60%	
2	RADAR	Interface Management	N	Request /Acknowledge	NTDS Fast	667 Kbytes/Sec	20%	25%	
3	Sonar	Interface Management	N	Request /Acknowledge	NTDS Fast	667 Kbytes/Sec	40%	75%	
4									
5	TMA	User	M	TCP/IP	ATM	155 Mbytes/Sec	10%	20%	

# (8) Messages

Message Name	Message Description	Link	N M E	System Use Case ID	Size (Bytes)
ESM_Track_Data	ESM Contact Tracker Data (contact id, bearing)	1	N	1	128 Bytes
RADAR_Track_Data	RADAR Contact Tracker Data (contact id, bearing)	2	M	1	64 Bytes
Sonar_Track_Data	Sonar Contact Tracker Data (contact id, bearing, base frequency)	3	M	1	256 Bytes

# (9) Test Scenarios

System Use Case Name	Test Scenario Name	Test Scenario Description	Test Case Names	Driving Application
Collect Organic Sensor Data	Test Organic DB Store/ Schema	Using simulated or real data from ESM, RADAR, and Sonar provide inputs to Interface Management to be forwarded to the Organic Sensor Database for storage and retrieval.	Test ESM Inputs Test RADAR Inputs Test Sonar Inputs	
Perform Localization	Test Localization	Using tracker data information contained in the Organic Sensor Database, validate the requirements for contact range and bearing localization	Test MATE Test KAST Test Polar Coordinates	

# (10) Requirements Verification Matrix



Requirement	Capability	SWIT	Unit Test	Factory Acceptance Test	Installation	Sea Trials
1	TMA	T		D		D
2	ASCM	T		D		D
3	ASCM			D		T
4						
5						
5.1						
6						
7						
8						

- **10 Elements are Required to Document an Architecture**
  - **1. Requirements Baseline – Technical Scope**
  - **2a. Physical View**
  - **2b. Functional Description**
  - **3. Data Initialization and Migration**
  - **4. Use Cases**
  - **5. Allocation of Requirements to Physical/Functional View**
  - **6. Interaction Diagrams (Data Flow)**
  - **7. Links**
  - **8. Messages**
  - **9. Test Scenarios**
  - **10. Requirements Verification Matrix (RVM)**

## • **Problem Statement:**

- **Current market trends in the commercial defence market**
  - **Reduced development cost**
  - **Increased use of sub-suppliers**
  - **Increased use of COTS components**
  - **Higher turnover rate in development department**
- **...suggest that traditional architectural and system design documentation during product development may no longer be sufficient for current and future long-term (and low volume) product sustainment.**
- **The objective is to develop a system capable of being produced in small volumes the coming 30 years, and capable of being maintained even longer,**
- **In light of constantly evolving functional capabilities, changing physical implementation and changing workforce**
- **In a cost effective manner**

- **The systems model – shall be a product of the entire development effort, and shall contain what was kept as well as what was discarded**
- **The systems model will act as the product’s collective memory**

**If the systems model is unknown (undocumented), it will be impossible to cope with future changes with reasonable resources given our production volume**

- **The following structures realize the system model:**
- **Requirements hierarchy**
  - **All contract requirements – with requirement breakdown are internally ordered in a traceable structure**
  - **Analyses performed as part of requirement breakdown are attached to all influenced requirements**
  - **Test- and verification information are attached directly to the requirements**
- **System hierarchy**
  - **All systems are ordered in a stringent hierarchy (a stringent hierarchy is defined as a hierarchy in which each object can have only one parent (as opposed to the requirement hierarchy, where one requirement can have multiple parents))**
  - **Textual information used in specifications and similar, are attached directly to the system**
- **Functional hierarchies**
  - **All functions are ordered in a set of stringent hierarchies**
  - **Each function shall be allocated to one system (applies to the leaf-node functions)**
  - **All functional requirements are allocated to the function**
  - **Data- and control flow between functions are given by their internal relations**

- **System physical hierarchies**
  - Models the physical architecture for each subsystem
  - System functions are allocated directly to the parts
  - Non-functional requirements are allocated to the part
- **Assembly hierarchy**
  - One stringent hierarchy containing all parts organized in assemblies
  - Physical requirements following from the "packaging process" are allocated to on applicable levels
  - Failure modes are attached to the component
  - Testpoints are identified in the structure
  - Signal tracing are given by the relations among the parts / assemblies, and their functional counterpart are given by the relations to the functional hierarchies

**If you are interested in this research, and would be willing to share your thoughts in a telephonic interview/ discussion, please give us your contact information:**

**Name:**

**Phone:**

**Email:**