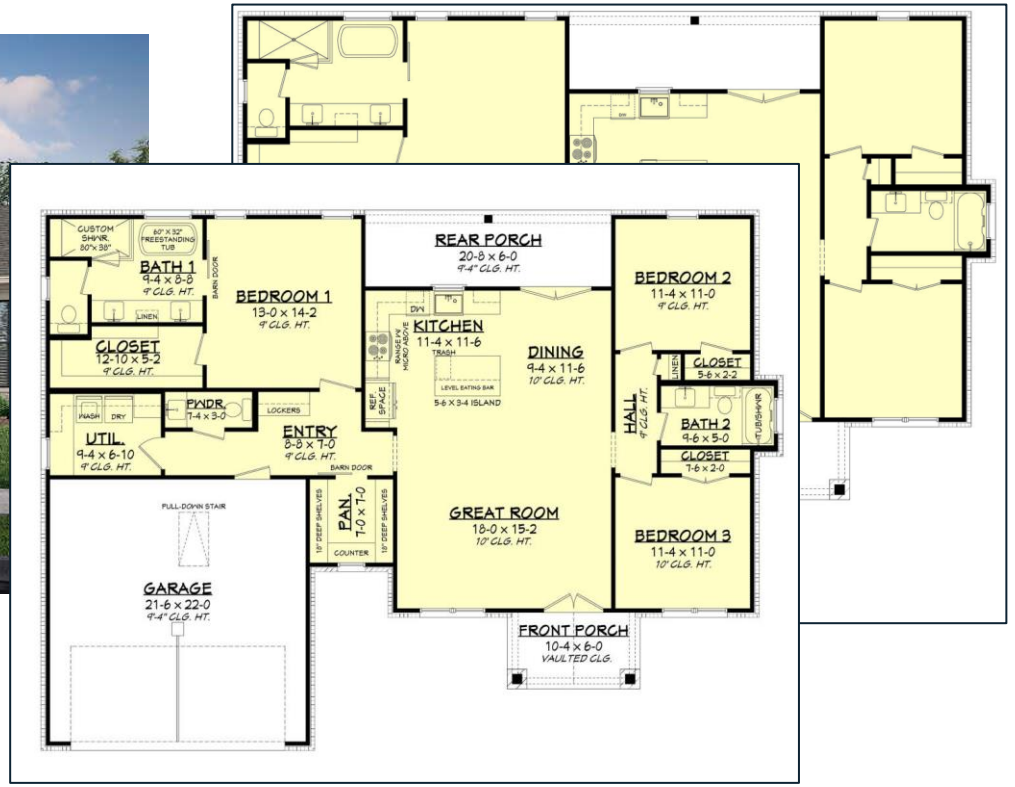
33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Architecture: More Than a Floor Plan

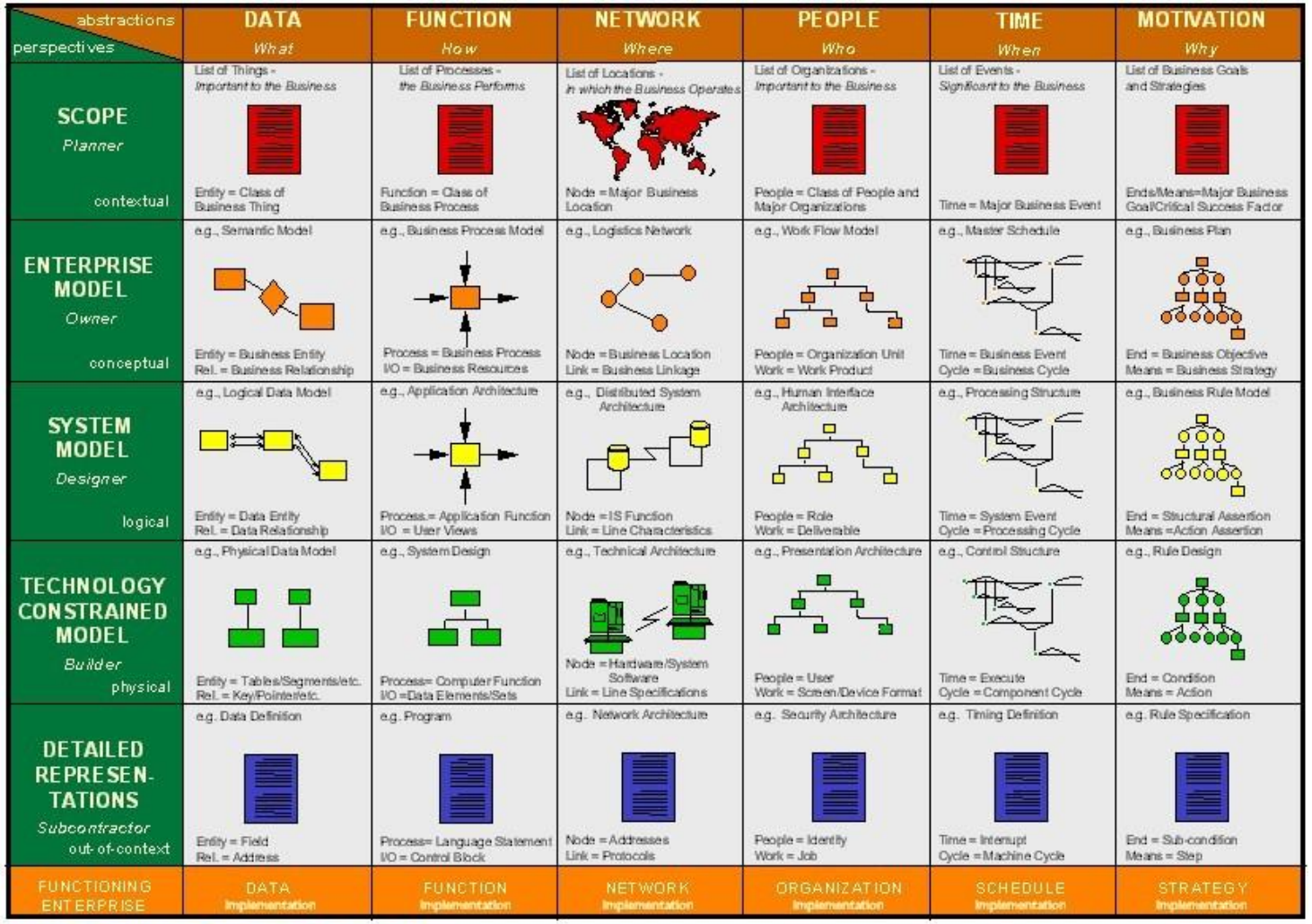


Jim Armstrong

Architecture Topics

- Architecture Frameworks
- Physical
- Functional
- Software
- Interfaces
- SOS
- Analysis
- Relationship to other SE areas
- Places to go for further information

Architecture Frameworks

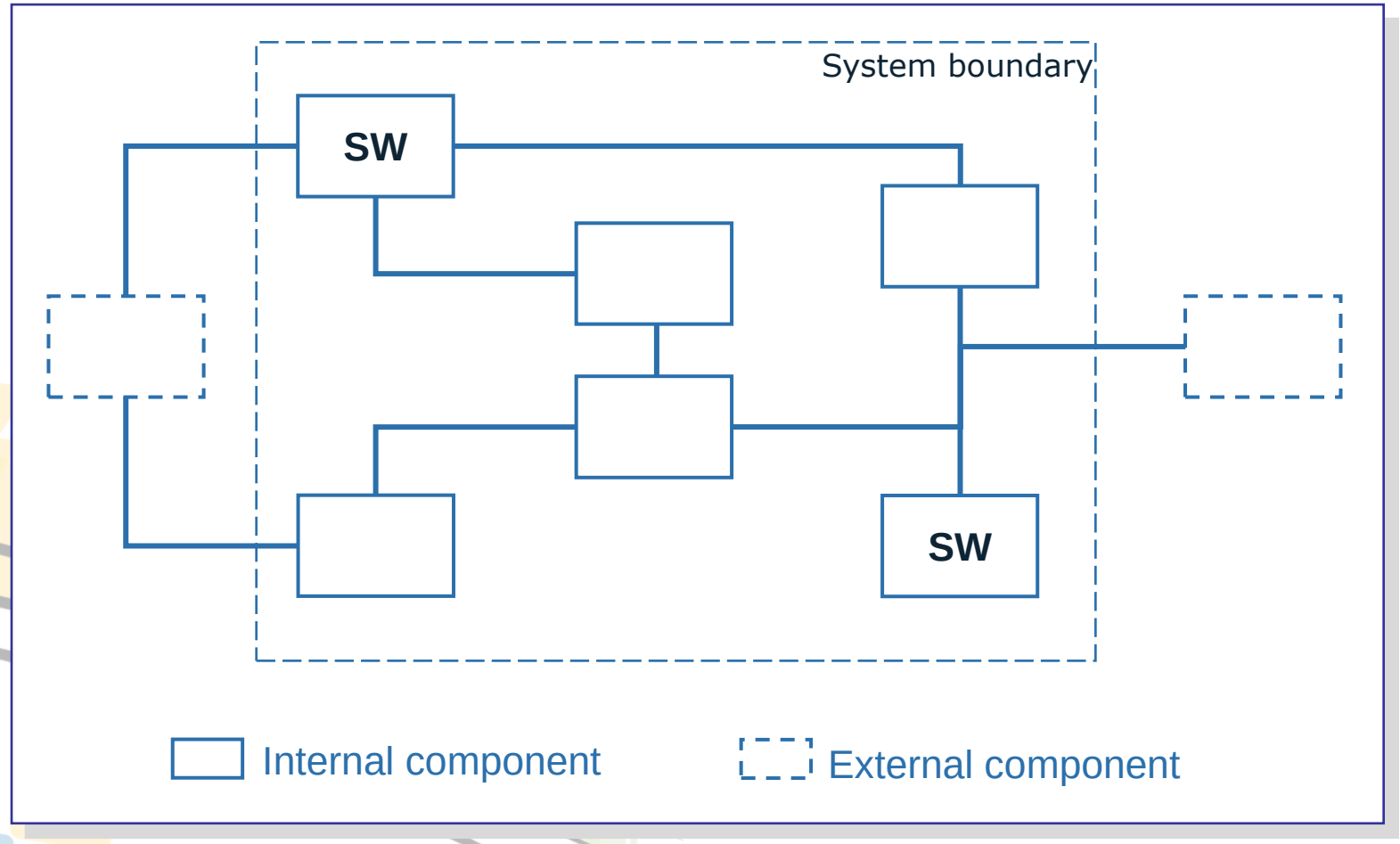


abstractions / perspectives	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*
SCOPE *Planner* contextual	List of Things - *Important to the Business*	List of Processes - *the Business Performs*	List of Locations - *in which the Business Operates*	List of Organizations - *Important to the Business*	List of Events - *Significant to the Business*	List of Business Goals and Strategies
	Entity = Class of Business Thing	Function = Class of Business Process	Node = Major Business Location	People = Class of People and Major Organizations	Time = Major Business Event	Ends/Means=Major Business Goal/Critical Success Factor
ENTERPRISE MODEL *Owner* conceptual	e.g., Semantic Model	e.g., Business Process Model	e.g., Logistics Network	e.g., Work Flow Model	e.g., Master Schedule	e.g., Business Plan
	Entity = Business Entity Rel. = Business Relationship	Process = Business Process I/O = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy
SYSTEM MODEL *Designer* logical	e.g., Logical Data Model	e.g., Application Architecture	e.g., Distributed System Architecture	e.g., Human Interface Architecture	e.g., Processing Structure	e.g., Business Rule Model
	Entity = Data Entity Rel. = Data Relationship	Process = Application Function I/O = User Views	Node = IS Function Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion
TECHNOLOGY CONSTRAINED MODEL *Builder* physical	e.g., Physical Data Model	e.g., System Design	e.g., Technical Architecture	e.g., Presentation Architecture	e.g., Control Structure	e.g., Rule Design
	Entity = Tables/Segments/etc. Rel. = Key/Pointer/etc.	Process= Computer Function I/O =Data Elements/Sets	Node = Hardware/System Software Link = Line Specifications	People = User Work = Screen/Device Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action
DETAILED REPRESEN-TATIONS *Subcontractor* out-of-context	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Specification
	Entity = Field Rel. = Address	Process= Language Statement I/O = Control Block	Node = Addresses Link = Protocols	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = Step
FUNCTIONING ENTERPRISE	DATA *Implementation*	FUNCTION *Implementation*	NETWORK *Implementation*	ORGANIZATION *Implementation*	SCHEDULE *Implementation*	STRATEGY *Implementation*

John A. Zachman, Zachman International

- Define various views of the architecture
- Columns are attributes
  - What
  - How
  - Where
  - Who
  - When
  - Why
- Rows are level of abstraction from system to component
- Other Frameworks
  - DODAF
  - MODAF
  - TOGAF

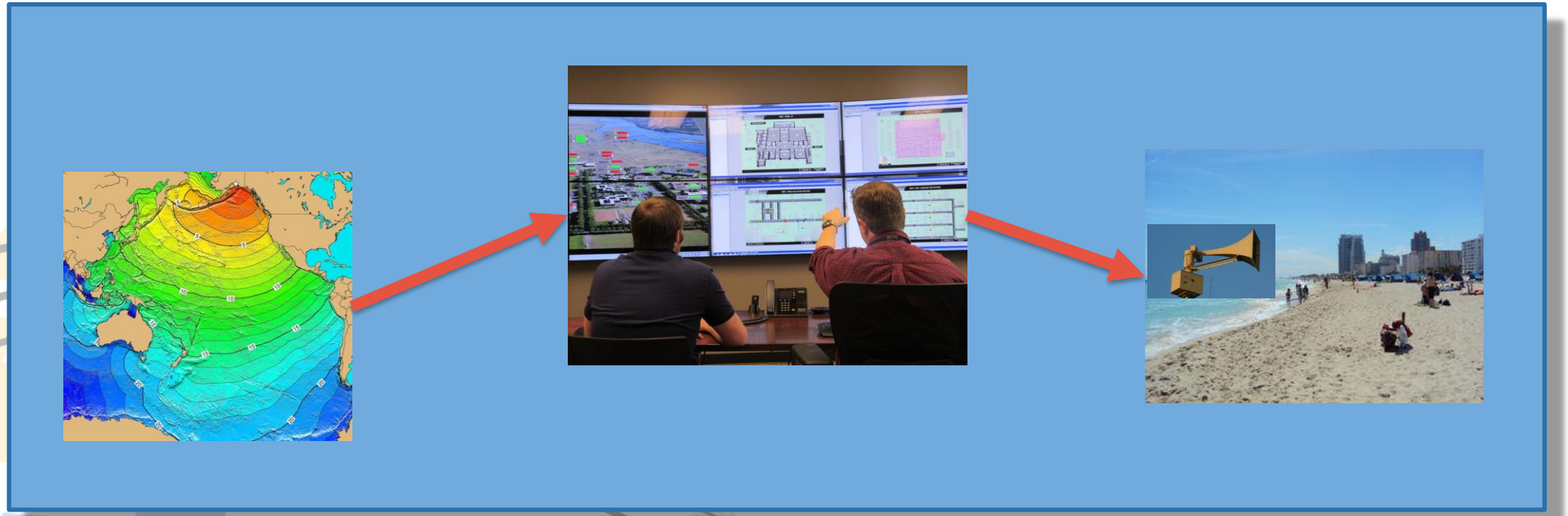
Physical Architecture



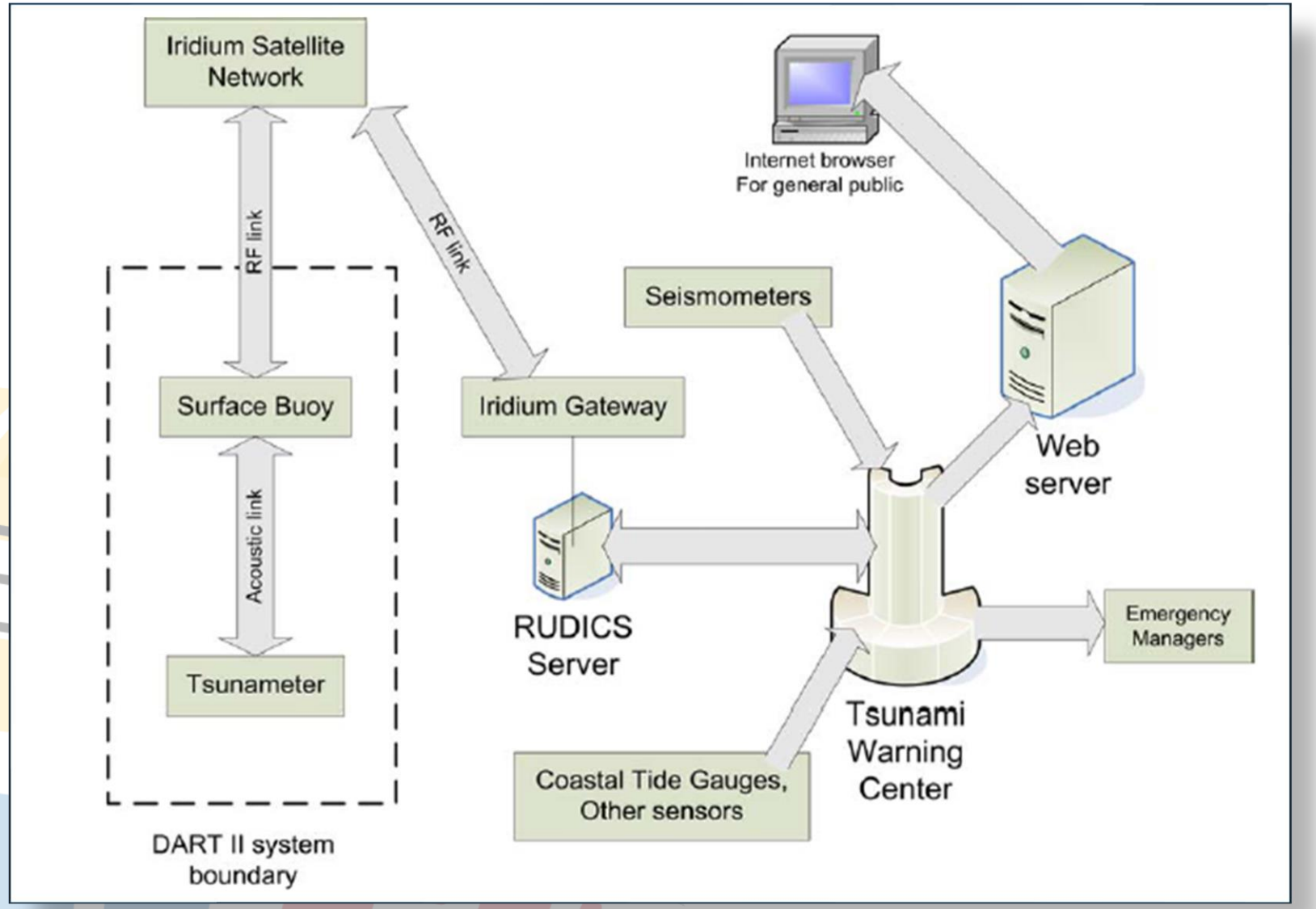
Schematic Block Diagram (SBD)

User View

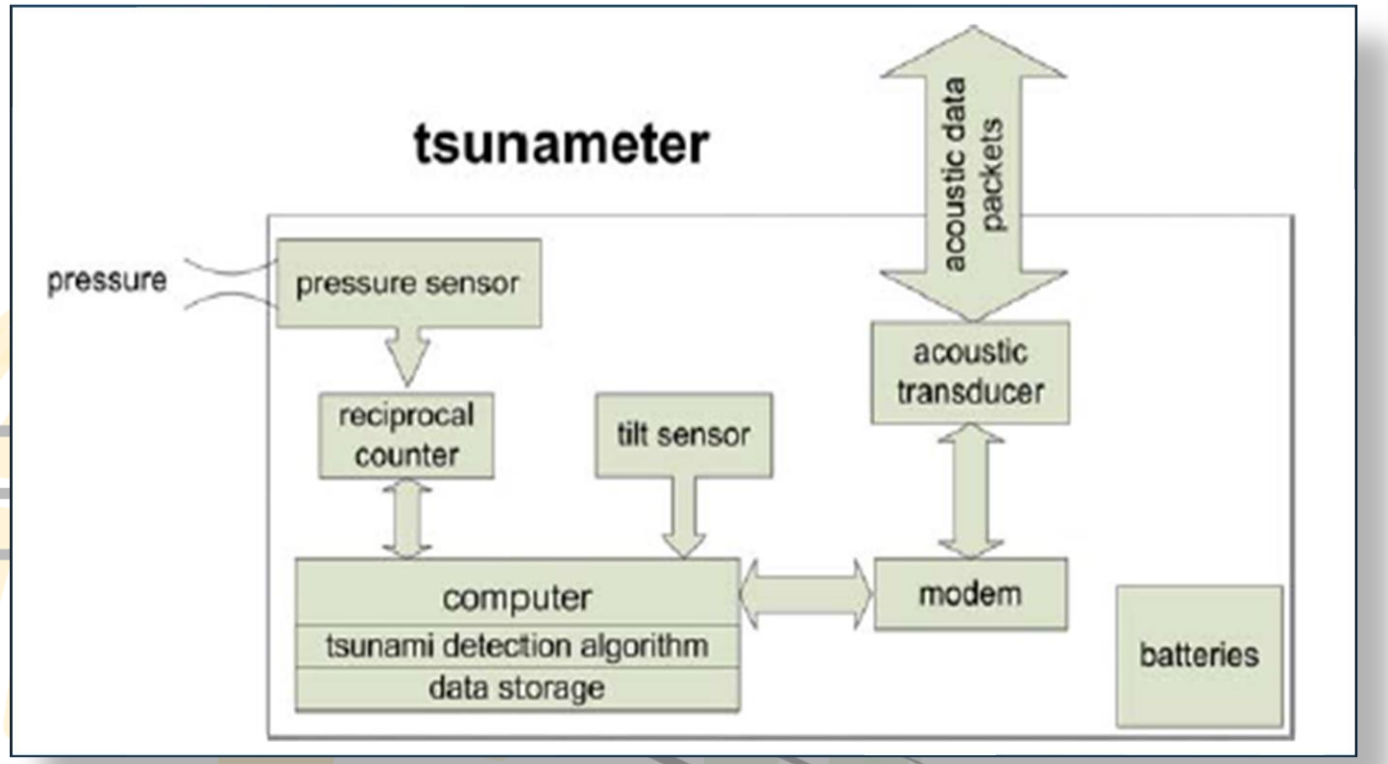
- Tsunami Alert System

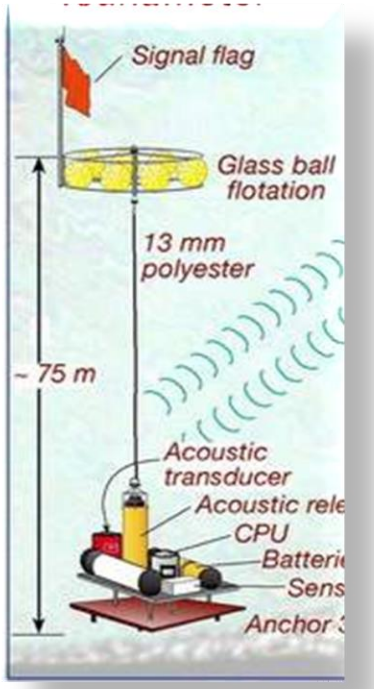


System View

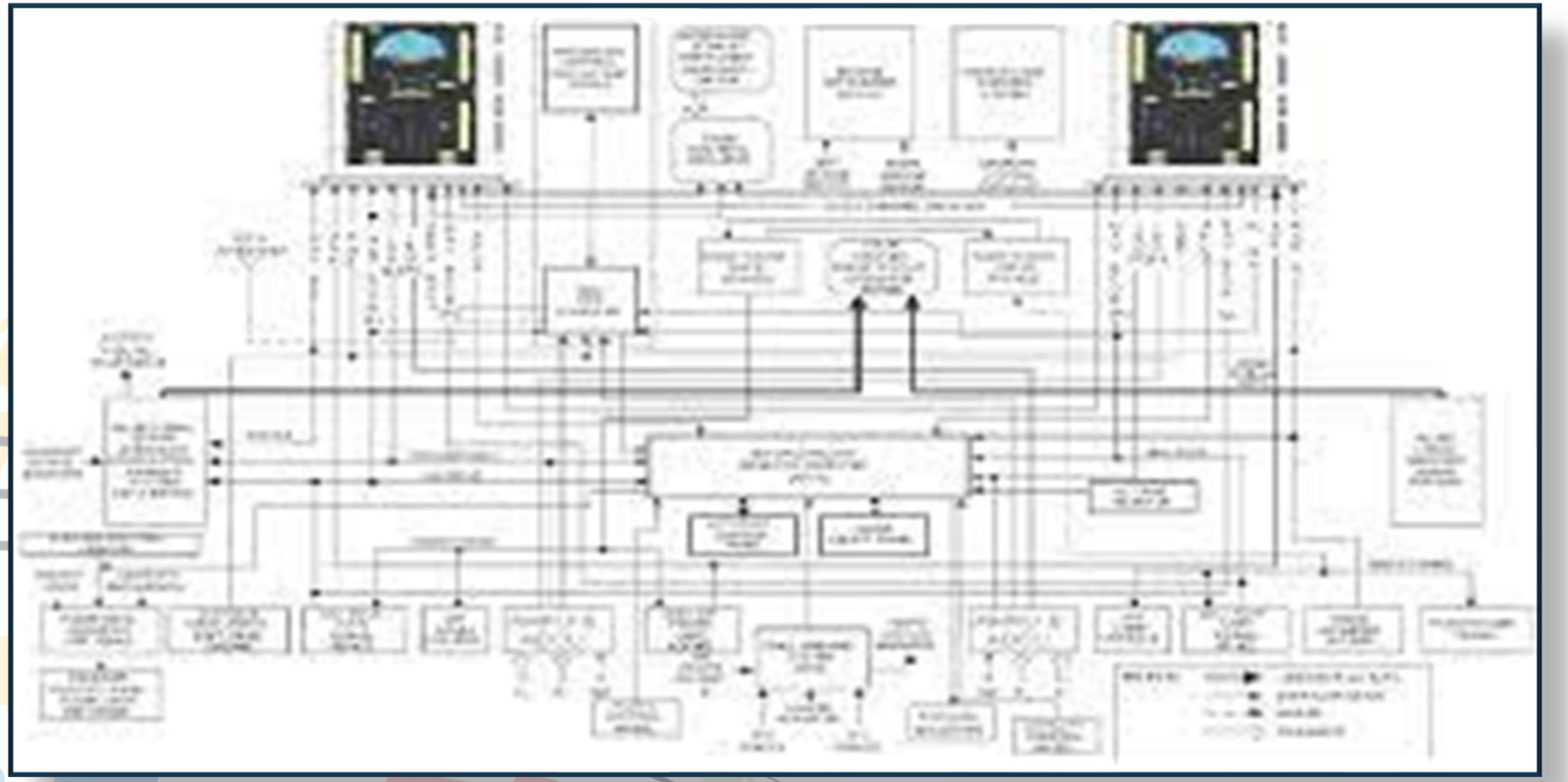

Iridium Satellite Network
RF link
RF link
Internet browser
For general public
Seismometers
Surface Buoy
Iridium Gateway
Web server
Acoustic link
RUDICS Server
Emergency Managers
Tsunameter
Tsunami Warning Center
Coastal Tide Gauges, Other sensors
DART II system boundary

Component View

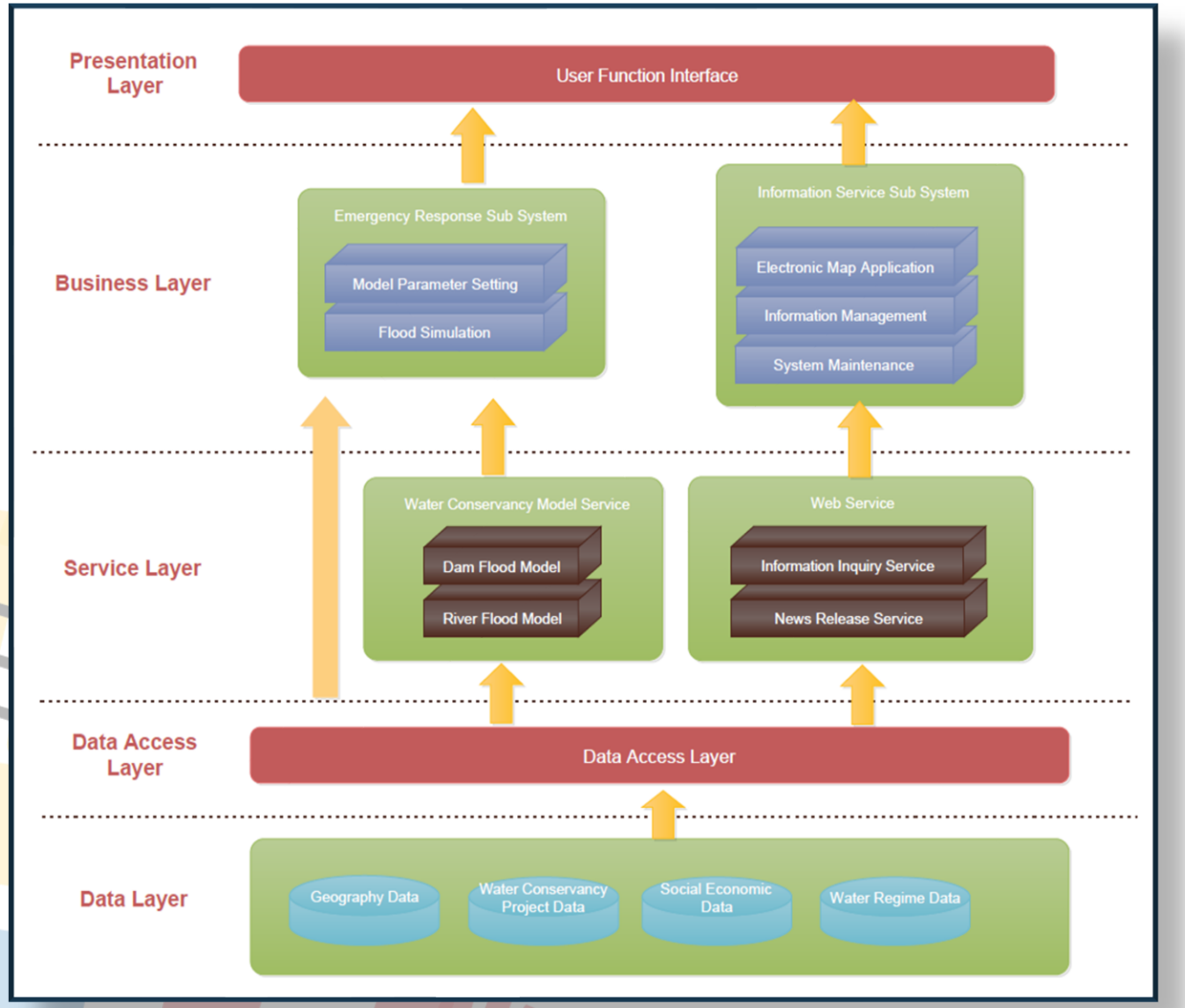




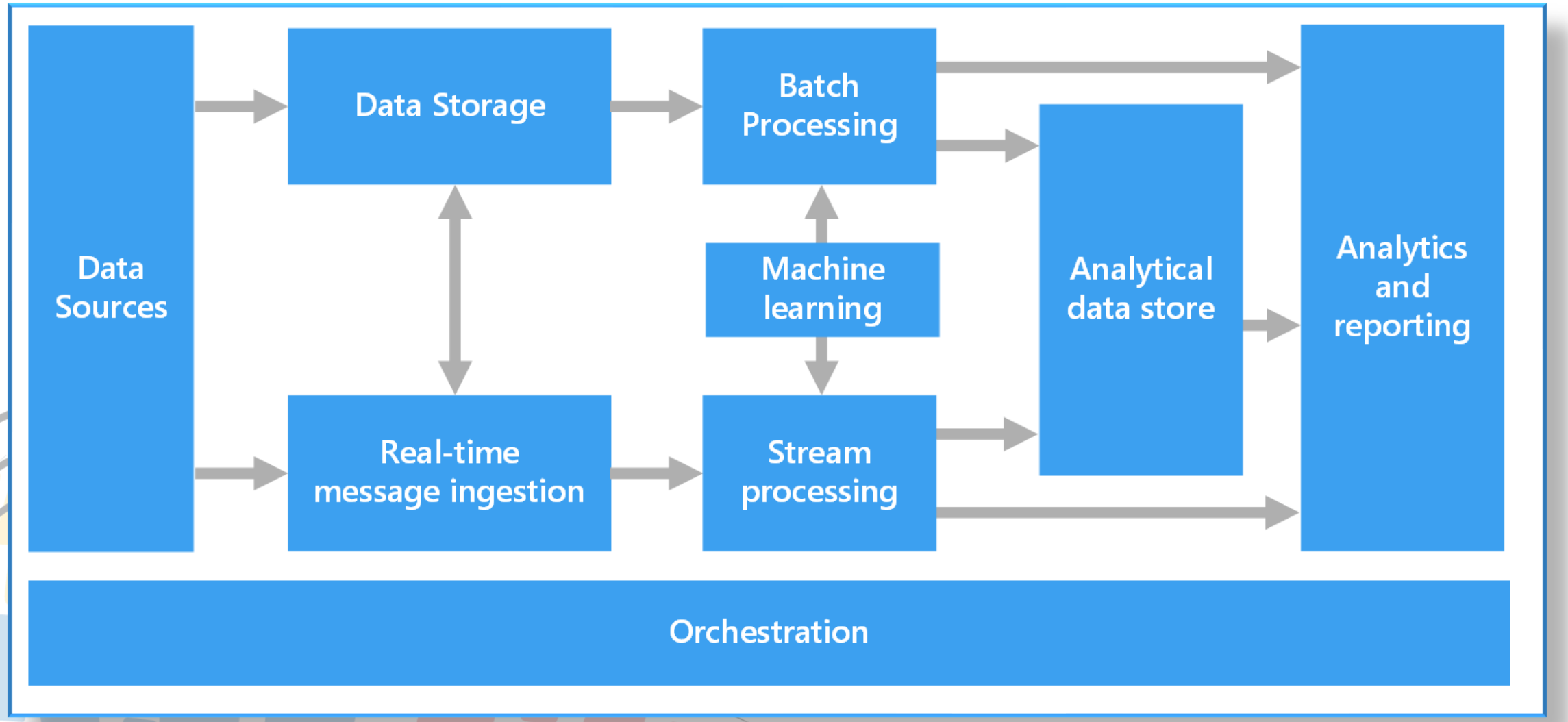
Typical Physical Architecture Diagram



SW Architecture

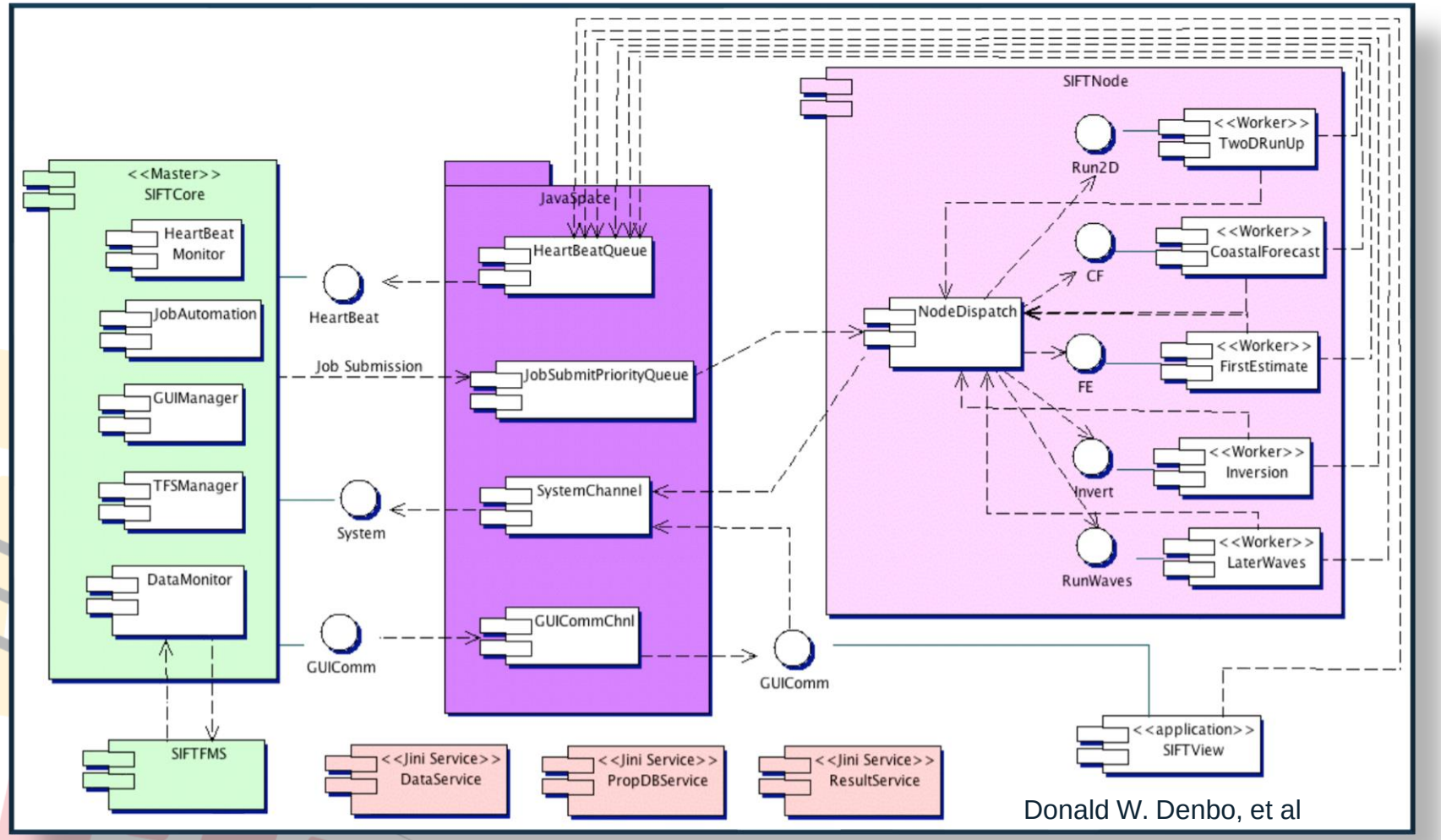


Data Architecture



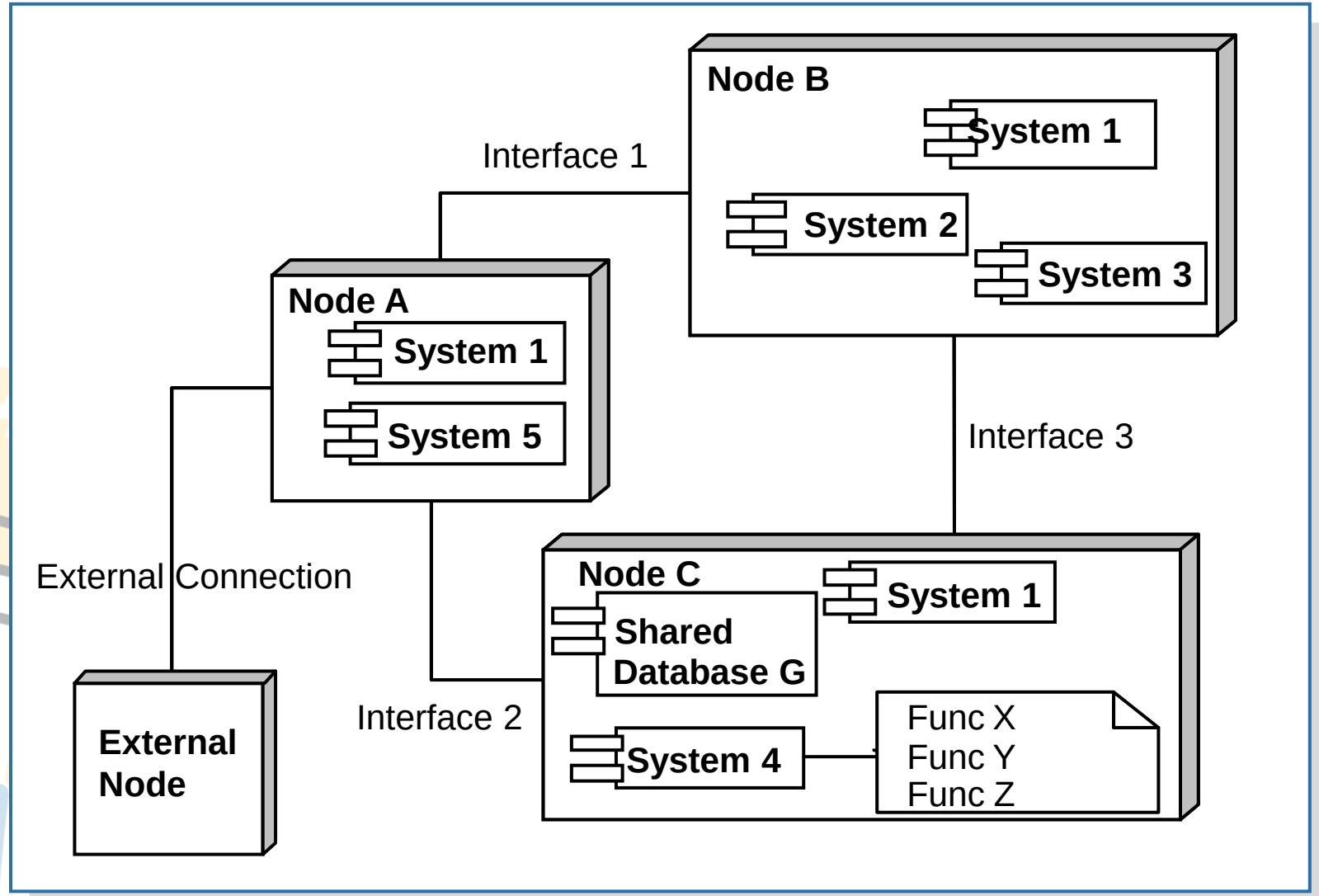
TSUNAMI FORECASTING SYSTEM

Software design and implementation using service oriented architecture

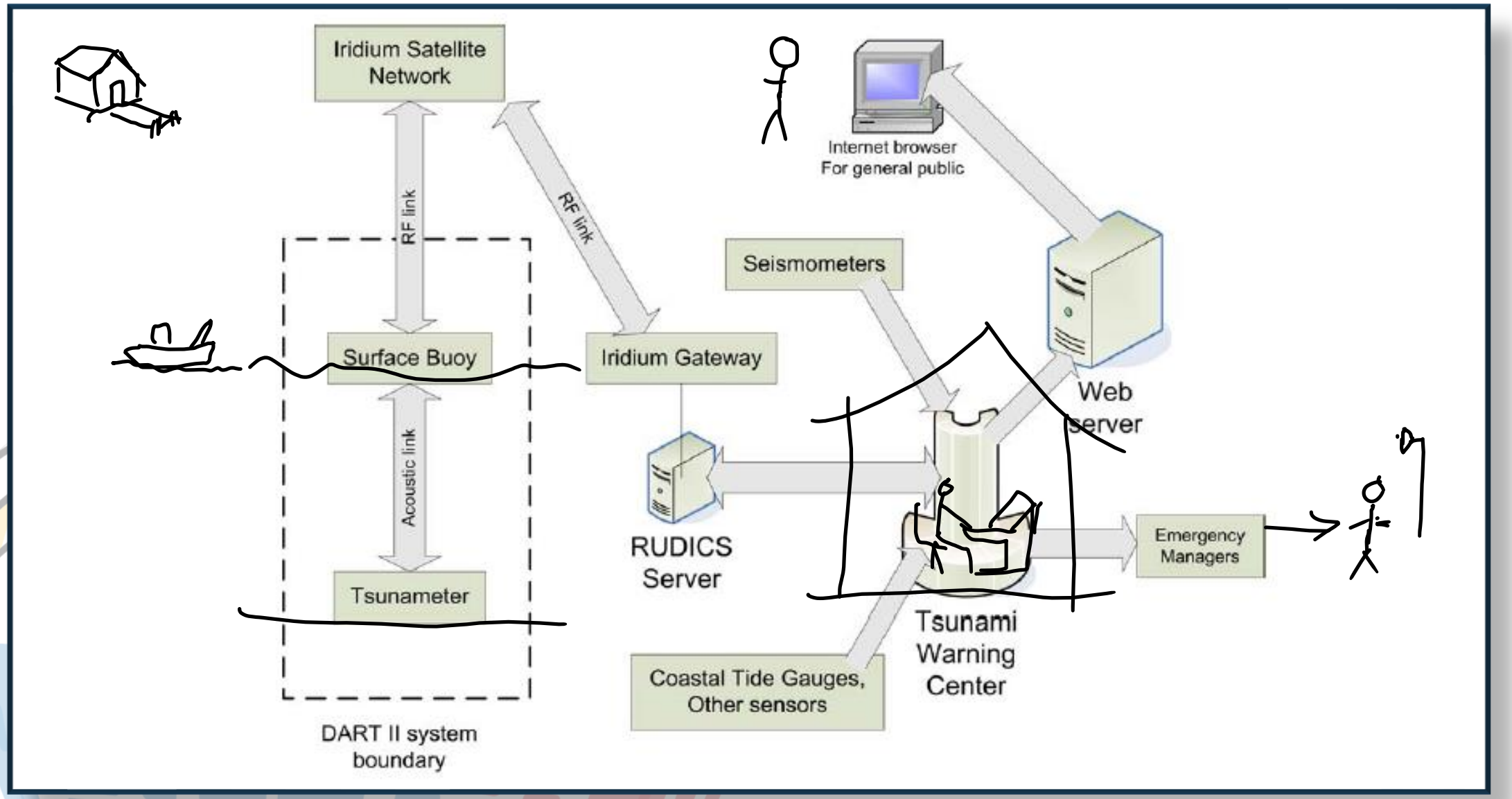


Donald W. Denbo, et al

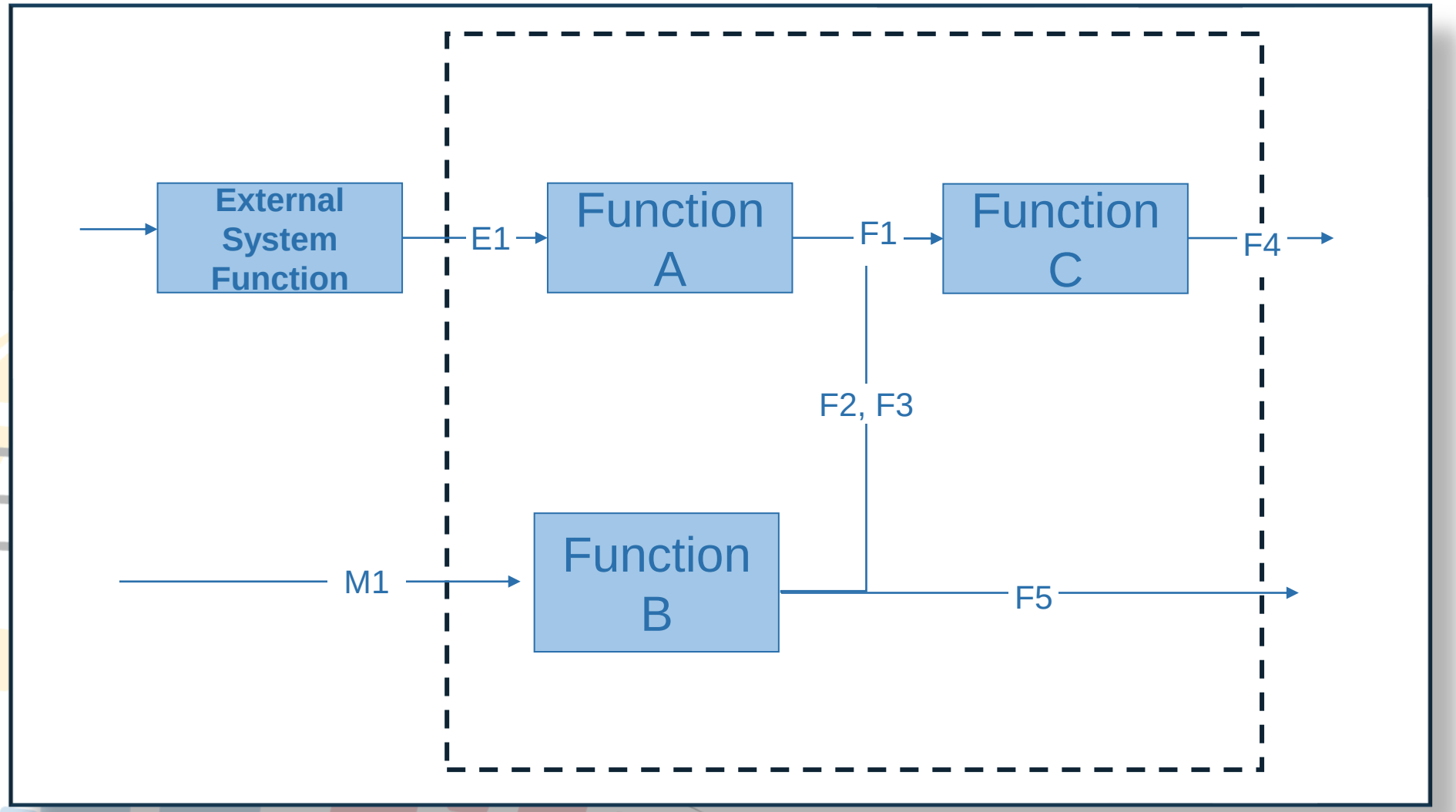
Deployment Diagram



But what is missing?



Functional Architecture



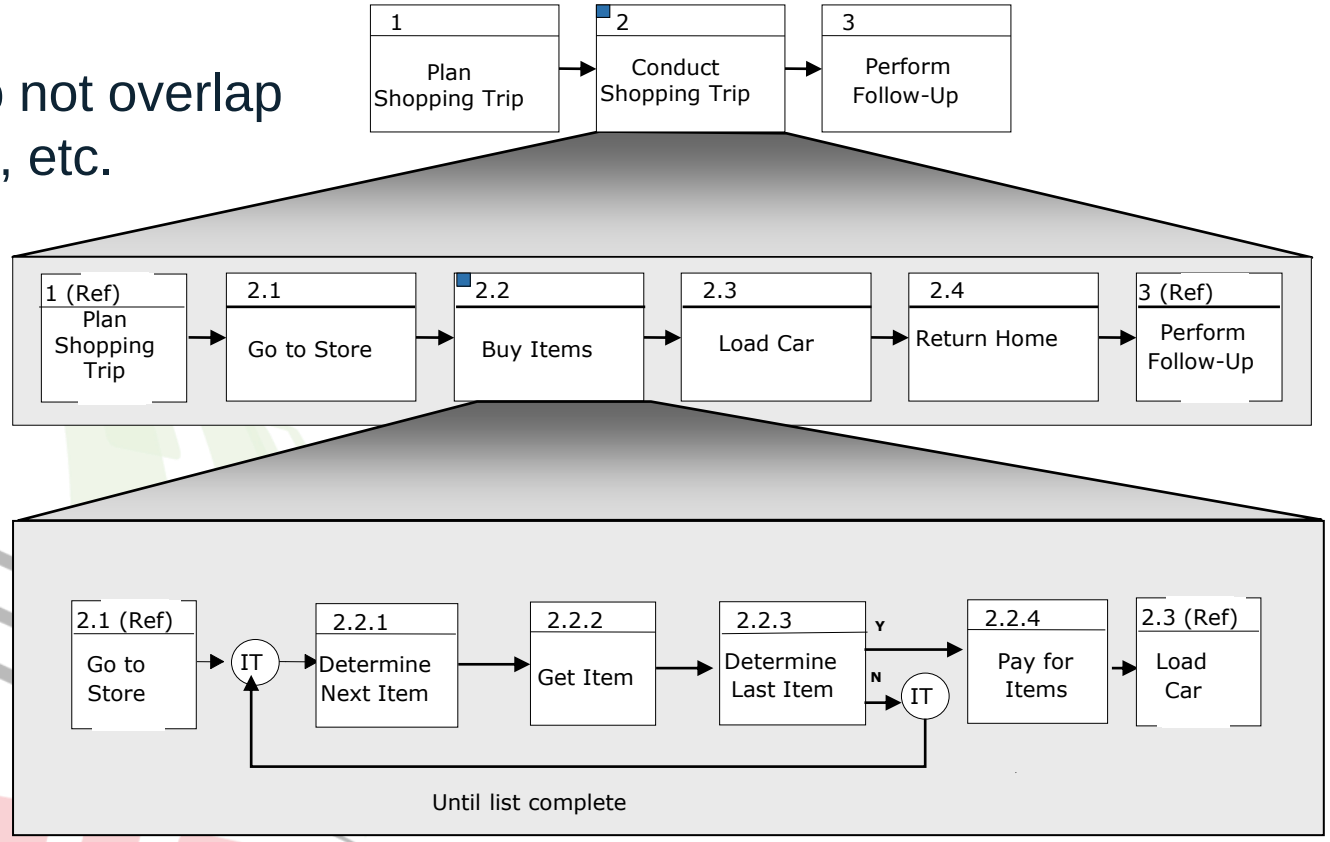
What the Functional Architecture Defines

- What the system does
- How (functionally) it does it
  - Series or parallel
  - Push or pull
  - Demand or schedule
  - Simple transmission or “ack-nack”
  - Data storage or process and discard
  - Command movement or exception-only control
  - Discrete/flood search/other (telephone)

Functional Flow Block Diagram

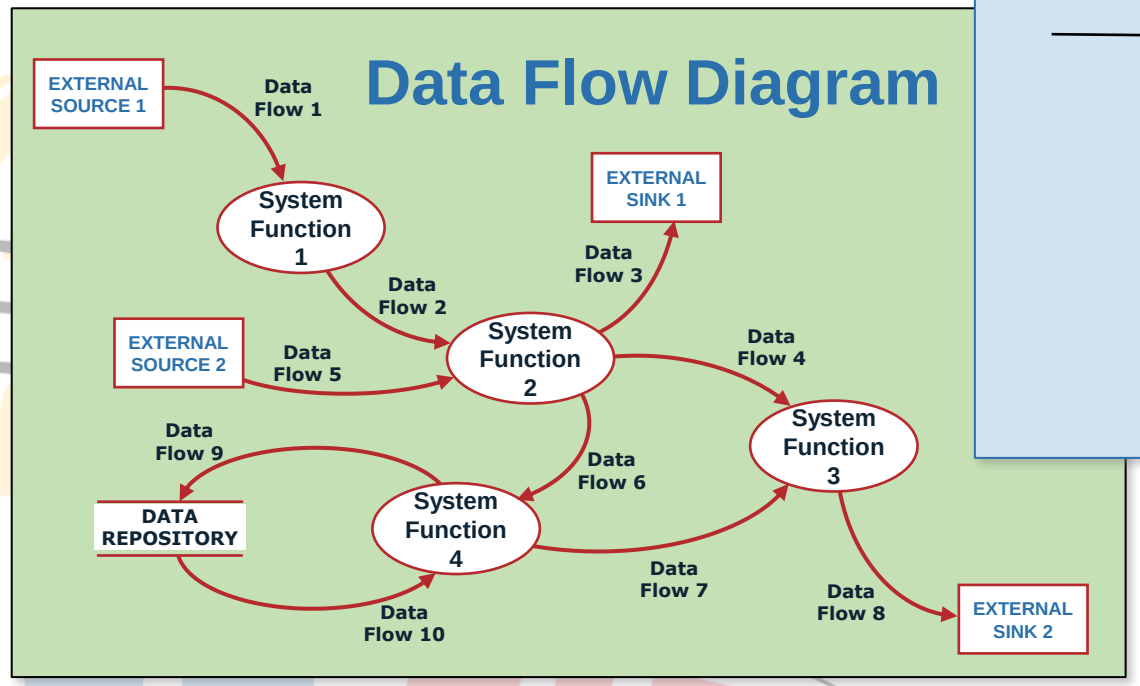
Sequence and control flow is defined in the basic FFBD

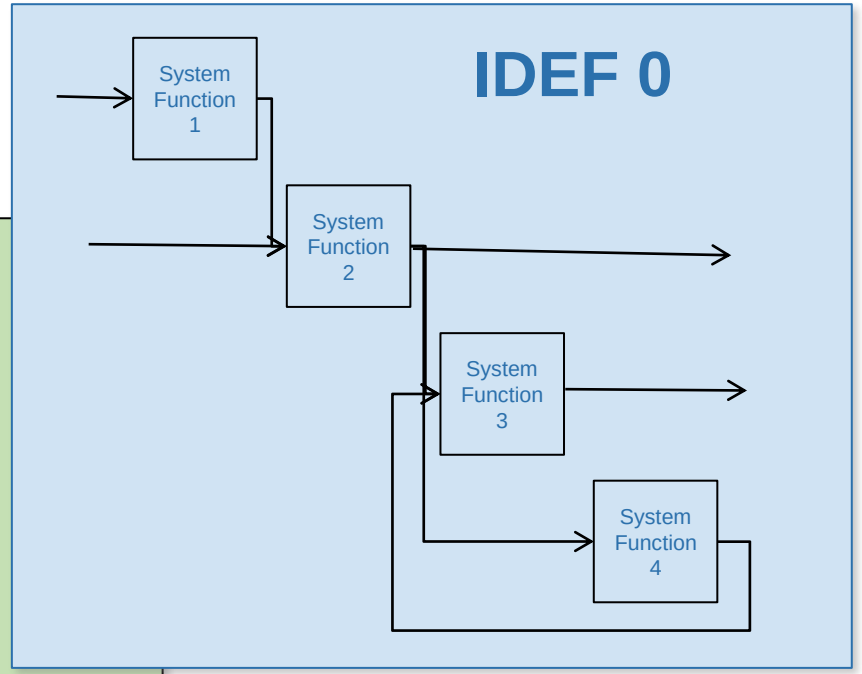
- Blocks in sequence do not overlap
- Parallel, iteration, loop, etc.



Data or Object Flow

Data flow is the principal concern of IDEF 0 and the Data Flow Diagram

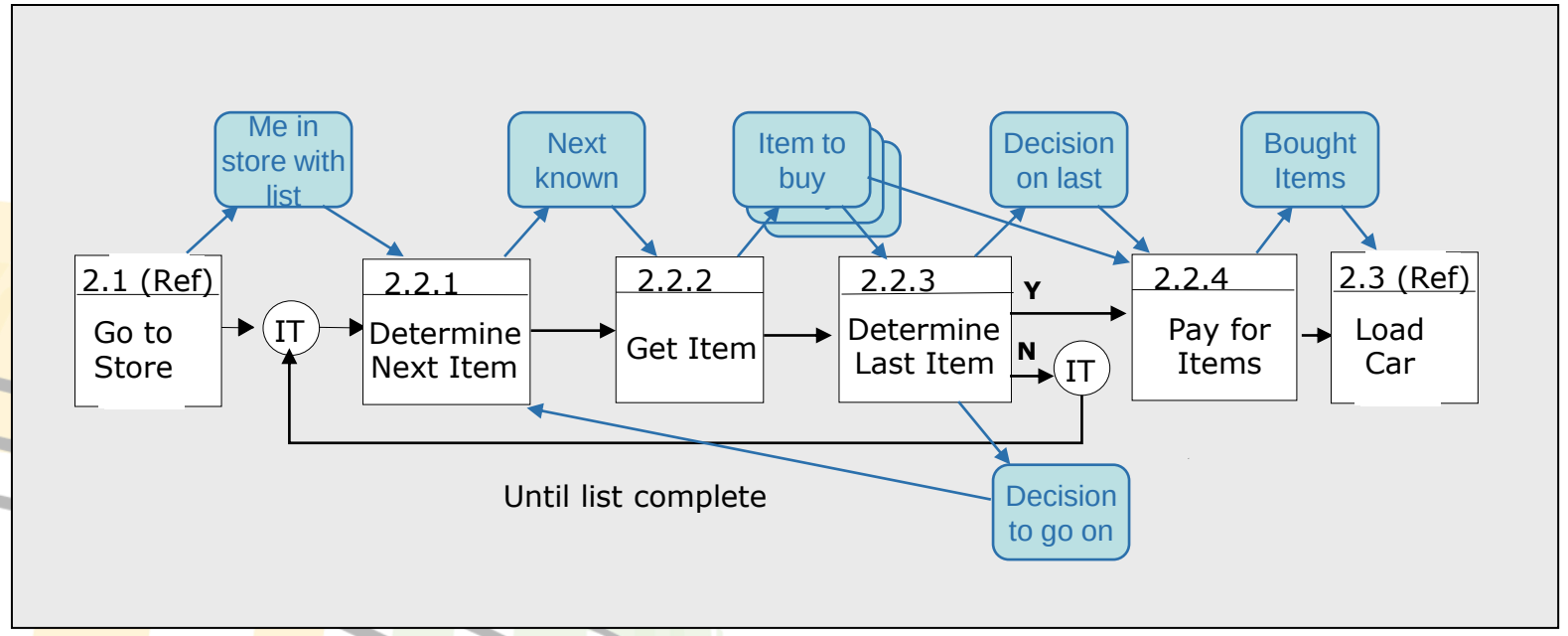




Sequence is only indirectly shown

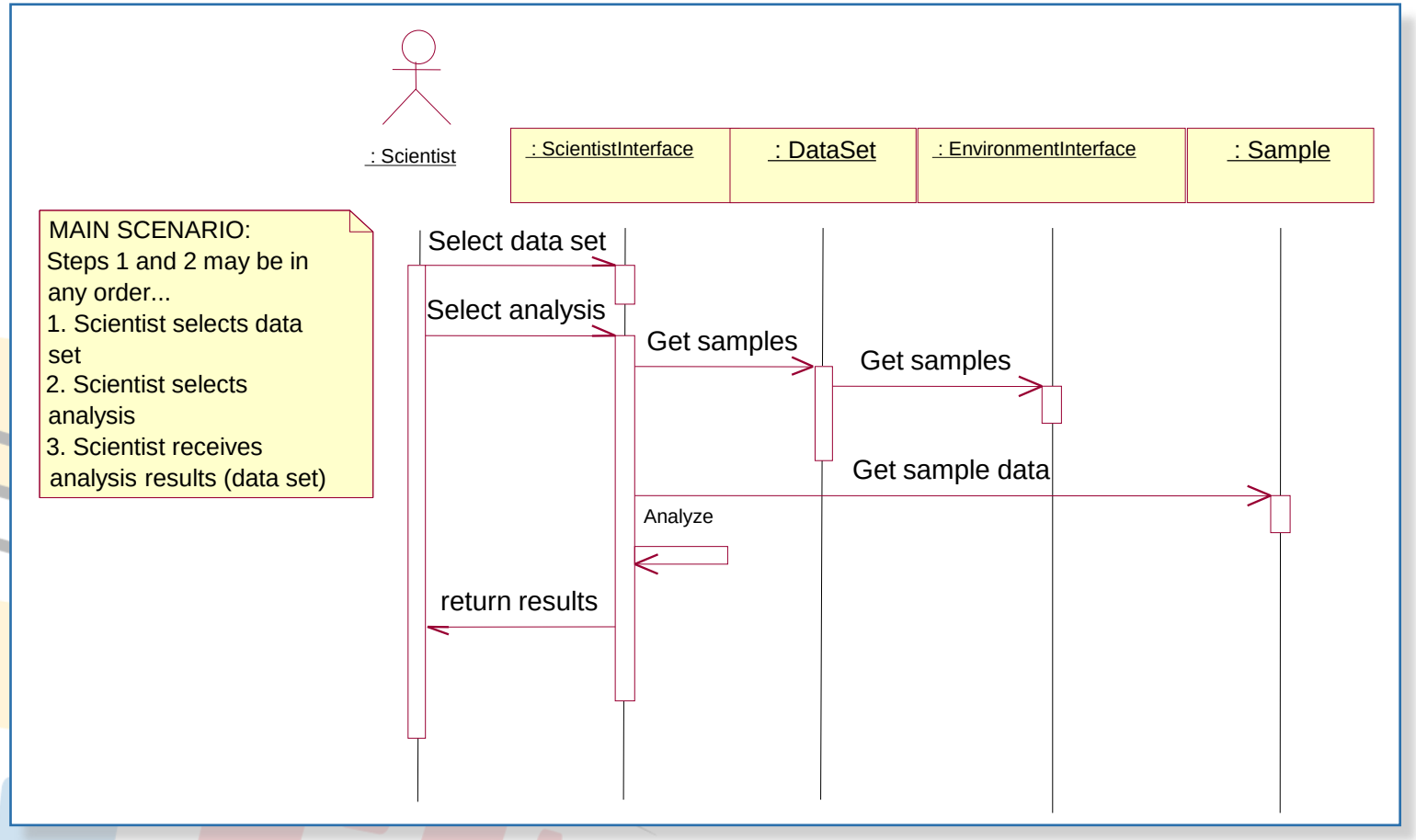
Enhanced FFBDs

- EFFBDs or Behavior Diagrams provide full, executable modeling of functionality with loops, decisions, replications, and other constructs.

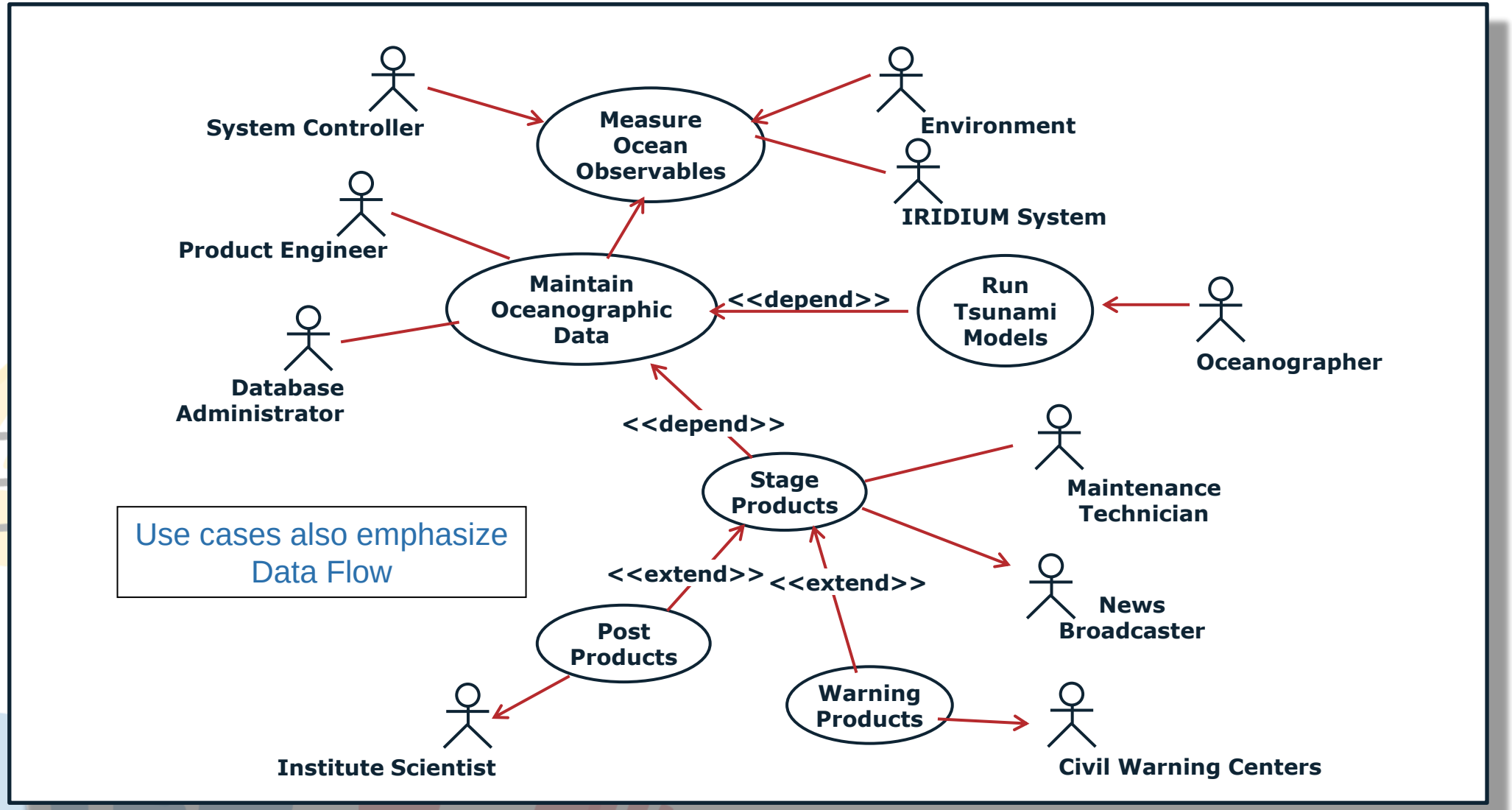


Sequence Diagram

Sequence Diagrams provide limited combination of sequence and data



System Summary Use Case



Use Cases

ID:	[Unique ID of this use case]
Title:	[Enter the goal of the use case - preferably as a short, active verb phrase]
Description:	[Describe the goal and context of this use case. This is usually an expanded version of what you entered in the "Title" field.]
Primary Actor:	[A person or a software/hardware system that interacts with your system to achieve the goal of this use case.]
Preconditions:	[Describe the state the system is in before the first event in this use case.]
Postconditions:	[Describe the state the system is in after all the events in this use case have taken place.]
Main Success Scenario:	[Describe the flow of events from preconditions to postconditions, when nothing goes wrong. This is the meat of the use case.]
Extensions:	[Describe all the other scenarios for this use case - including exceptions and error cases.]
Frequency of Use:	[How often will this use case be used?]
Status:	[Development status]
Owner:	[Who owns this use case, in your project team?]
Priority:	[Priority of this use case]

Interfaces

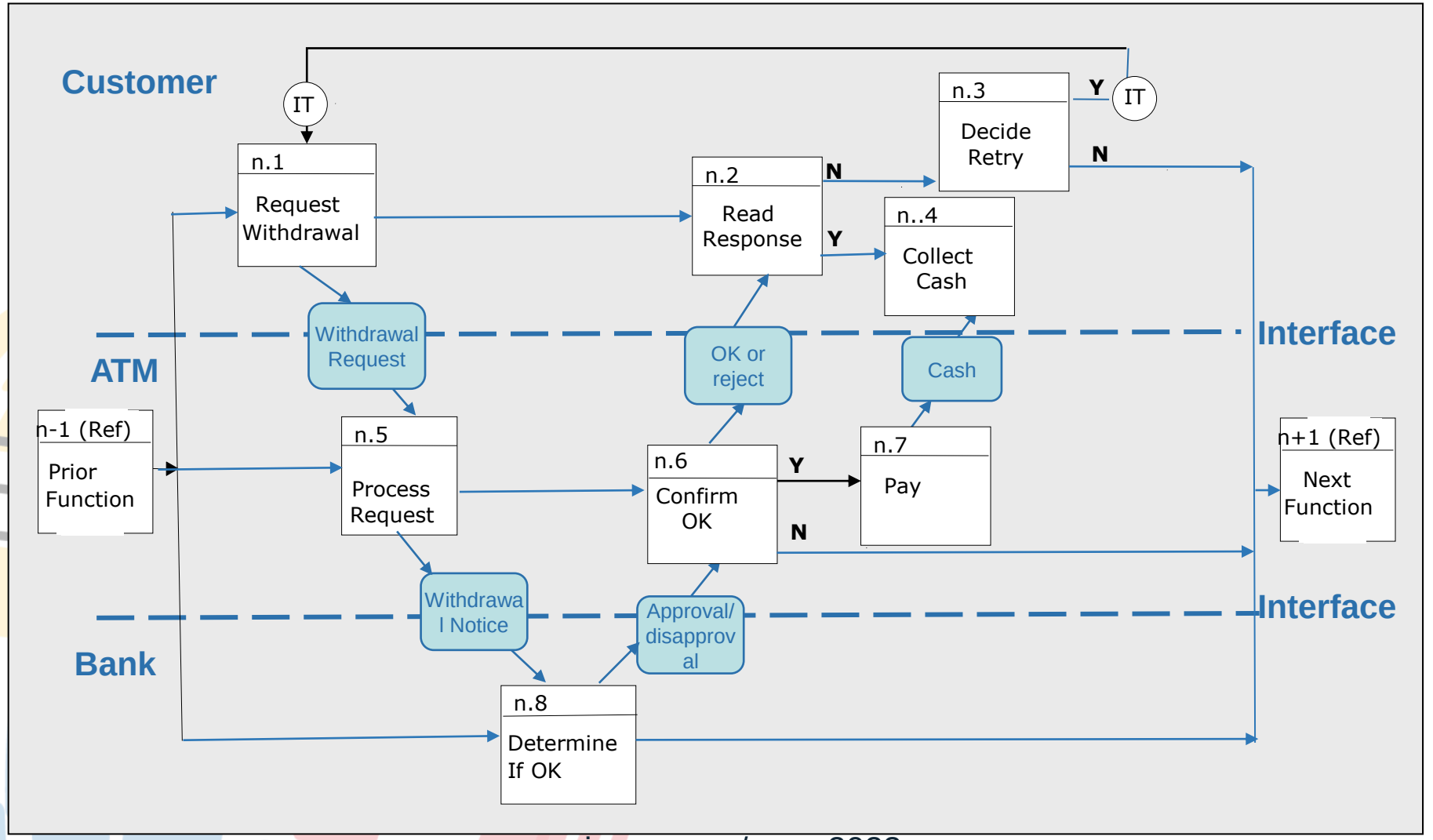
Types

- Physical
- Functional
- Human
- Enterprise
- Logistics
- Manufacturing
- …

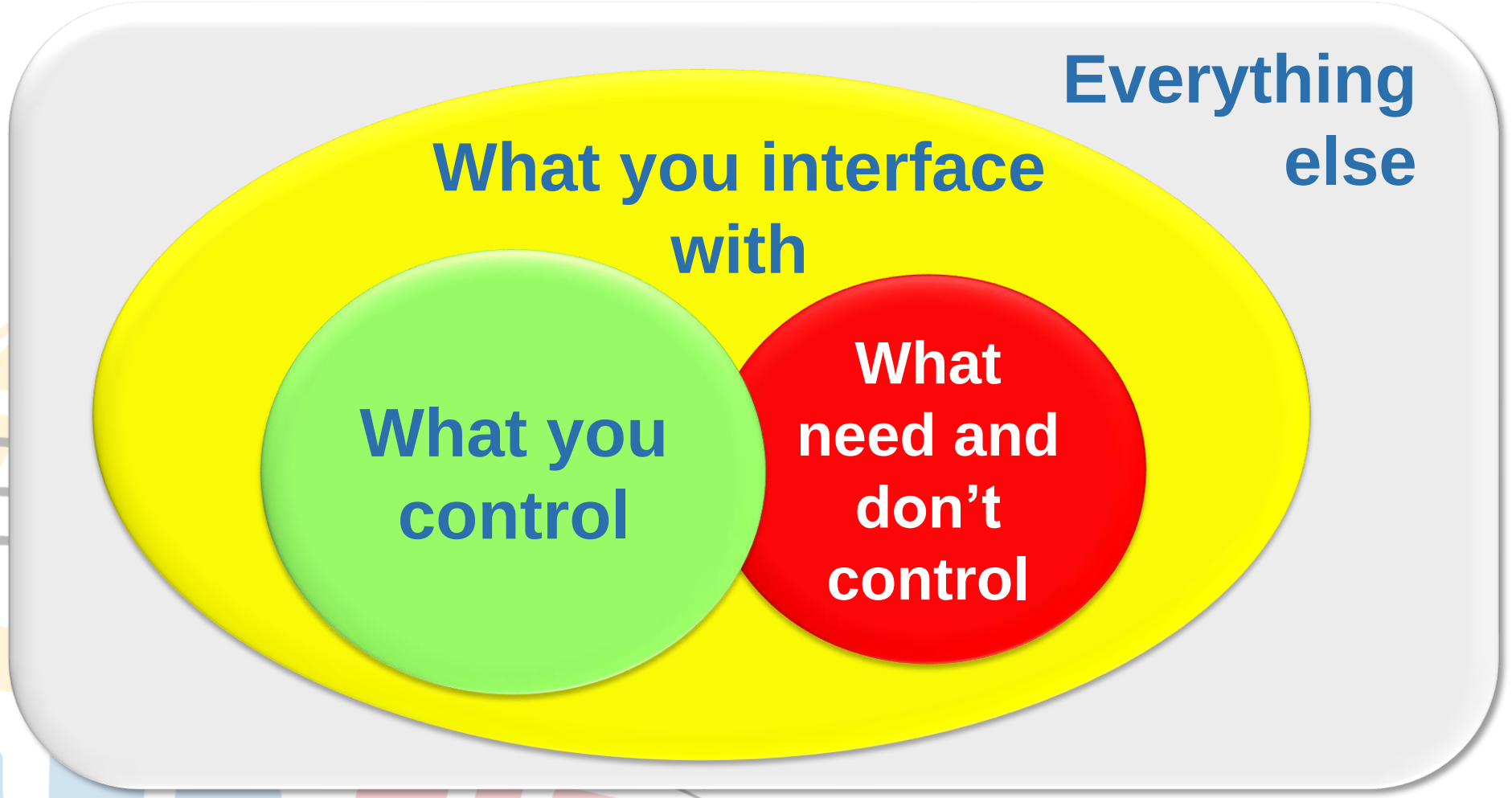
Considerations

- Direct/Indirect
- Control/don’t control
- Standards
- …

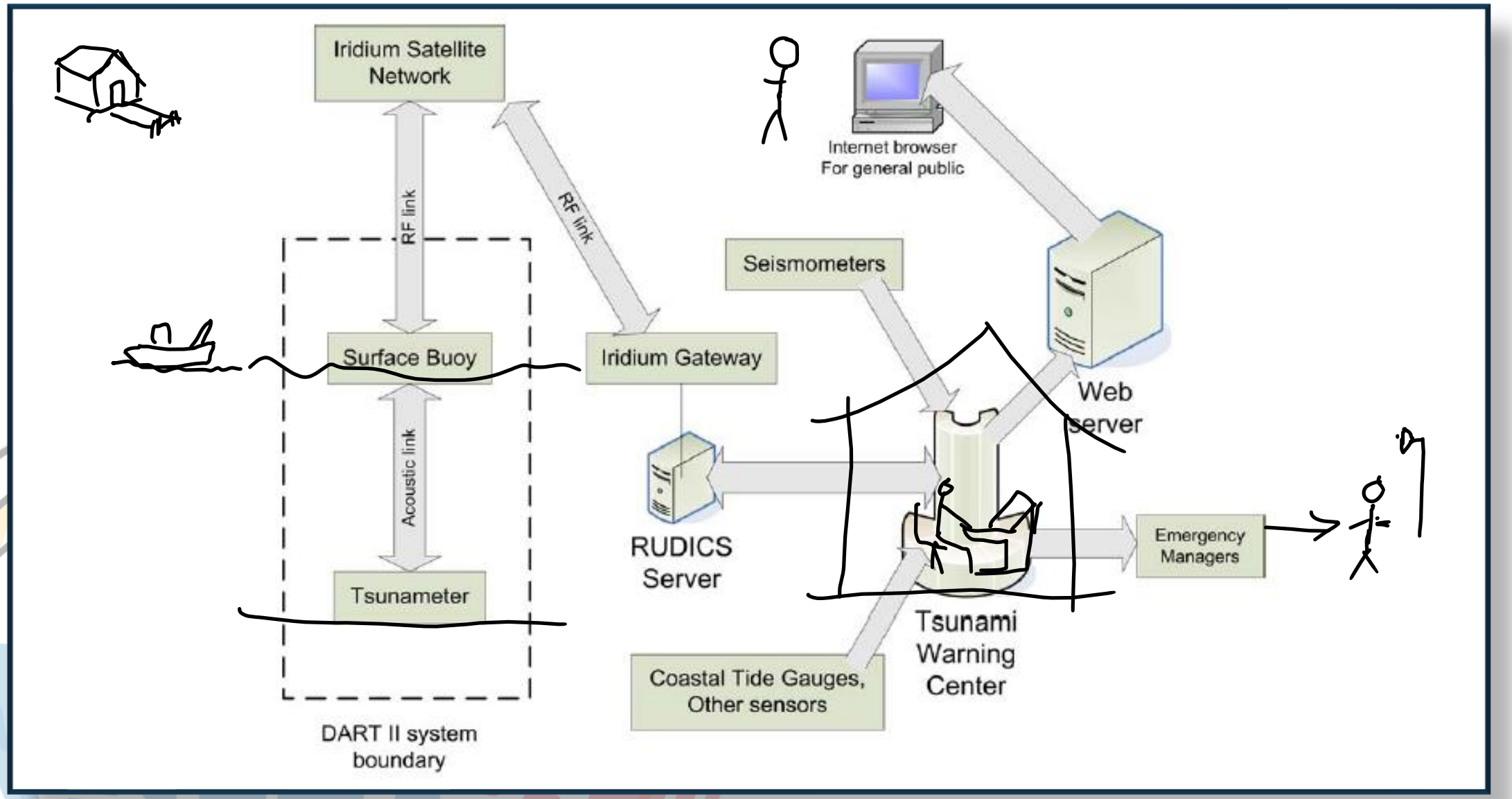
Defining Interfaces with the EFFBD



System of Systems



SOS Example


Iridium Satellite Network
RF link
RF link
Internet browser
For general public
Seismometers
Surface Buoy
Iridium Gateway
Web server
Acoustic link
RUDICS Server
Emergency Managers
Tsunameter
Tsunami Warning Center
Coastal Tide Gauges, Other sensors
DART II system boundary

Architecture Analysis

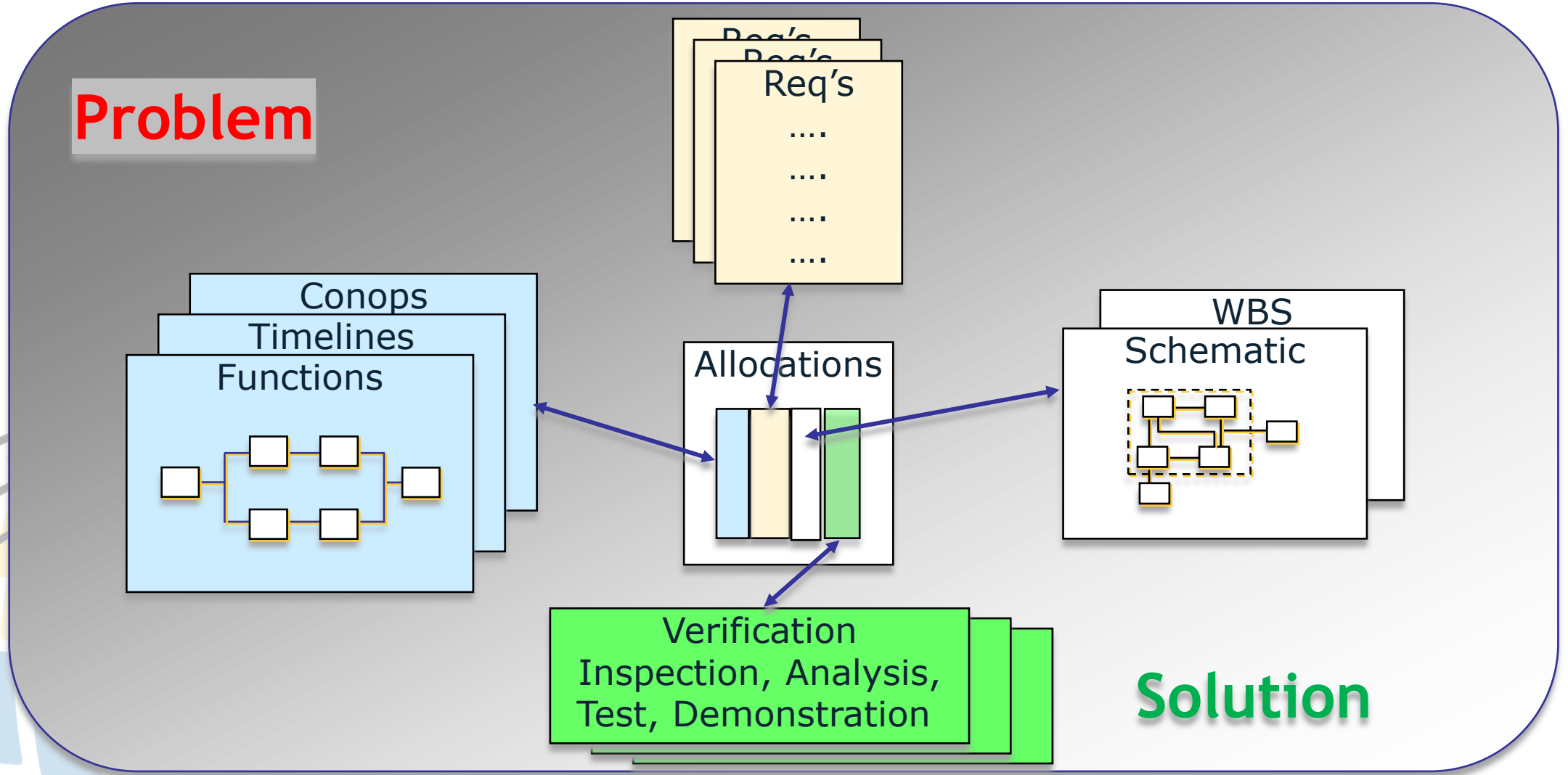
- **Commonality – internal reuse level**
- **Modularity – ease in upgrade to component or function**
- **Standardization**
- **Complexity – cyclomatic complexity, McCabe**
- **Failure analysis – physical, functional**
- **Maturity**
  - **Technology Readiness Levels**
  - **System Readiness Levels (TRLs + Integration Readiness Levels)**
- **Timing**
- **Peak loading**
- **Risks**

Don’t wait until built to find out it won’t work

Conway’s Law

- Generalized as:
  - Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.
- Main point:
  - System interface issues are likely to occur where organizational interfaces have issues
- Example of problem: Mars Climate Orbiter
  - Spacecraft expected metrics
  - Ground sent ‘English’ (feet)
- Lesson: Pay close attention to the Enterprise Architecture

Putting the Pieces Together



Architecture and other SE activities

- Requirements – Defines what a successful Architecture does and how well
  - Architecture choices may change or add to requirements
- Integration – Putting the pieces together
  - starts with integration analysis of the architecture
- Verification & Validation – is it correct
  - Meets both requirements and user needs
- Risk/Opportunity – likelihood of success
- MBSE – A way of expressing, communicating, and analyzing architectures

Places to go for more information

- INCOSE SE Handbook
- ISO/IEC/IEEE 15288:2015 Systems and software engineering -- System life cycle processes
- ISO/IEC/IEEE 12207 Systems and software engineering – Software life cycle processesThe Art of Systems Architecting – Maier
- Most SE textbooks
- Various Frameworks
- https://sebokwiki.org

Jim's contributions

- SE Methods Compared, INCOSE IS 1993
- Failure Mode Analysis in SE, INCOSE IS 1995
- Functional Architecture's Mental Roadblocks and Other Things Your Mother Didn't Tell You, INCOSE IS 2013
- Contact for questions
  - jimarmstrong29@aol.com

33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

www.incose.org/symp2023