Following an SE Process with an SE Tool

Jim Cunningham
JFC35@Comcast.net
26 February 2019
New England INCOSE Chapter

Goals and Audience

- Understand the architecture of the Systems Engineering process
  - Step through the SE process
  - Show the process architecture via a graphical view
- A Key Skill:
  - Does the SE know how to approach / organize a complex problem?
- Audience
  - Persons doing system type tasks but may be unfamiliar with the traditional SE process and the use of a tool to assist in managing the process and developing the system solution.

Define the Project Characteristics

- Does the Organization:
  - Define the project?
  - Receive a contract?
  - Contract and ideation?
  - And what role does the organization play?
  - What is the Enterprise perspective?
  - What is the Operational architecture?
- What is the Approach:
  - Top-down – new system
  - **Middle-out – new version or update of a known system**
  - Bottom-up – understand and document a existing system (reverse engineering)
- Define Scope of Project / Appropriate Tools / Personnel Skills
  - **Mini-System of Systems**? Subsystem?
  - Mostly Requirements?
  - All Software?
  - Size and Constraints?
- Mission / Problem Statement

Types of Models

- **“Descriptive” - includes requirements, functions, components, interfaces, etc.** (today’s discussion)
- “Analytical” – MATLAB, Simulink, custom – usually models the functional behavior
- Design Level – e.g., electrical board layout, structural, thermal

Value of the “Descriptive” Tool

- **Get the team on the same page**
- A template for project development
  - Helps with thinking and discovery
  - Helps with following a process
- A place to capture project system information
- Documentation done up front
  - More useful up front (system, less so for the design level), flows to the design teams
  - Not neglected at end of project
- The various classes, attributes, and relationships plus diagnostics aid in:
  - Completeness (eg, functional i/o items trace to source)
  - Correctness (eg, output documents for subsystems are consistent)
- Outputs (Reports, Tables, Diagrams) help with communicating with the team and customer.
  - Get everyone on the same page
  - Easier to find flaws
  - Communication is a key role of the systems engineer

A Specific Systems Tool:
Vitech's CORE (or Genesys)

- A System-Wide model with schemas and capabilities that include:
  - Basic engineering – Requirements, functions, components, interfaces
  - Verification and Test
  - Program Management
  - DoDAF
  - Flow and Performance analysis of discrete functions

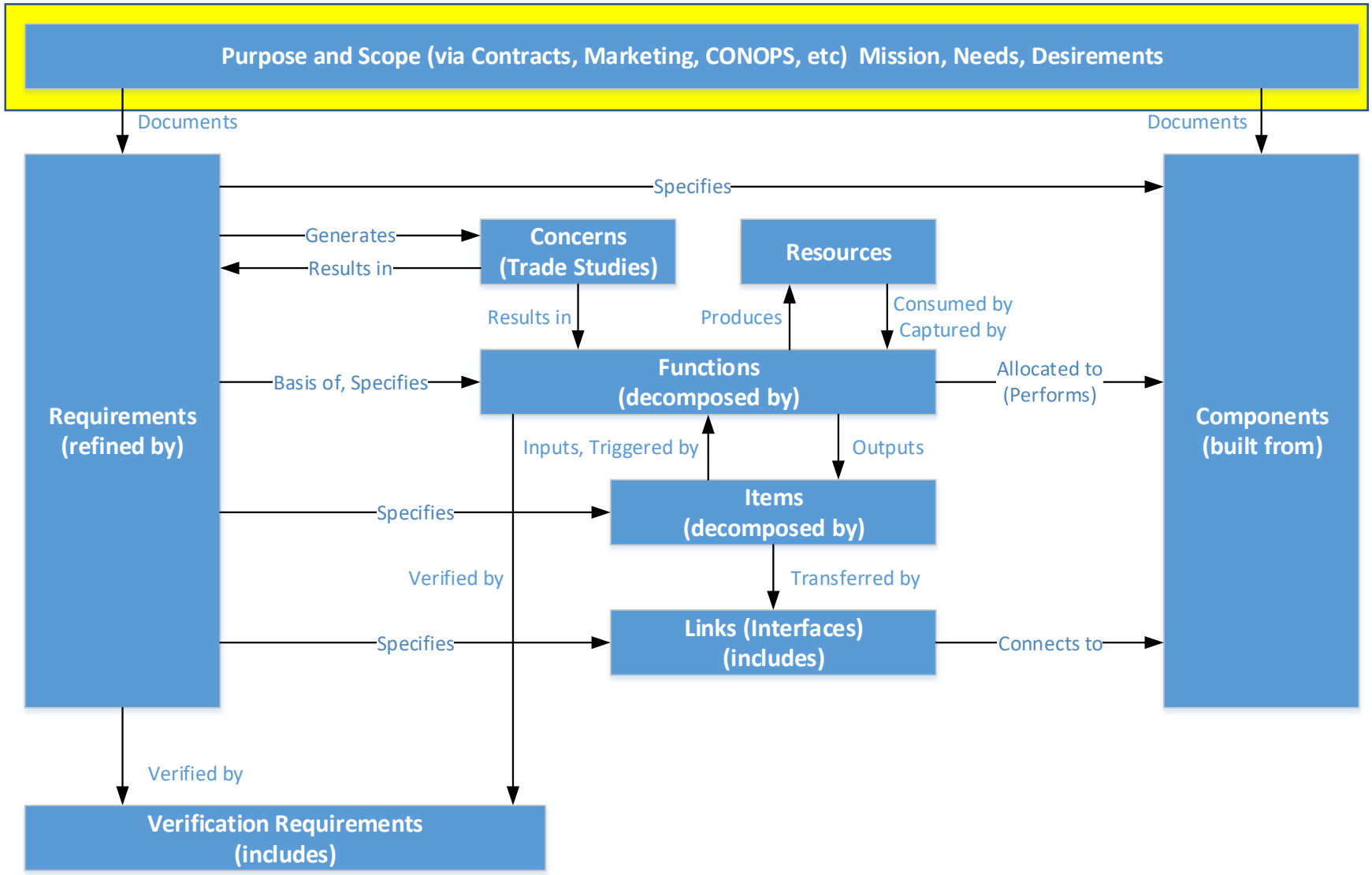
- Internal Structure is "ERA" / "object oriented"
  - Entity (Elements) (Components, Functions, Documents, …)
  - Relationships (specifies, allocated to, includes, verified by) between elements
  - Attributes (type, number, origin, cost, …) of an element

- Generates Views and Reports (artifacts) from the database

Define the Context for the Model

- Life-Cycle
- Marketing/Contract/Business/Product Line
- Engineering
- Manufacturing
- Deployment
- **Operational** - a good starting point
- Maintenance
- End-of-Life

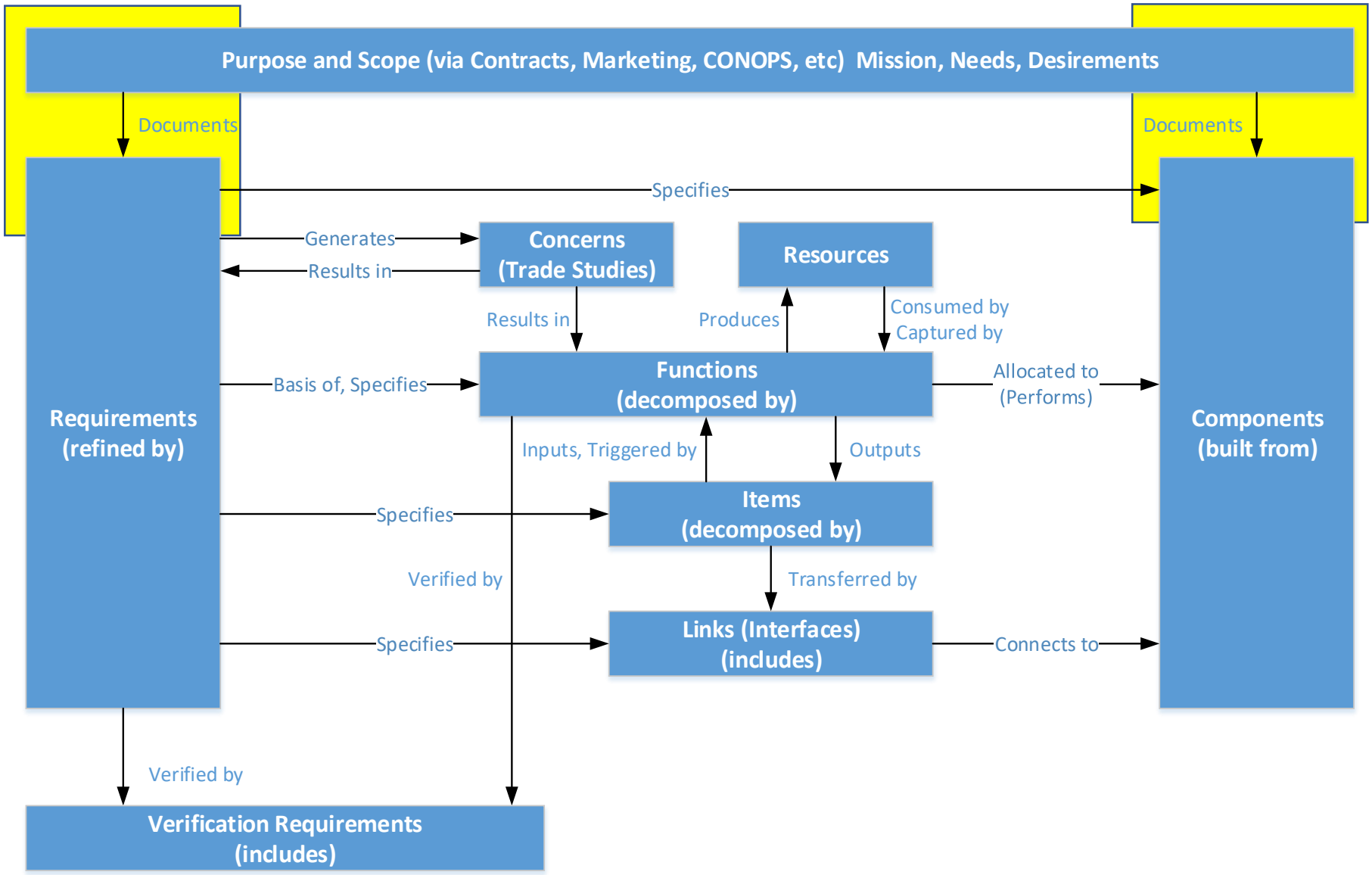
Get the Starting Point (Documents, Ideas, …)



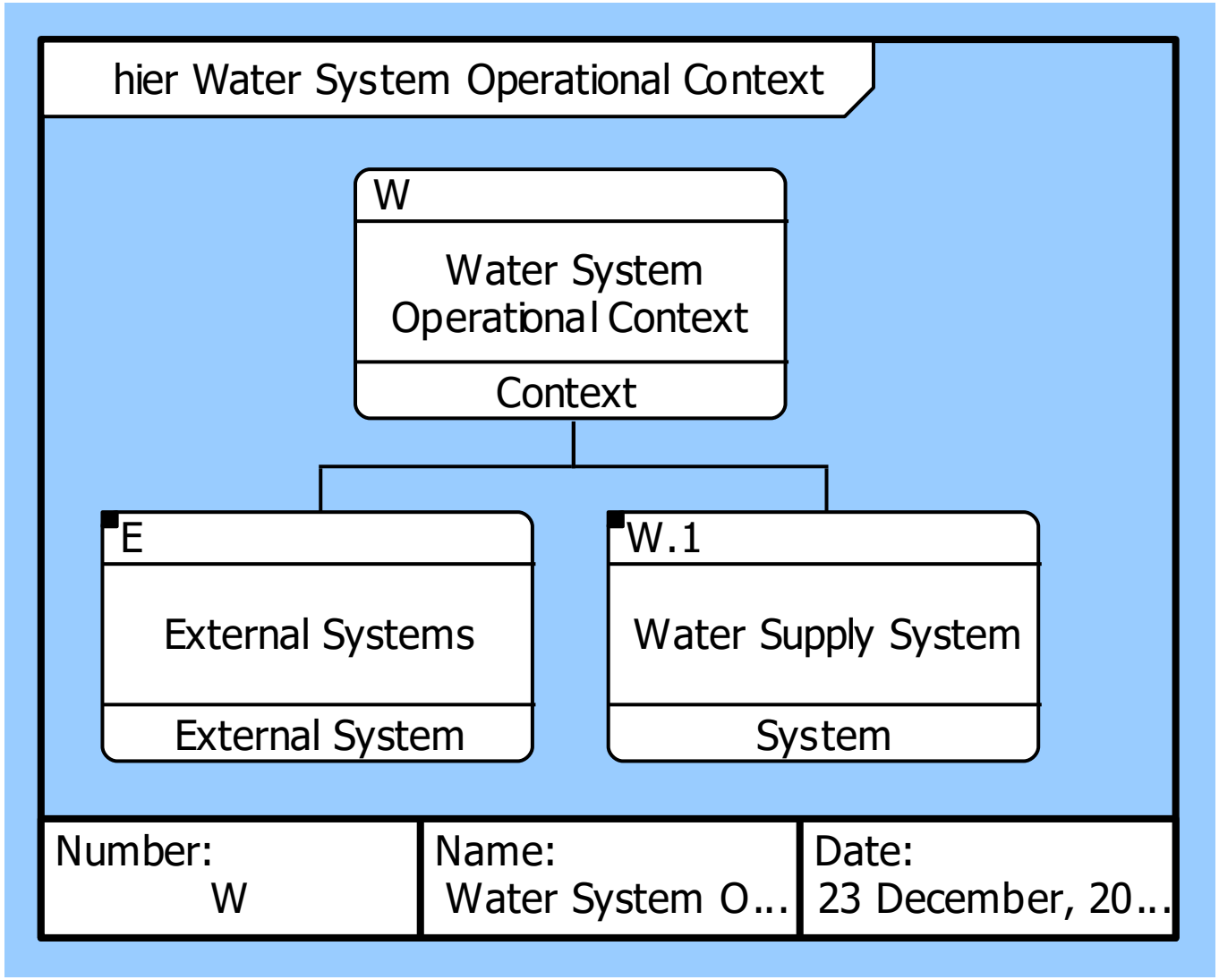
Next Steps Depend on Project Character

- Capture the Easy, Already Defined Data
  - Top-Down, New – System Definition, Key Functionality. Derive Requirements.
  - Top-Down, Contract – System Definition, Requirements from Contract
  - Bottom-Up or Existing – System Definition, **Components and Interfaces, Functionality**
- Top-Down Example:
  - Need = Housing
  - Top Level Function = Provide Housing
  - Top Level Component/System = House, Tent, Cave, Igloo, Yurt, …
  - Define Environmental Components
  - Develop Specific Requirements

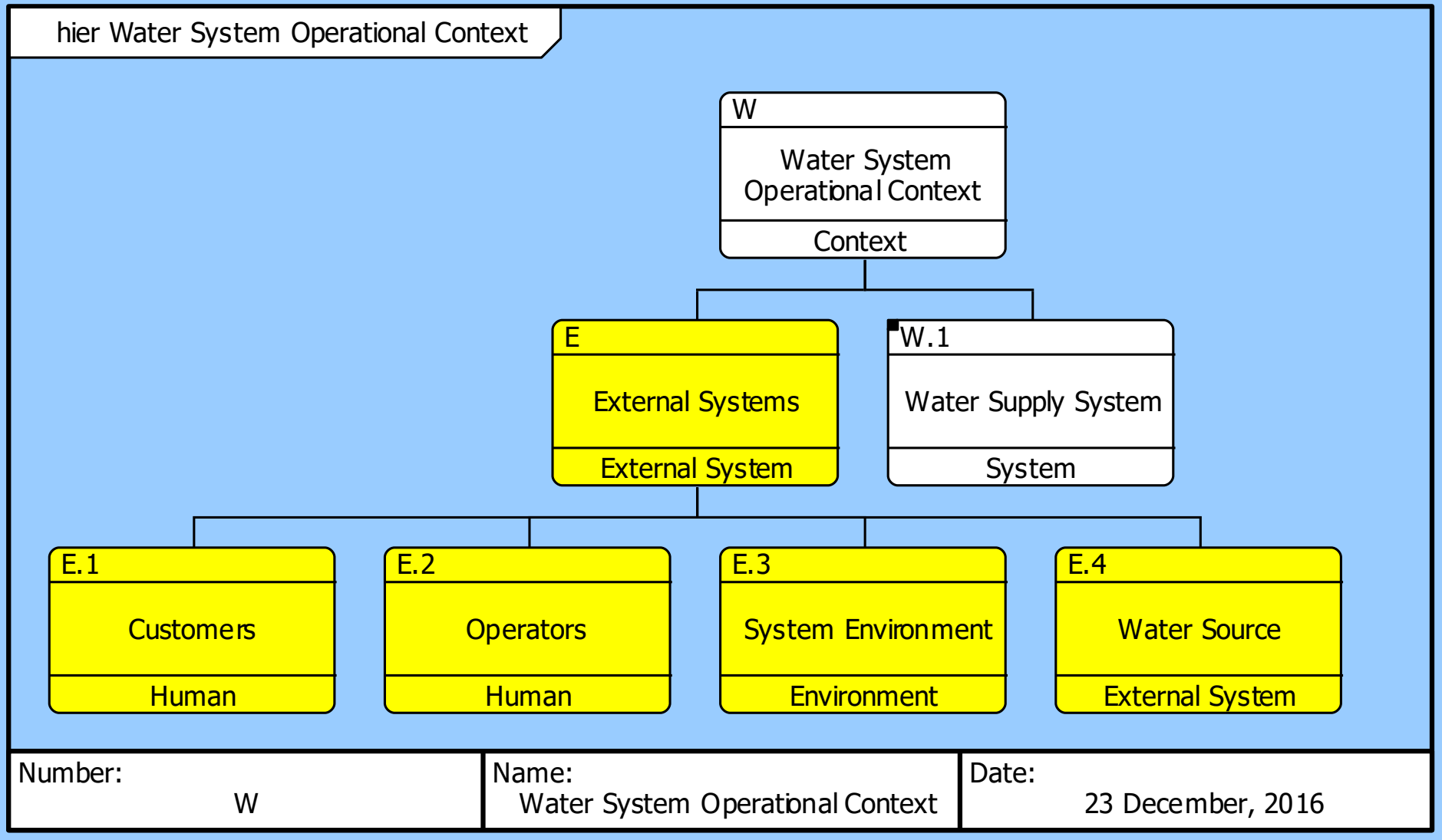
Define the System Boundary, Internals, Externals, Requirements



Define the System Boundary Internals, Externals



Define the System Boundary - Externals

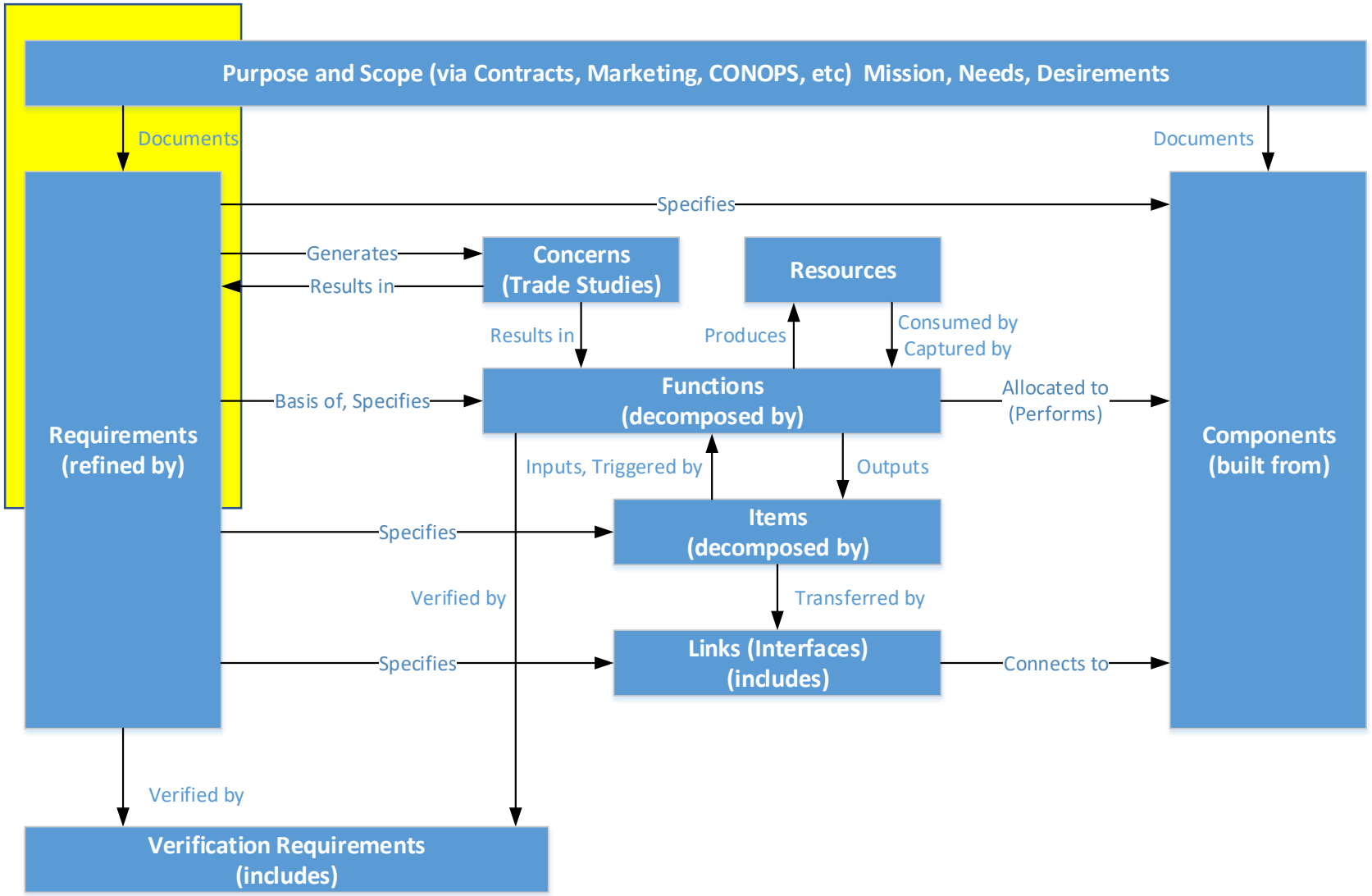


Generate the Requirements

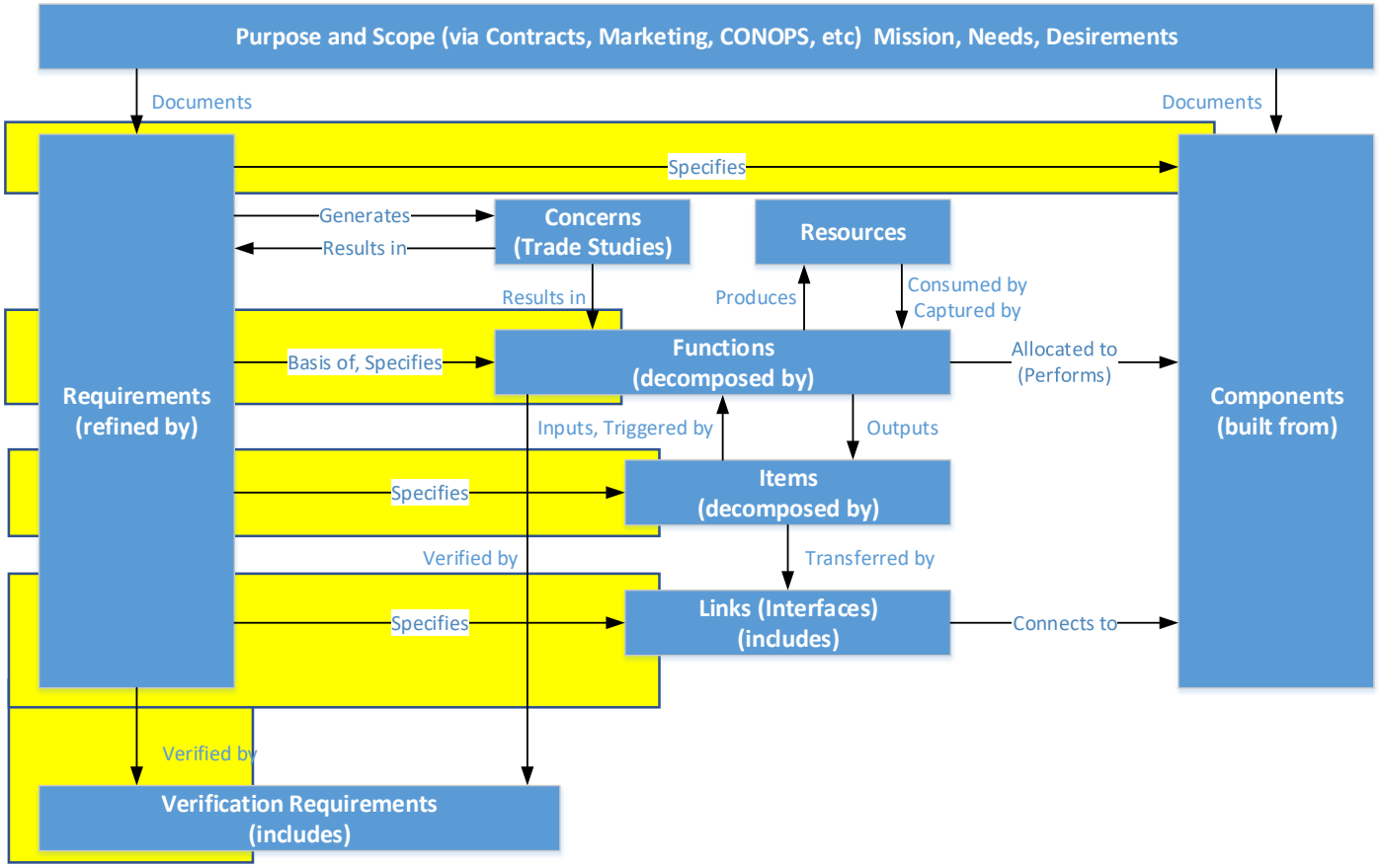
Start with “desirements”, Develop Requirements

Type them(Constraint, Function, Performance, Verification, I/F, …)

Add Rationale



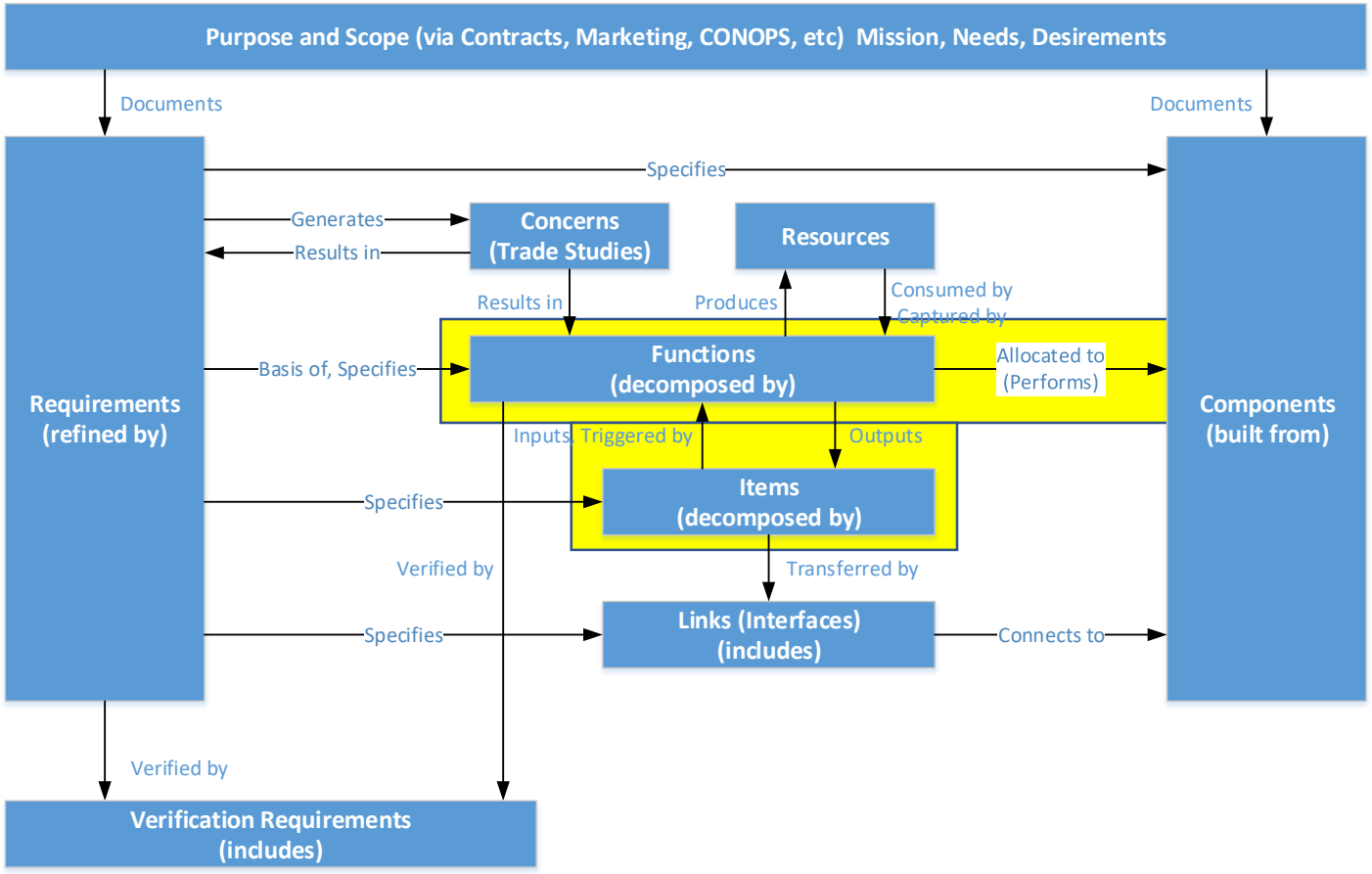
Refine and Relate the Requirements
Start Developing the Architectures



Requirements with Attributes and Relationships

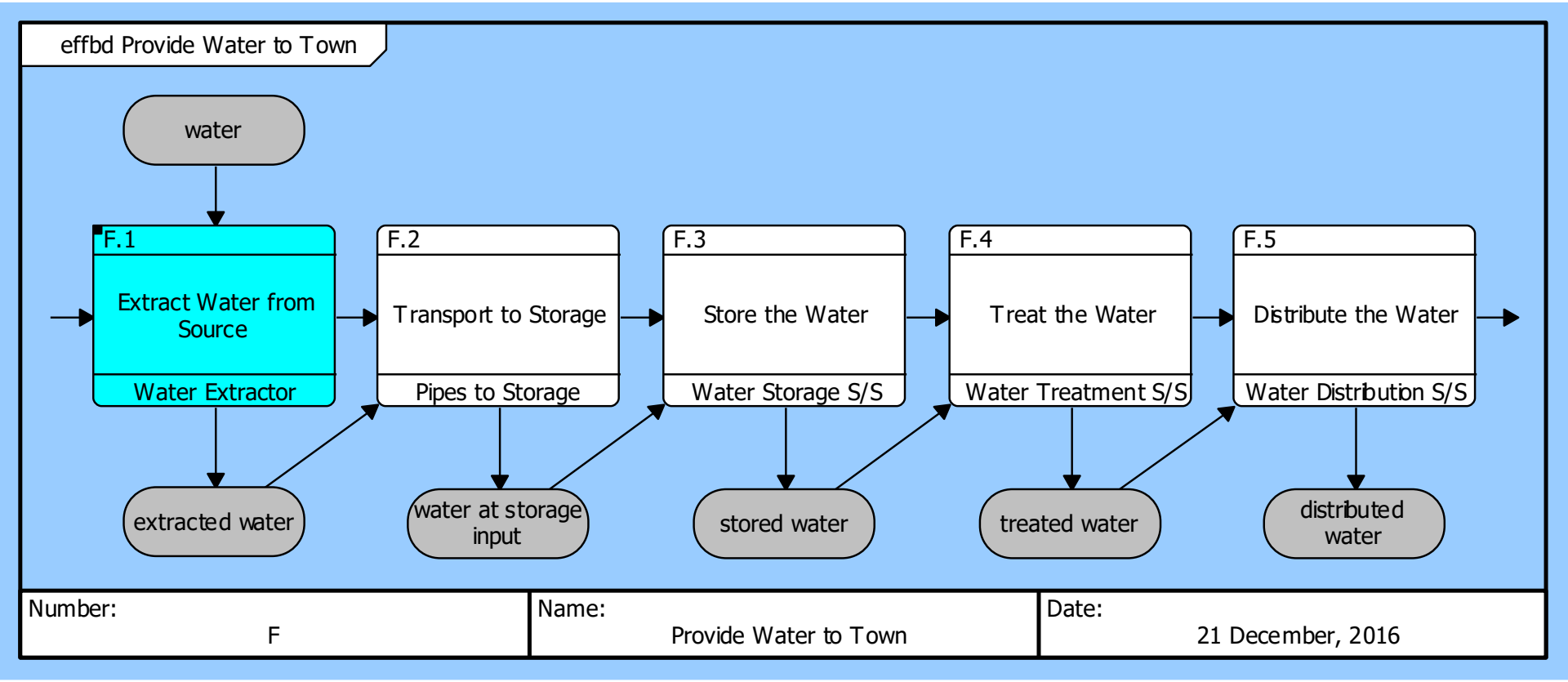
Number & Name	Description	Rationale	Type	basis of	categorized by	specifies
RSA Acquisition Subsystem Requirements (Seals)	A set of requirements fot the image acquisition subsystem in the seals enviornment.		Composite			
RSA.1 Acquisition Subsystem Environment (Seals)	The image acquisition subsystems shall operate in a marine environment, year-round, typical of the Gulf of Maine.	This is where the seals are to be studied. The Shoals Marine Lab is situated near the southern end of the Gulf of Maine.	Constraint		3.2.6 Environmental Conditions	X.4.1 Duck Island Equipment X.4.2.2 SML Antenna X.4.2.4 SML Radio X.4.2.5 SML Tower
RSA.2 Automatic Image Capture (Seals)	The image acquisition subsystem shall be capable of acquiring images without manual operator control.	The system is to work year round at all times of daylight corresponding to lower tide conditions. This also reduces operator labor costs.	Functional	S.1 Capture Seals Images		
RSA.3 Daylight Operation (Seals)	The acquisition system shall operate during daylight hours.	Nighttime operation is not necessary for project goals. Nighttime technology (IR) is not adequate or currently cost effective for identification.	Functional	S.1 Capture Seals Images		
RSA.4 Image Resolution (Seals)	Acquired images shall be of adequate resolution to identify individual seals.	Need to identify individual seals for: - eventual tracking when multiple sites are implemented - change in physical condition - behavioral analysis	Performance			S.1 Capture Seals Images IT.6 image at specific az & el IT.16 set of images
RSA.5 Lightning Protection (Seals)	The exterior system elements shall be protected from lighting.	Minimizes the risk of lightning strikes and minimizes damage from lightning strikes.	Constraint		3.2.6 Environmental Conditions	X.4.1 Duck Island Equipment X.4.2 SML Site Equipment
RSA.6 Manual Image Capture (Seals)	The image acquisition subsystem shall be capable of acquiring images using manual operator control.	- Aides in the setup of the automatic image capture. - Allows for random, operator controlled viewing of the site	Functional	S.1.2 Acquire Set of Images (manual)		
RSA.7 Number of Images per Day	The number of images to collect shall be at least (tbd) per day.	The number of images to gather each day depends on: - area covered - turnover in seal population in given area - adequacy of identification per seal per image - biological research needs	Performance			S.1 Capture Seals Images
RSA.8 Remote Operation (Seals)	The acquisition system shall be operated by remote control.	Humans at the camera location near the seals would disturb the seal population and impact the results. Access to the viewing locations can be difficult and may be restricted by permit conditions.	Functional	S.1 Capture Seals Images		
Harbor Seals	The Seals shall be Harbor Seal species.	This is the seals species that have spots for identification.	Constraint		3.2.1 Physical Characteristics	X.2 DI Seals

Develop the Functional Architecture (Behavior) (System Level)

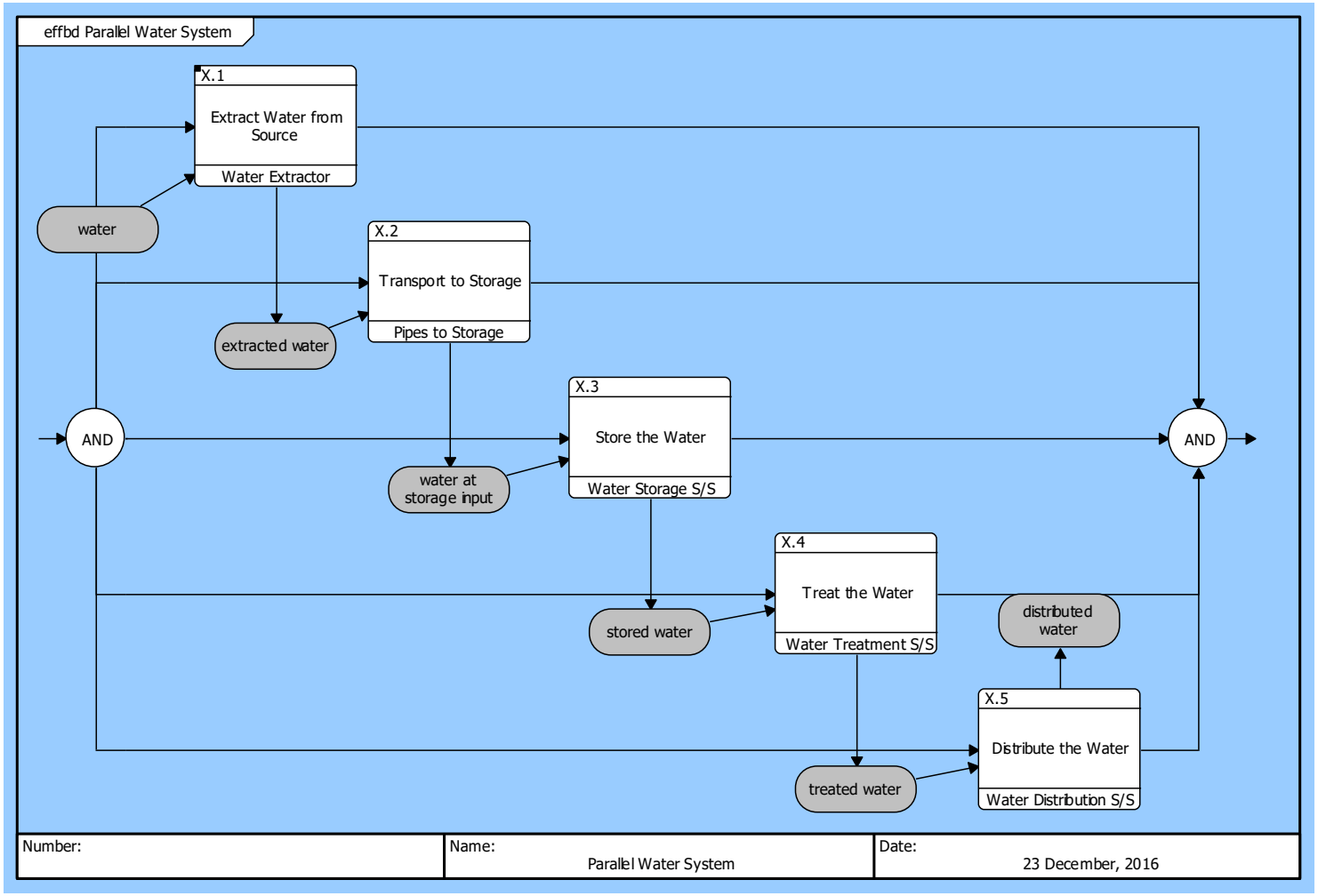


Define the Functional Behavior

Items are critical to well defined behavior



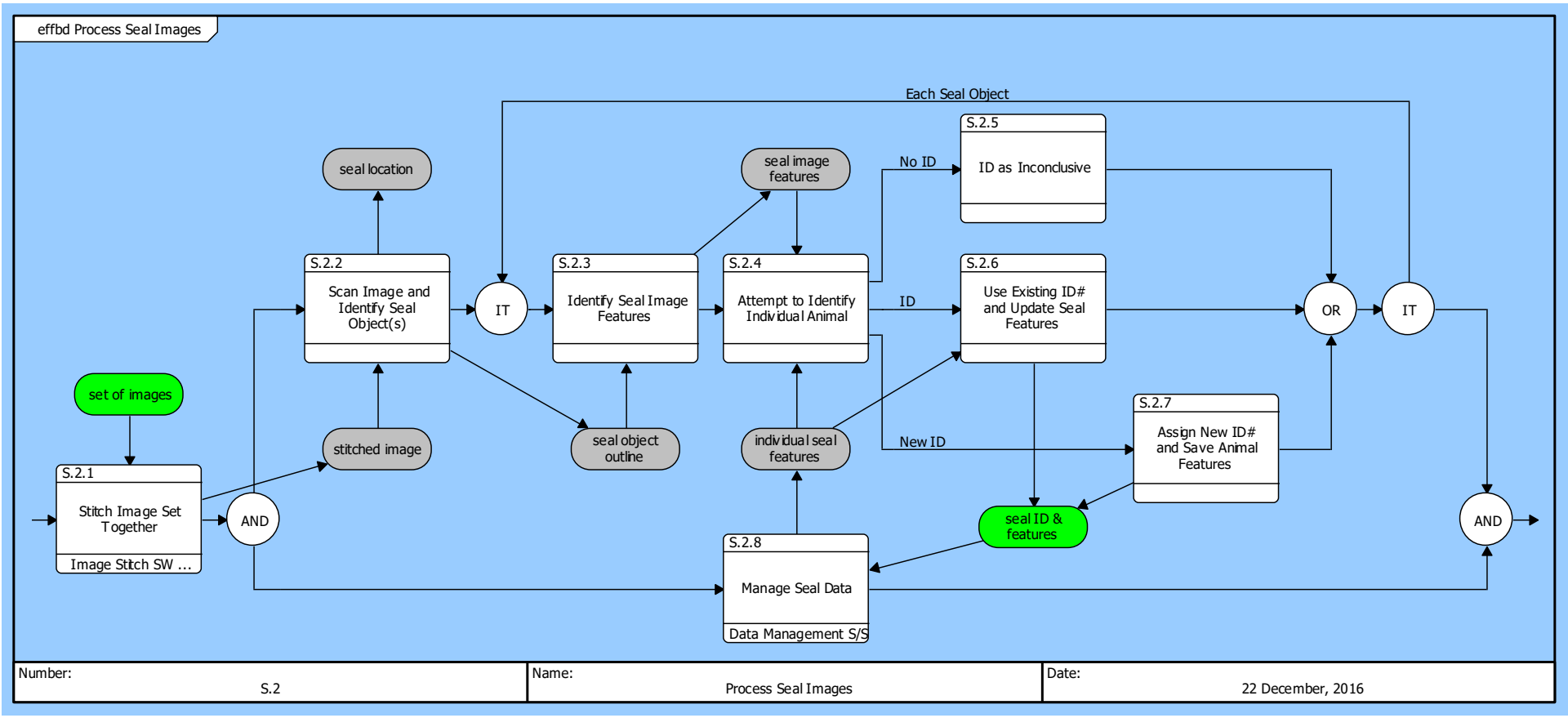
A Parallel Representation



Who is your audience? What is the purpose of the representation?

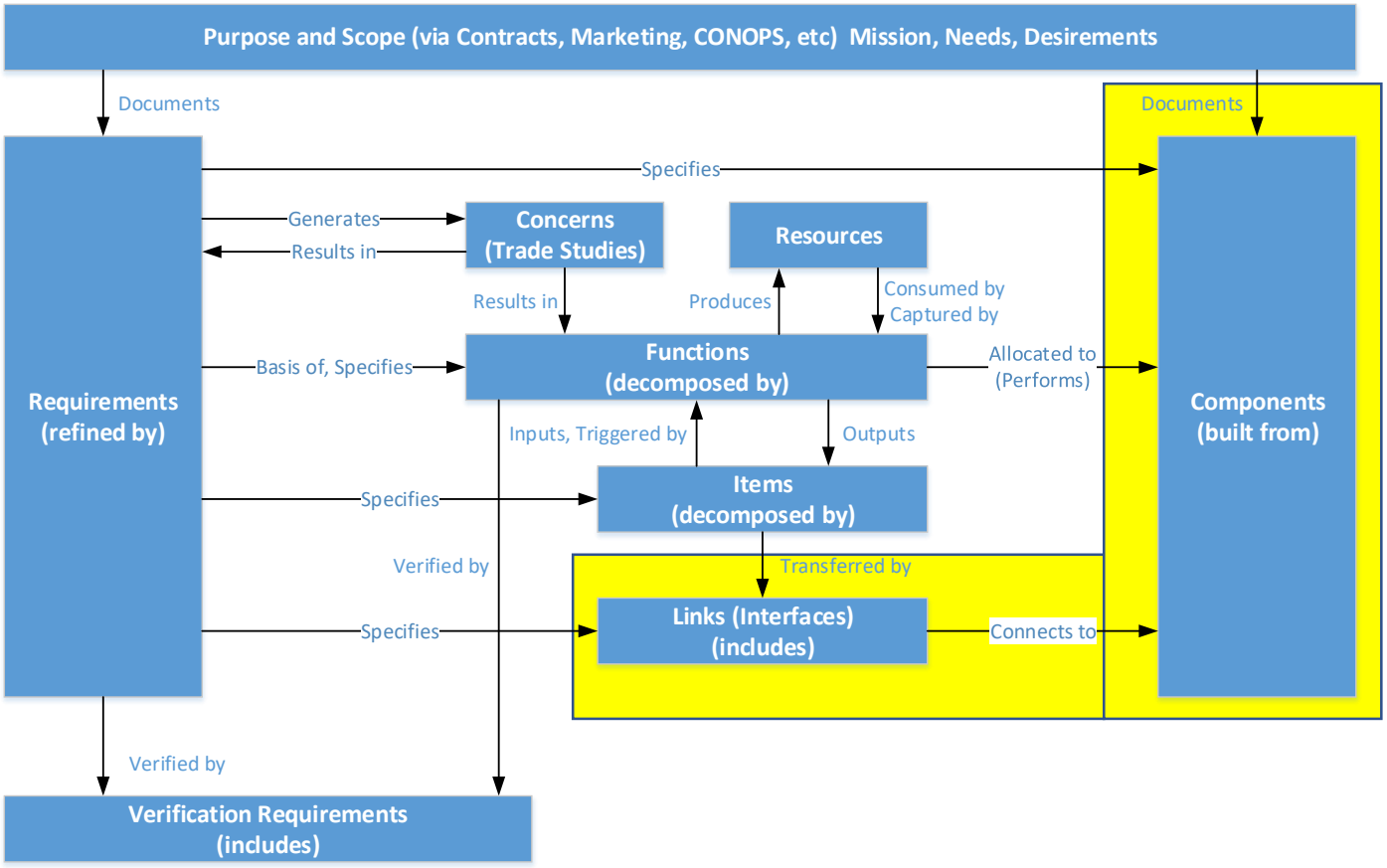
- Audience – understand the flow?
- Simulation – should be functionally correct

Lower Level, More Complex Functional Behavior

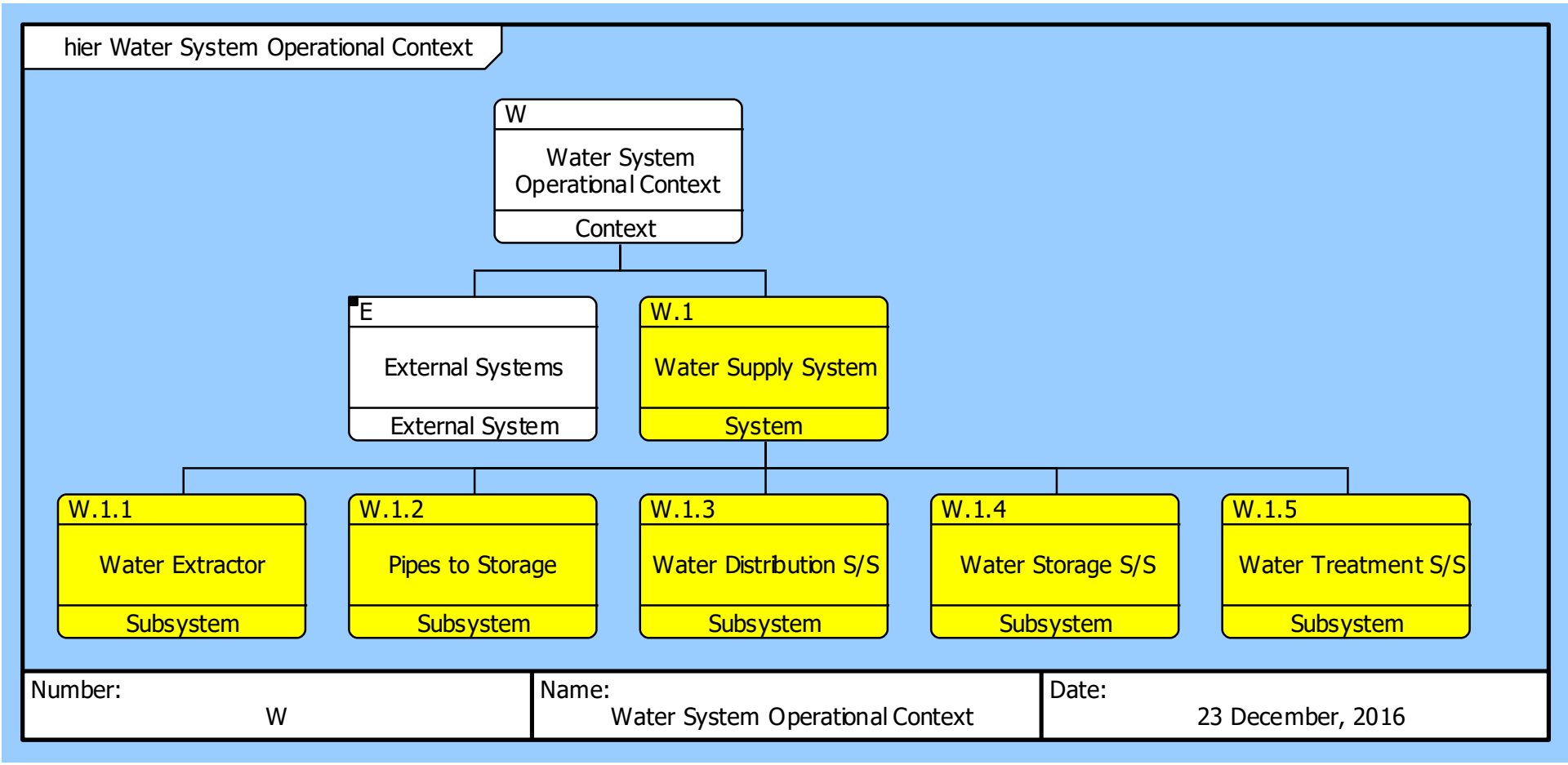


Adds parallel logic (AND)
Adds iteration (IT)
Adds multiple exits to a function

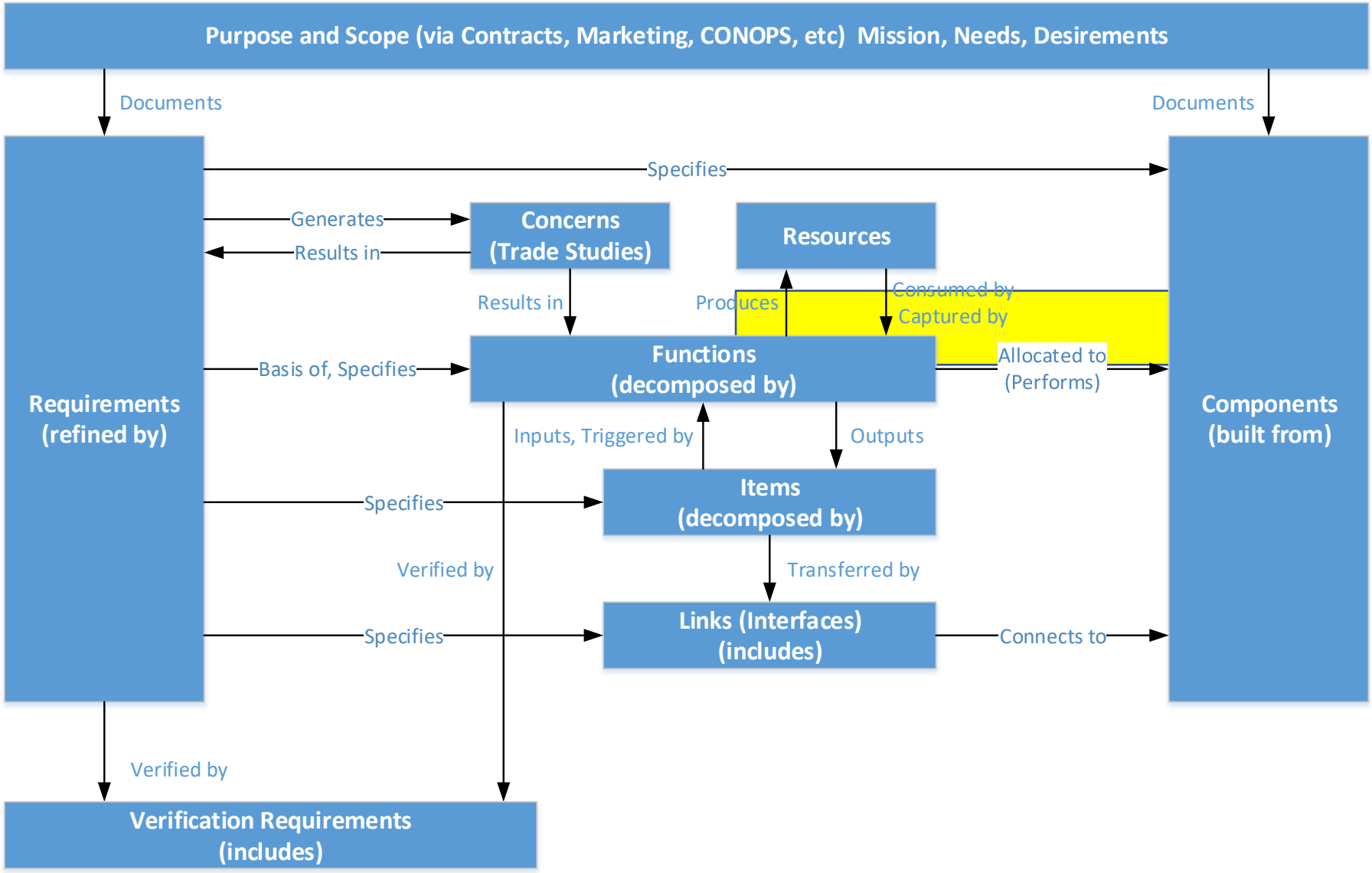
Develop the Physical Architecture



Define the System Components



Allocate the Functional Behavior



Functions Allocated to Components

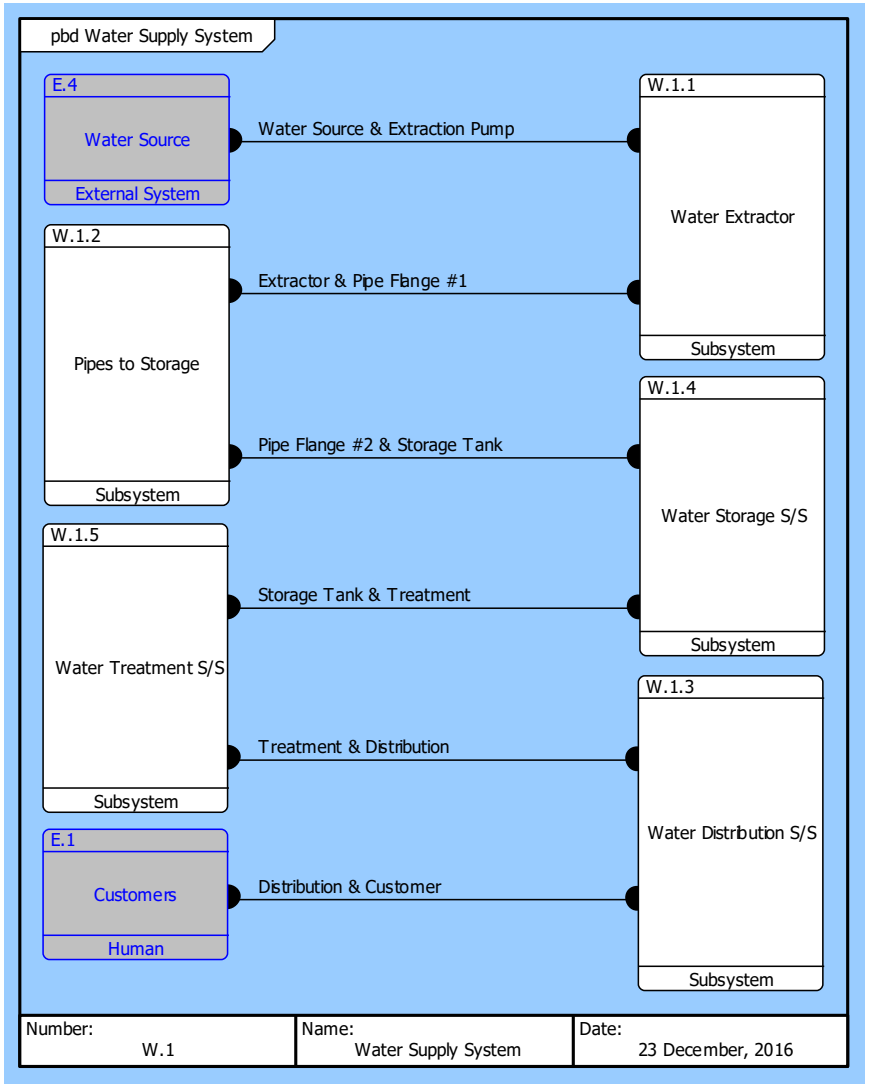
Number & Name	Description	allocated to
S · · Capture, Process, and Analyze Data (Seals)	Capture images of seals in their natural habitat. · · Process the images to obtain time, location, individual seal identification and features. · · Analyze the individual seal data to provide biological information such as behavior, migration, spread of disease, etc.	X Seals at SML Operational Context
S.1 · · Capture Seals Images	Capture images of areas inhabited by seals.	X.4 Seal Image Acq System
S.1.1 · · Acquire Set of Images (automatic)	Capture the seal images in an automatic fashion. · · Pre-programmed date, time, pointing location, and optical zoom controls the automated operation.	
S.1.1.1 · · Acquire Specific Az & El Image	Capture an image at a specific azimuth and elevation for the pointed camera.	
S.1.2 · · Acquire Set of Images (manual)	Capture images under human control. · · Control includes azimuth, elevation, optical zoom, and capture triggering.	
S.2 · · Process Seal Images	Process the sets of images to obtain data on seal population and individual seal identification and activity.	X.1.3 Image Processing S/S
S.2.1 · · Stitch Image Set Together	Stich the images together, where appropriate, so that objects (seals) crossing image edges are combined and image analysis can proceed without boundary condition problems.	X.1.3.1 Image Stitch SW Module
S.2.2 · · Scan Image and Identify Seal Object(s)	Scan the image with appropriate algorithms to identify possible seal objects.	
S.2.3 · · Identify Seal Image Features	Analyze possible seal objects and identify specific seal features.	
S.2.4 · · Attempt to Identify Individual Animal	Use the seal features to attempt to identify an individual seal.	
S.2.5 · · ID as Inconclusive	If the seal object cannot be clearly identified as an individual seal, mark as inconclusive.	
S.2.6 · · Use Existing ID# and Update Seal Features	If an individual seal has been identified, update the individual seal identification history and features.	
S.2.7 · · Assign New ID# and Save Animal Features	If a seal is not individually identified, but is adequate to identify it as a seal, then add the seal and its features to the identified seal database.	
S.2.8 · · Manage Seal Data	Manage the seal database and its interactions.	X.1.2 Data Management S/S
S.3 · · Analyze Seal Data	Analyze the seal data (time, location, identification, characteristics) to obtain: - · local location behavior of individual seals - · migration behavior of individual seals - · identification of disease (eg, seal pox) - · migration and transmission of disease	X.1.1 Data Analysis S/S

Define the Links (view as “PBD”)

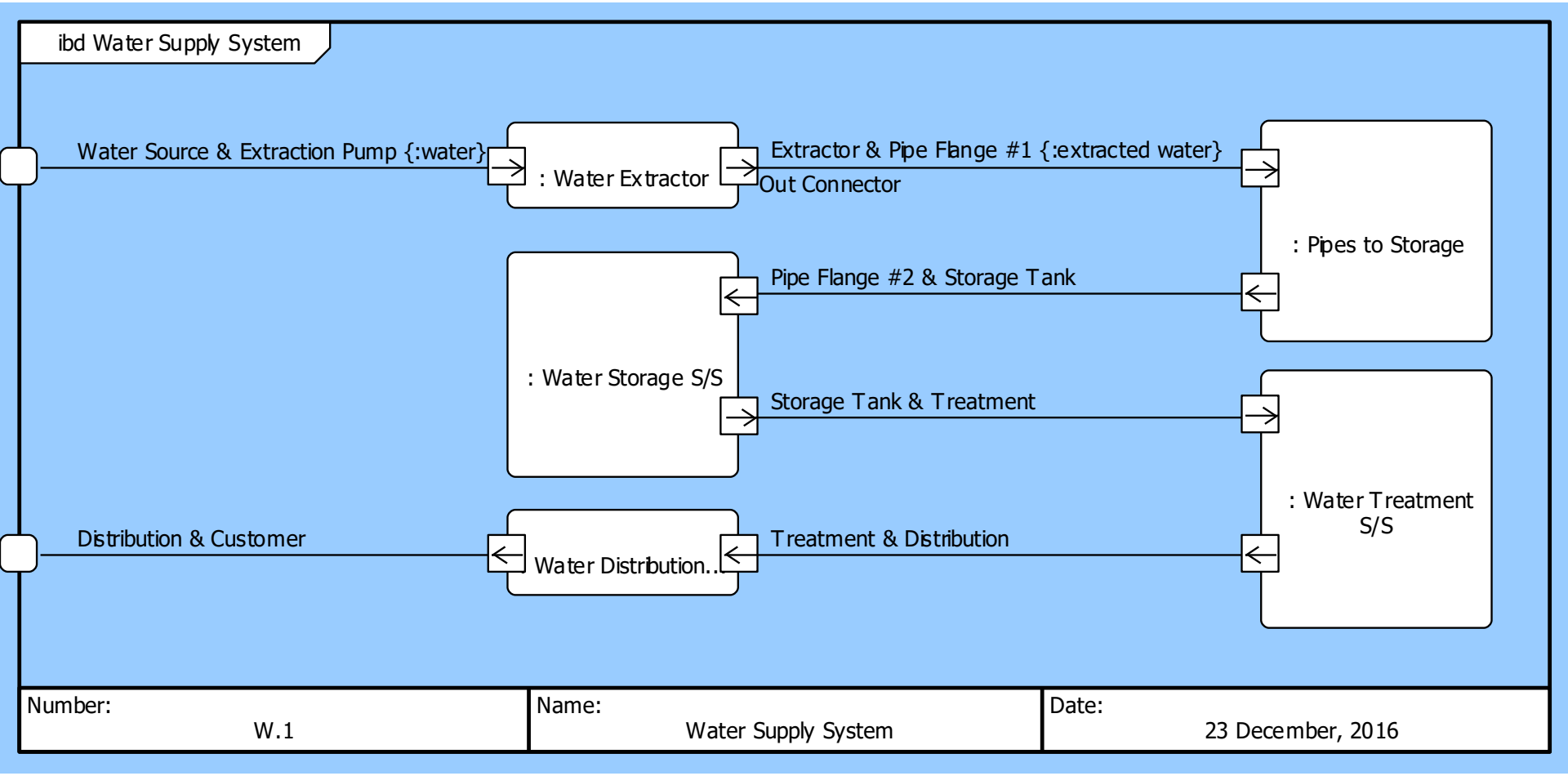
(the external-to-view components are gray)

Interfaces are the higher level, more abstract connections between components.

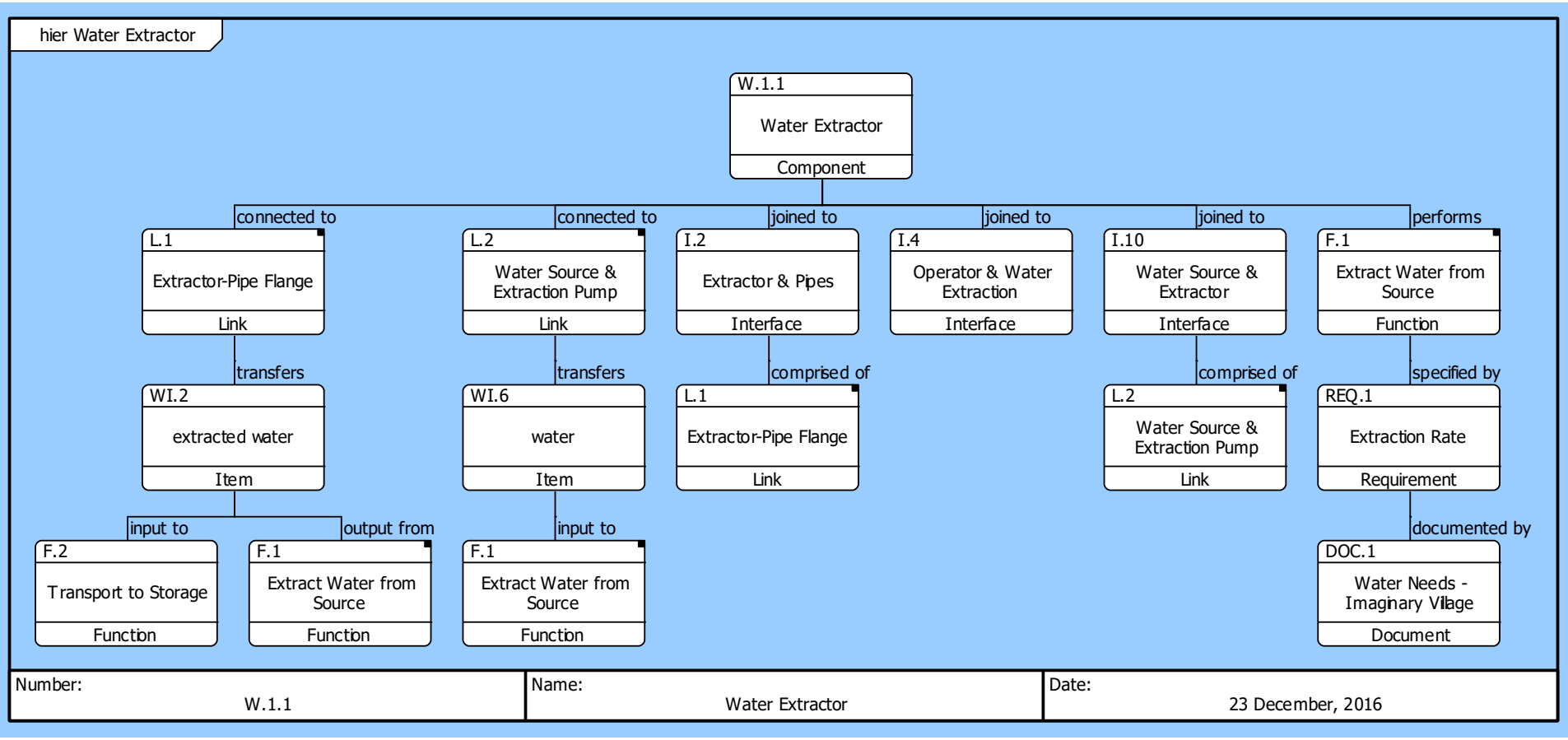
Links “comprise” Interfaces and are the more detailed connections between components.



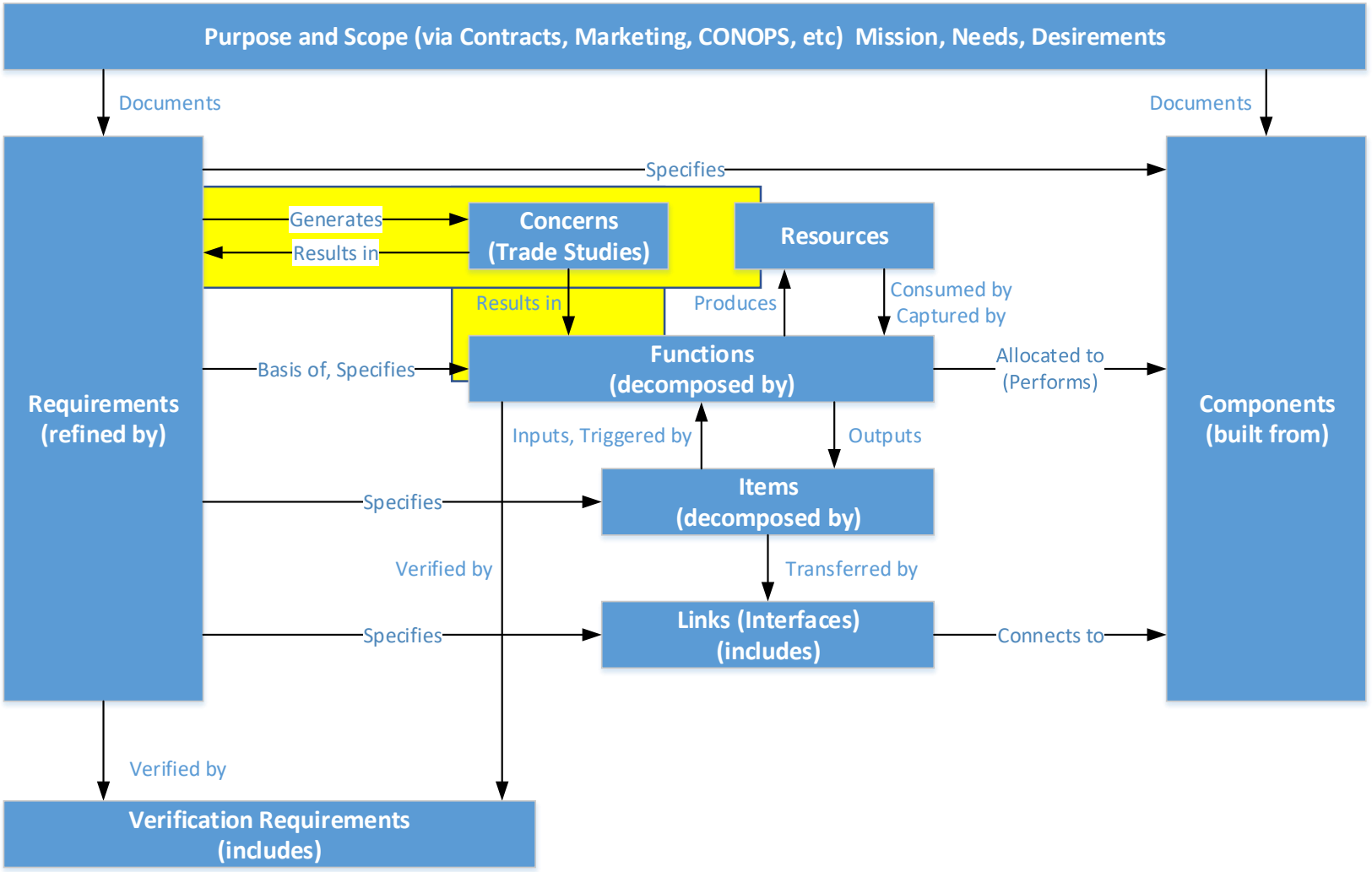
View Link “PBD” as a SysML “Flow IBD” Diagram



Traceability – with Selected Relationships



Develop the Next Layer
Concerns = Issues & Decisions / Trade Studies



Issues and Decisions (aka Concern)

generated by	Number & Name	Description	Alternatives	Assumptions	results in
RSA.4 Image Resolution (Seals)	Define DI Tower & Imaging Configuration	Define: - tower location and height on Duck Island. Must have line of sight to Appledore tower and areas of interest on Duck Island. - image pointing (az & el) and range.		- tower and camera on Duck Island - seals on Shagg Island segment in view of Duck Island	
RSA.4 Image Resolution (Seals)	Define Equipment Requirements for Image Resolution	Acquired images must be adequate to identify individual seals. An important input to this is what spatial resolution is required to capture the individual markings that aid in that identification. Other inputs are the geography of the site to be imaged. Outputs should include camera pixel resolution and optical zoom capabilities.			
S Capture, Process, and Analyze Data (Seals)	Generic Issue for Program Activities				
S Capture, Process, and Analyze Data (Seals)	How to capture images	Images must be captured of the seals on a nearby island. Initial technique has been to take a boat over to the island and take photos with a 500mm lens. Desire is to take more photos, more often, without the Wx and sea condition constraints.	- use boat with manual capture with telephoto lens. - tower with camera mounted and remote control - a number of smaller towers with cameras		S.1.1 Acquire Set of Images (automatic) S.1.2 Acquire Set of Images (manual)
RSA Acquisition Subsystem Requirements (Seals)	Video Capture				

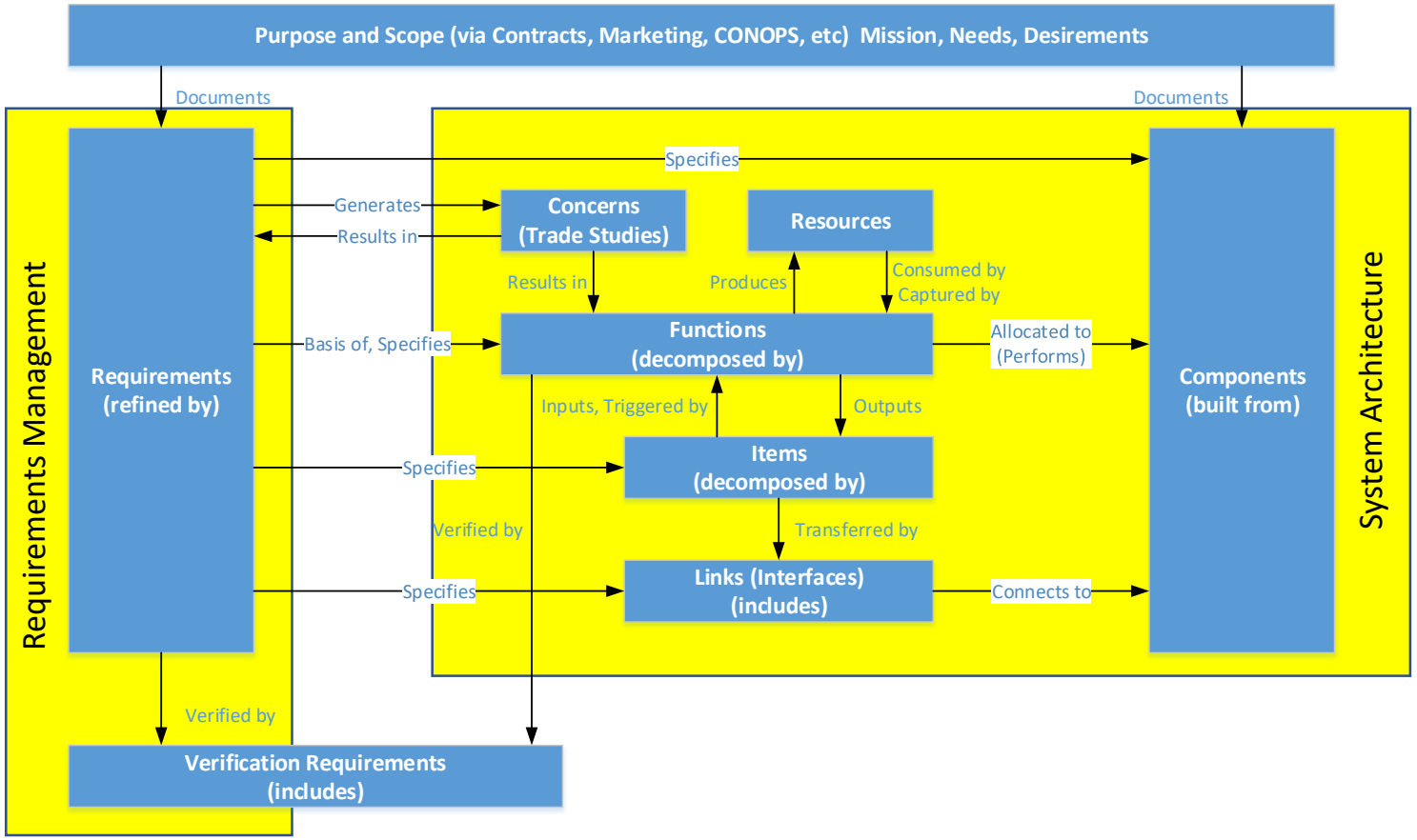
The Model vs Views

- The MODEL is a database that contains:
  - All the ERA data (Entity – Relationship – Attribute)
  - Logic and layouts of the produced diagram views

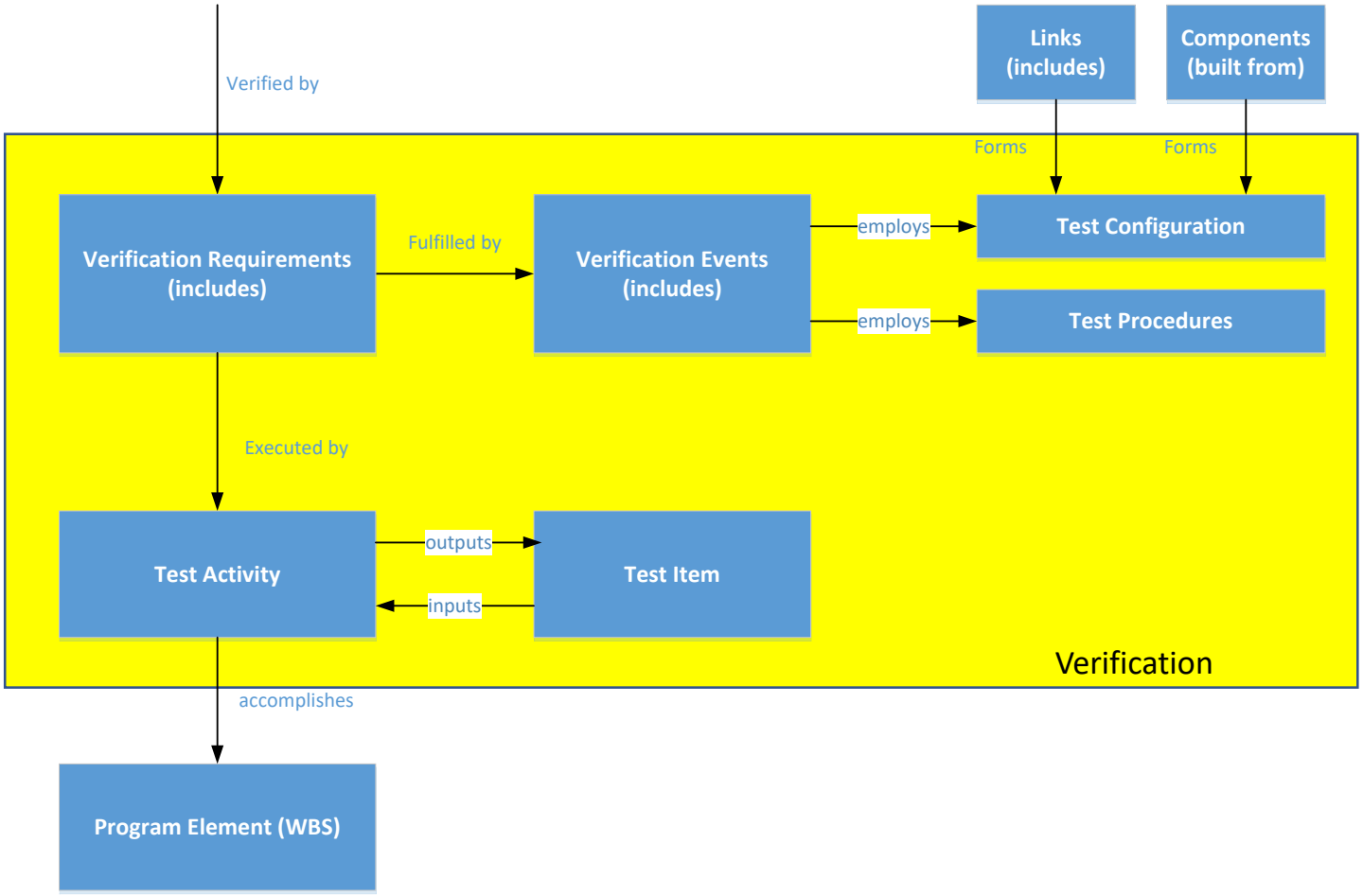
- The VIEWS:
  - Produced from the database
  - Can update the database from the views
  - Traditional system engineering views
  - SysML views
  - An update to one view is reflected in all views

- The INFORMATION STRUCTURES (aka SCHEMAS):
  - Requirements Management and System Architecture
  - Verification
  - Program Management
  - Concerns and Risk
  - Operational Architecture (DoDAF Department of Defense Architectural Framework)

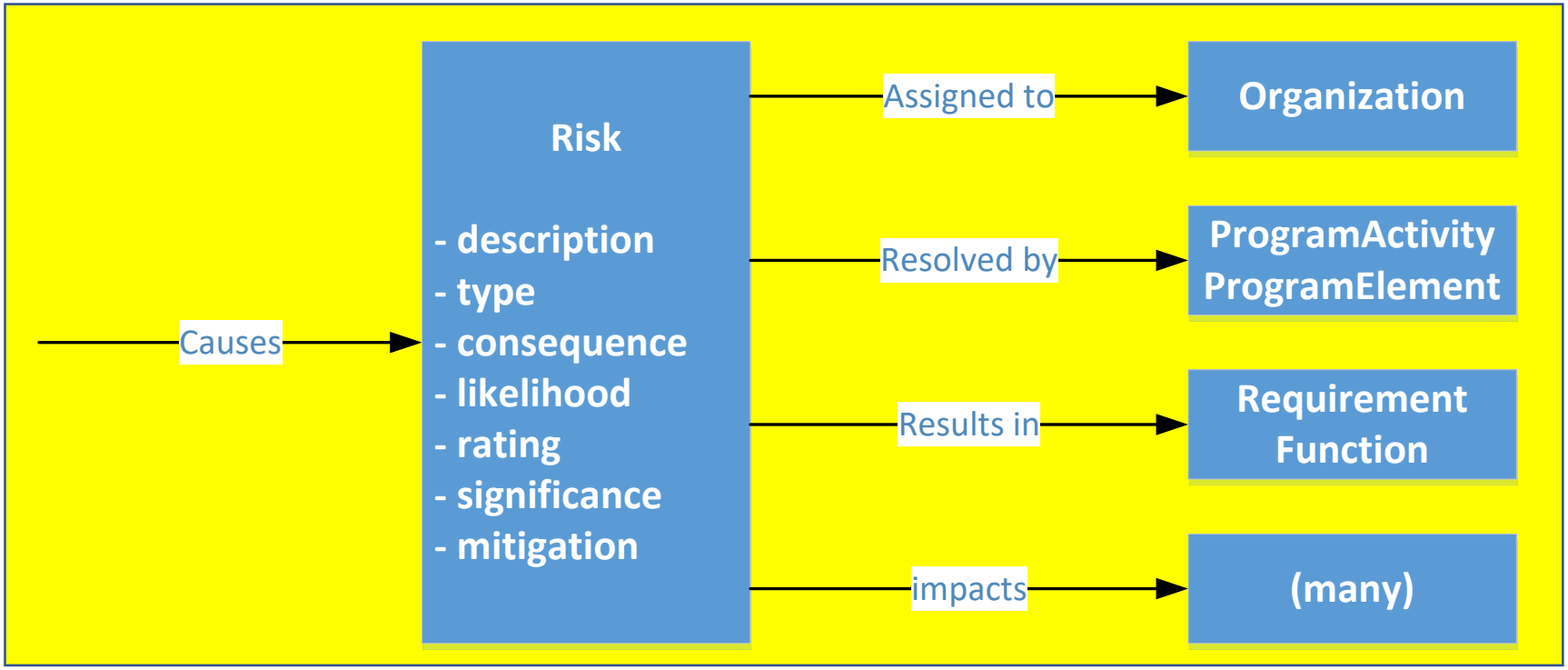
Requirement Management and Systems Architecture Information Structure



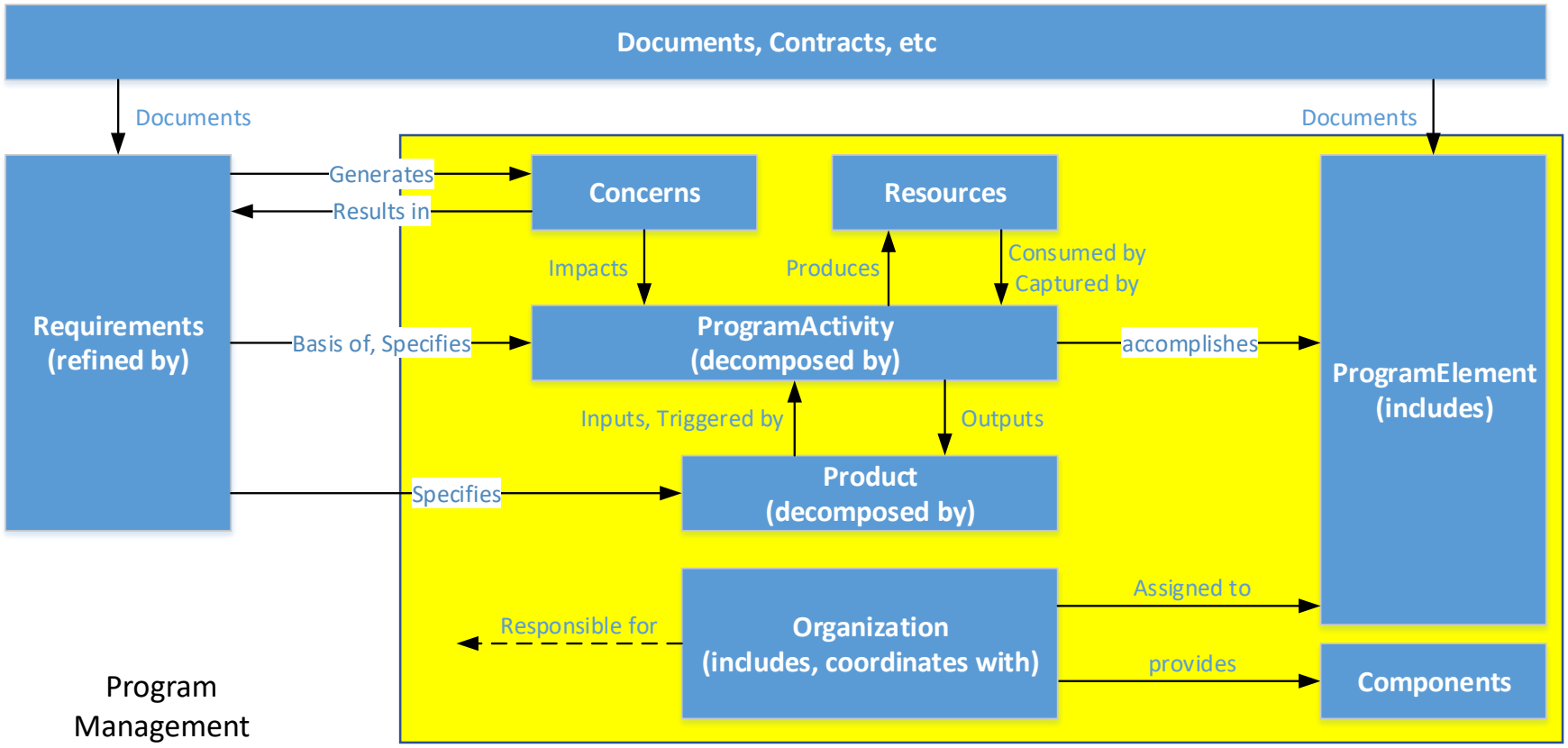
The Verification Information Structure



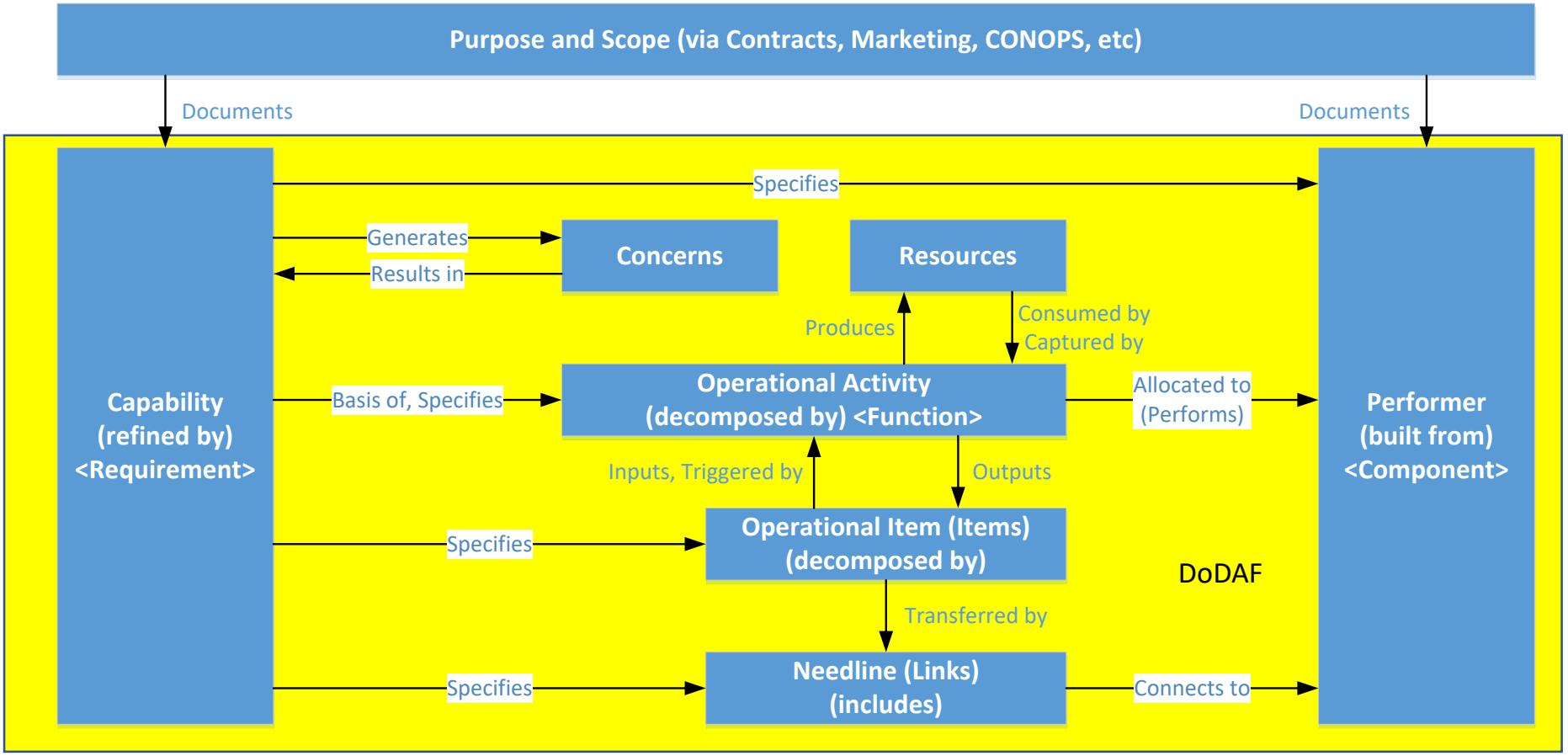
The Risk Information Structure



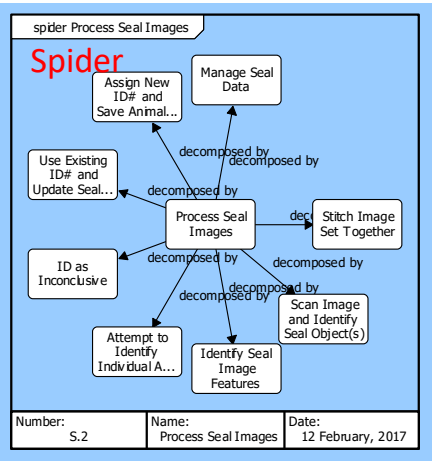
The Program Management Information Structure

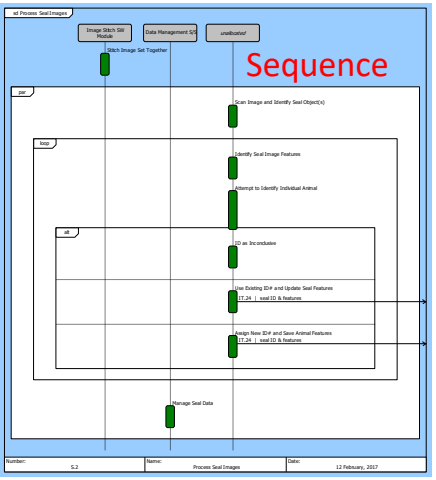


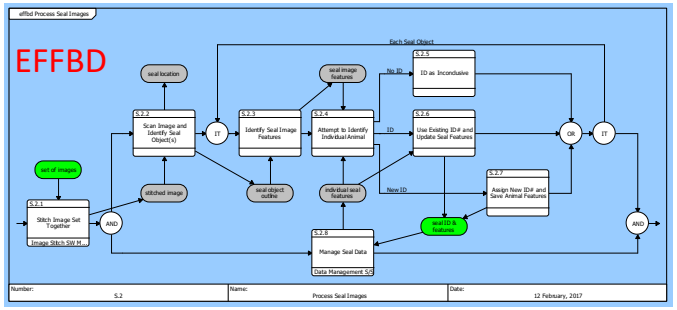
The Operational Architecture (aka DoDAF) Information Structure

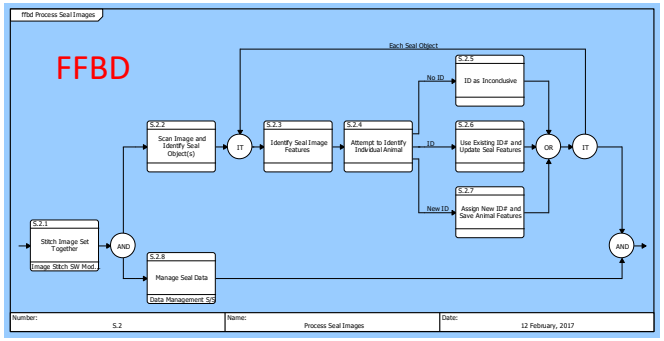


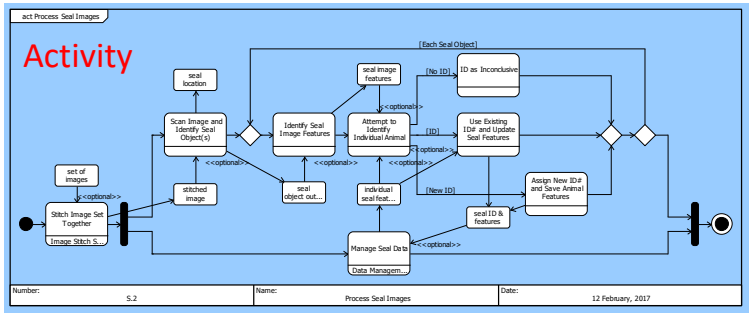
Many Views Available – Function Example

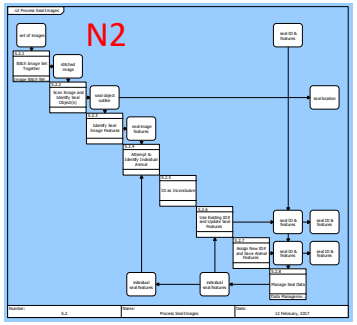


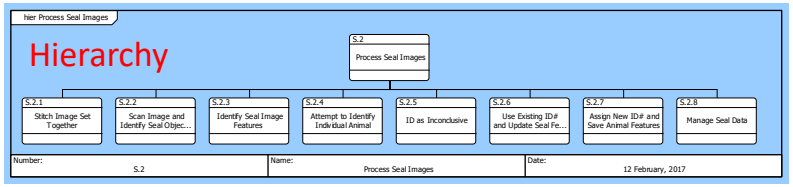




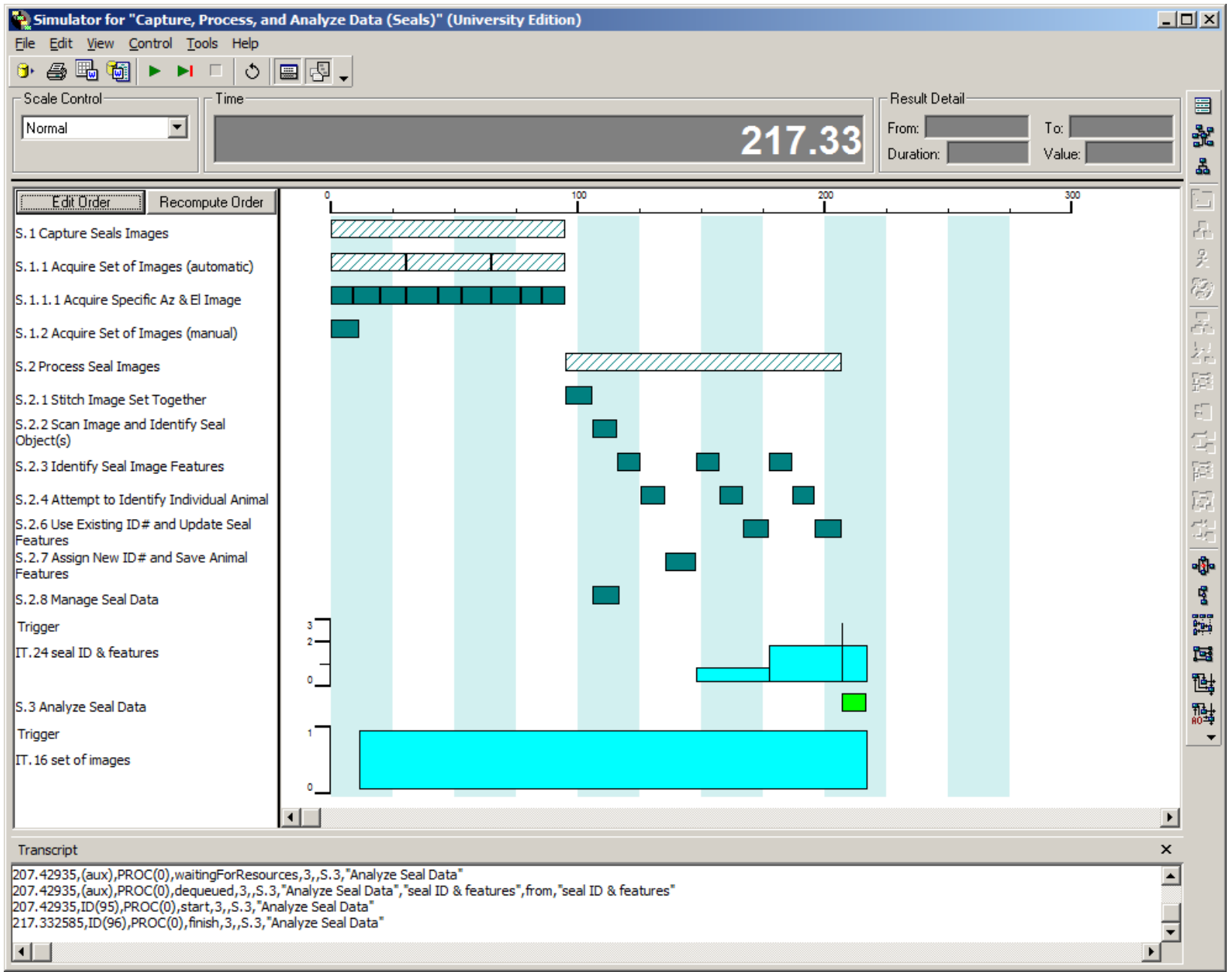








Discrete Simulation with the Tool



Guidelines / Advice

- Key Diagrams:
  - Hierarchy
  - Functional Behavior – EFFBD/Activity
  - Interface
- EFFBD/Activity/Hierarchy – No more than 9 equivalent blocks wide
  - Can it be easily read when printed in portrait mode?
- Requirements – use the Rationale attribute
- Keep connecting lines, elements, items aligned and straight
  - This makes a big difference in clarity of the diagram

Some Benefits of Learning and Using an SE Tool

- Learn and Apply System Engineering Skills and Techniques
  - How to approach and structure complex and larger projects
- Project Impact
  - Keep track of Requirements, Physical and Functional Architectures, Interfaces, Trade Studies, Verification (testing)
  - Communicate with Team members for a consistent understanding (be on the same page)
  - Produce documentation from the database (consistency)
- Individual Benefits
  - Know the whole system – how it works and goes together
  - An important step to managing a section or a whole project
  - Makes for a much better overall engineer or manager

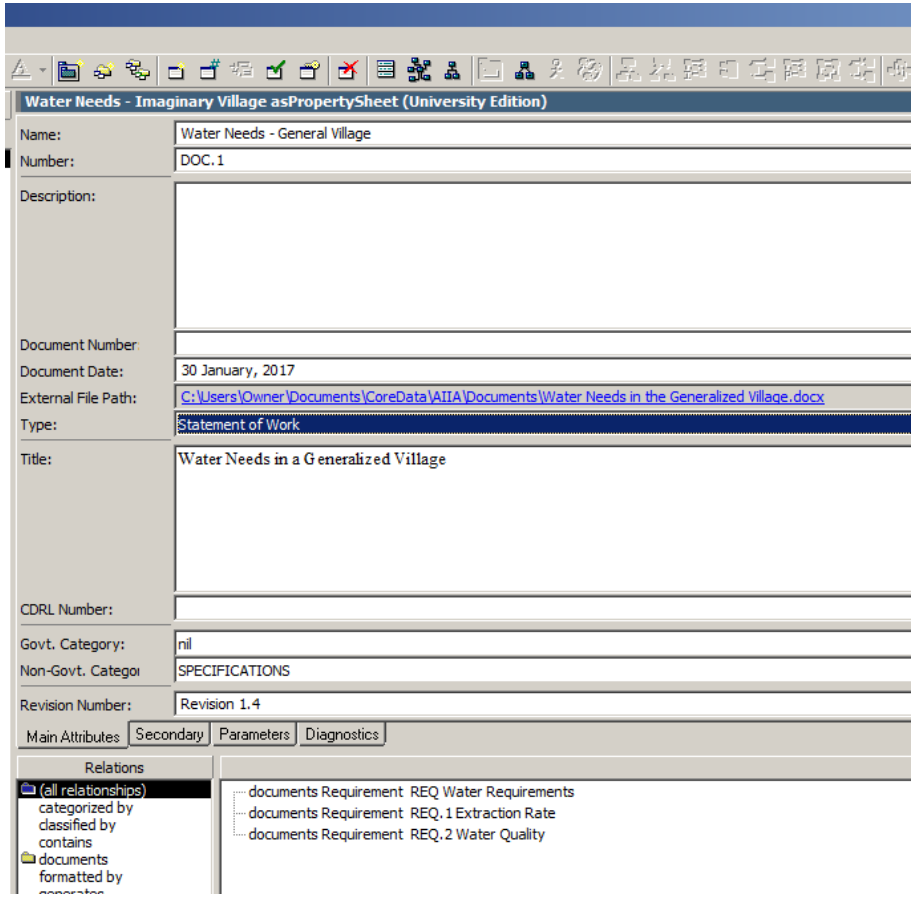
The End

Key Classes and their Attributes

- All Classes - Name, Number, Description, Audit Log, Parameters
- Requirements - type, rationale, …
- Function - Duration, …
- Items - type, range, units, size, priority, accuracy, fields, …
- Components - type, cost, purpose, operations, receptions, …
- Concerns - assumptions, alternatives, decision, rationale, …
- Risk - Mitigation Plan, type, Consequence, Likelihood, ..
- Links - capacity, delay, protocol, …
- VerificationRequirements - status, method, level, …
- …
- State, Mode, Resource, Transition, UseCase, VerificationEvent, …

Many Supporting Classes: Document, ExternalFile, Text, …

- Document Attributes
  - Name and Number
  - Description, Document Number and Date
  - File Path
  - Type, Title, Category, Rev#
  - External Text or Graphics
- Document Key Relationships
  - “Documents”
  - “References” another document
- ExternalFile, Text
  - “augments” another element – This allows inclusion of external text or graphics into a report



Start-Up Information

- A primer for Model-Based Systems Engineering
  - http://www.vitechcorp.com/downloads/index.shtml
- Vitech's Resource Library

Where to Start?

- The order depends on the nature of the project and what things are known.
  - System context and top level components
  - Requirements
  - Components
  - Links (Interfaces)
  - Verification Requirements and Testing
  - Functional Behavior

Set the System Context and Top-Level Components

- Select the COMPONENT Class
- Add component: "*project* Operational Context", Type = context
- Add lower level components (built from) "*project* system" and "Externals"
- Under Externals, add external components
  - E.g., operator, environment, road, …
- Under system, add major sub-systems
- Detail to more levels (*built from* relationship)

Add Interfaces (“Links”) between Components

- The relationship is “Connects to”
- Connect to the lowest level connected component (can change later)
- Can do from:
  - Selecting Links, then “connects” components
  - Select the higher level component, show PBD physical block diagram and select the components, then the connects icon.

Add Requirements

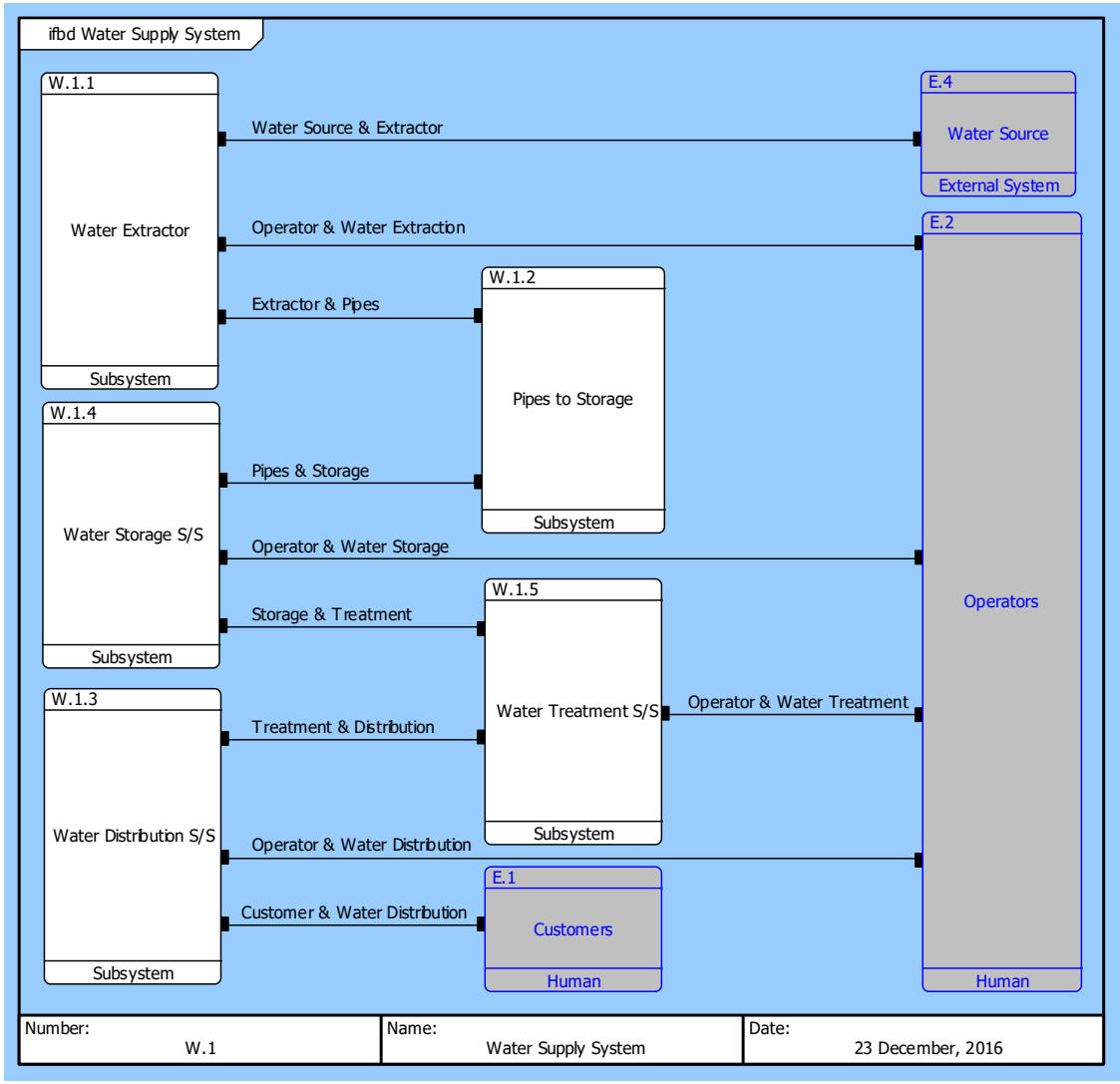
- Identify the requirements
- For each source (contest, business, university, …) make a folder
- Enter the requirements
  - Type them in, Starting with a name
  - Cut and paste from a document
  - Use *Home, Document Parser*
  - Use Excel connector
- Names - Many requirements are formatted as just paragraph statements. This tool needs a unique name for each requirement, so some prep for import may be needed.
- Set the type of requirement – composite, constraint, functional, performance, or verification
- If a constraint type, categorize by the type of constraint.

Define the Interfaces

(the external-to-view components are gray)

Interfaces are the higher level, more abstract connections between components.

Links “comprise” Interfaces and are the more detailed connections between components.



Components and Links Example