

# Rethinking the Integrative Foundation of Enterprises - The Activity Domain Theory

Lars Taxén

Linköping University

[www.neana.se](http://www.neana.se)

[lars.taxen@telia.com](mailto:lars.taxen@telia.com)



# My background



34 yrs

Methods, processes, IT, Information systems  
project management

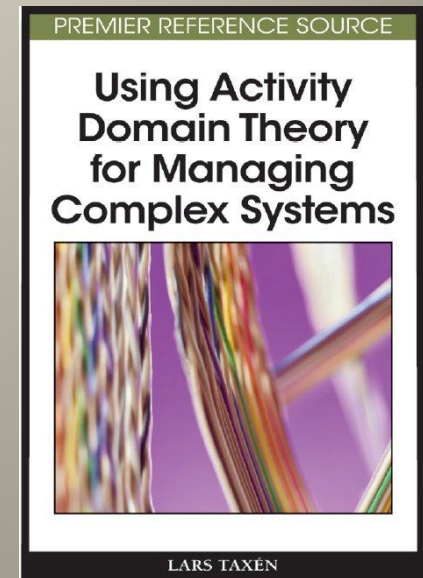


>15 yrs

Coordination & Integration  
PhD (2003), Associate prof. (2007)



> 10 yrs Consulting

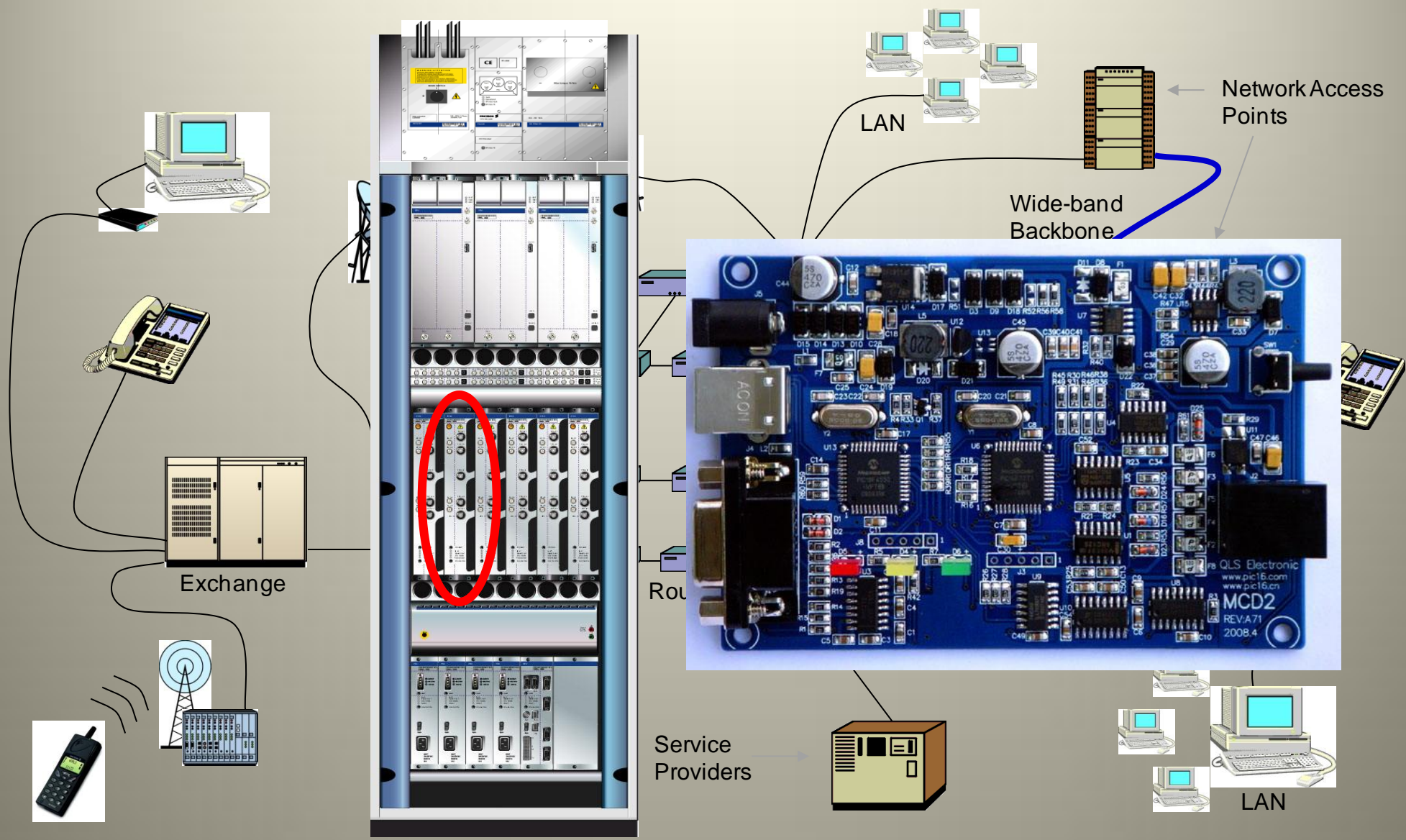


(2009)

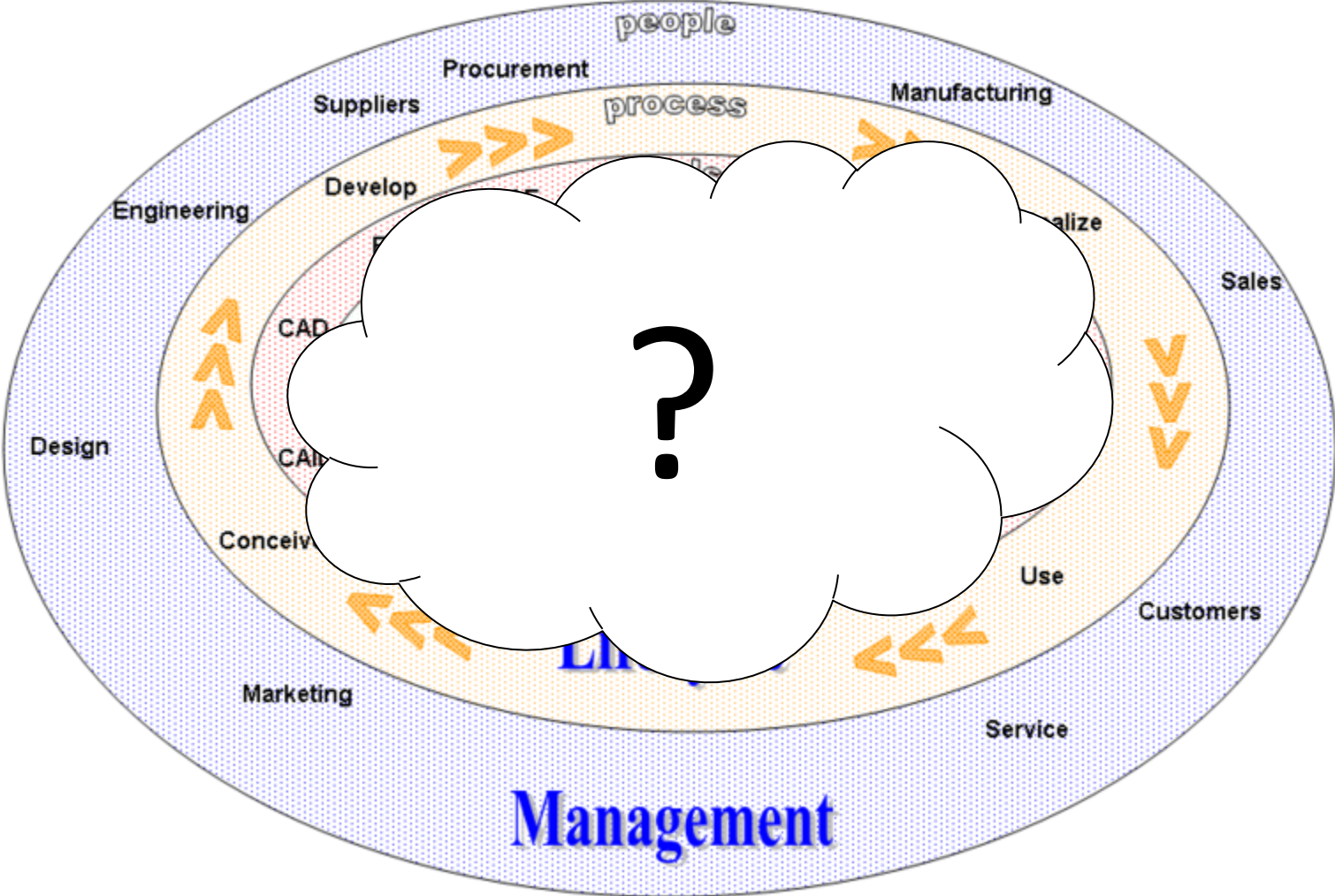
# Outline

- **My professional context**
- **Some existing integrational approaches**
- **The Activity Domain**
- **Integrating product development**
- **The System Anatomy**
- **The Enterprise Anatomy**
- **IT system implementation**
- **Some takeaways**

# **My professional context**

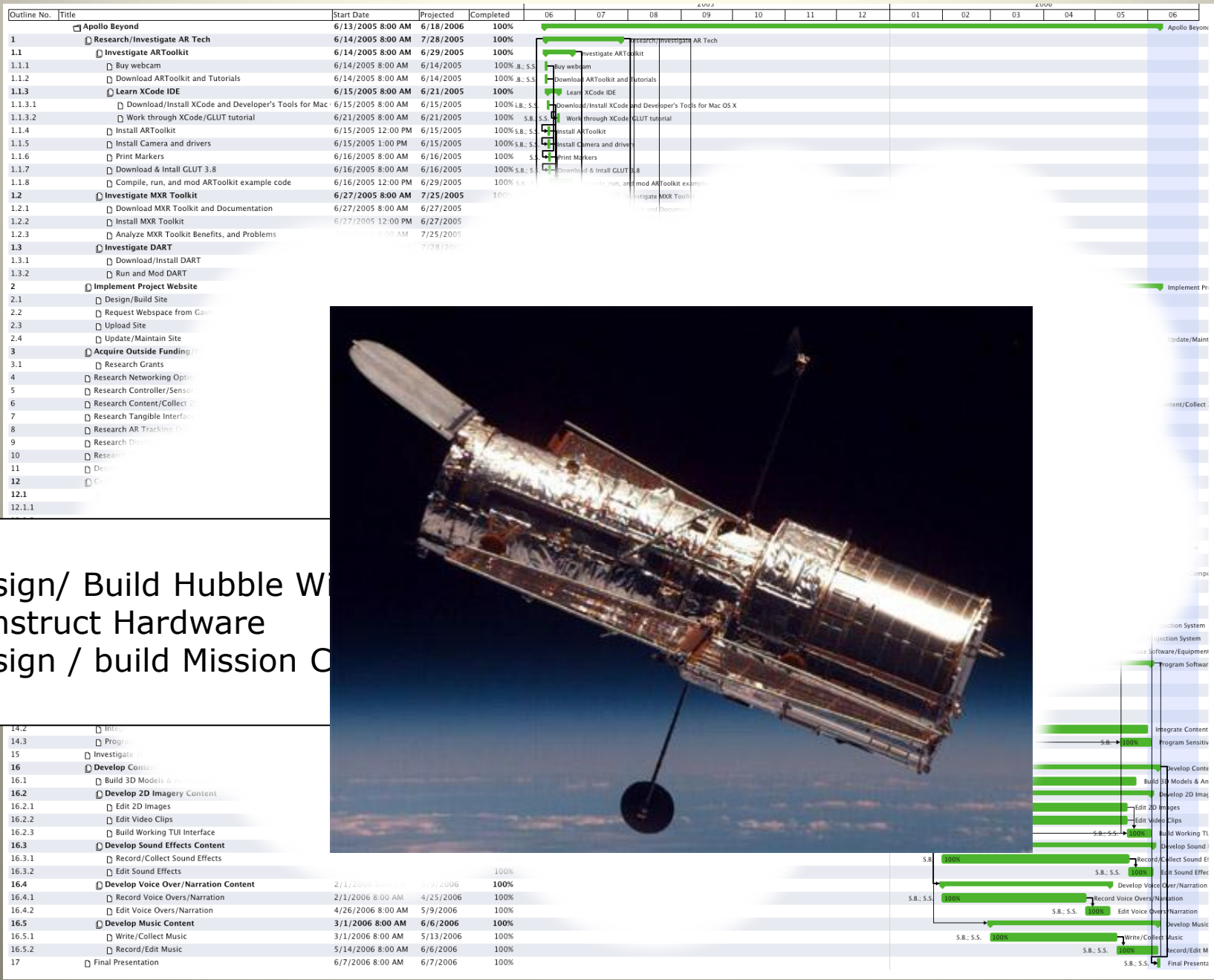


# Enterprise Integration!



Wikipedia Creative Commons

# Some existing integrational approaches



- ....
- Design/ Build Hubble Website
- Construct Hardware
- Design / build Mission Control
- ....





## Activities

- Task**  
A Task is a unit of work, the job to be performed. When marked with a symbol it indicates a Sub-Process, an activity that can be refined.
- Transaction**  
A Transaction is a set of activities that logically belong together; it might follow a specified transaction protocol.
- Event Sub-Process**  
An Event Sub-Process is placed into a Process Sub-Process. It is activated when its start event is triggered and can interrupt the main process context or run in parallel (non-interrupting) depending on the start event.
- Call Activity**  
A Call Activity is a wrapper for another Sub-Process or Task that is reused in a process.

### Activity Markers

Markers indicate execution behavior of activities:

- Sub-Process Marker
- Loop Marker
- Parallel MI Marker
- Sequential MI Marker
- Ad Hoc Marker
- Compensation Marker

### Task Types

Types specify the action to be performed:

- Sequence Flow**  
defines the execution order of activities.
- Default Flow**  
is the default flow to be chosen if no other conditions evaluate to true.

## Conversations

A **Conversation** defines a set of logically related message exchanges. When marked with a symbol it indicates a Sub-Conversation, a compound conversation element.

A **Conversation Link** connects a Conversation and Participants.

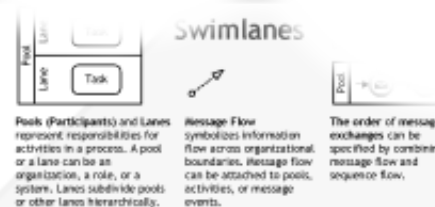
## Choreographies



## Gateways

- Exclusive Gateway**  
When splitting, it routes the outgoing flow to one of the outgoing branches. When merging, it waits for all incoming branches to complete before triggering the outgoing flow.
- Event-based Gateway**  
It is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.
- Parallel Gateway**  
When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.
- Inclusive Gateway**  
When splitting, one or more branches are activated. All active incoming branches must complete before merging.
- Exclusive Event-based Gateway (Inclusive)**  
Each occurrence of a subsequent event starts a new process instance.
- Complex Gateway**  
Complex merging and branching behavior that is not captured by other gateways.
- Parallel Event-based Gateway (Inclusive)**  
The occurrence of all subsequent events starts a new process instance.

## Swimlanes



**Pool (Participants) and Lanes** represent responsibilities for activities in a process. A pool or a lane can be an organization, a role, or a system. Lanes subdivide pools or other lanes hierarchically.

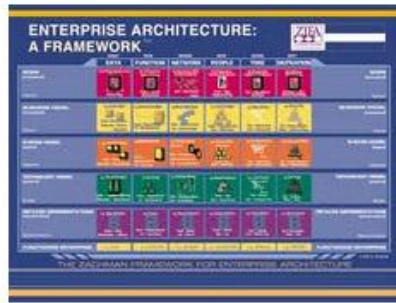
**Message Flow** symbolizes information flow across organizational boundaries. Message flow can be attached to pools, activities, or message events.

The order of message exchanges can be specified by combining message flow and sequence flow.

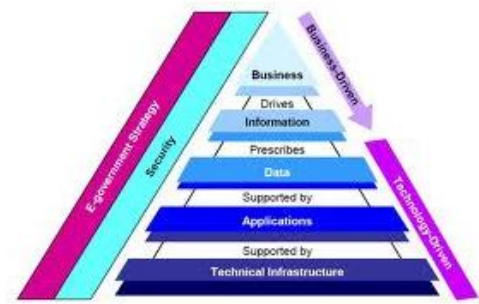
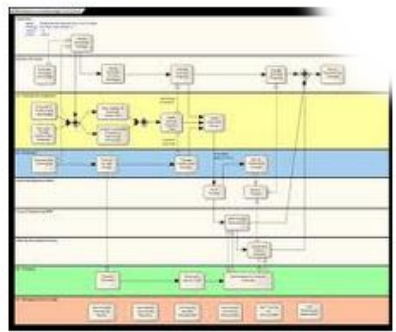
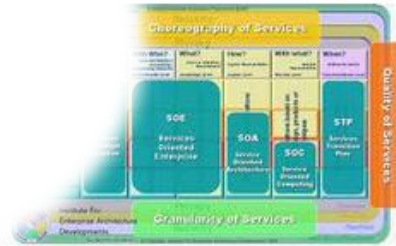
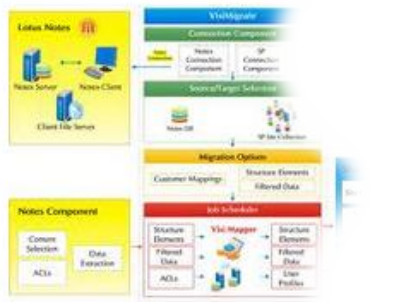
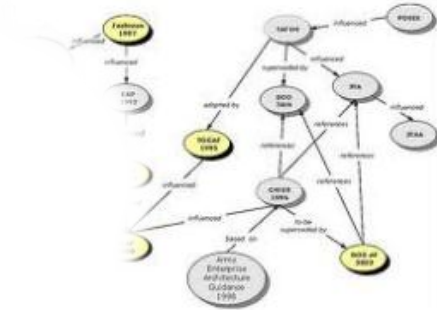
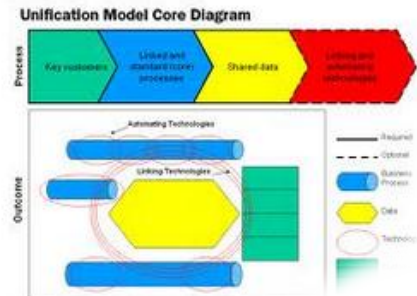
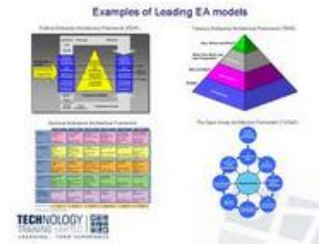
Log-Level	Start	Intermediate	End
Event Sub-Process Interrupting			
Event Sub-Process Non-Interrupting			
Catching			
Boundary Interrupting			
Boundary Non-Interrupting			
Throwing			

- Data Input** is an external input for the entire process. It can be read by an activity.
- Data Output** is a variable available as result of the entire process.
- Data Object** represents information flowing through the process, such as business documents, e-mails, or letters.
- Collection Data Object** represents a collection of information, e.g., a list of order items.
- Data Store** is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.
- Message** is used to depict the contents of a communication between two Participants.





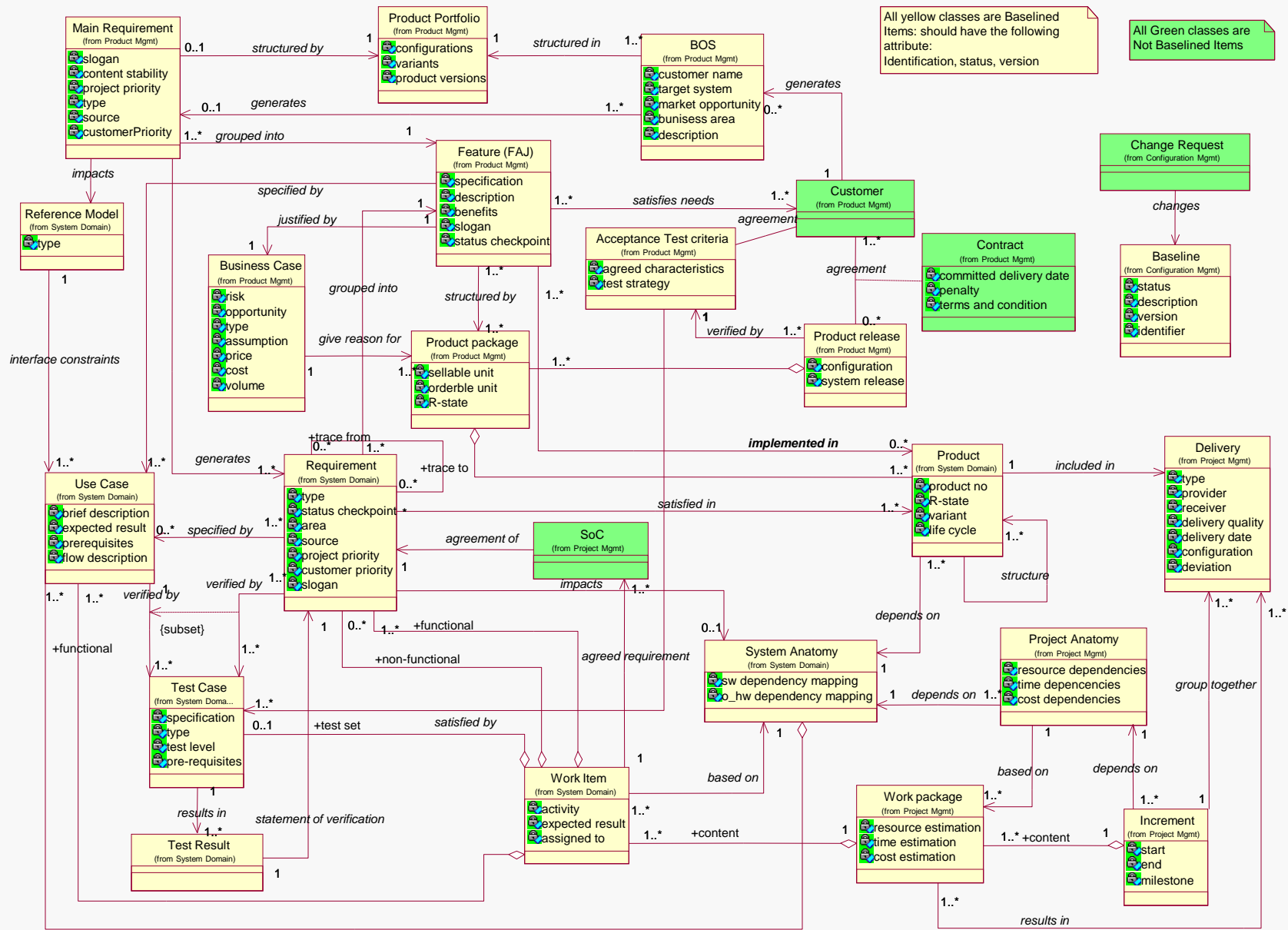
- Business Processes
- Information Landscape
- Application Software
- Infrastructure



# Zachman EA

		<b>WHAT</b> A	<b>HOW</b> B	<b>WHERE</b> C	<b>WHO</b> D	<b>WHEN</b> E	<b>WHY</b> F	
<b>SCOPE</b>	1	List of things important to business	List of processes the business performs	List of locations which the business operates	List of organisations / agents that are important	List of significant events	List of business goals / strategies	<b>Planner</b>
<b>BUSINESS MODEL</b>	2	Semantic Model	Business Process Model	Business Logistics Syst.	Work Flow Model	Master Schedule	Business Plan	<b>Owner</b>
<b>SYSTEM MODEL</b>	3	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model	<b>Designer</b>
<b>TECHNOLOGY MODEL</b>	4	Physical Data Model	System Design	Technology Architecture	Presentation Architecture	Control Structure	Rule Design	<b>Builder</b>
<b>DETAILED REPR.</b>	5	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification	<b>Programmer</b>
<b>FUNCTIONING ENTERPRISE</b>	6	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy	<b>User</b>
		<b>DATA</b>	<b>FUNCTION</b>	<b>NETWORK</b>	<b>PEOPLE</b>	<b>TIME</b>	<b>MOTIVATION</b>	

# A Logical Data Model in practice...



All yellow classes are Baselined Items: should have the following attribute: Identification, status, version

All Green classes are Not Baselined Items

Change Request (from Configuration Mgmt) changes Baseline (from Configuration Mgmt)

Contract (from Product Mgmt) committed delivery date, penalty, terms and condition

Baseline (from Configuration Mgmt) status, description, version, identifier

Acceptance Test criteria (from Product Mgmt) agreed characteristics, test strategy

Business Case (from Product Mgmt) risk, opportunity, type, assumption, price, cost, volume

Product package (from Product Mgmt) sellable unit, orderable unit, R-state

Product release (from Product Mgmt) configuration, system release

Reference Model (from System Domain) type

Use Case (from System Domain) brief description, expected result, prerequisites, flow description

Requirement (from System Domain) type, status checkpoint, area, source, project priority, customer priority, slogan

SoC (from Project Mgmt) impacts

System Anatomy (from System Domain) sw\_dependency mapping, hw\_dependency mapping

Work Item (from System Domain) activity, expected result, assigned to

Test Case (from System Domain) specification, type, test level, pre-requisites

Test Result (from System Domain)

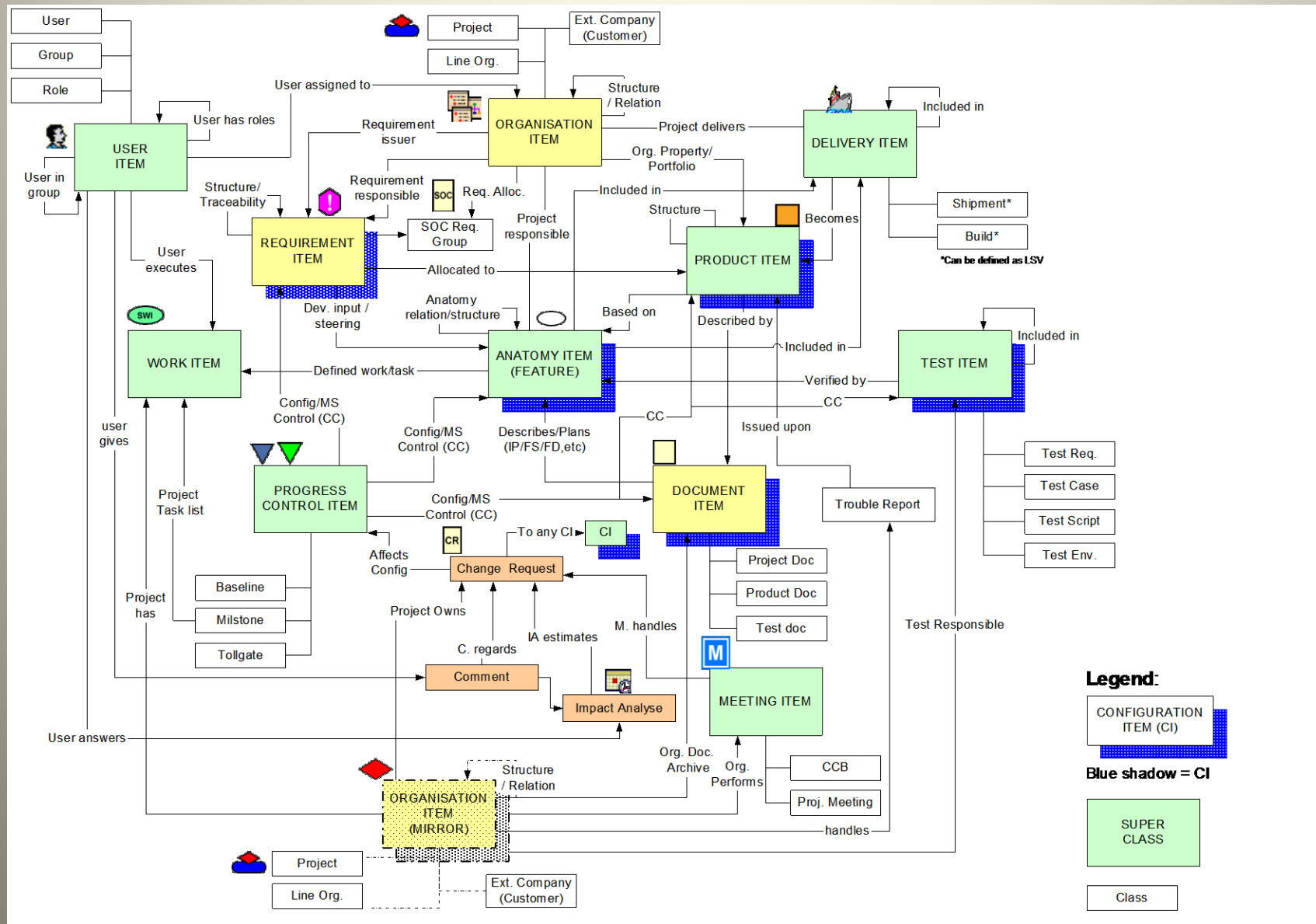
Project Anatomy (from Project Mgmt) resource dependencies, time dependencies, cost dependencies

Work package (from Project Mgmt) resource estimation, time estimation, cost estimation

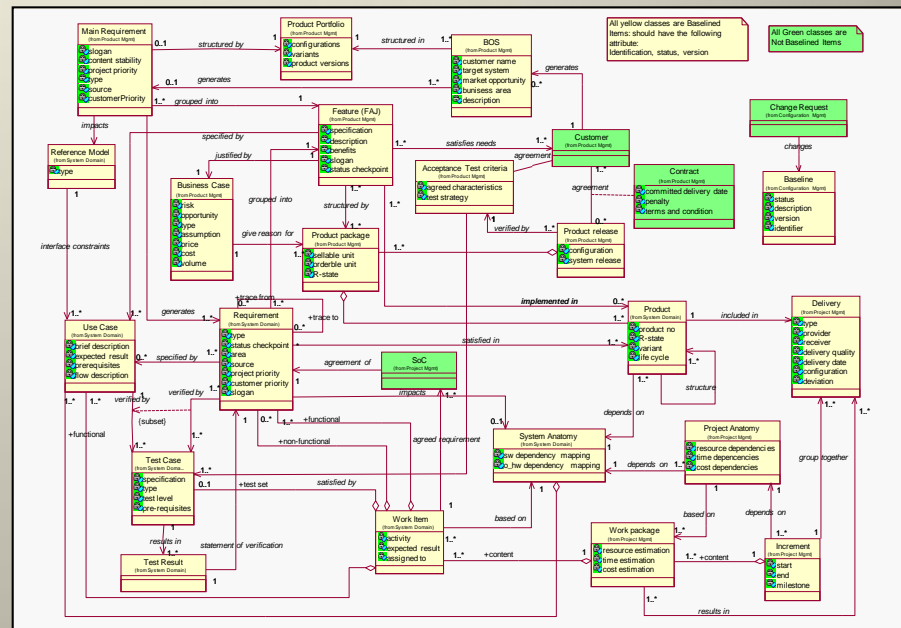
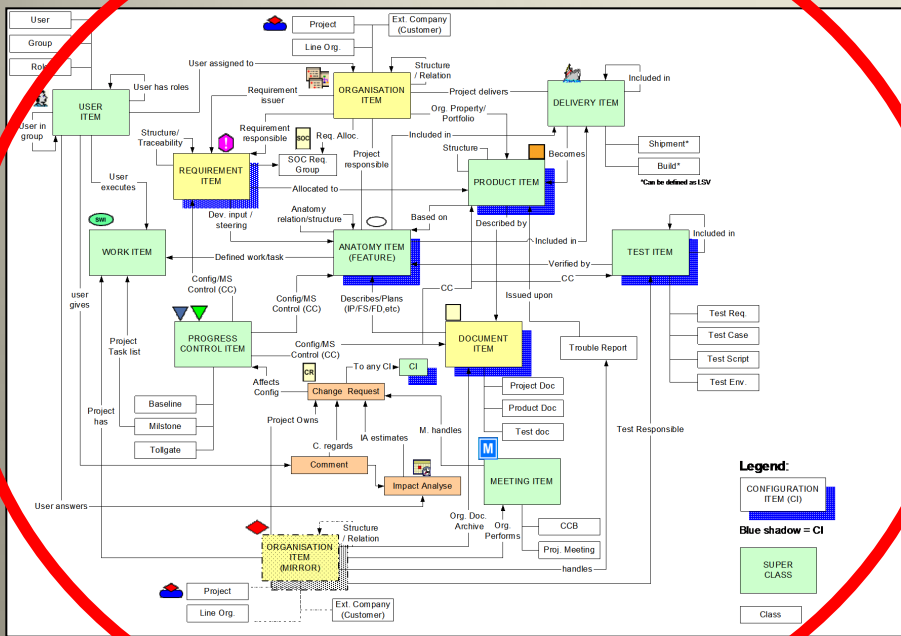
Increment (from Project Mgmt) start, end, milestone

Delivery (from Project Mgmt) type, provider, receiver, delivery quality, delivery date, configuration, deviation

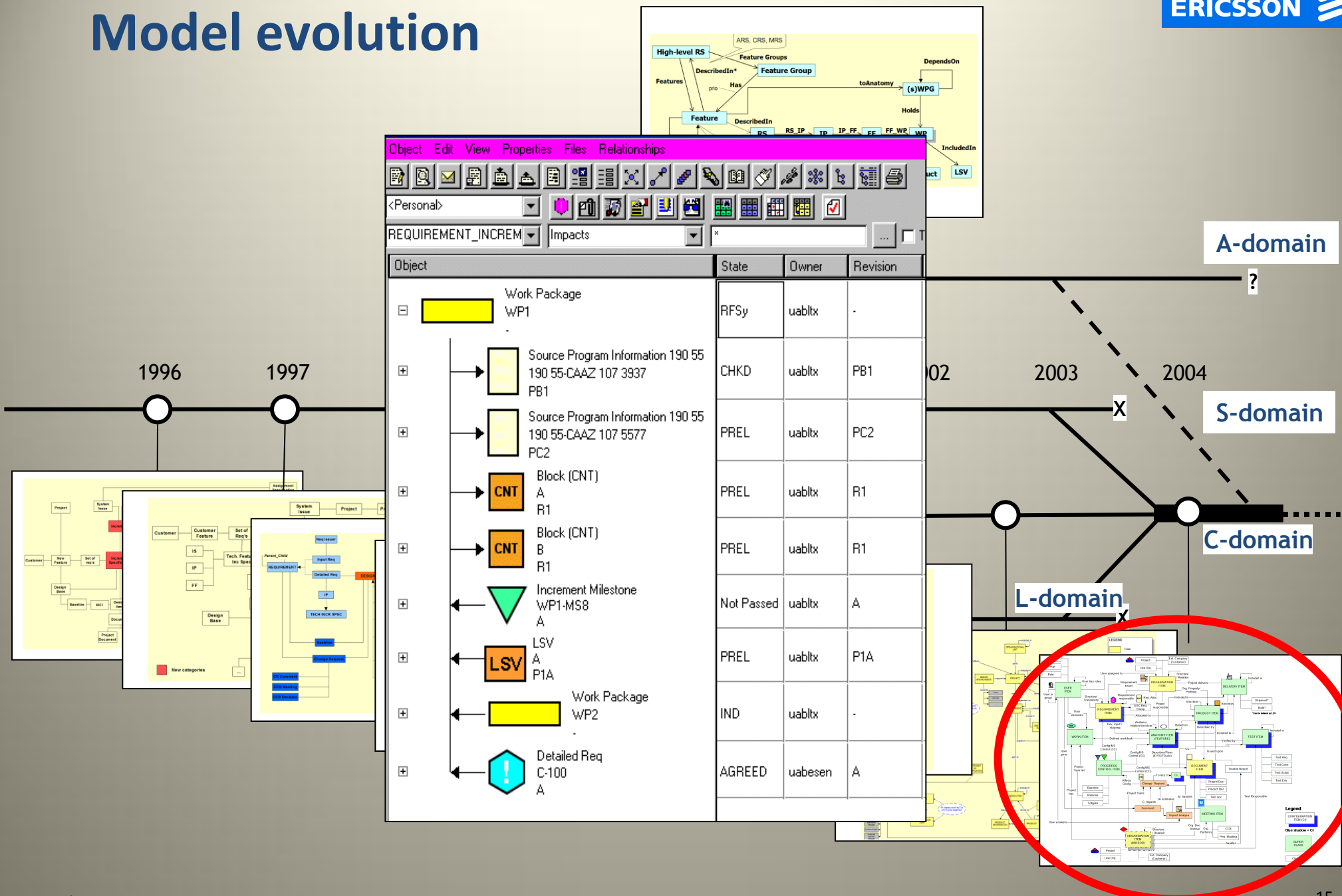
# Another Logical Data model ...

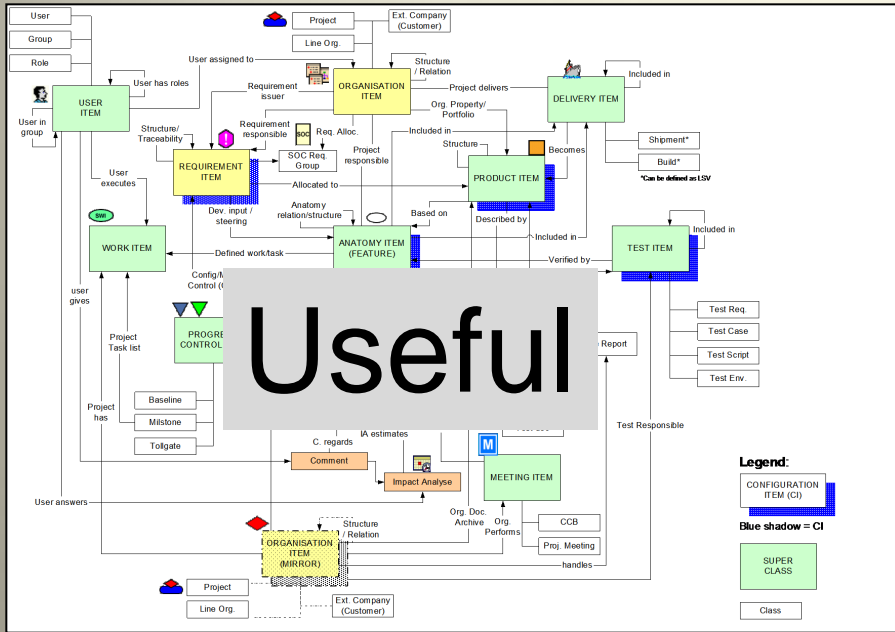


# What's the difference?

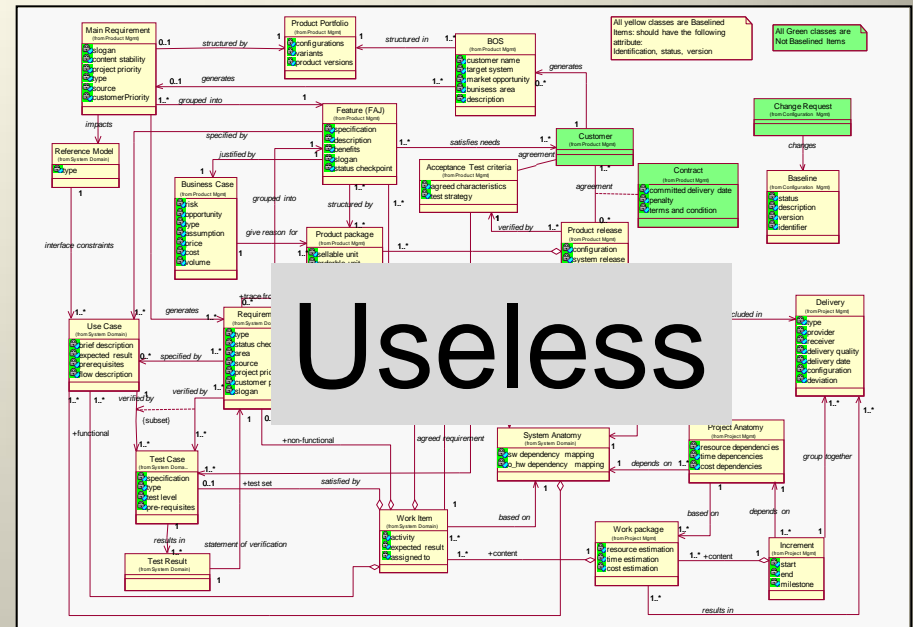
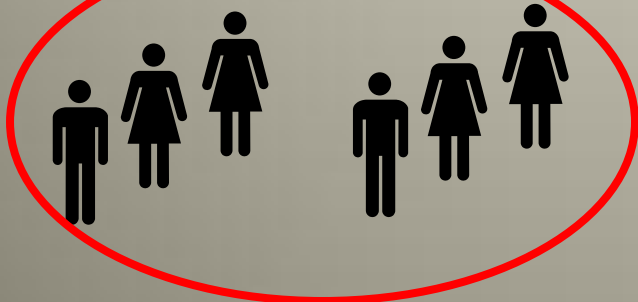


# Model evolution





Chiseled out “on the combat field” between 1996 - 2003



Defined by a consultant “in the chamber”





Existing integrating approaches  
not particularly successful

# The Activity Domain

- Back to basics
- What does it take to integrate an activity?

# The Activity Domain

## Activity Modalities

Focus on a target, motivated by a need

- objectivation

Frame a context of relevance

- contextualization

Orient ourselves in space

- spatialization

Conceive of actions leading to the goal

- temporalization

Learn how to act relevantly

- stabilization

Change focus

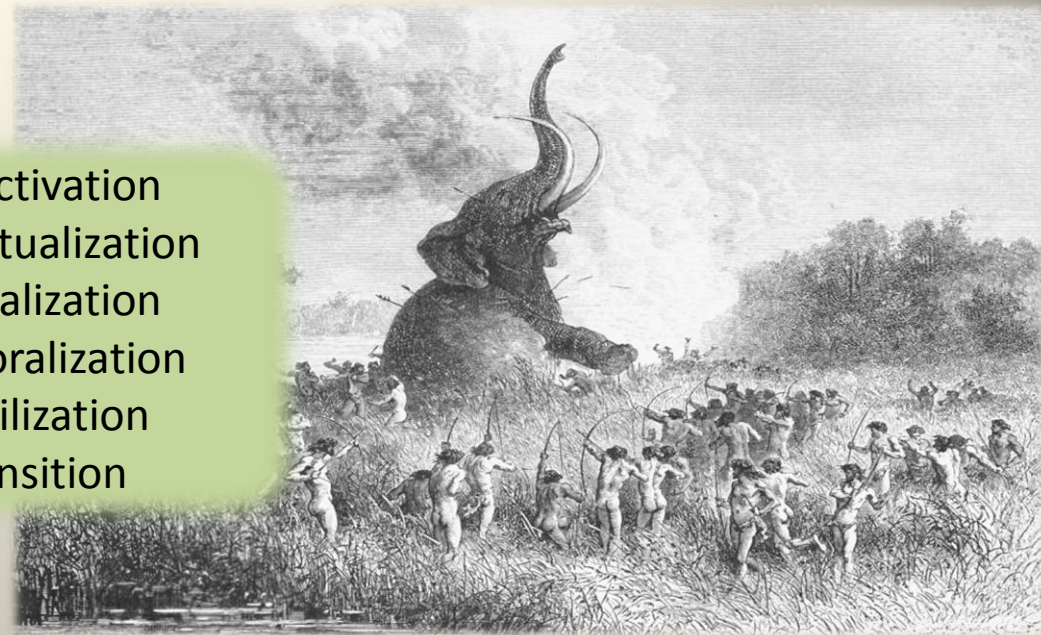
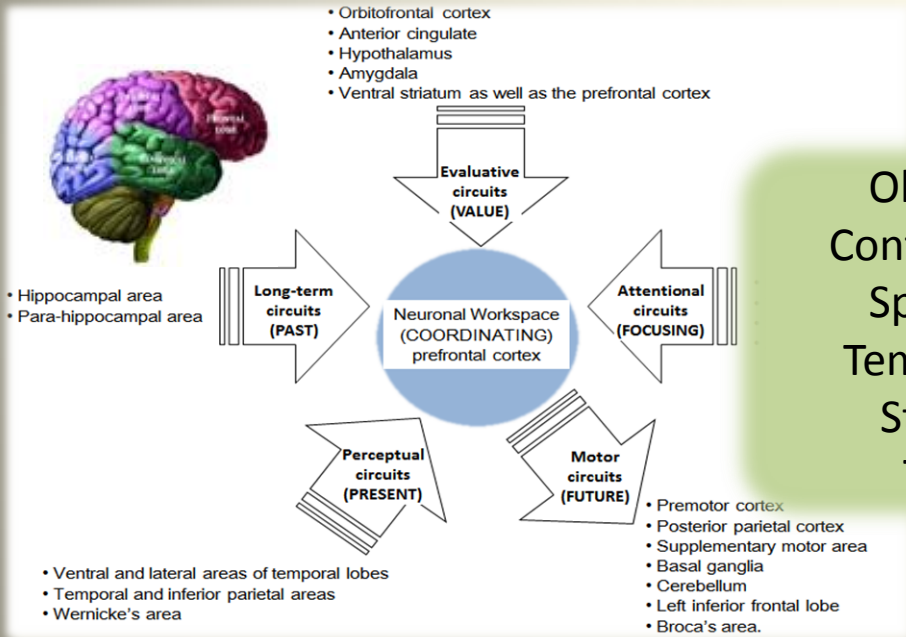
- transition

Enact means

Align individual meanings

# Cognitive –Neural, inner realm

# Social, external realm



Objectivation  
Contextualization  
Spatialization  
Temporalization  
Stabilization  
Transition



The Activity Domain is regarded as the **integrative foundation** for all human activities, including organizational ones

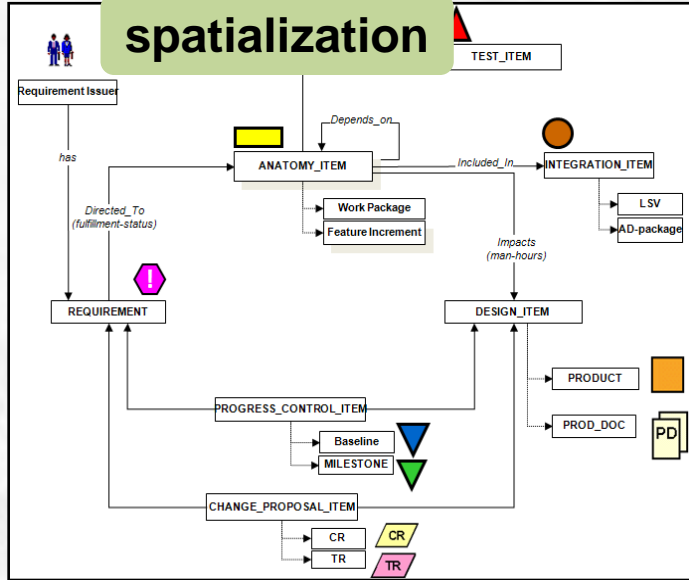
Why?

Because we still employ the same biological abilities in all situations we encounter in every-day life, including organizations

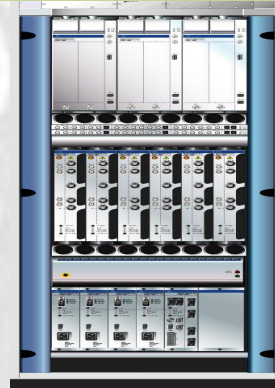
# A contemporary Activity Domain



## Information models



Target  
objectivation



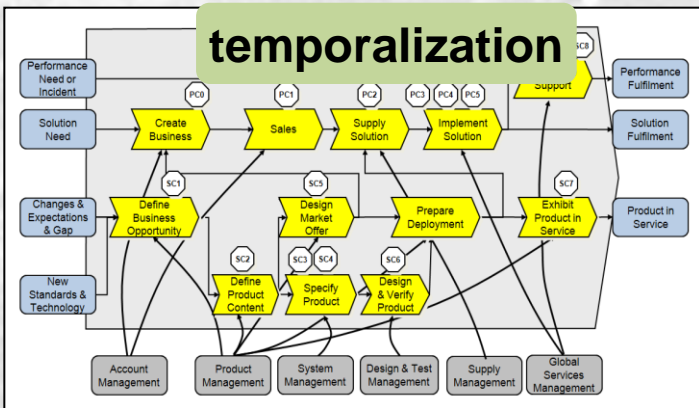
Enact means

Align individual meanings

## PLM systems

Object	State	Owner	Revision
Work Package WP1	RFSy	uabltx	-
Source Program Information 190 55 190 55-CAAZ 107 3337 PB1	CHKD	uabltx	PB1
Source Program Information 190 55 190 55-CAAZ 107 5577 PC2	PREL	uabltx	PC2
Block (CNT) A R1	PREL	uabltx	R1
Block (CNT) B R1	PREL	uabltx	R1
Increment Milestone WP1-M58 A	Not Passed	uabltx	A
LSV A P1A	PREL	uabltx	P1A
Work Package WP2	IND	uabltx	-
Detailed Req C-100 A	AGREED	uabesen	A

## Process models



Cooperation

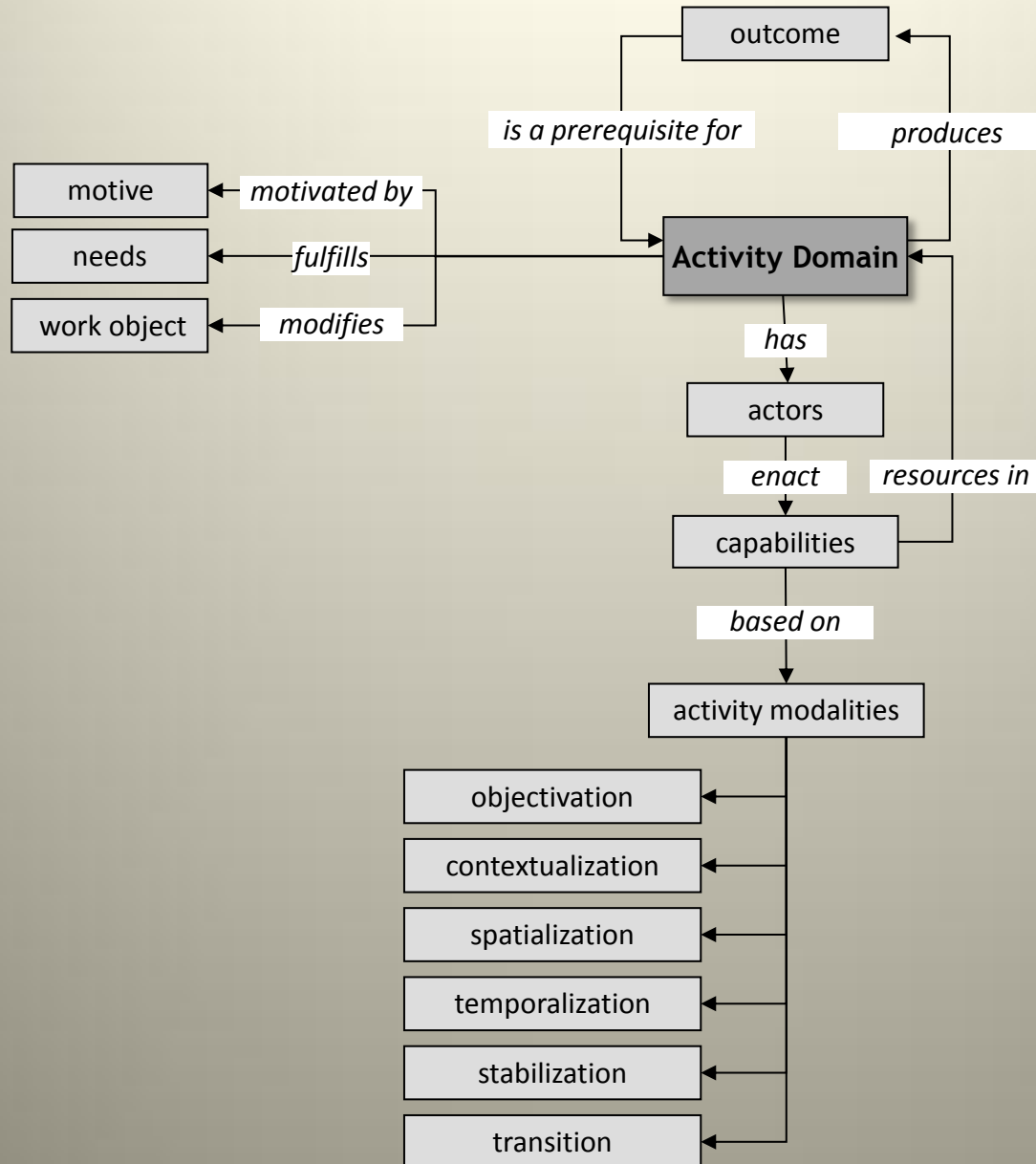
transition



Business rules



# The structure of the Activity Domain



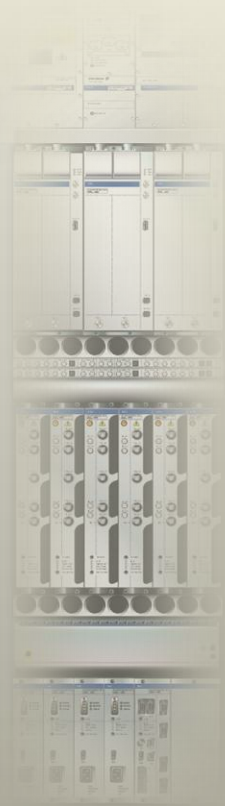
# **Integrating product development**



# Management based on Activity Domains

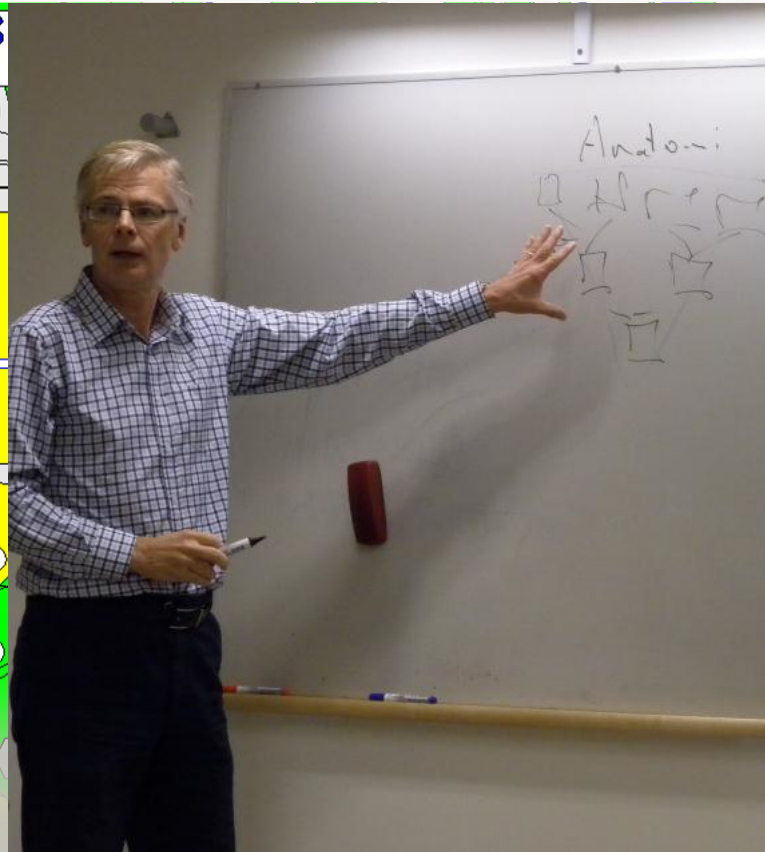


# Early phases?



“The most important thing when working with complex systems is to *manage dependencies*”

Jack Järkvik (1990s)



# **The System Anatomy**

- an integrating image in early phases



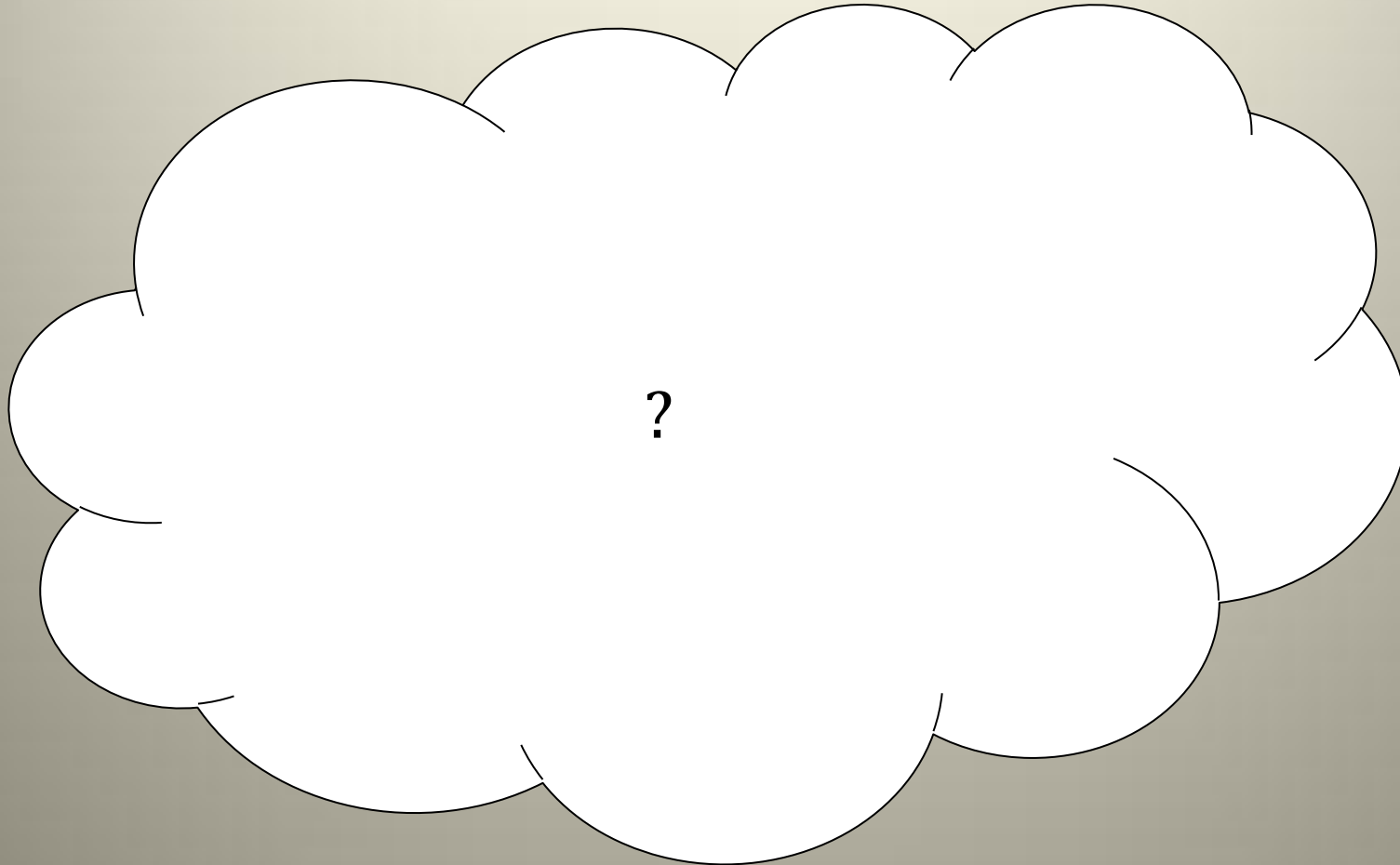
Entrance

Comfort

Cleaning

Sanitation

Nutrition



Ground



Entrance

Comfort

Cleaning

Sanitation

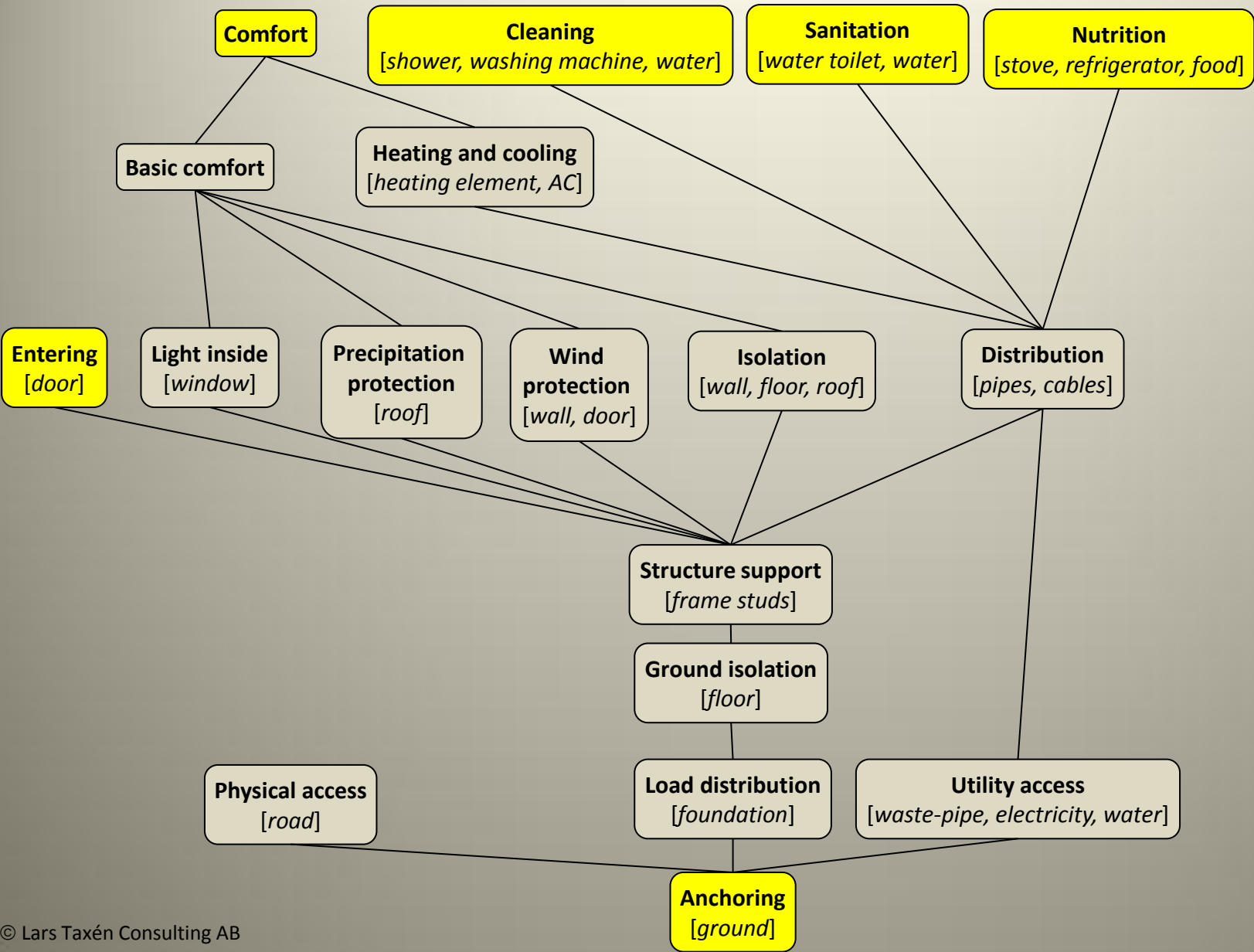
Nutrition

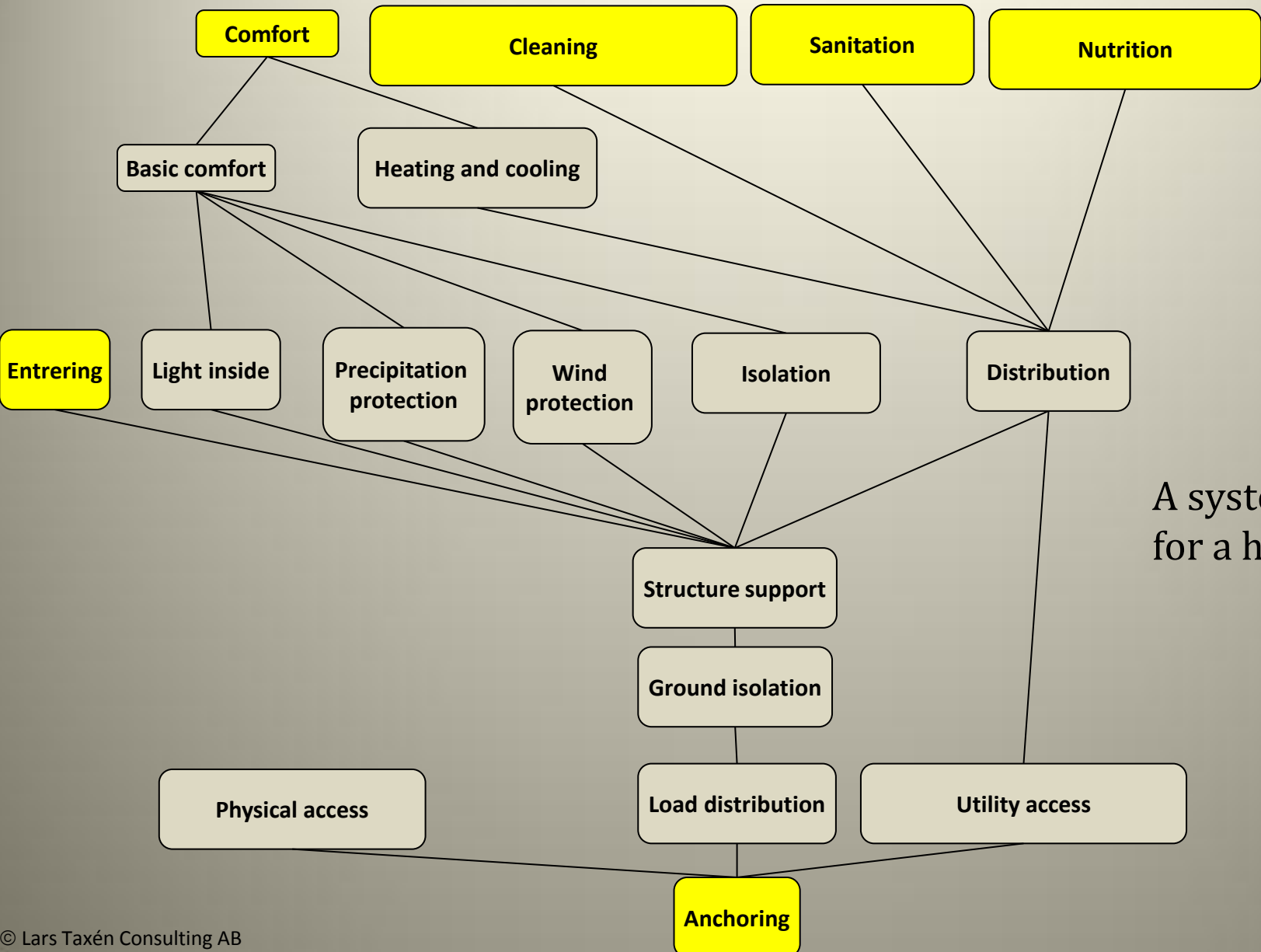
## The System Anatomy

What capabilities are needed?

How do they depend on each other ?

Ground

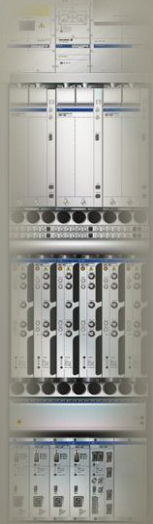
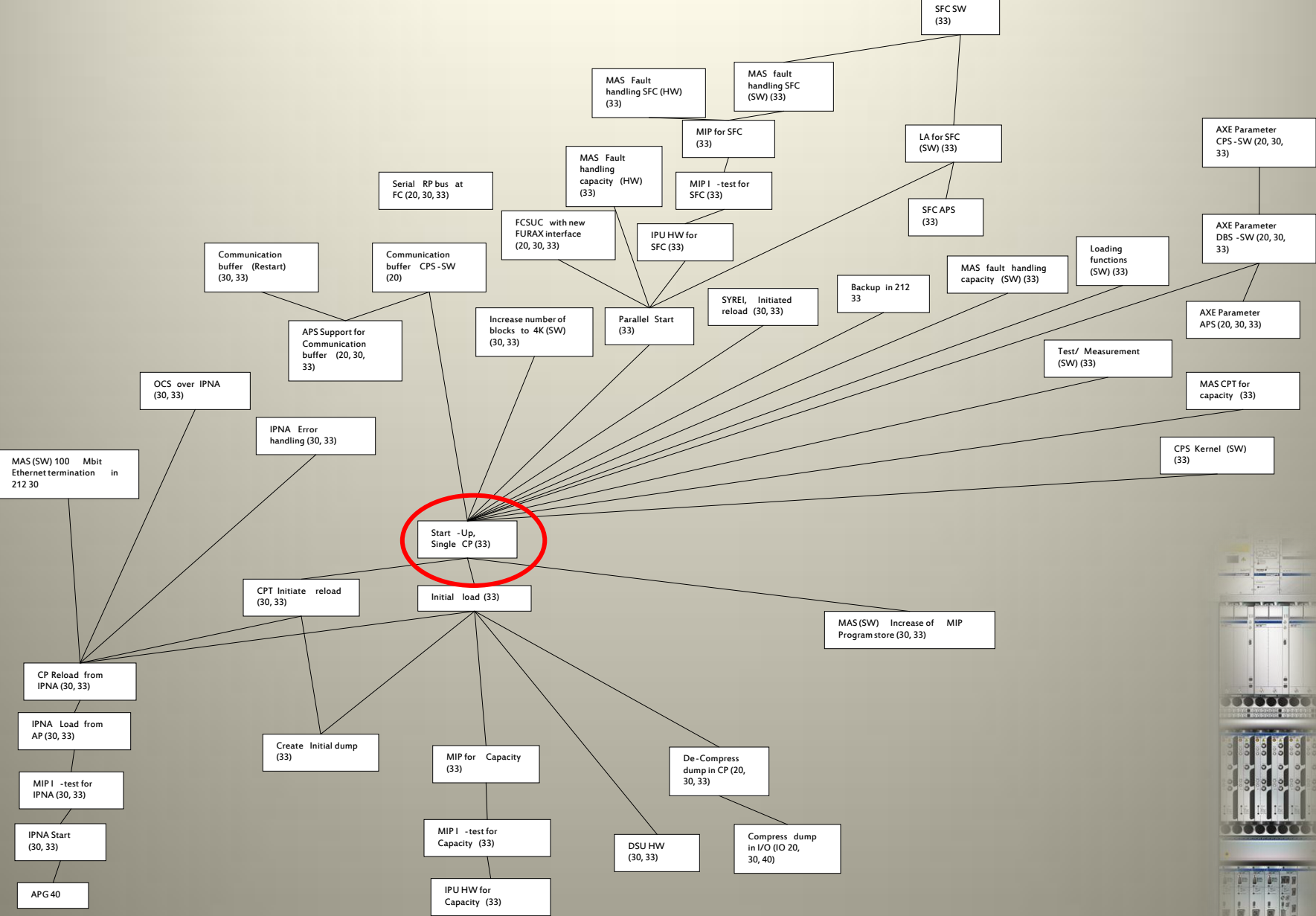




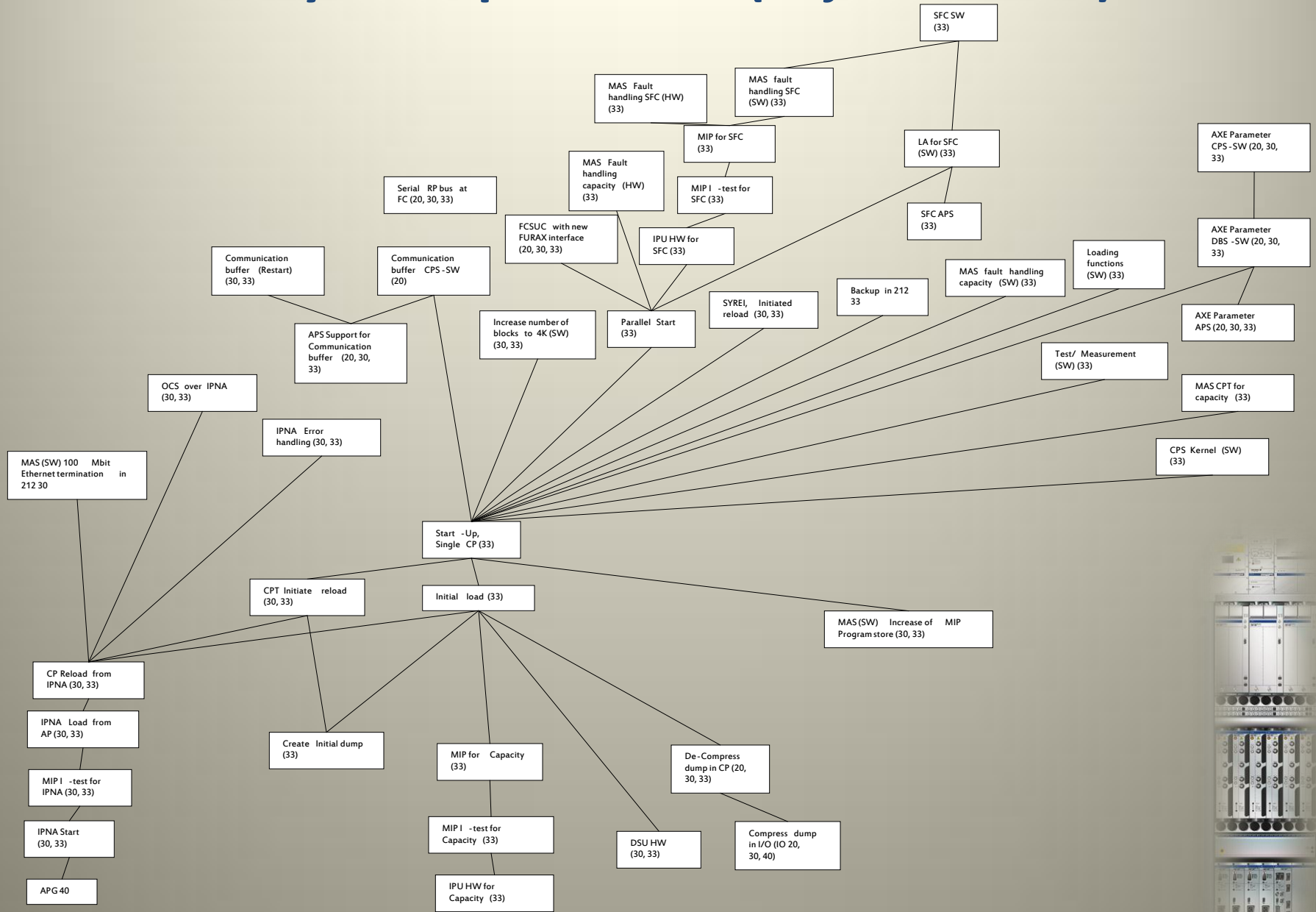
A system anatomy for a house



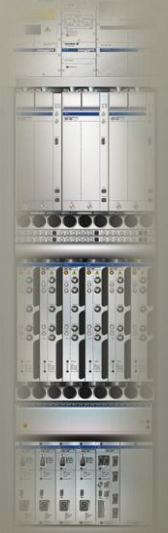
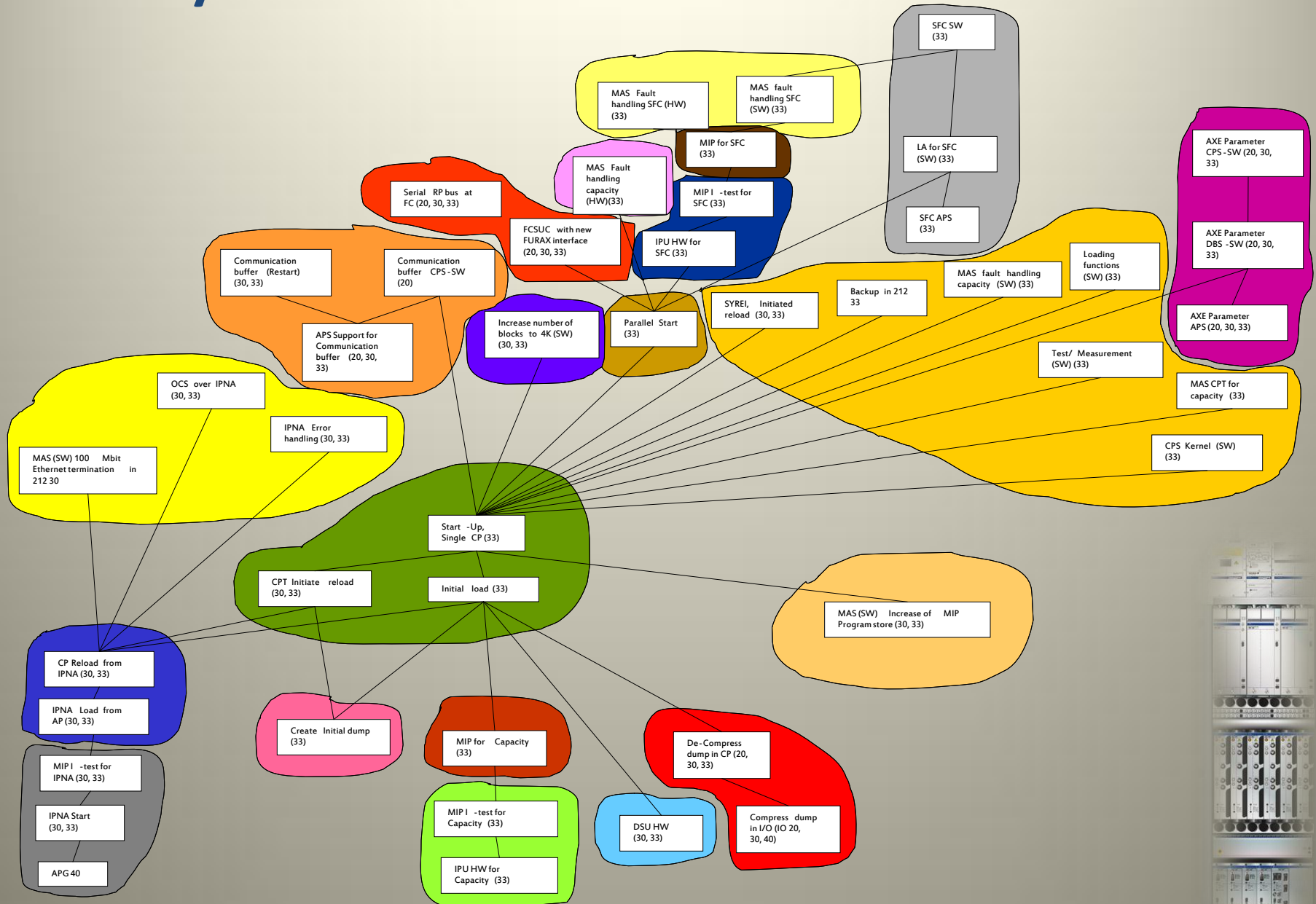
# An anatomy for a processor



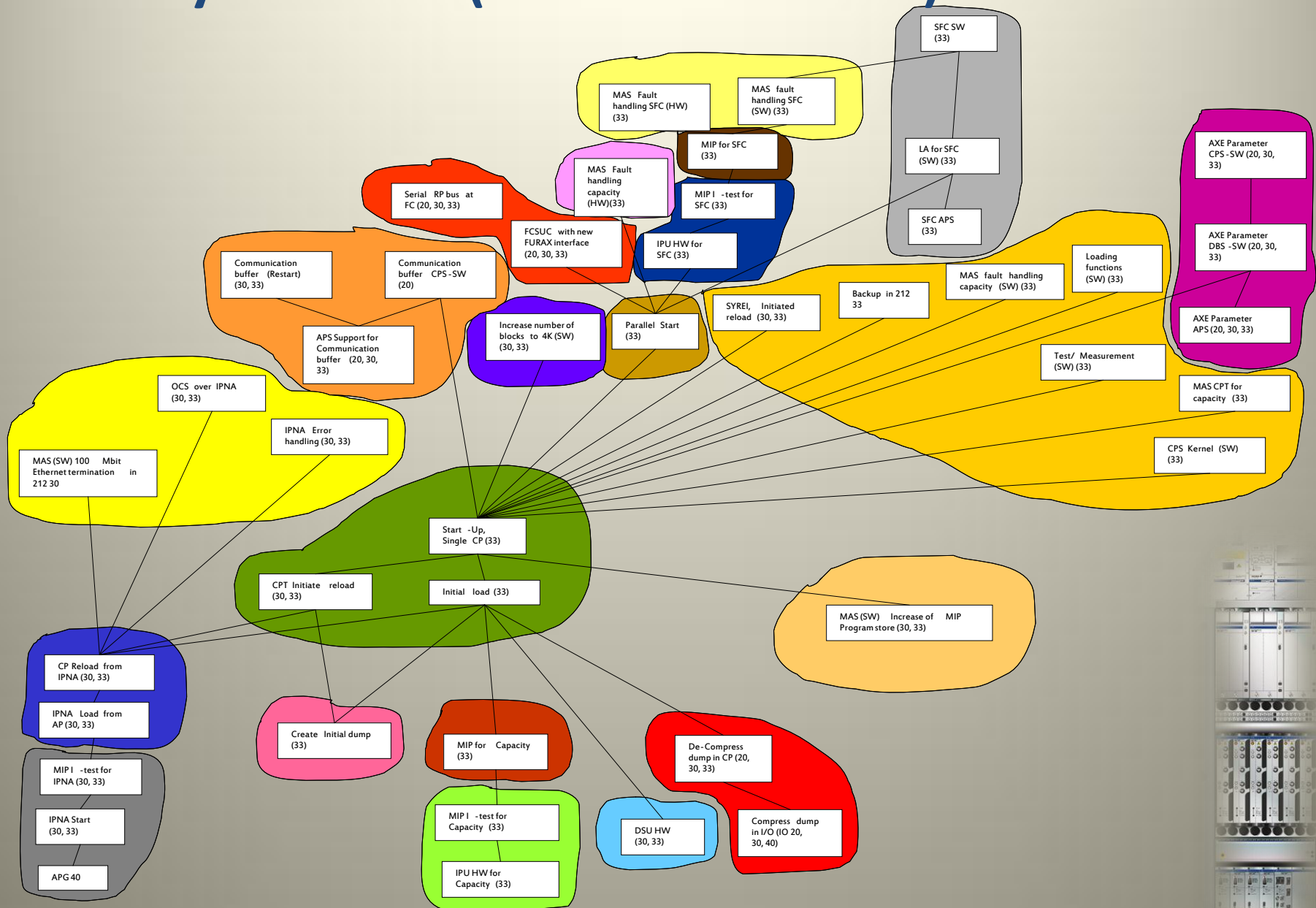
# An anatomy for a processor (*objectivation*)



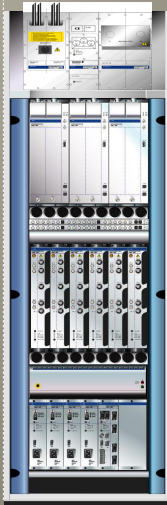
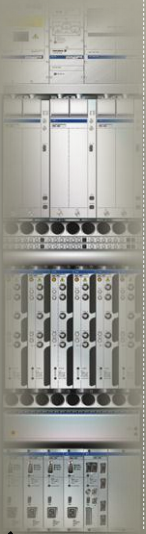
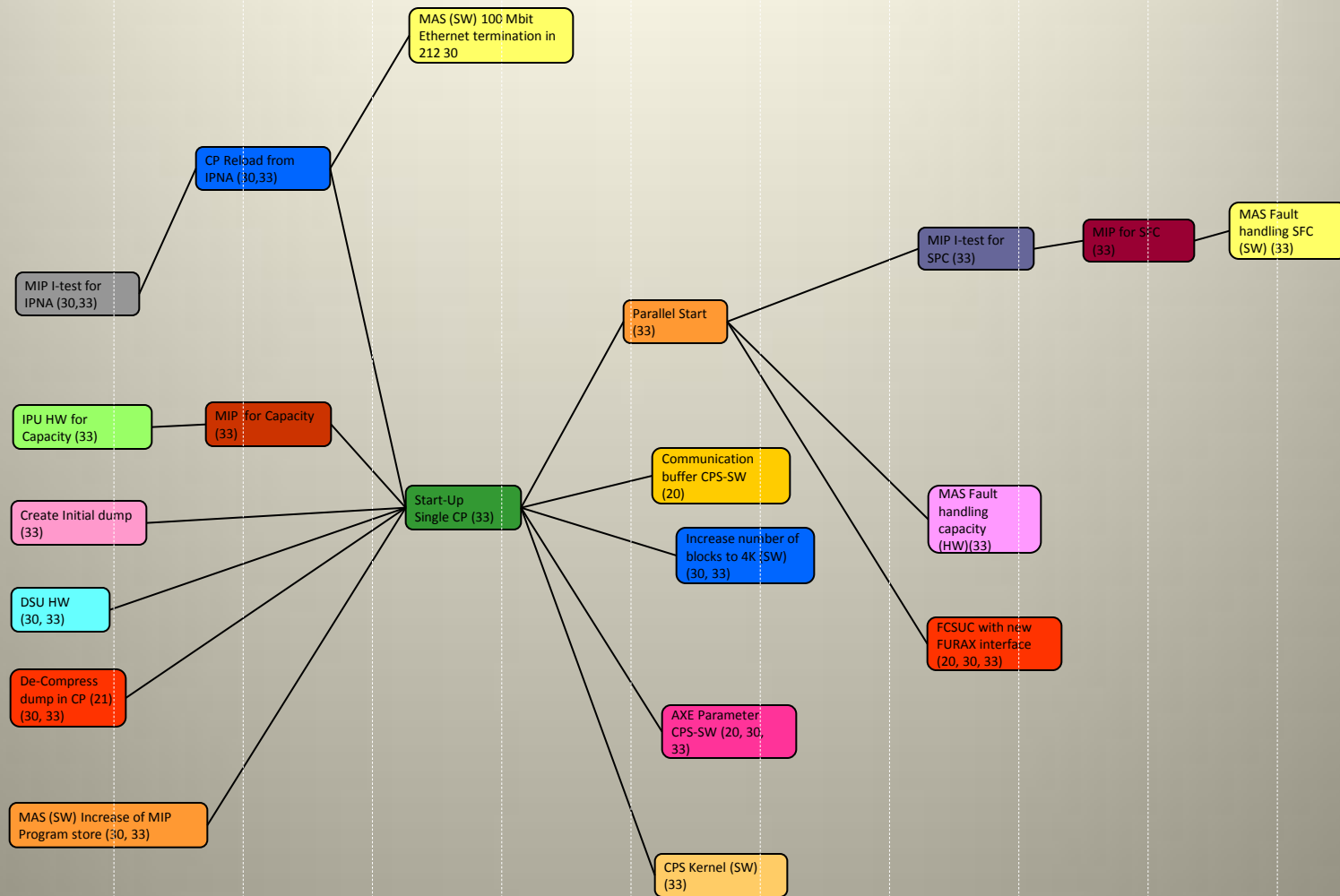
# Activity domains



# Activity domains (contextualization)

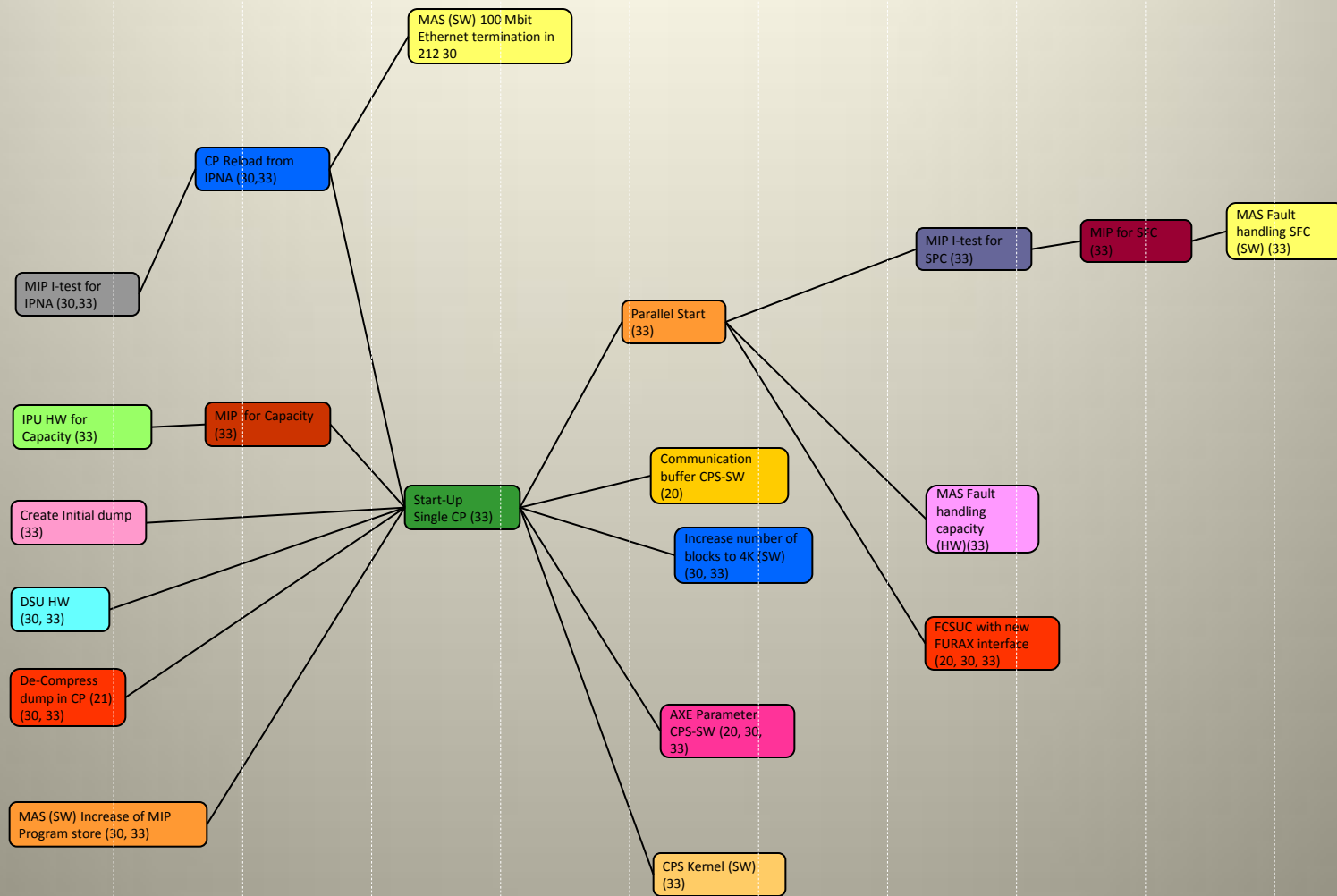


# Integration plan



v13    v14    v15    v16    v17    v18    v19    v20    v21    v22    v23    v24    v25    v26

# Integration plan (*temporalization*)



v13	v14	v15	v16	v17	v18	v19	v20	v21	v22	v23	v24	v25	v26
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

# Target trajectory through Activity Domains

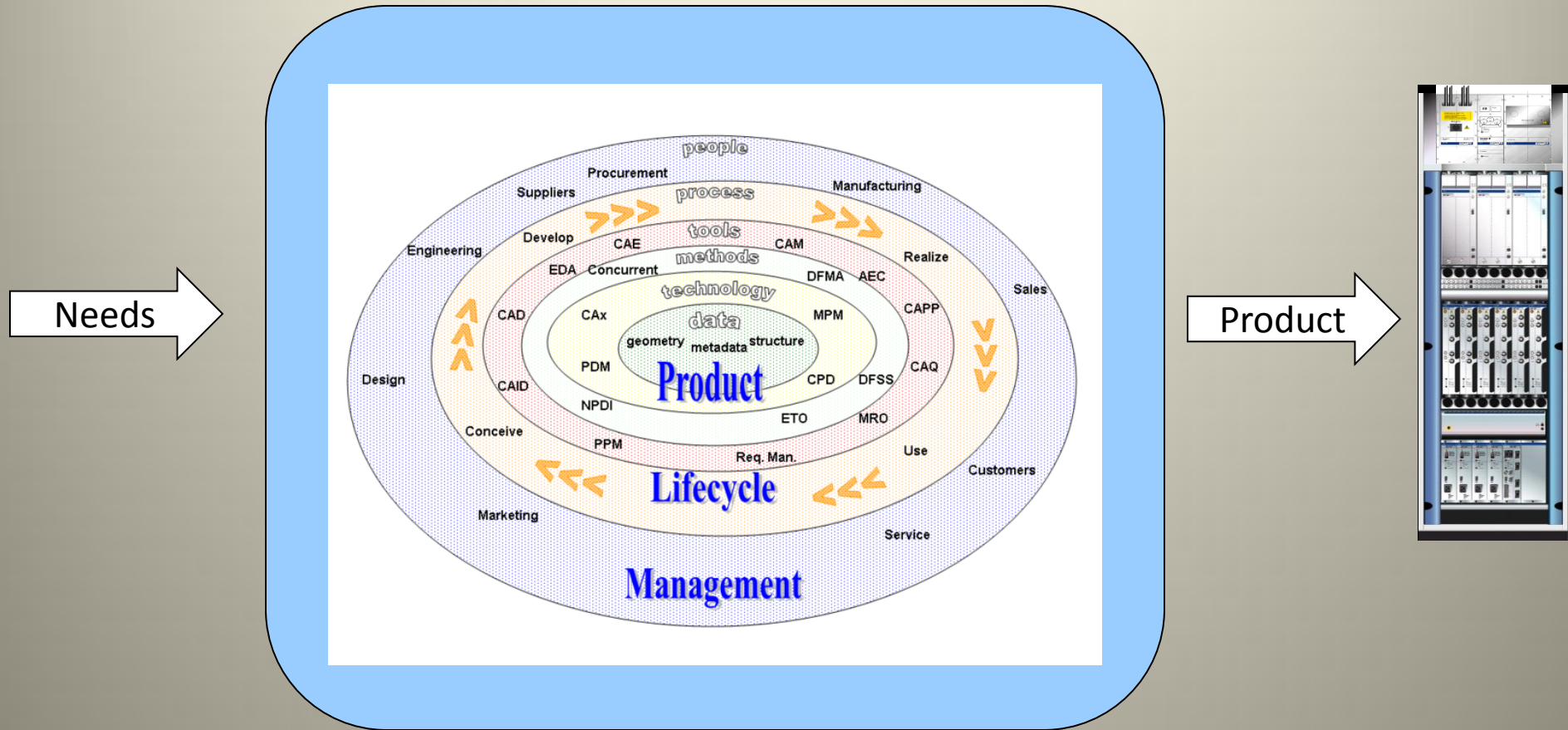


Technical diagrams and server rack. The left diagram shows 'SW upgrade during traffic' and 'DCP connection responsiveness'. The right diagram shows 'Performance and Management' and 'Data Center'. The server rack is a multi-bay unit with various modules.

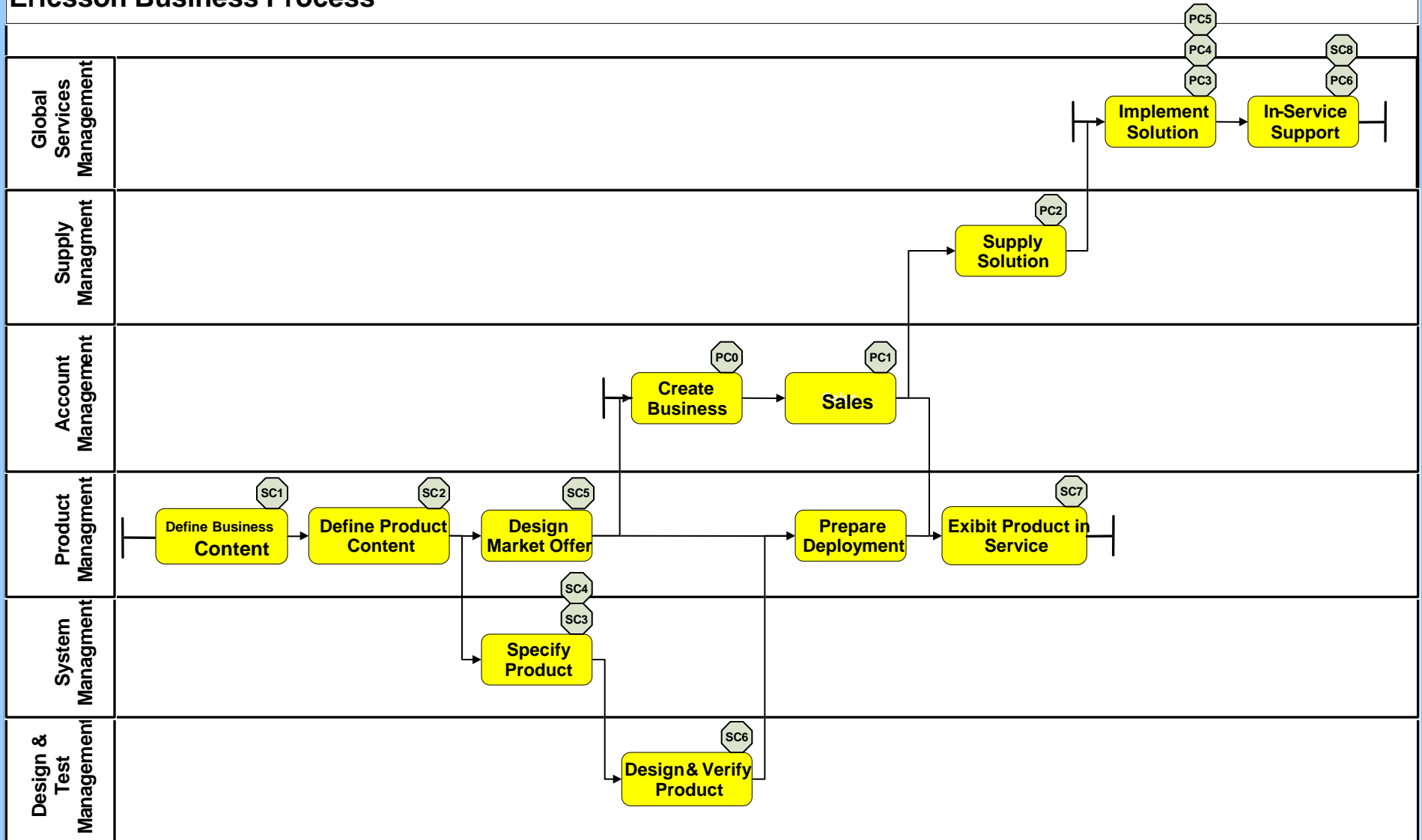
# The Enterprise Anatomy



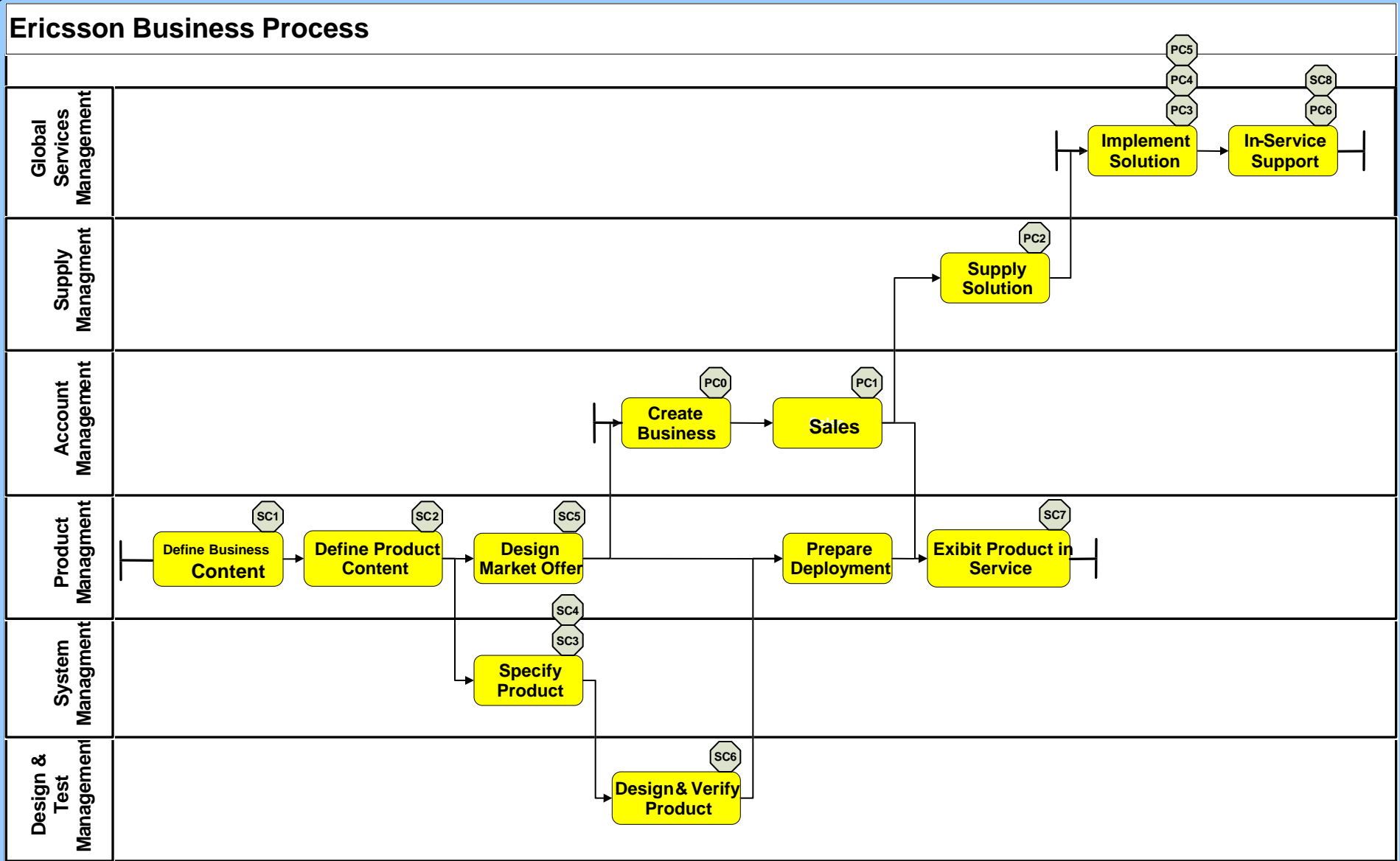
# The Activity Domain



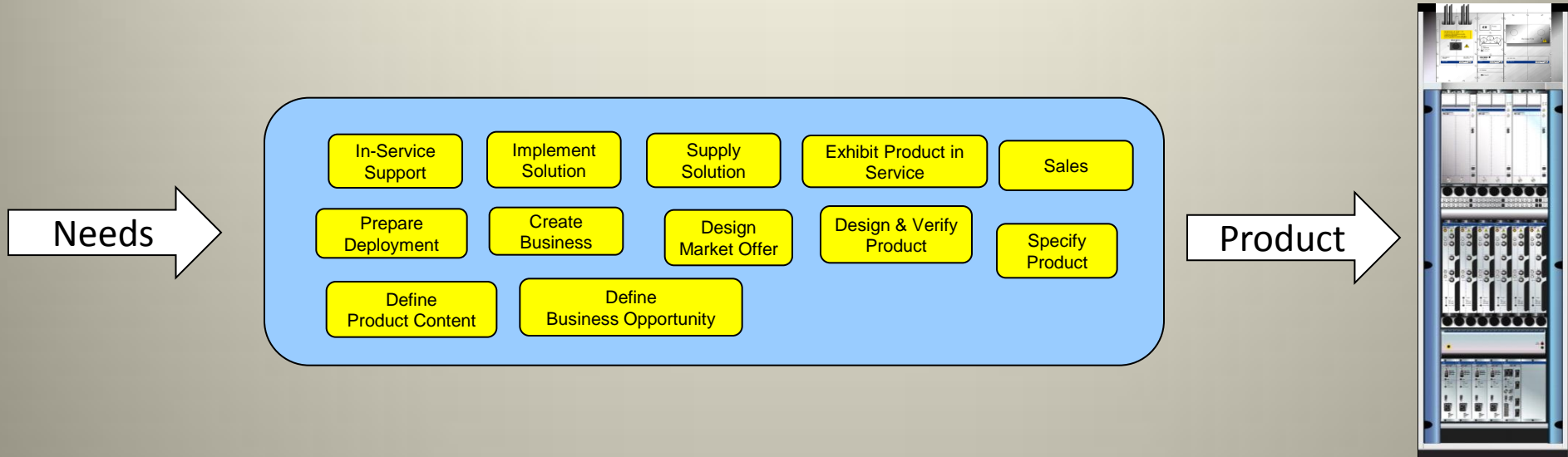
## Ericsson Business Process



# Business process (*temporalization*)

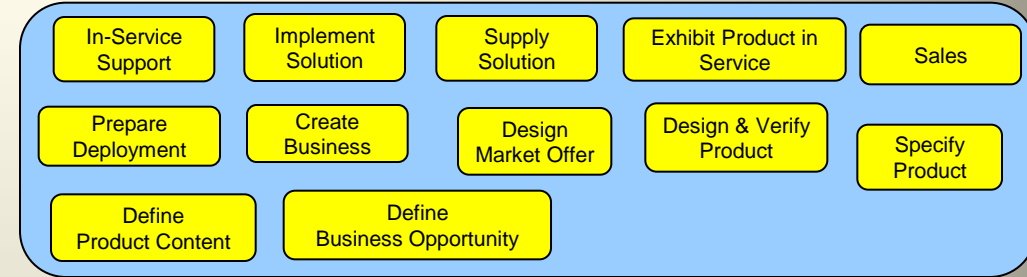


# Activity Domains as Business Capabilities



# Define activity domains

- **Activity Domains**
  - Target
  - Motive

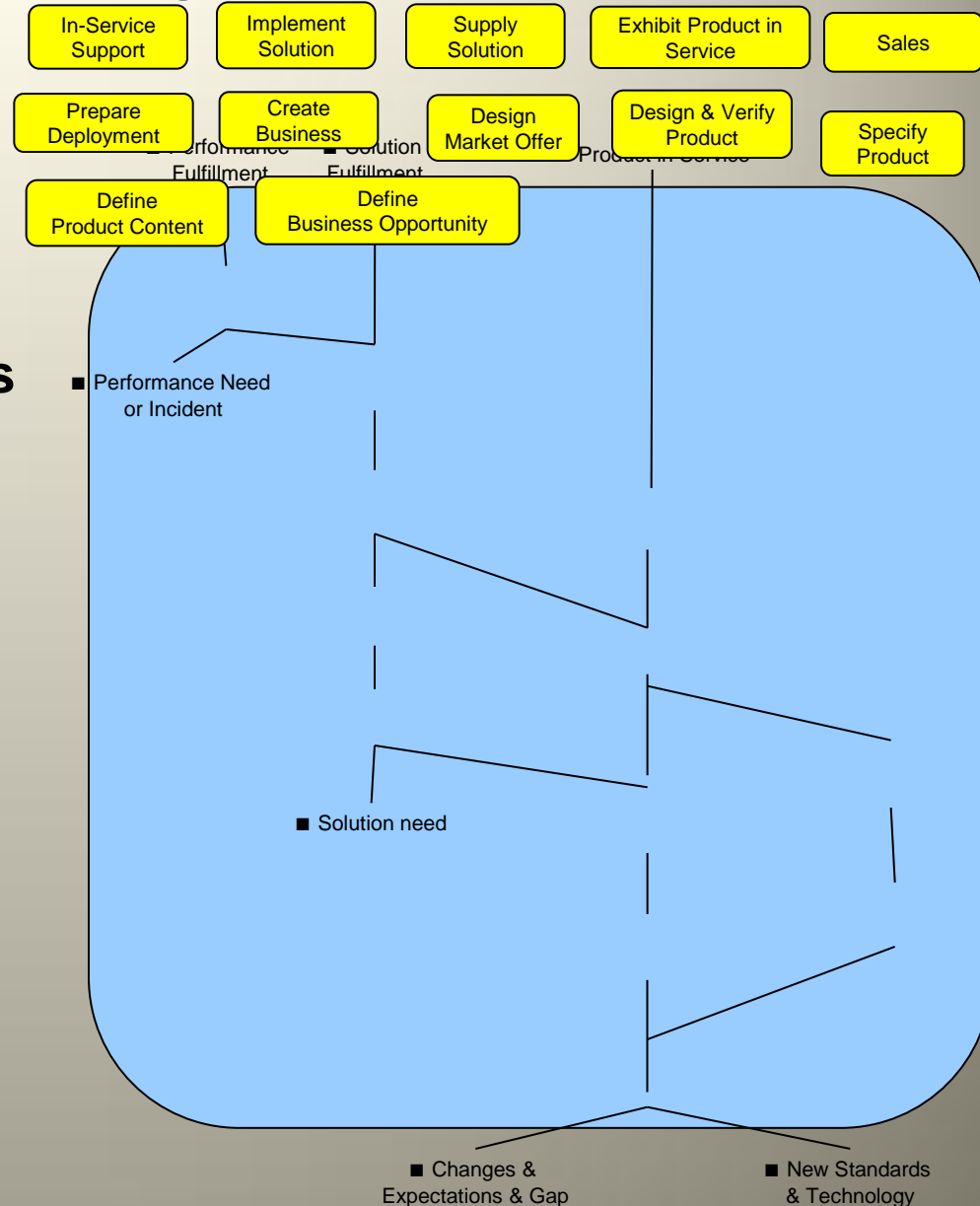


# Define the enterprise anatomy

- **Activity Domains**

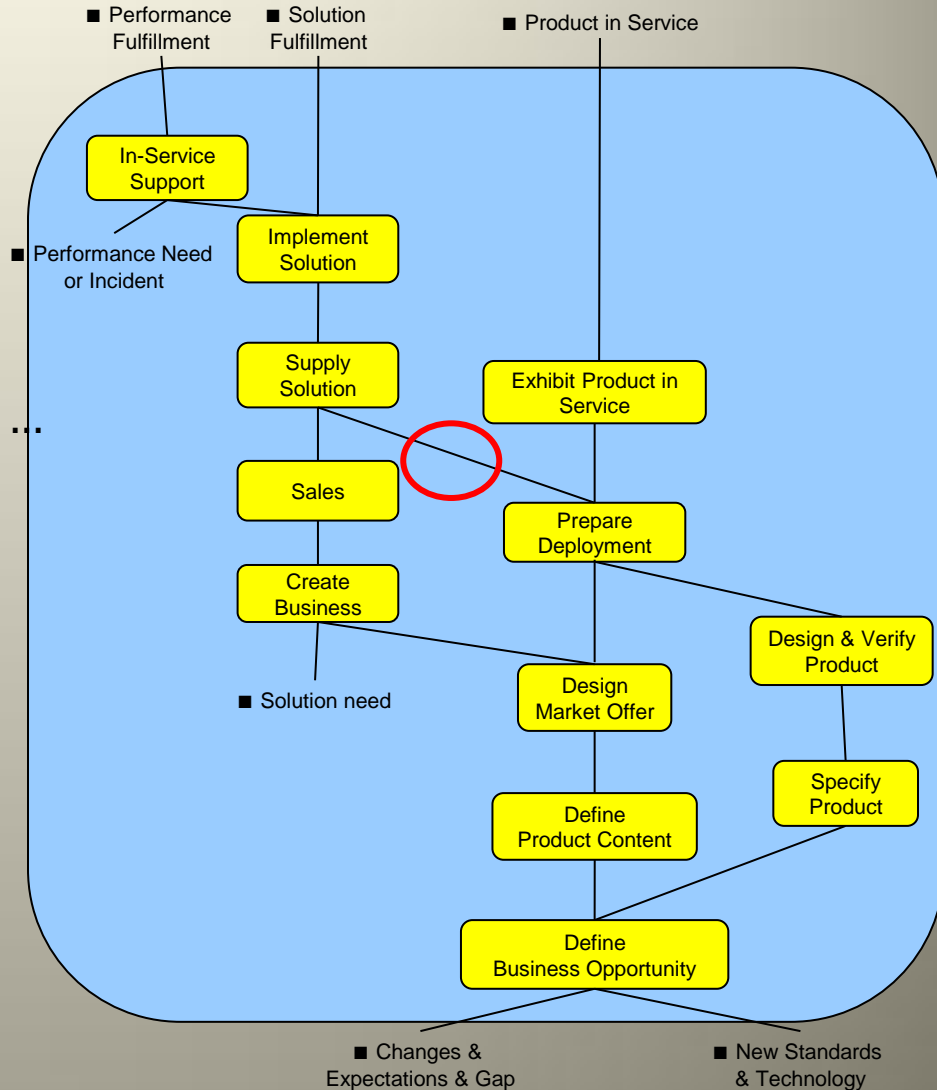
- Target
- Motive

- **Dependencies between domains**



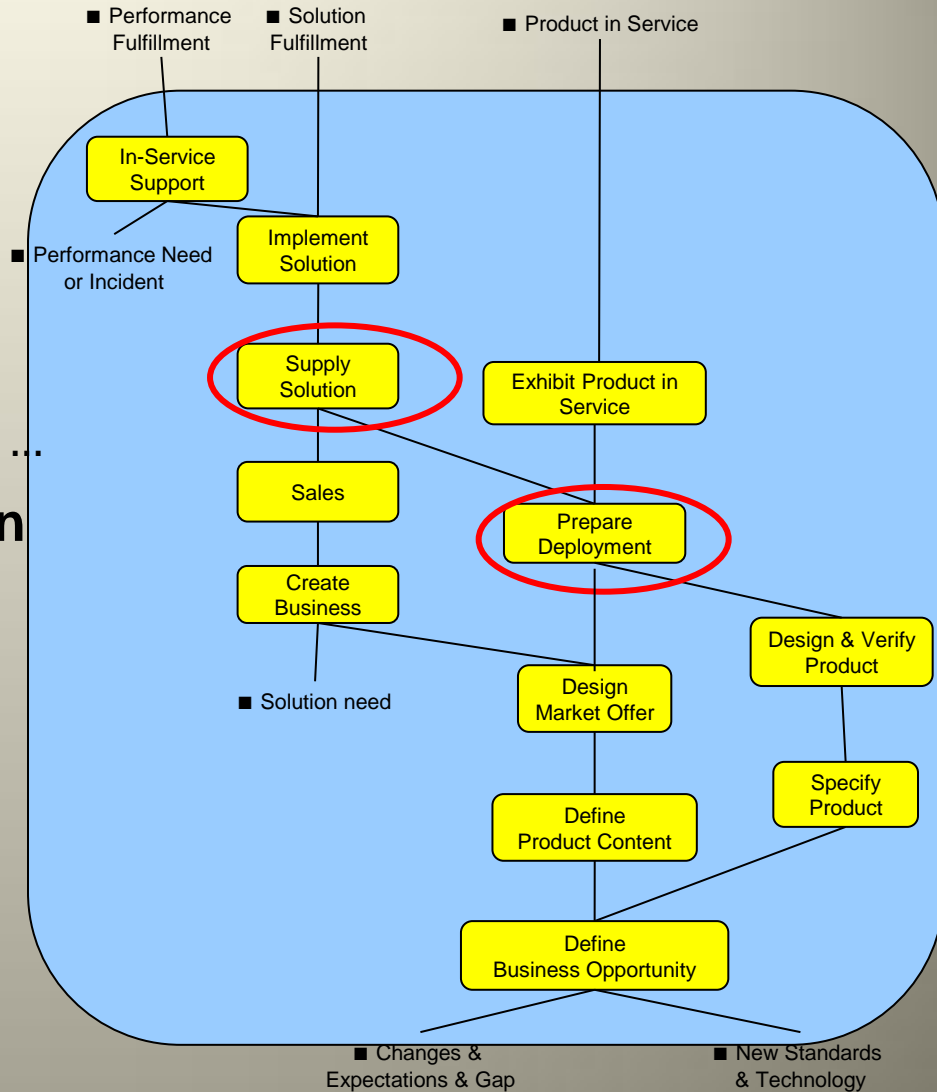
# Transitions

- **Activity Domains**
  - Target
  - Motive
- **Dependencies between domains**
- **Transitions between domains**
  - Mapping rules, translations, interfaces ...



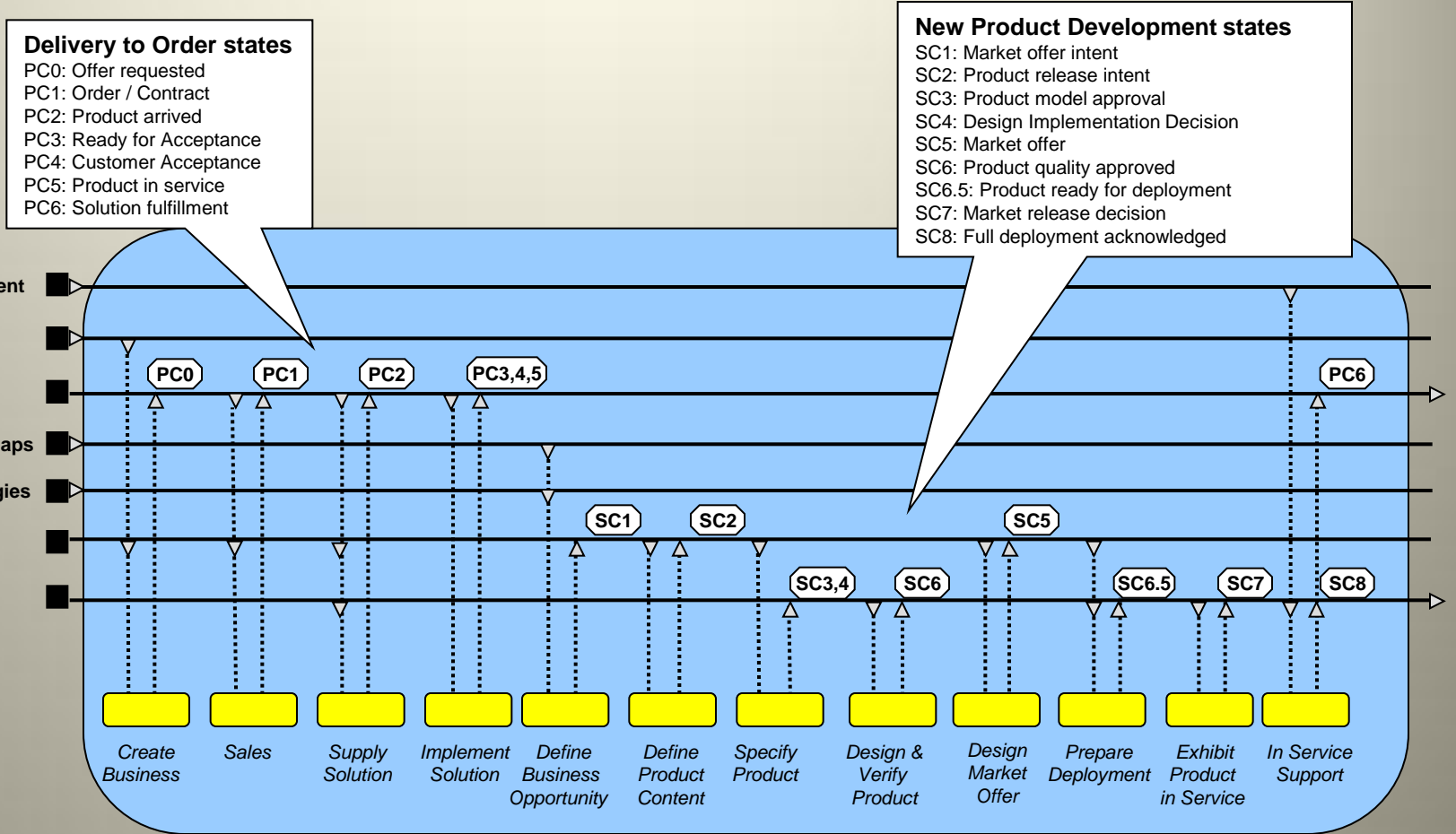
# The internals of each domain

- **Activity Domains**
  - Target
  - Motive
- **Dependencies between domains**
- **Transitions between domains**
  - Mapping rules, translations, interfaces ...
- **Activity modalities for each domain**
  - Information Model    - relevant things
  - Process Models        - ordered actions
  - Business rules         - valid actions
  - PLM, ERP                - means
  - Enacting means
  - Aligning individual meanings

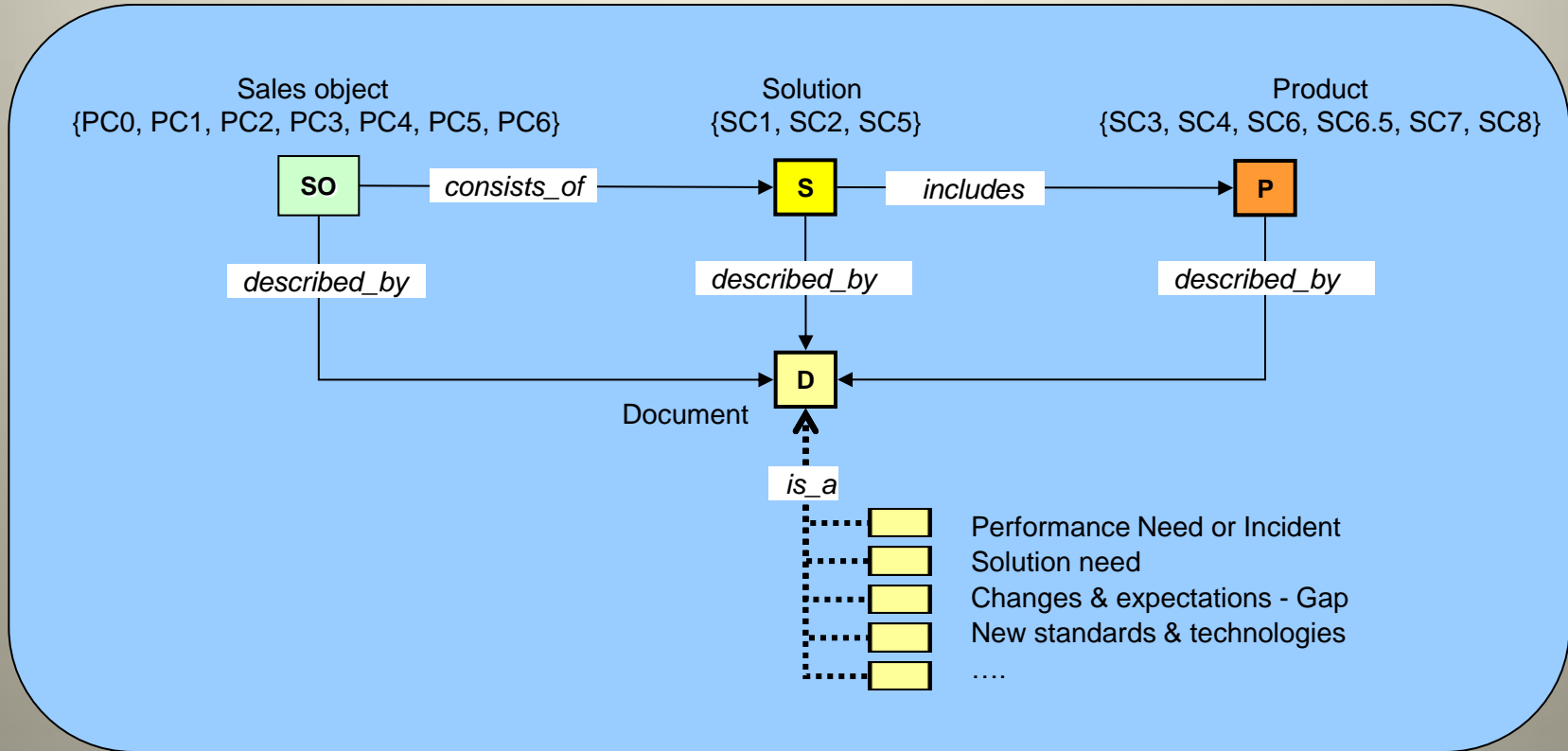




# Activity Domain process model (*temporalization*)

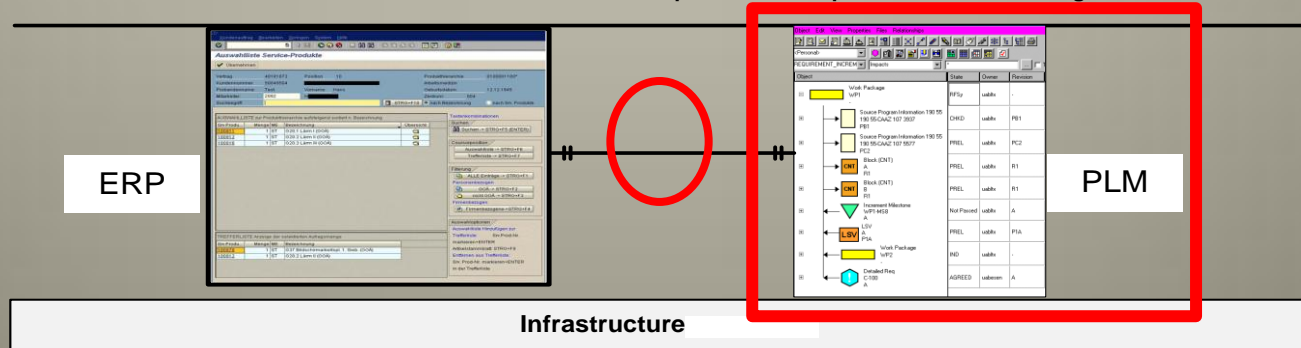
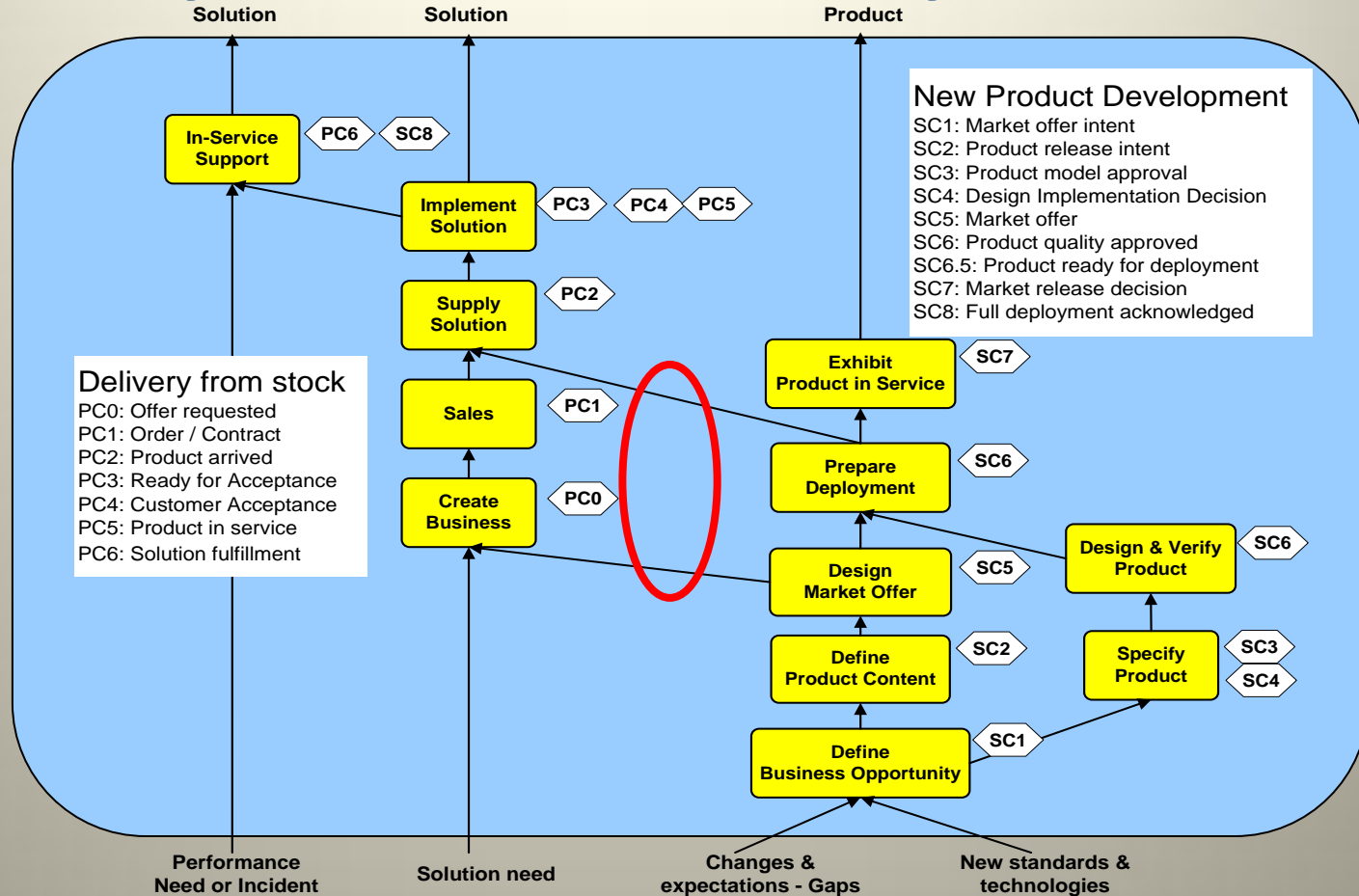


# Activity Domain information model (*spatialization*)

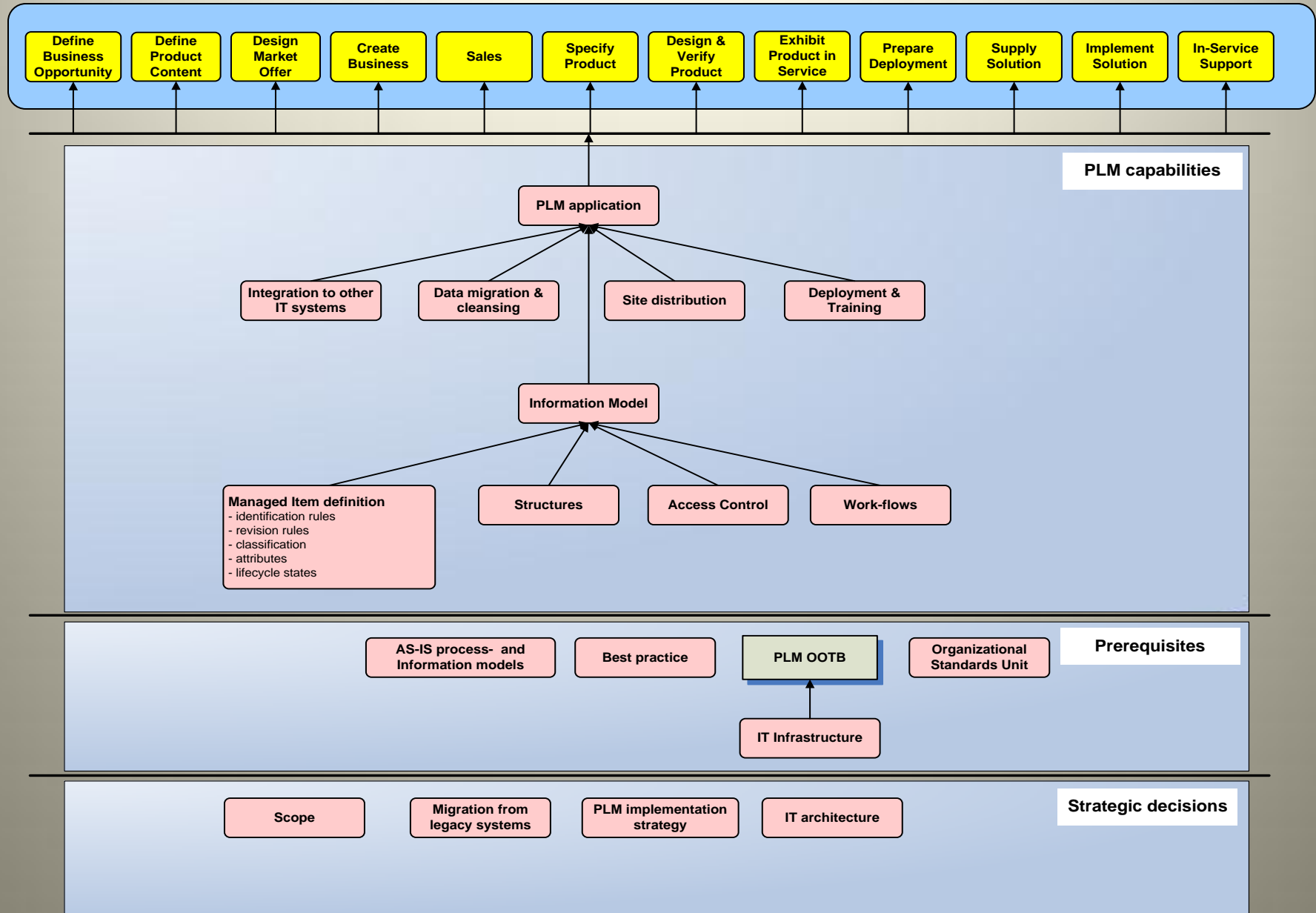


# IT system implementation

# Enterprise systems in the anatomy

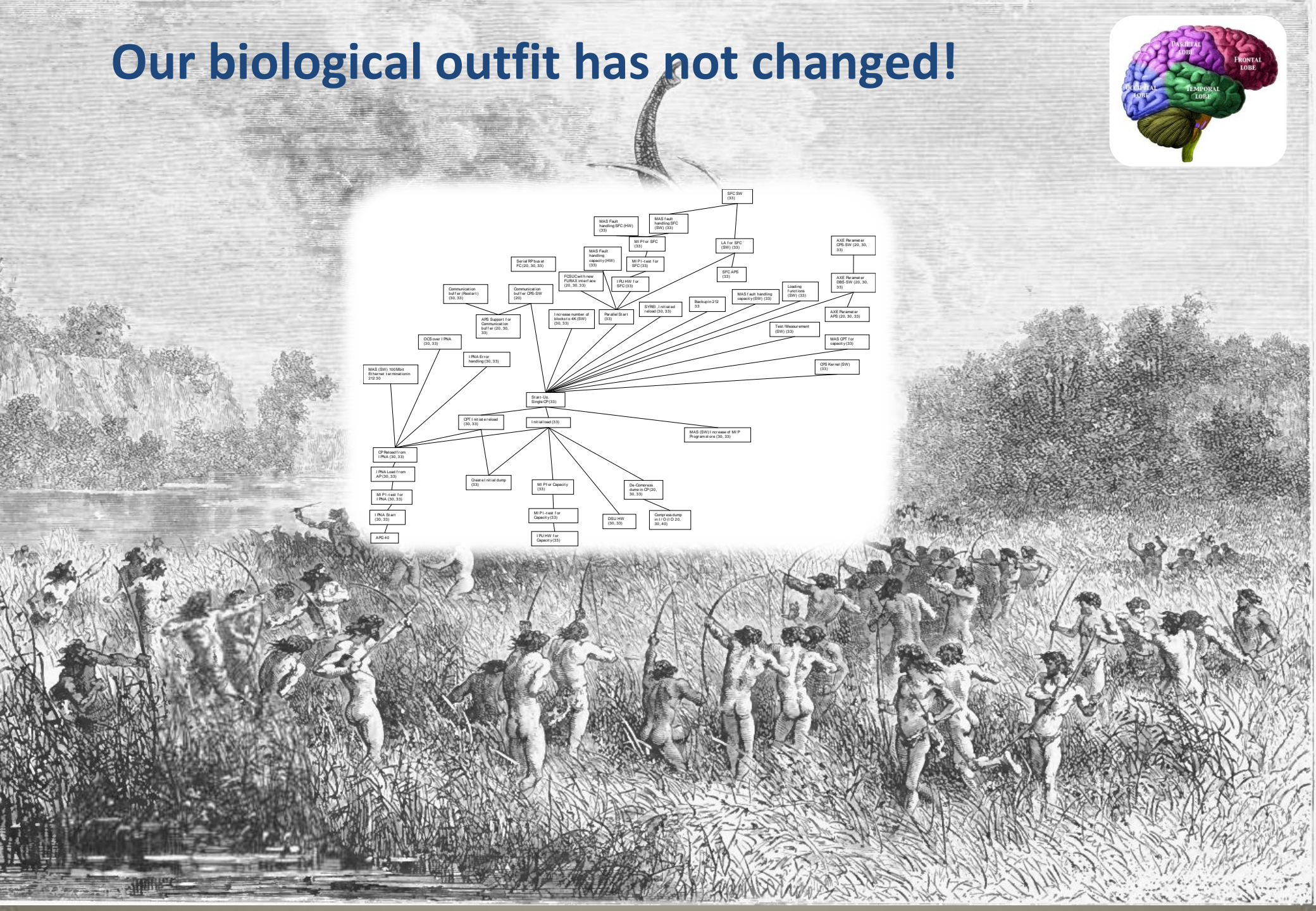
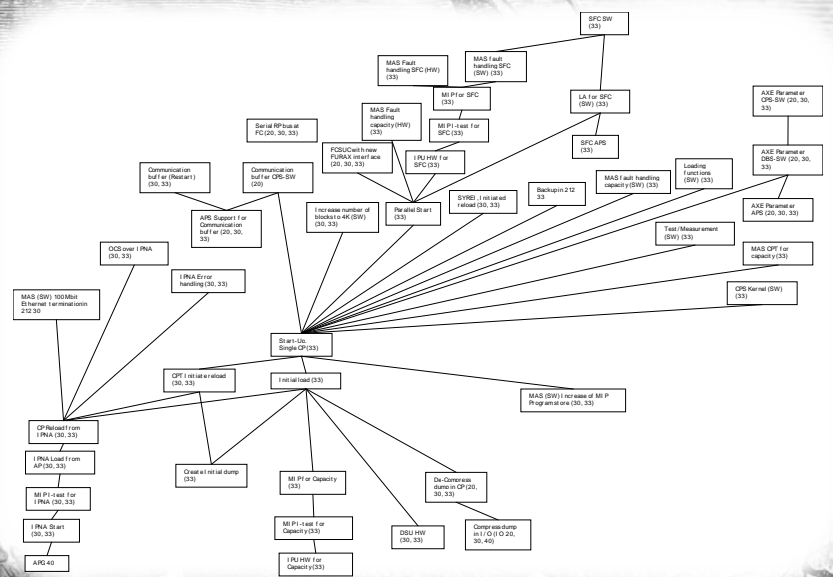
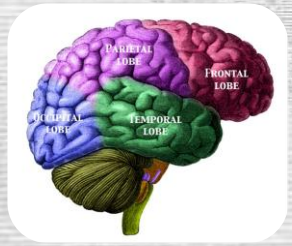


# PLM system anatomy



# Some takeaways

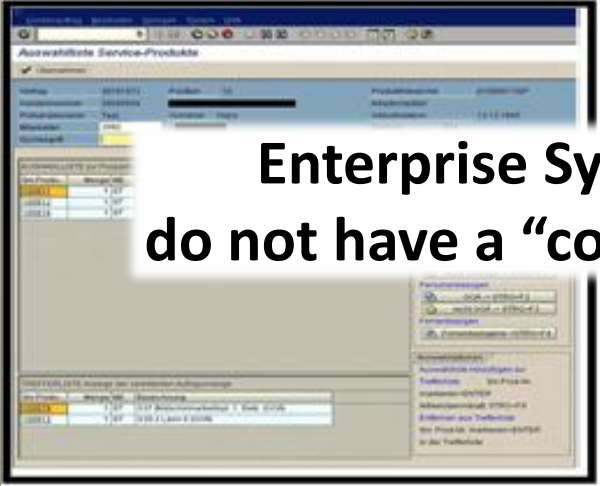
# Our biological outfit has not changed!



# Context is King!

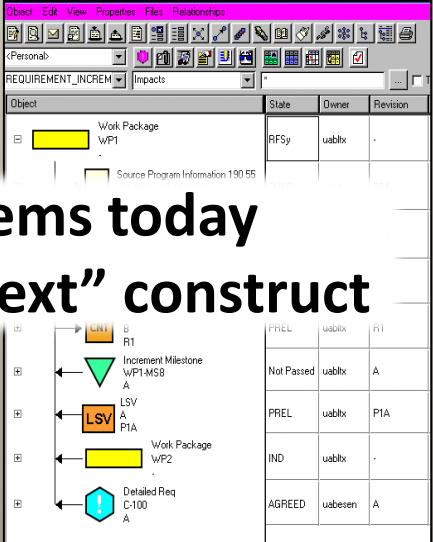


Tones!



**Enterprise Systems today do not have a "context" construct**

Tons!

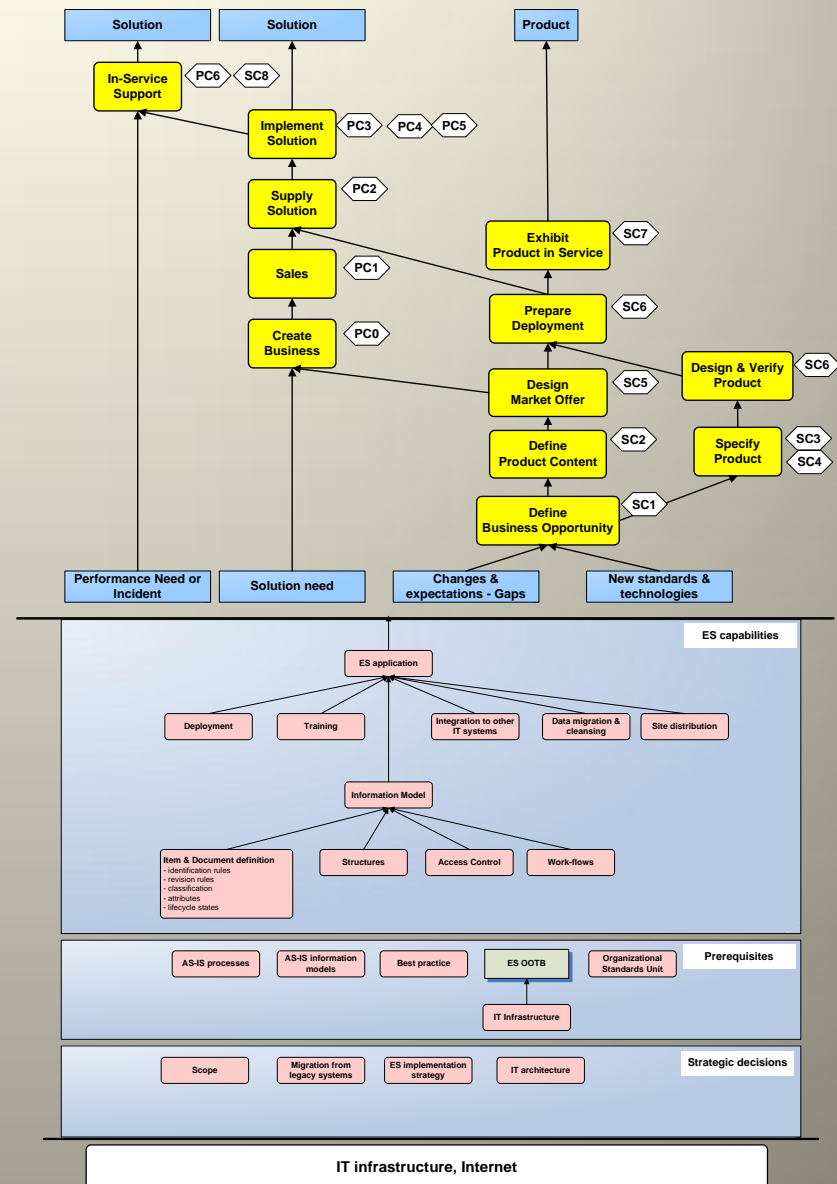




# An integrating visualization of the EA

One image where business and IT are visualized in a common notation: dependencies between capabilities

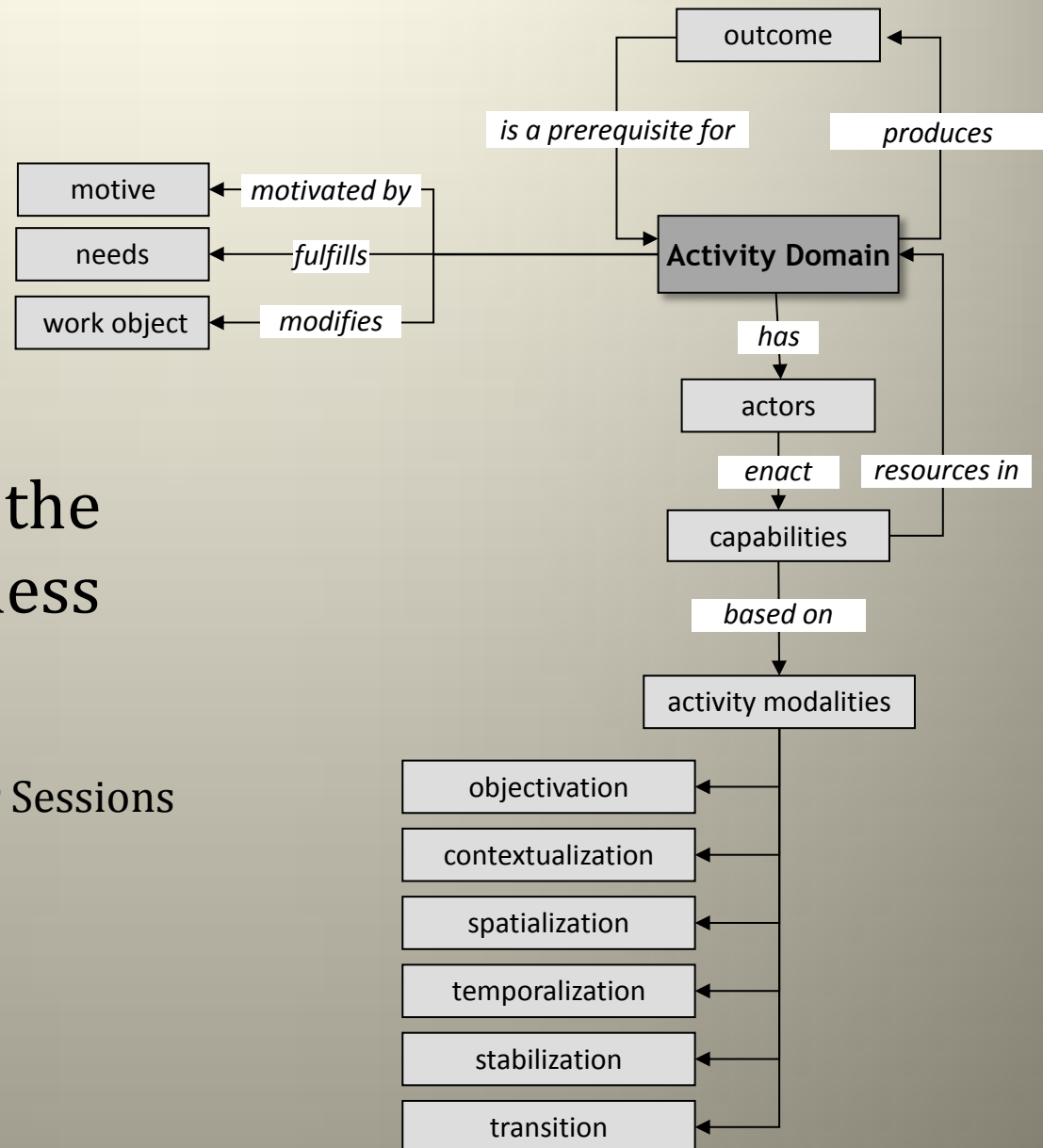
- Enterprise anatomy
- IT system anatomy
- Infrastructure anatomy



# A core Enterprise Architecture

Enterprise architecture, the science of aligning business needs and IT solutions

Roger Sessions



# **The Activity Domain Theory**

- An alternative approach to EA**
- Great potential**
- Work in progress!**

# The end is near ...



[lars.taxen@telia.com](mailto:lars.taxen@telia.com)

+46 73 09 77864