



# Functional Architecture

Session 11

Course ICT-Entrepreneurship

Prof.dr. Sjaak Brinkkemper



# Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture

# Architecture



## Definition:

*An **Architecture** is a high-level decomposition of a system into major components together with a characterization of the interaction of the components (van Vliet 2000)*

## Purpose:

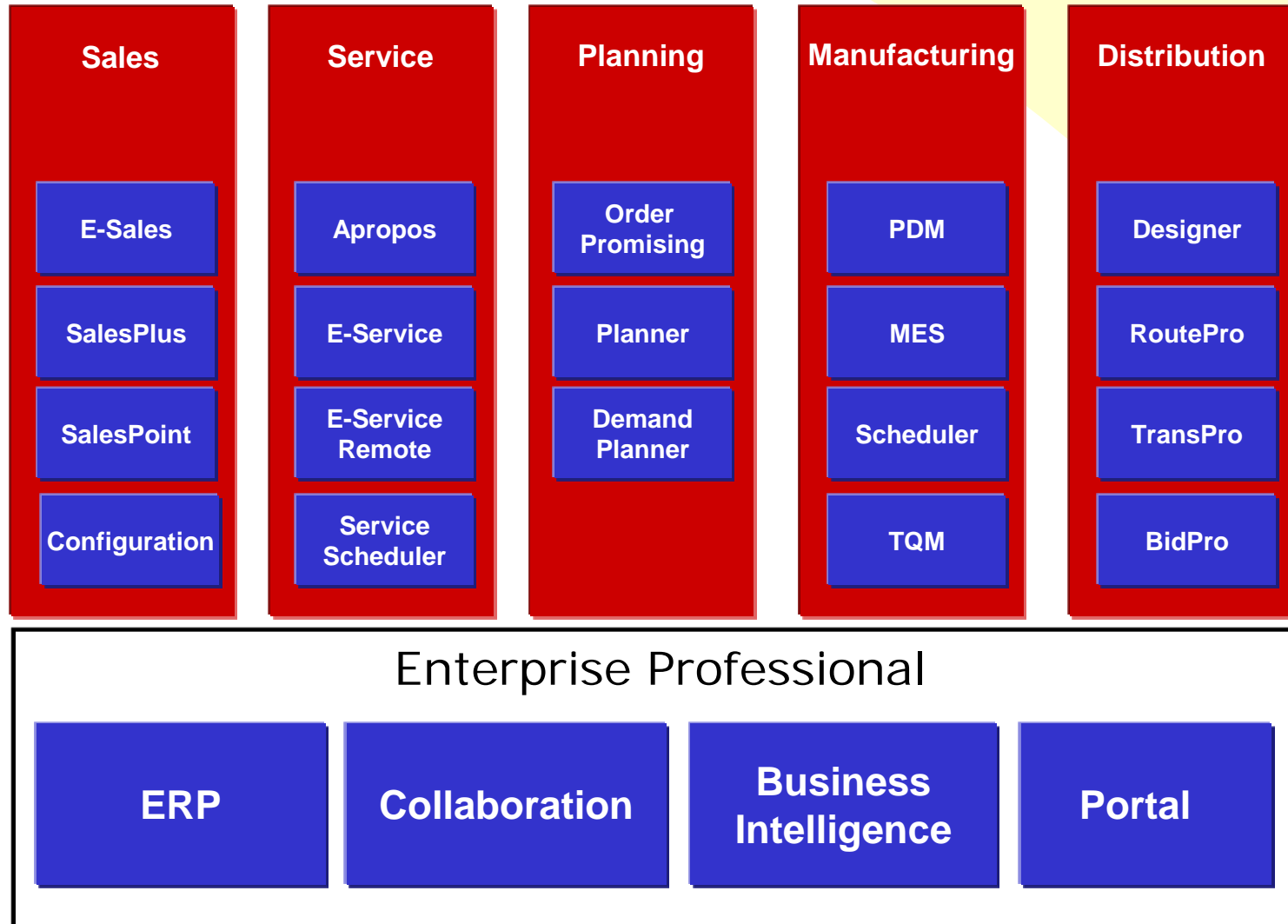
1. Communication among stakeholders  
Architects, Product Mgrs, Software Engineers, Testers, Sales&Marketing, Consultants, Customers
2. Capturing design decisions  
Rationale for later reference
3. Transferable abstraction for reuse  
Generalized constructs to be applied in other contexts

# Types of architecture



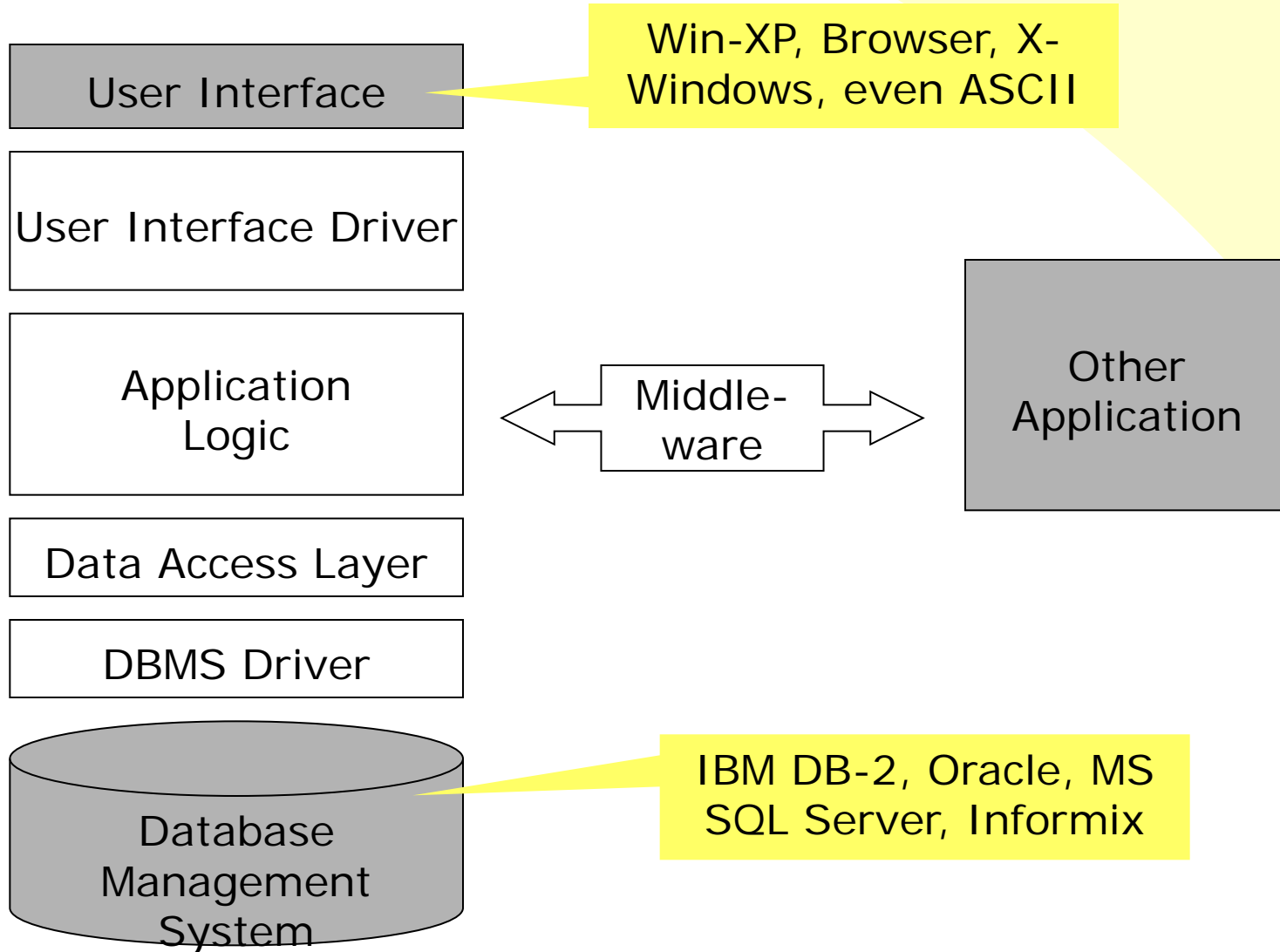
- Functionele architecture: usage perspective
  - Enterprise architecture
  - Application architecture
  
- Technical architecture: development perspective
  - Software architecture
  - Component architecture
  - System architecture
  - Data architecture
  - Data flow architecture
  
- Examples of different architectures on the next slides

# Functional architecture BaanERP



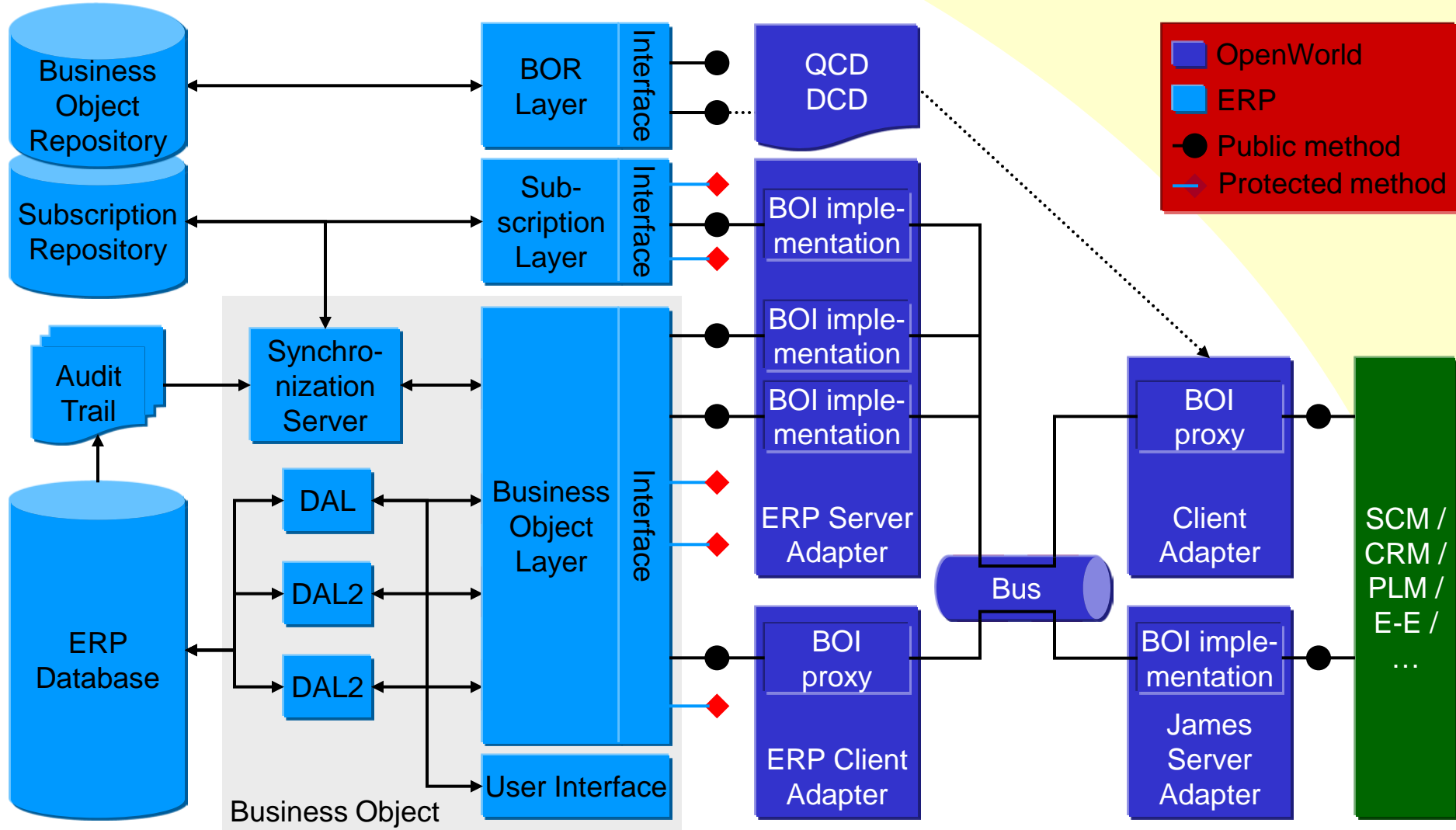


# High level technical architecture



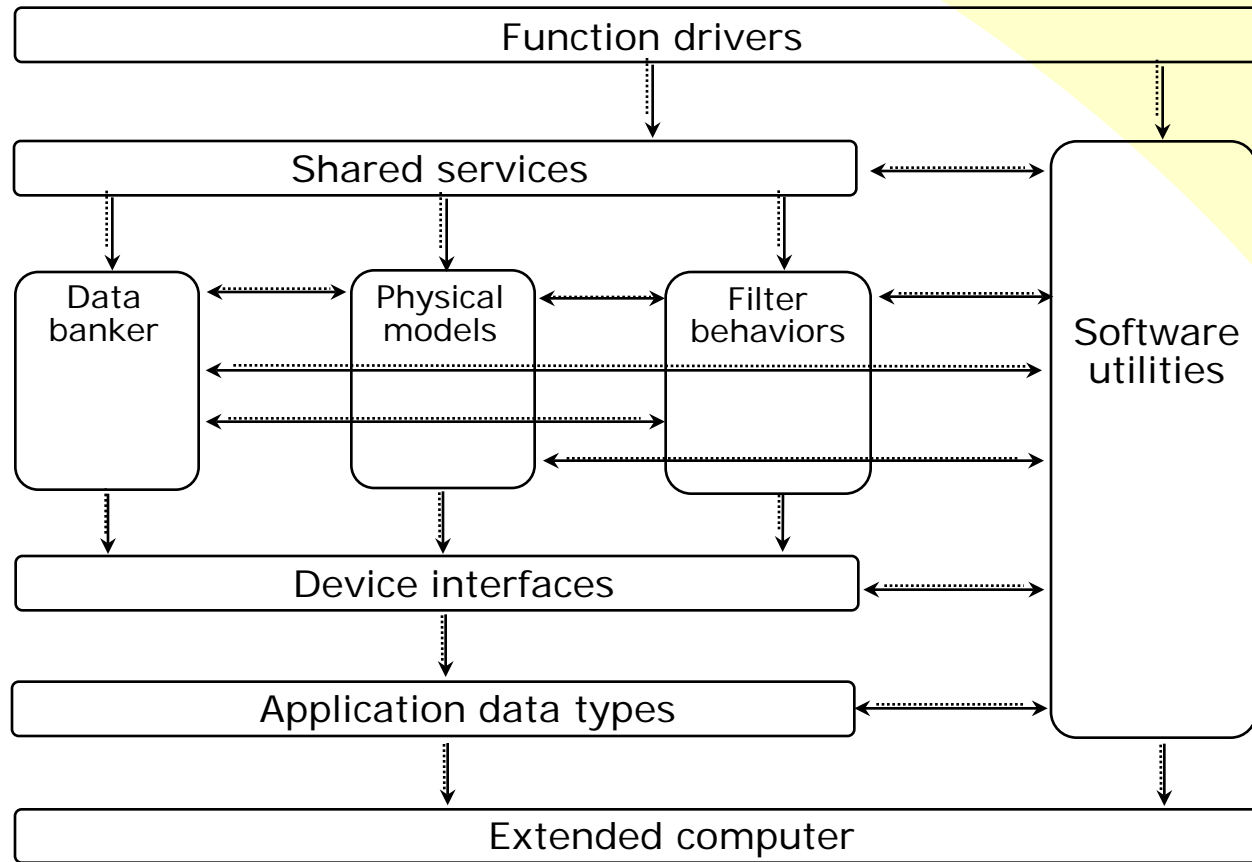


# Software architecture

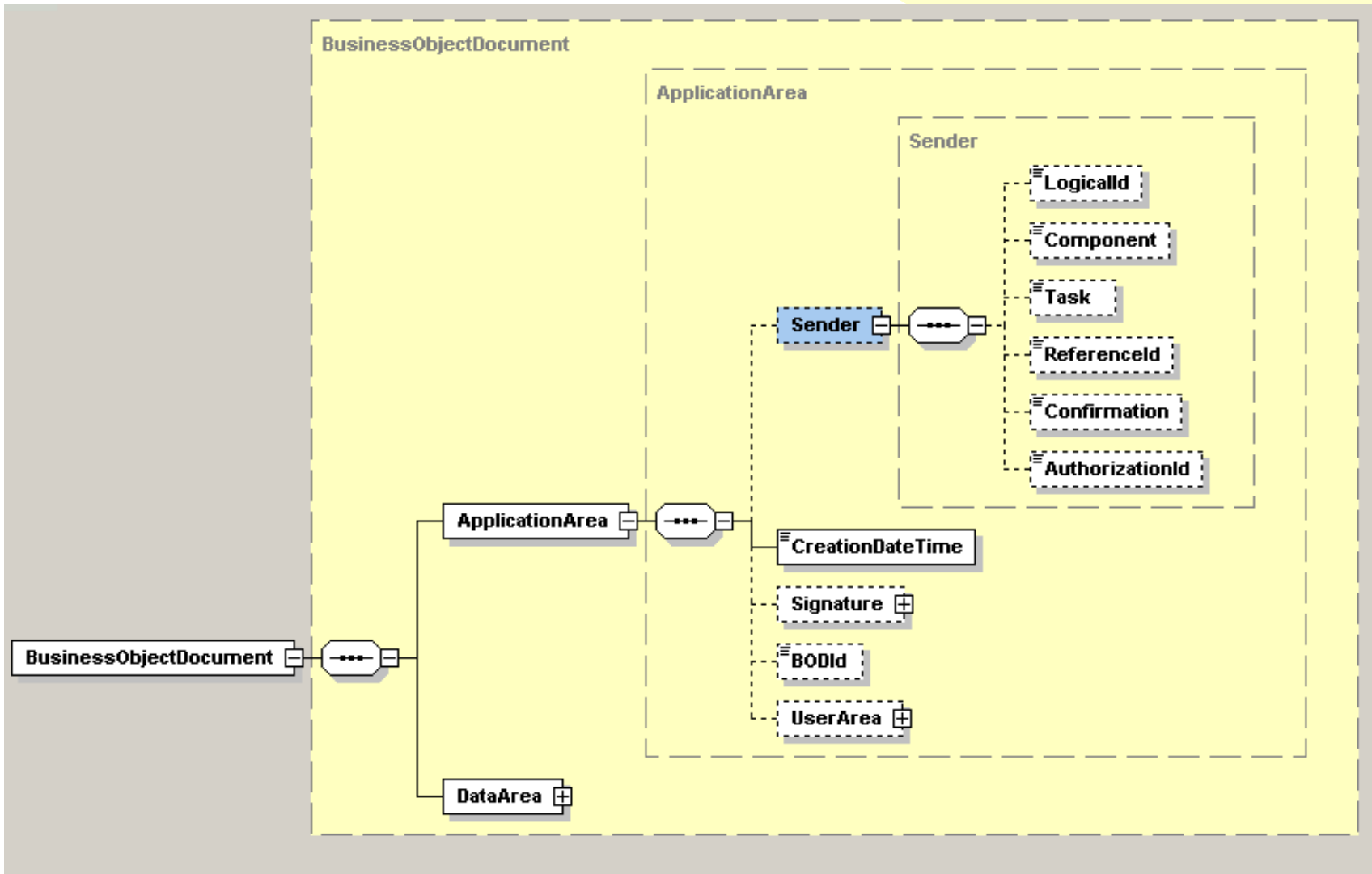




# Component architecture



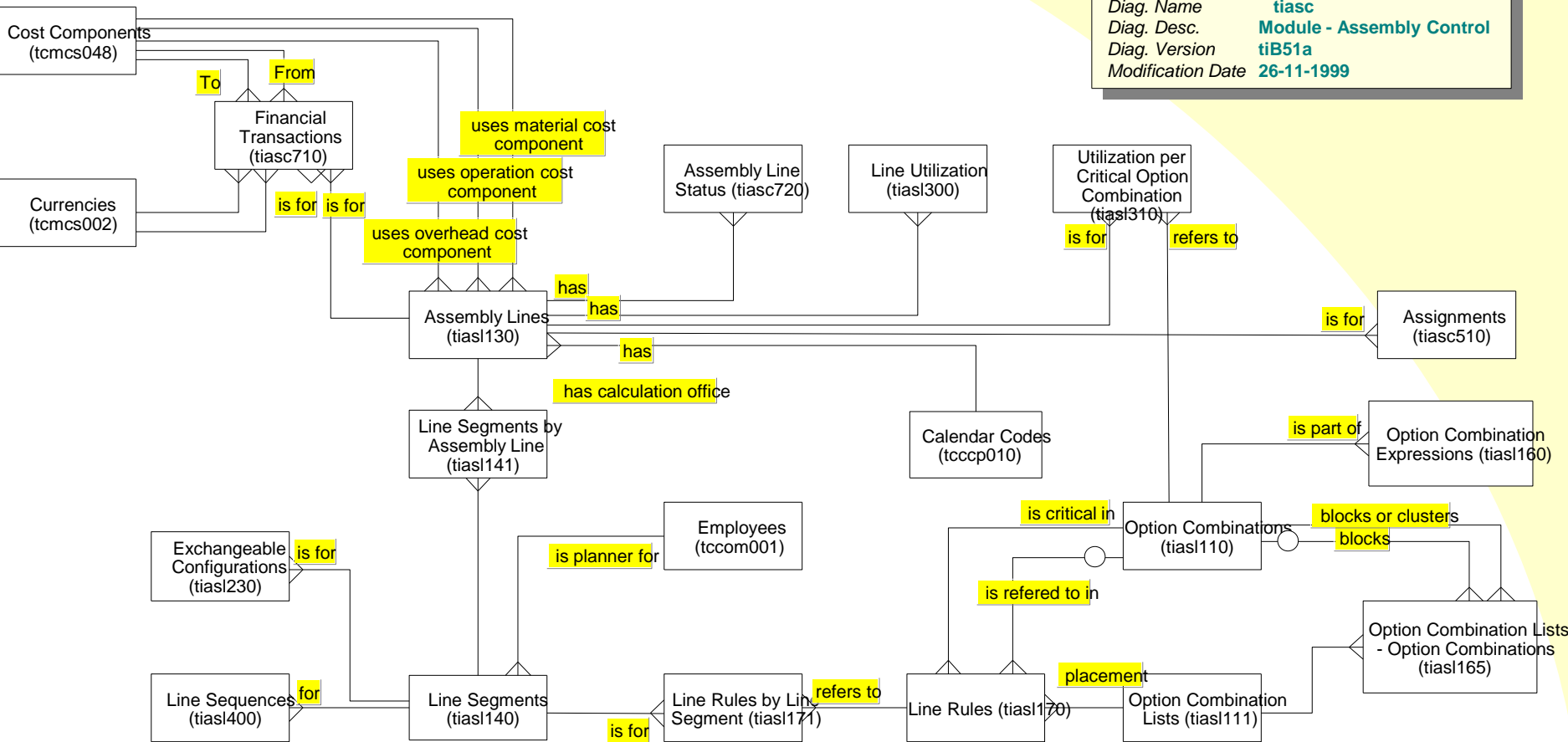
# Component Architecture



# Data Architecture

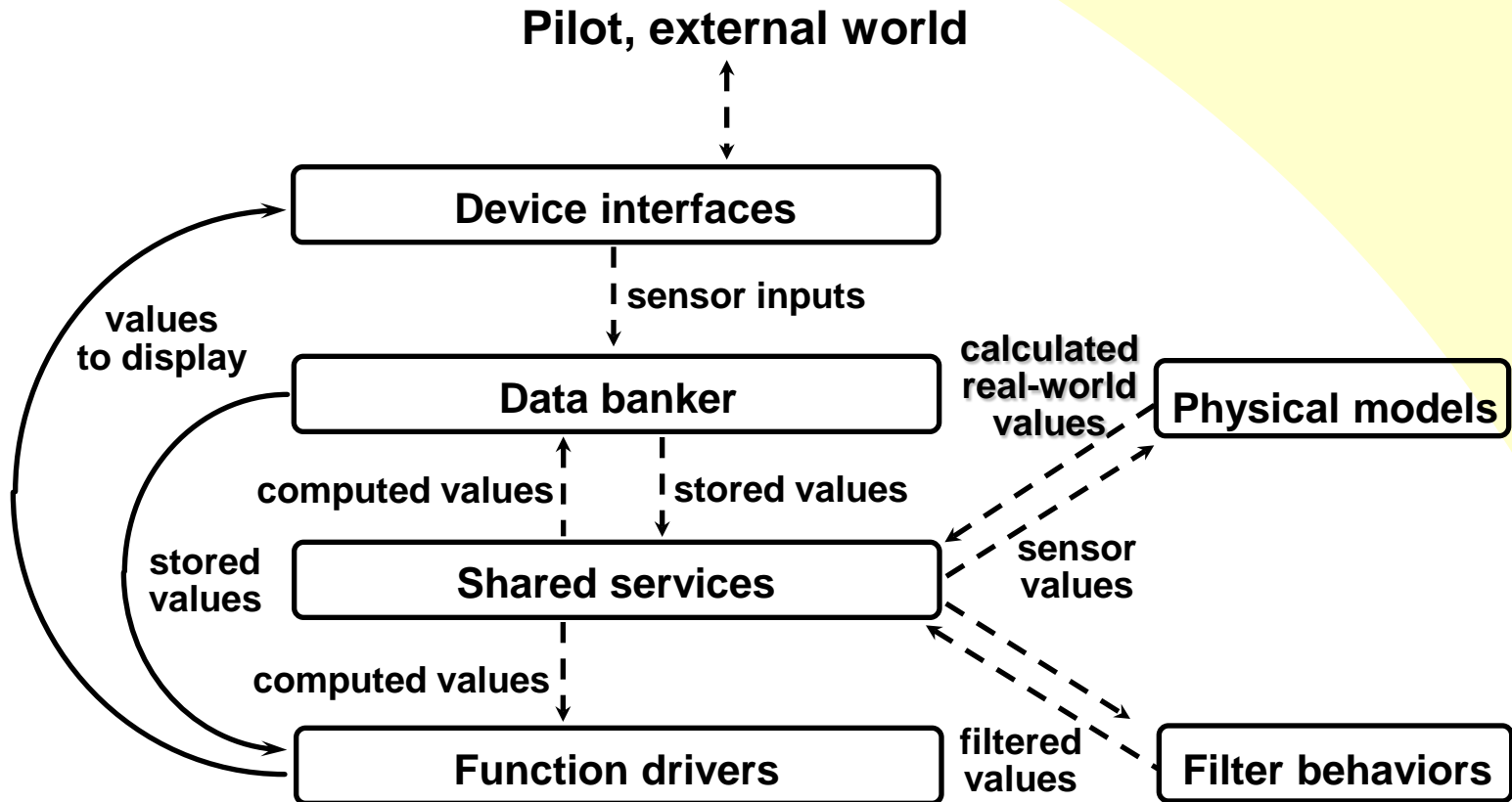


*Diag. Name*            **tiasc**  
*Diag. Desc.*          **Module - Assembly Control**  
*Diag. Version*        **tiB51a**  
*Modification Date*   **26-11-1999**





# Data Flow Architecture





# Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture

# What makes a good architecture?



# What makes a good architecture?



- Architecture is a premier key to the success of a product
  - Good architectures survive many releases
  - New functionality can easily be incorporated
  - Elegancy of a product is reflected in the elegancy of the architecture
- What makes a good design?
  - SAP R3, Google SE, MS Windows, Linux, ...
  - No scientific evidence of good architectures
  - Some evidence for other domains: electronics, building, civil engineering

# Influences on architecture



- Requirements of product
  - Functionality
  - Technical context
- Development organization and history
  - Earlier versions
  - Other systems
- Background and expertise of architect
- Technical environment
  - Design tools available
  - Development platform

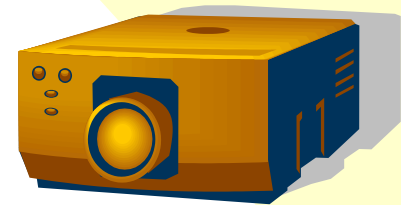


# Design structures



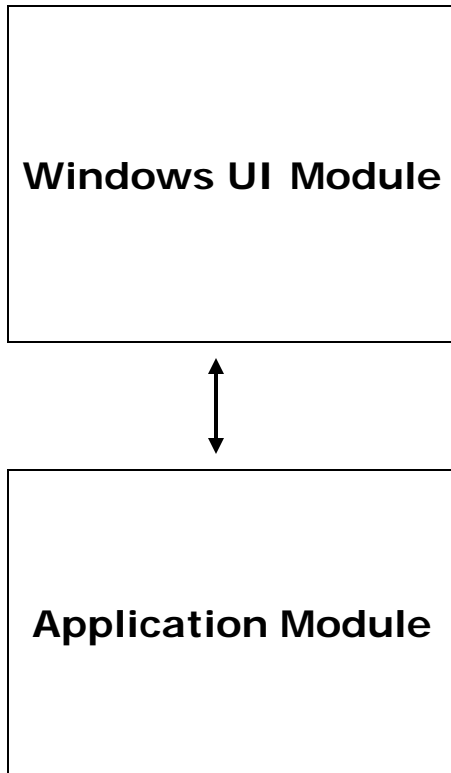
Design structures for the functional architecture

- A. Modularity: structure and flexibility
  - What are the modules?
  - For now and for in the future?
  
- B. Variability: multiple platforms and products
  - What variability is to be distinguished?
  - Both technical and functional
  - In which modules resides the variability?
  
- C. Interoperability: interfacing externally
  - What interfaces are needed?
  - Standards or dominant players available?
  - Positioning of interfaces in architecture?

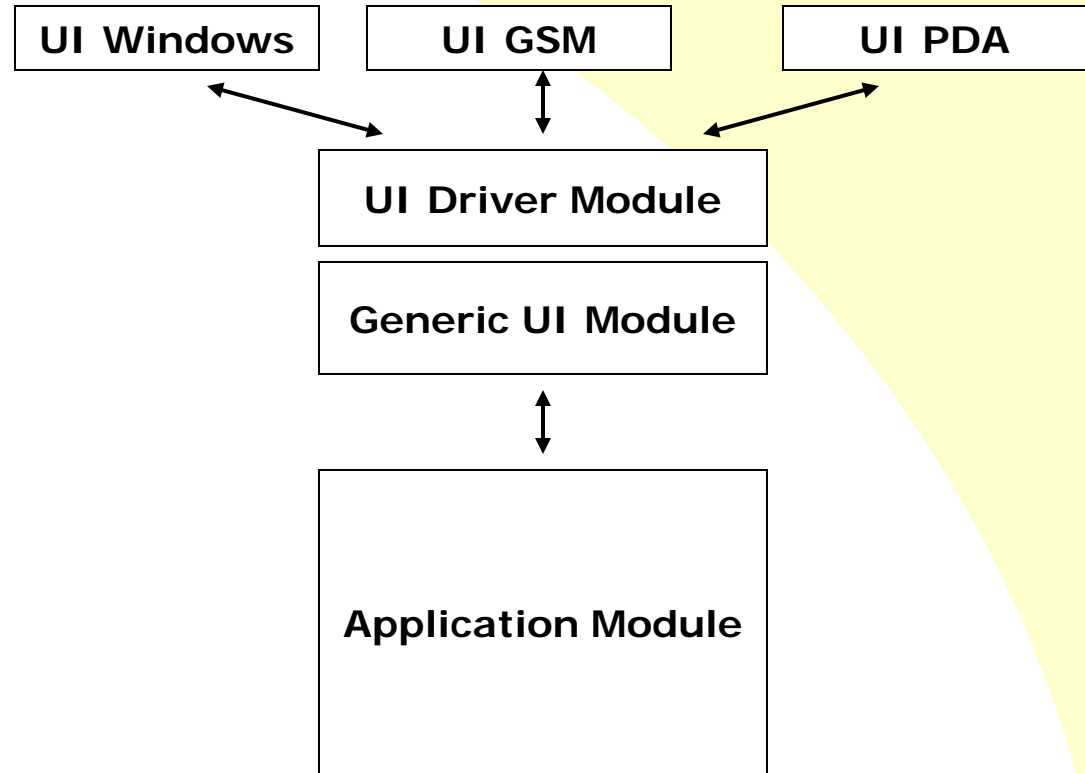


*Discuss these on a beamer*

# Example: Technical modularity

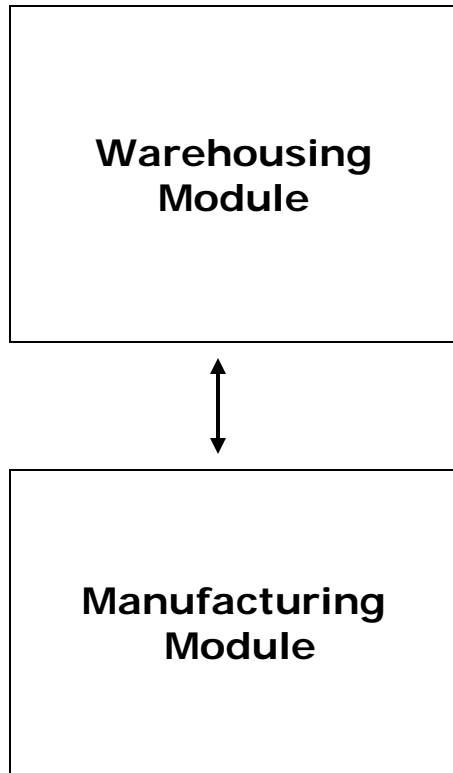


A. Single UI platform

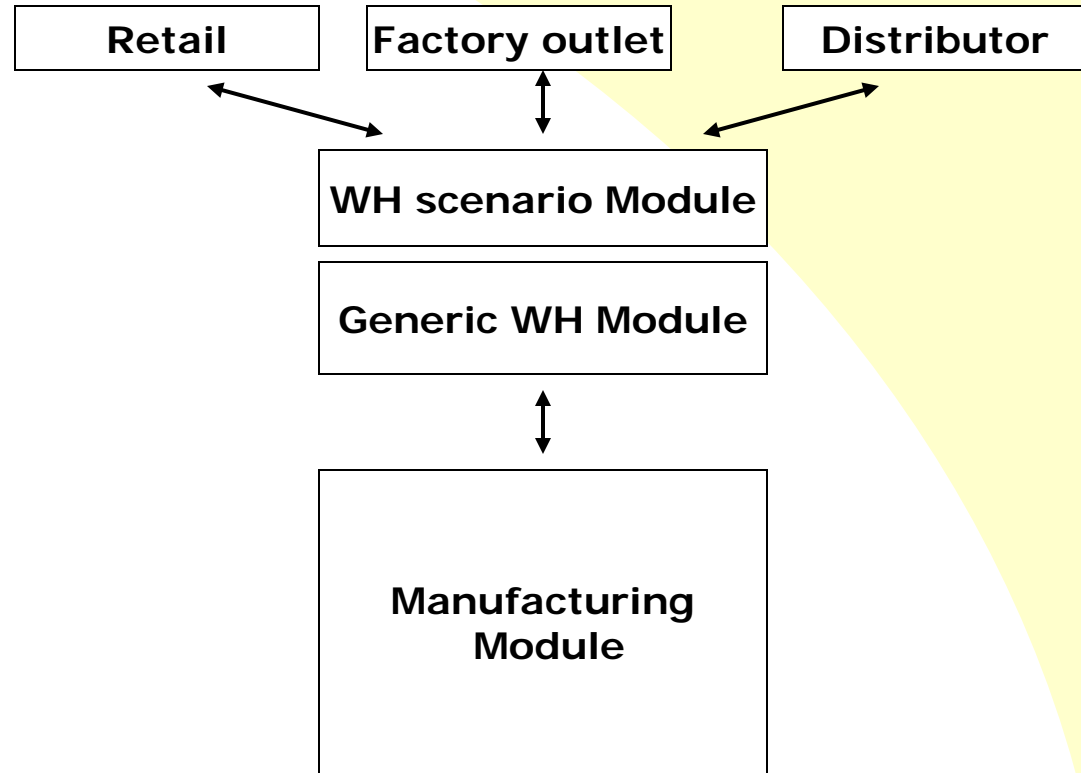


B. Multiple UI platforms

# Example: Functional modularity



A. Single Warehousing scenario



B. Multiple Warehousing scenarios



# Contents

- What is architecture?
- Design structures in architecture
- **Functional Architecture**
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture

# Product Software Architecture



## 1. Functional Architecture

- use Enterprise Function Diagram

## 2. Technical Architecture

- use techniques from other courses
- UML
- software architecture diagram



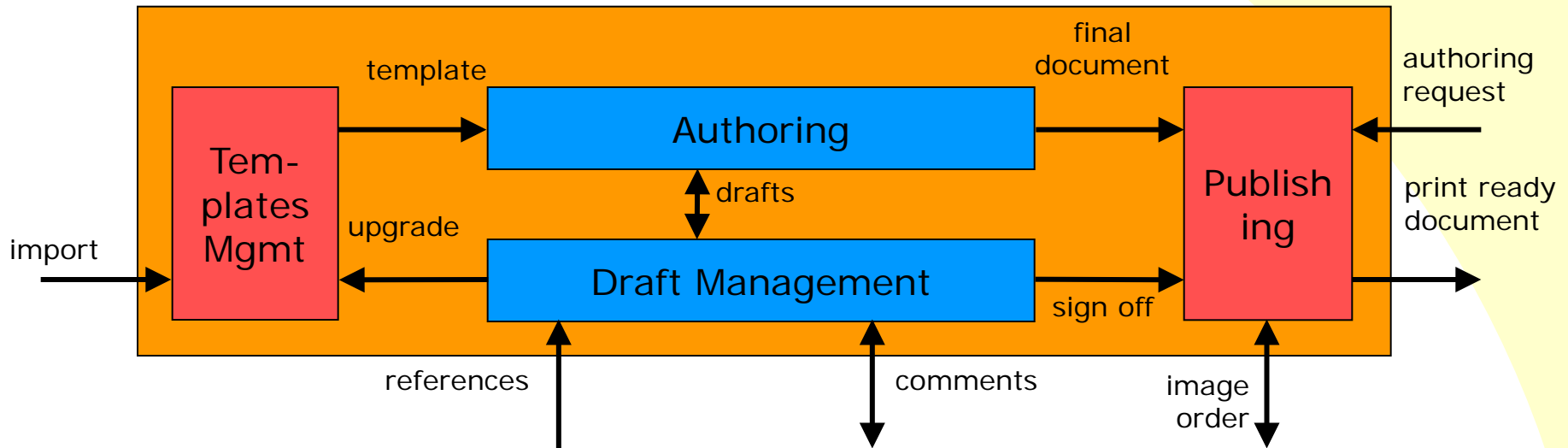
# Functional Architectures

- A **functional architecture** is an architectural model from a **usage** perspective
- So the functional architecture should **resemble** the enterprise functions of the **customer organisation or user context**
  - Names of **modules** should resemble the names of **enterprise functions**
  - **Flows** in functional architecture resemble the **interactions** in the customer domain
- Standard functional architecture is called **Reference Architecture**
- More theory on these types of models is presented in the course **Enterprise Architecture**

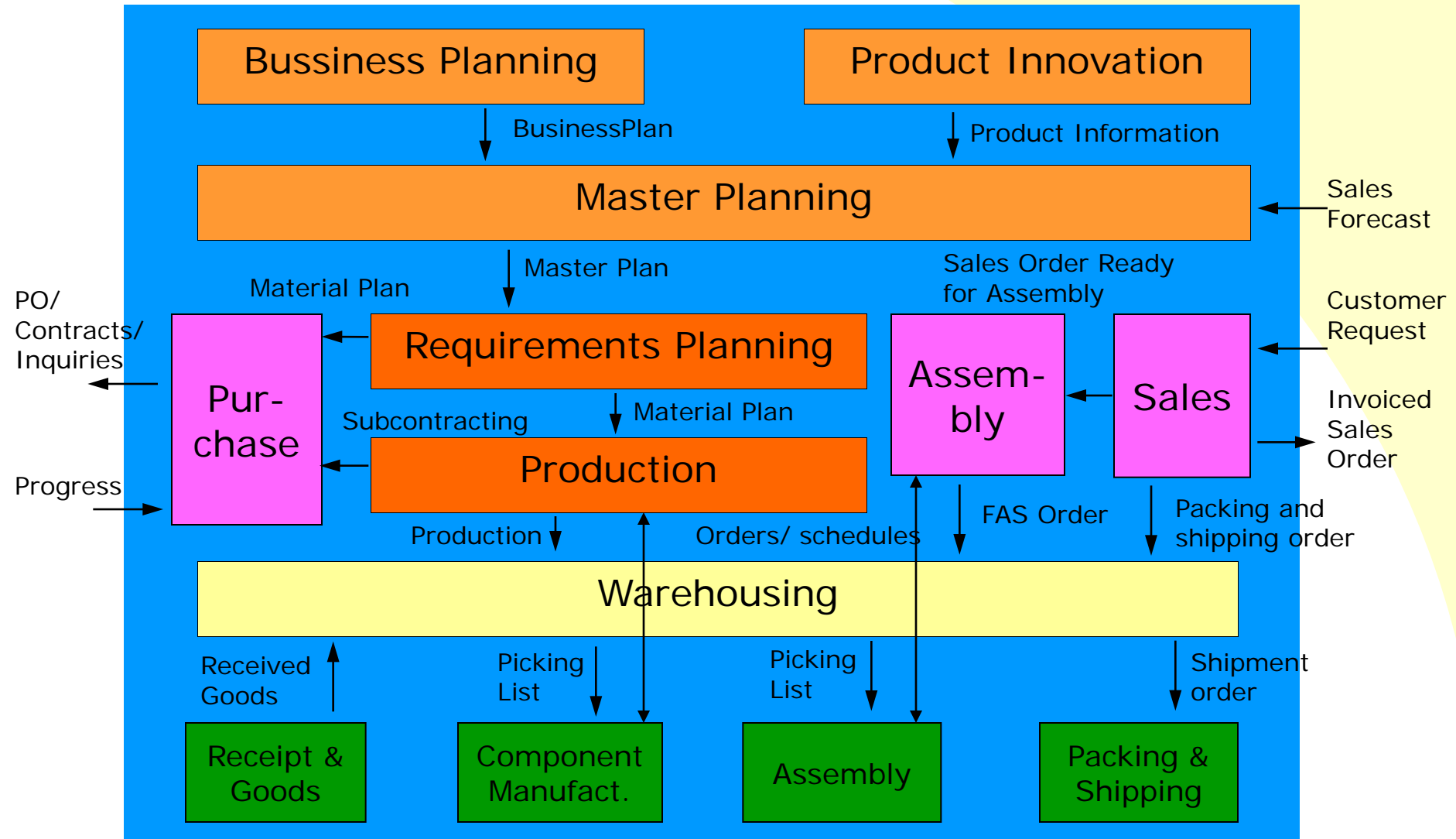
# Example of Functional Architecture



On-line authoring tool



# Example of Functional Architecture for ERP product



# Enterprise Function Model definition



## Enterprise Function Model

*An Enterprise Function Model is the representation of the primary process of an enterprise, consisting of its physical and administrative functions*

The Enterprise Function Model also helps to:

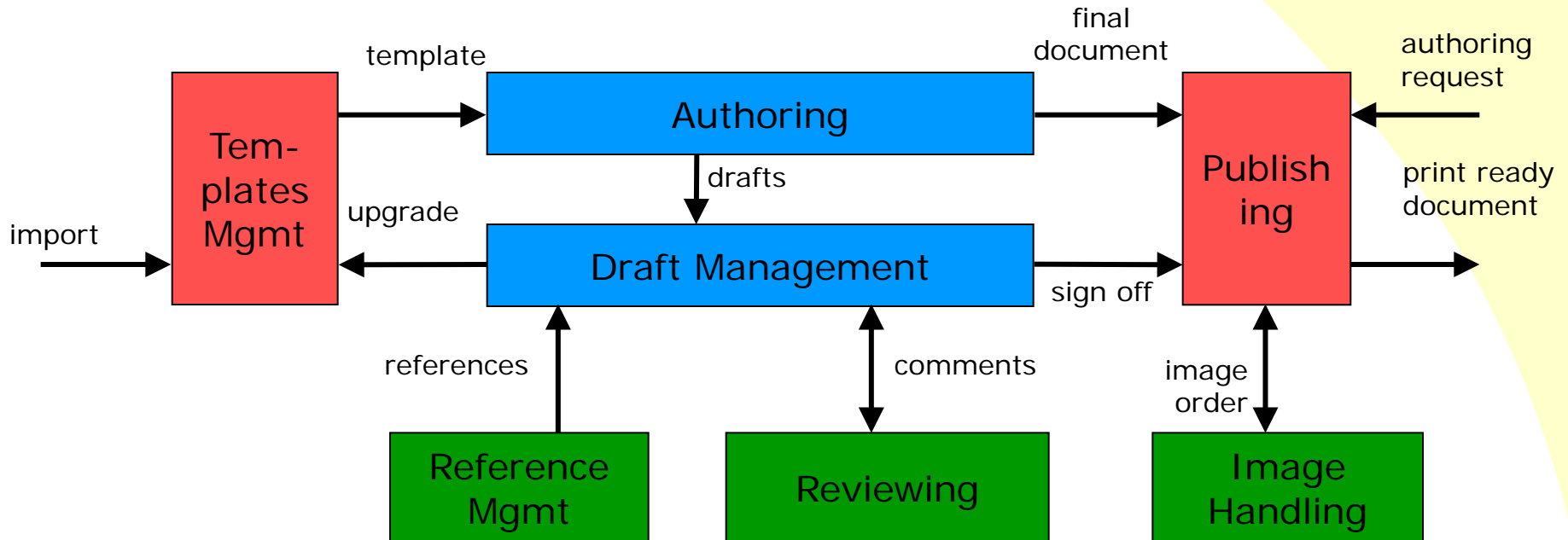
- identify the main functions
- show the interaction of the enterprise functions
- create a clear overview of the enterprise as a whole

EFD: Enterprise Function Diagram



# Product usage scope

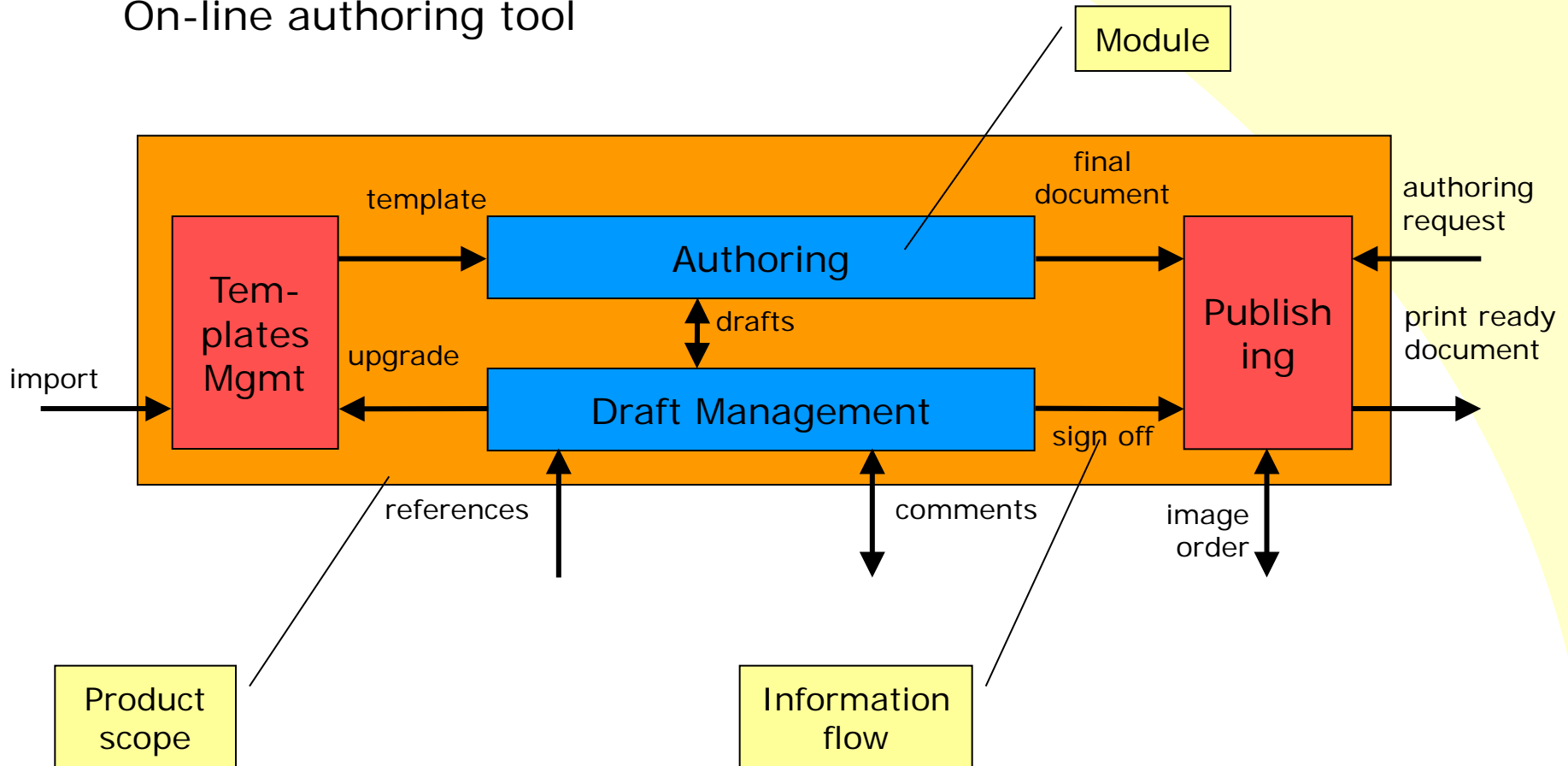
Example: Collaborative authoring





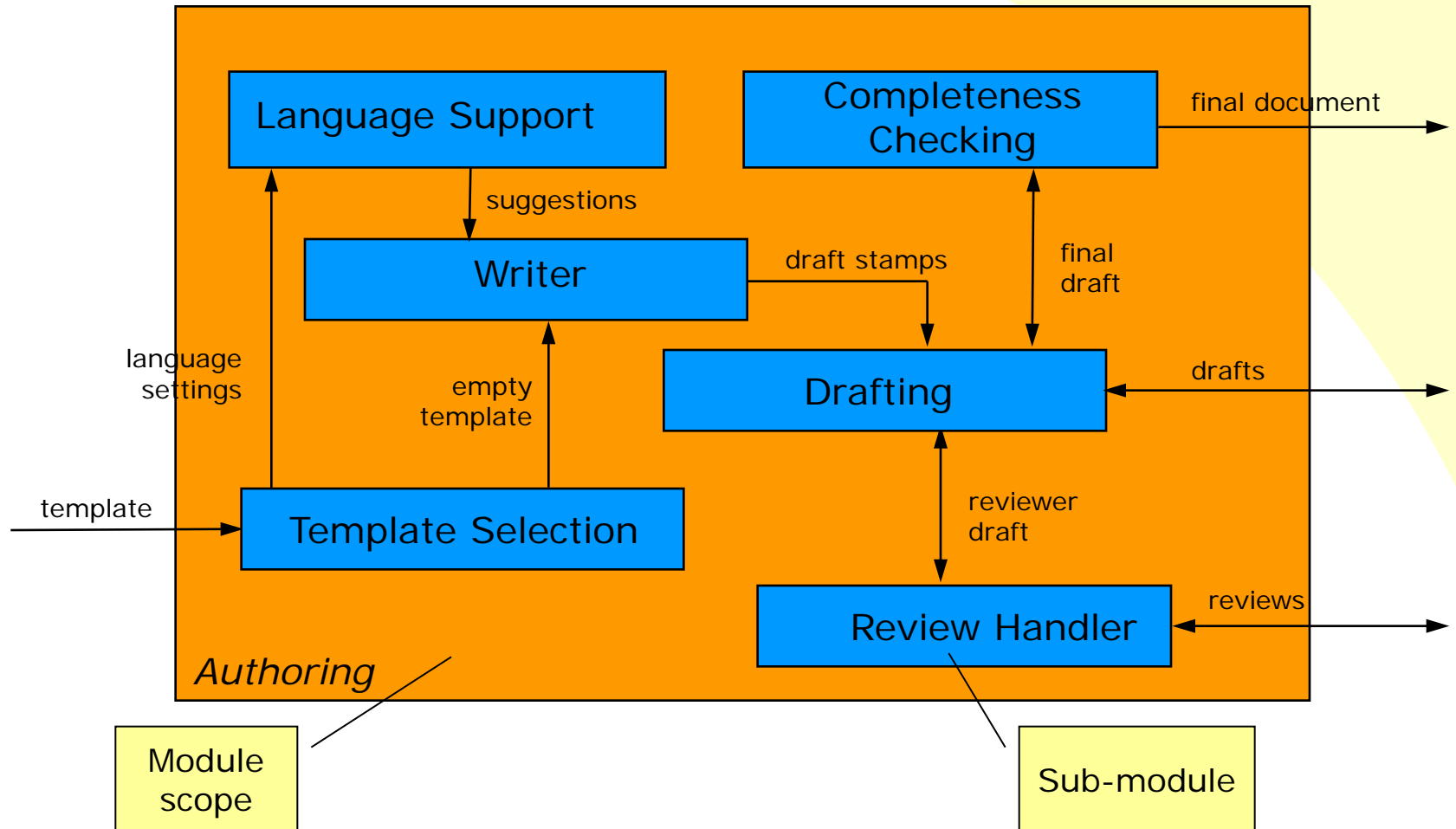
# Functional Architecture

On-line authoring tool



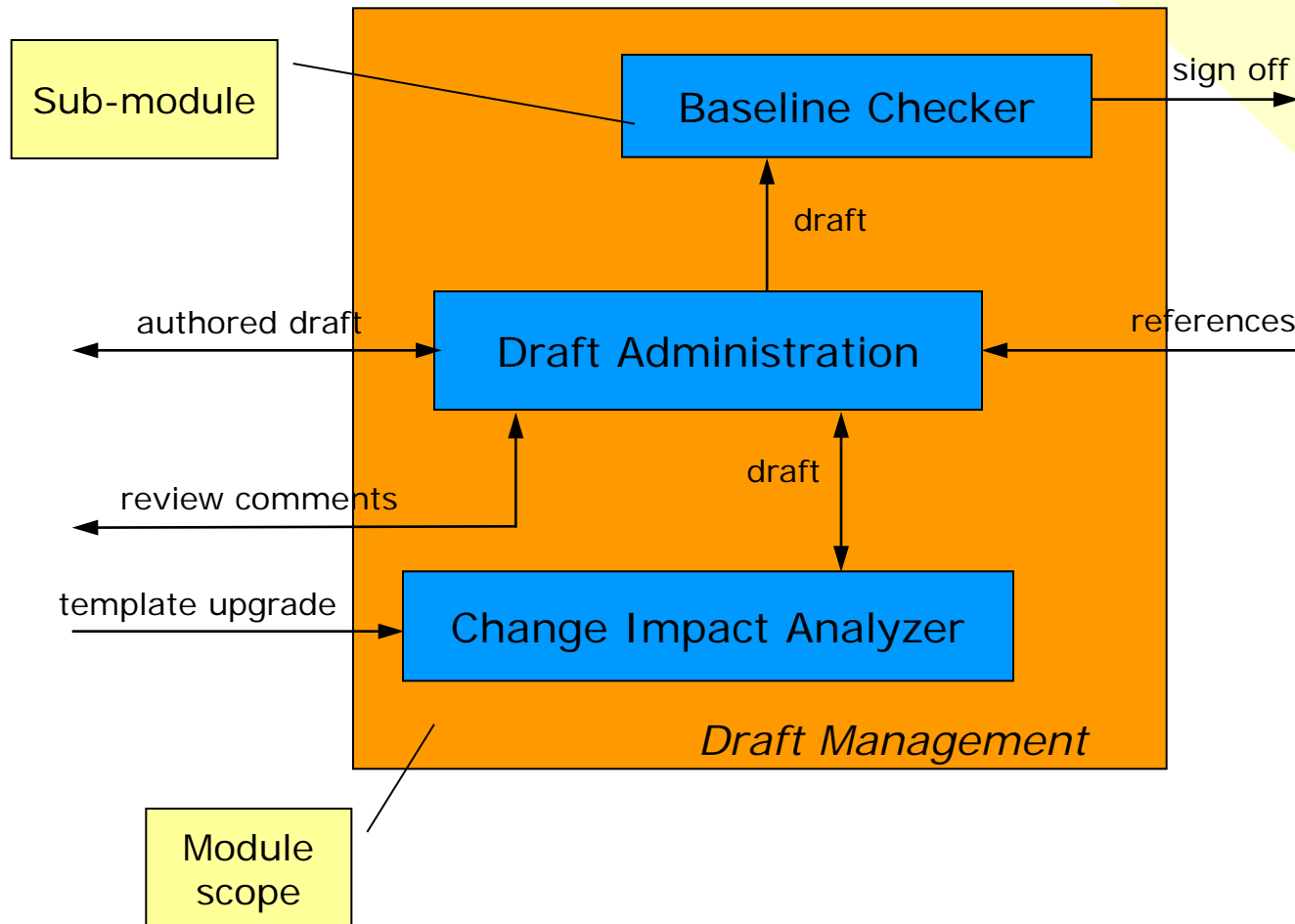
*Note: We assume it has been decided that Reference management, Reviewing and Image handling are not in the scope of the product*

# FA on module level: Authoring



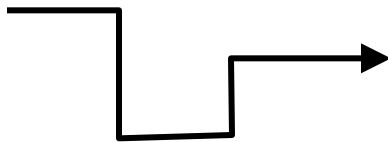
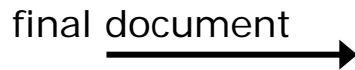


# FA of Draft management





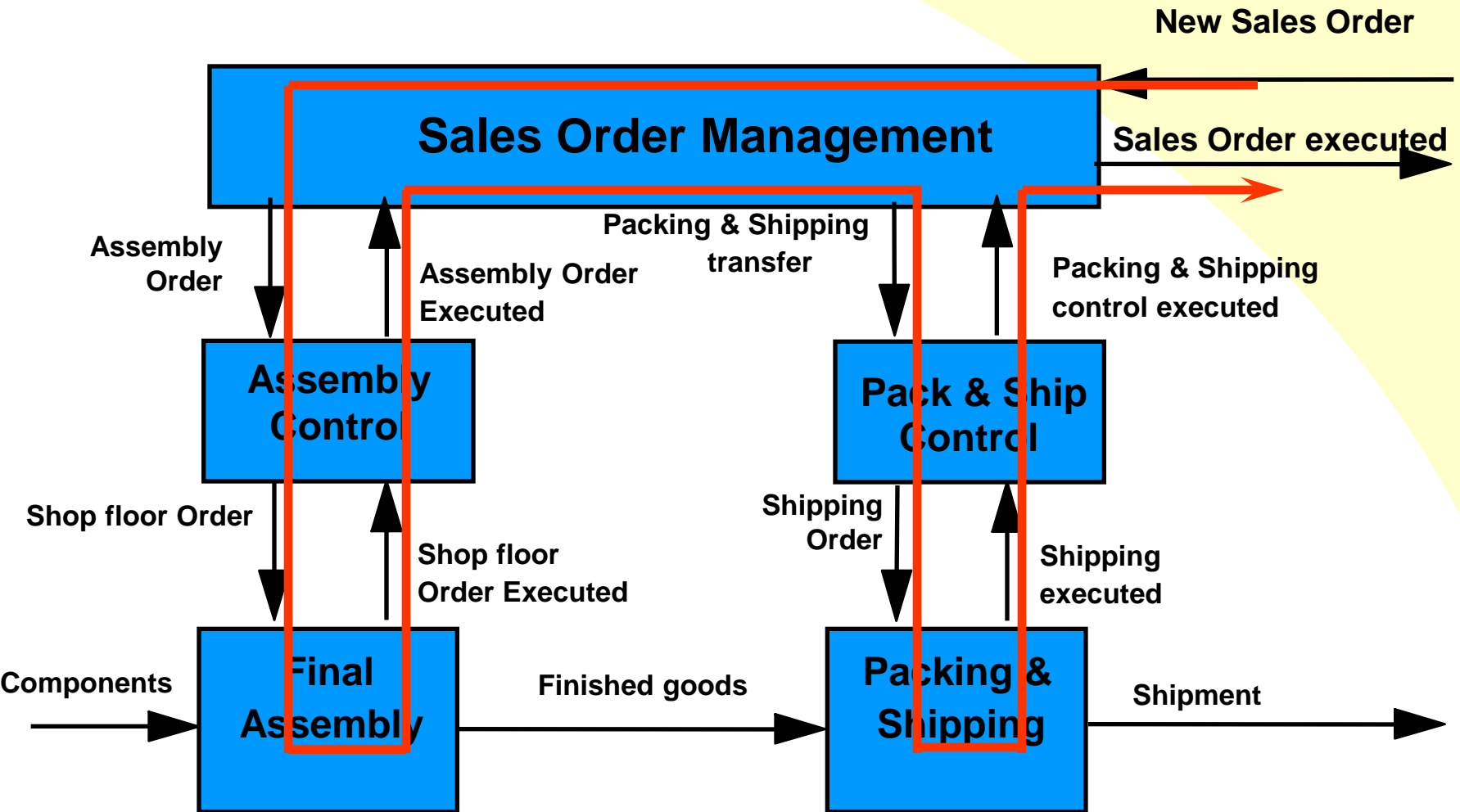
# Notation of EFDs



- Module or sub-module
  - Color is used for categorization
  - All words start with Capital
- Flow
  - all lower case
- Scenario
  - as overlay on EFD



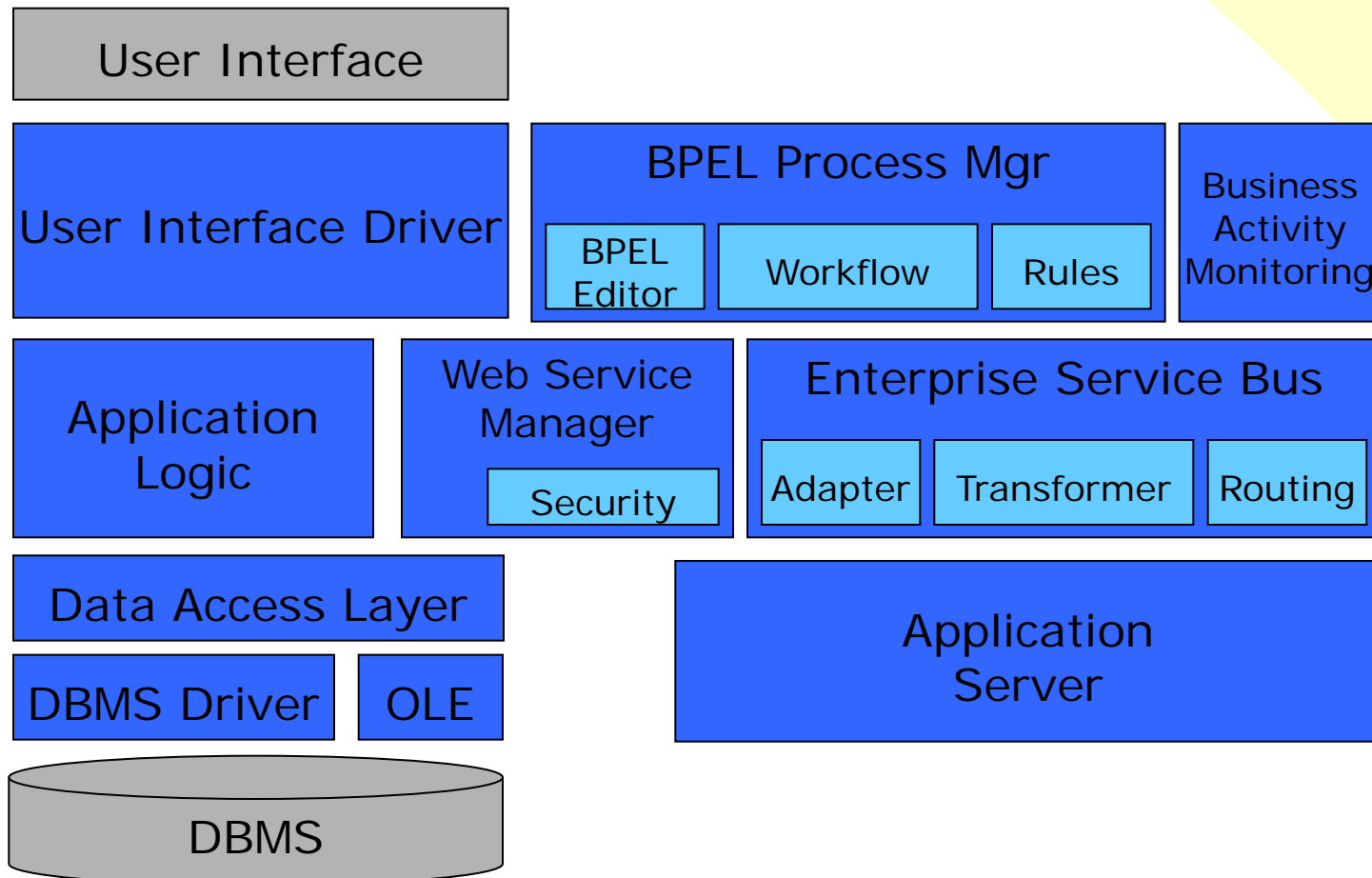
# Scenario on EFDs





# Technical systems

*FA can also be provided for technical systems*



# Identifying Enterprise Functions



- What is an Enterprise function?

*Definition: An enterprise function is a collection of **coherent processes, continuously performed** within an enterprise and supporting its mission*

- Examples:

Corporate Planning

Human Resource Management

Supplier Contract Management

Shop Floor Control

- Naming standards:

– Use substantivised nouns: Planning in stead of Plan

– Precise, determining words known in the business domain

– Name is Capitalized

# Identifying Functions (2)



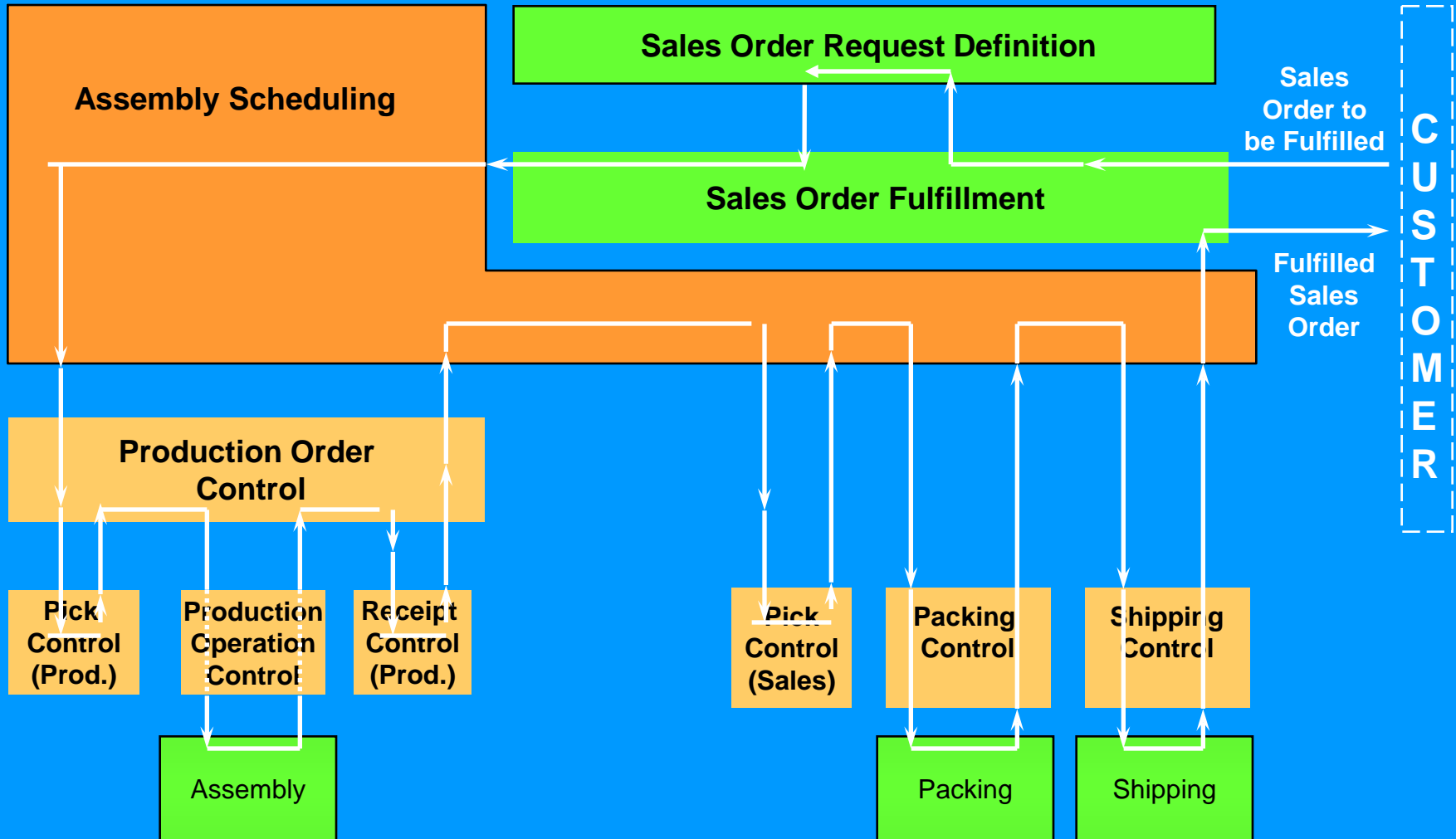
## Function Typology

- Life cycle stadium of a product or service:  
Design, Manufacturing, Distribution, Purchase
- Supporting resources:  
Human Resource Management, Facility Management
- Control and planning of the company core activities:  
Administration, Finance, Credit Control

Usually two levels are enough for Functional Architectures



# Identification of interfaces





# Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - **Decomposition**
  - Positioning
- Reference Architectures
- Creating a Functional Architecture

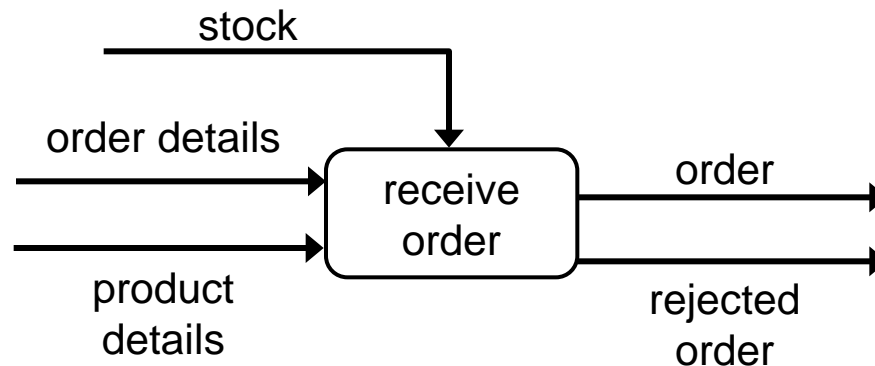


# Decomposition

- What is a process?

*"A business process is an activity of which the **execution** can be described in terms of **needed** and **delivered** data and of which the **start and end** can be determined."*

- A process is the WHAT a company does
- Not HOW it is being done.





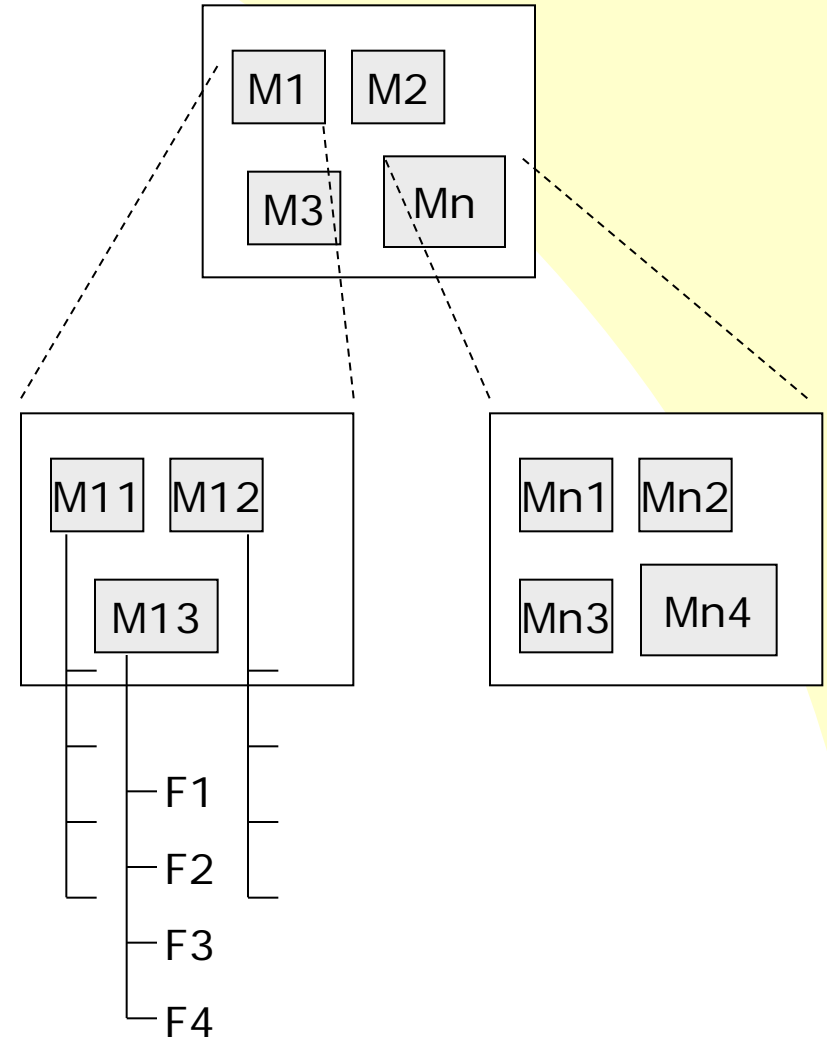
# Functions versus Processes

- Examples of Processes:
  - accept order
  - create delivery schedule
  - receive payment
  - pay employee
  - verify customer credit rating
- Lower level of detail compared to a function
- No realization (how) information
- Processes are modeled in the Feature Model



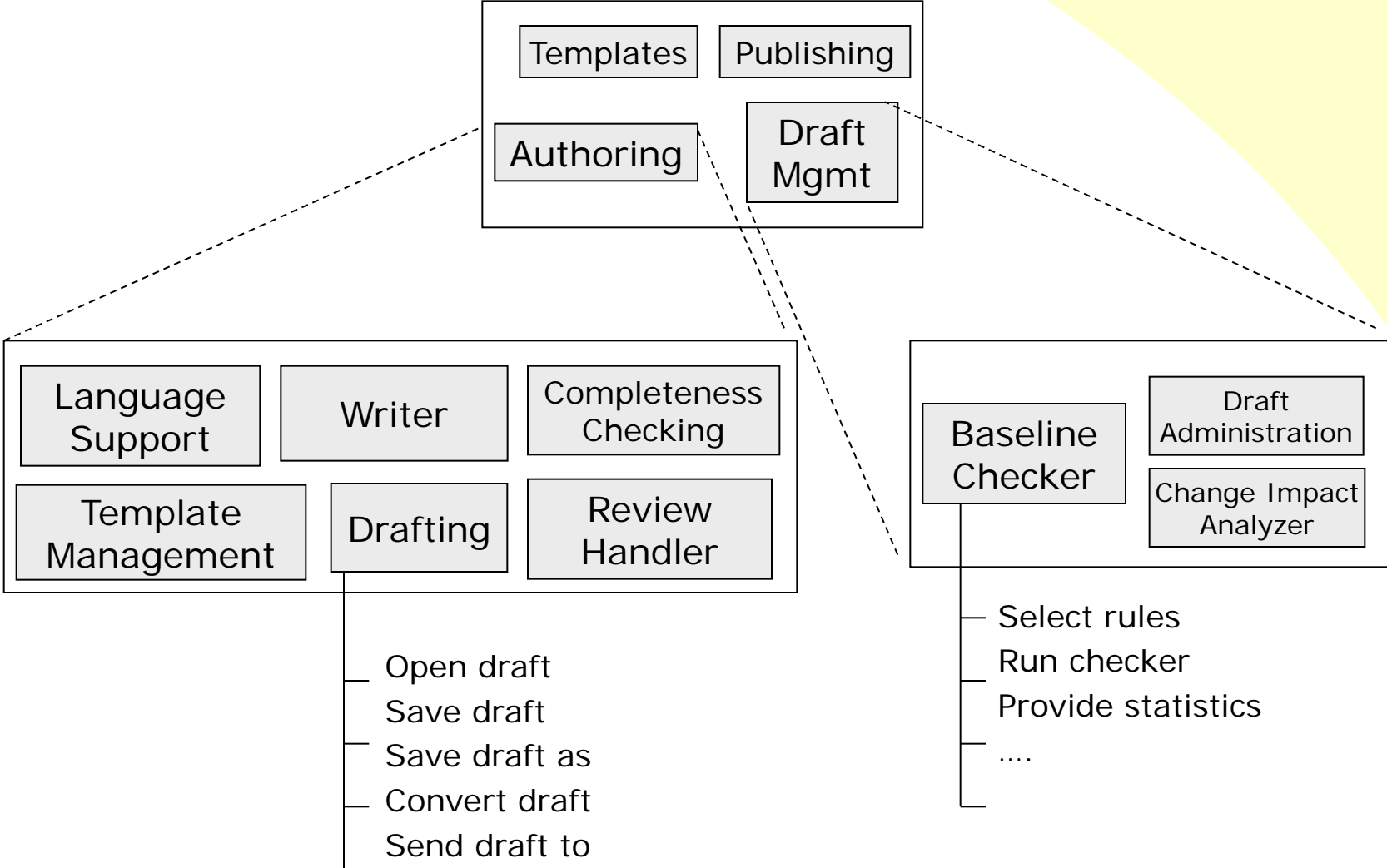
# Functional Decomposition

- The modules in a functional architecture are modelled in **2 or 3 layers**
- At the lowest level the module consists of **features**
- Definition: **Feature** is a **discrete unit** of unique and attractive **functionality** of a product that delivers measurable **benefit** to customers
- The lowest level modules are elaborated in a **feature model**



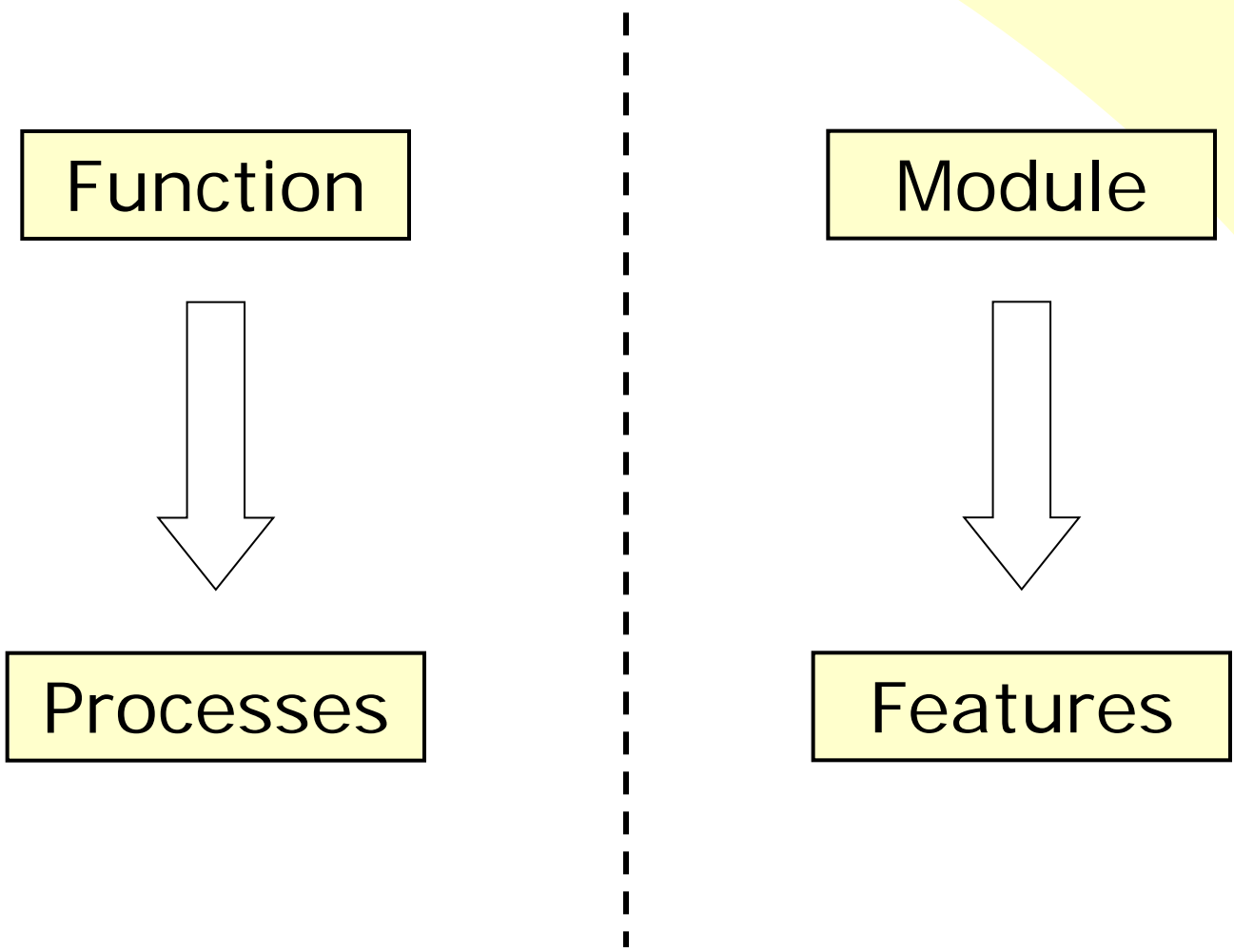


# Example: Authoring tool



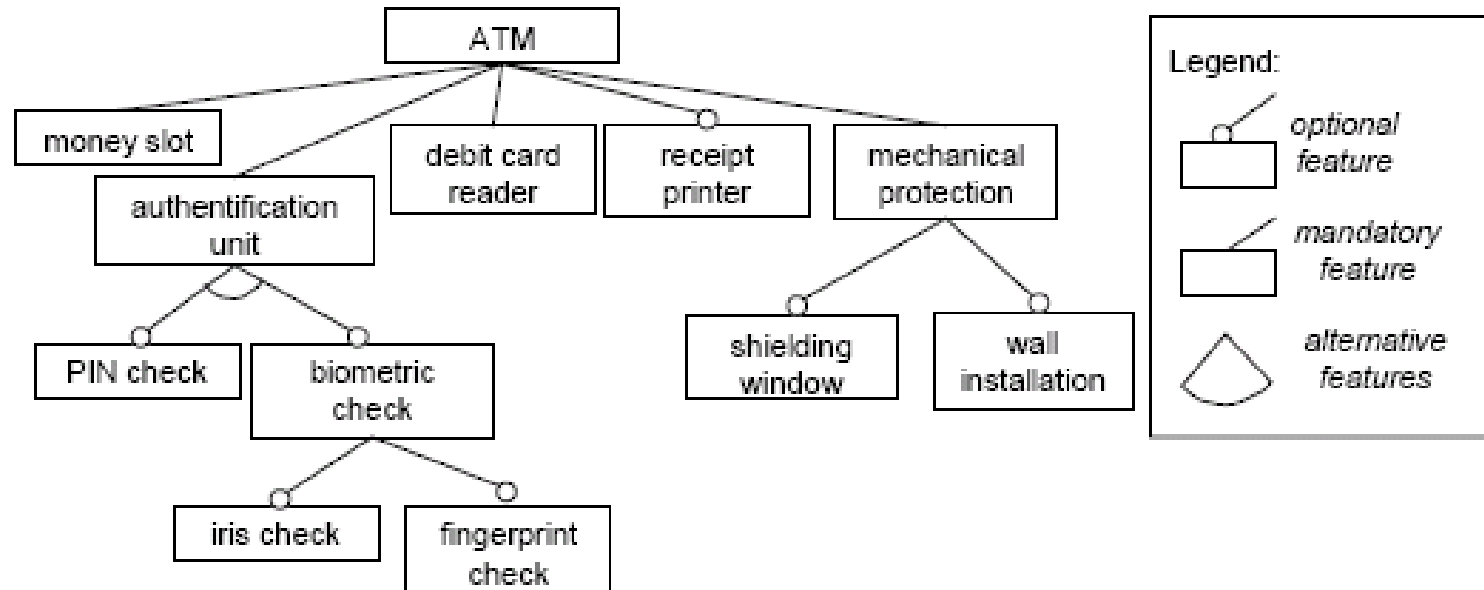


# Reality vs. Product





# Features

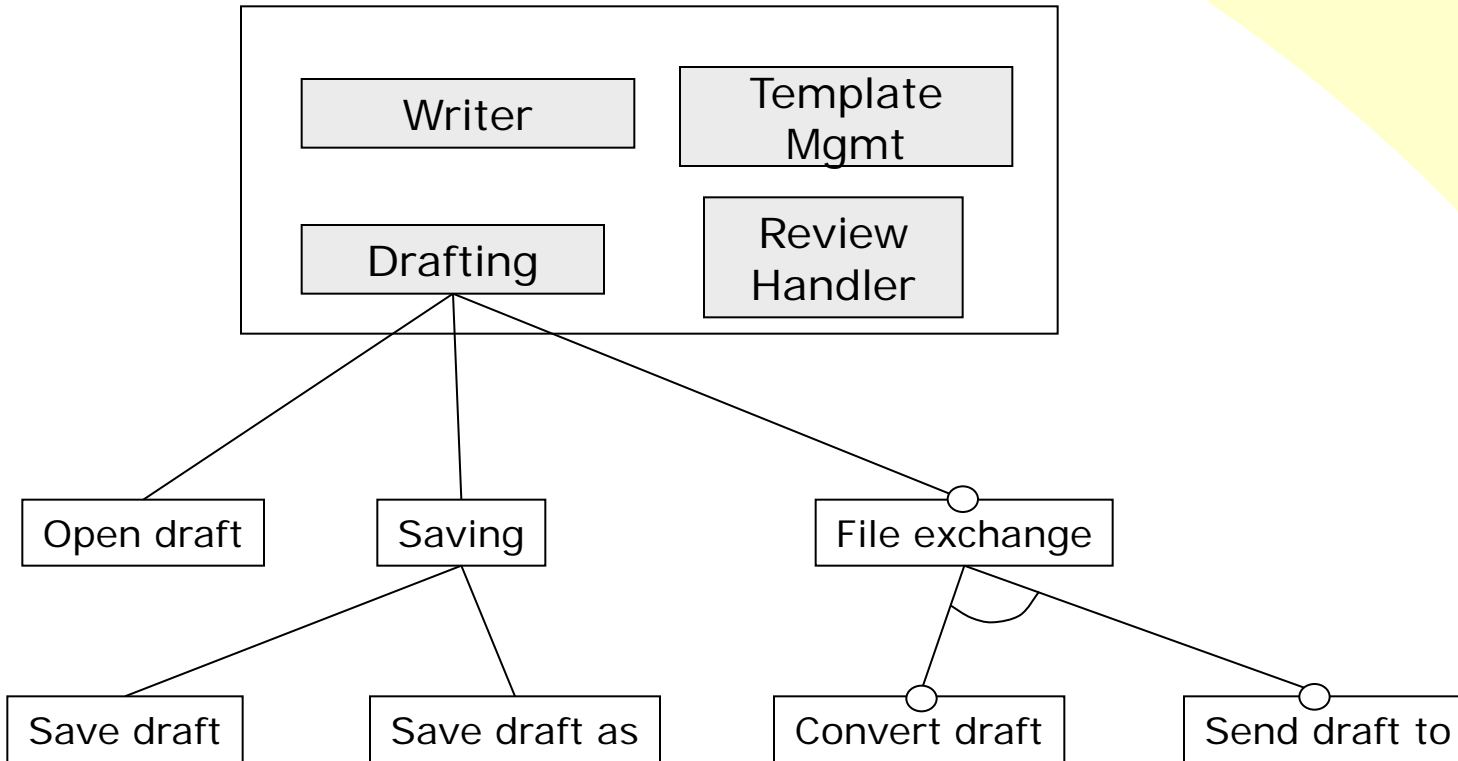


Feature models are a modeling tool for the process support functionality in a software product.

See: Matthias Riebisch – Towards a more Precise Definition of Feature Models - Modelling Variability for Object-Oriented Product Lines. BookOnDemand Publ. Co., Norderstedt, 2003. pp. 64-76.



# Feature models



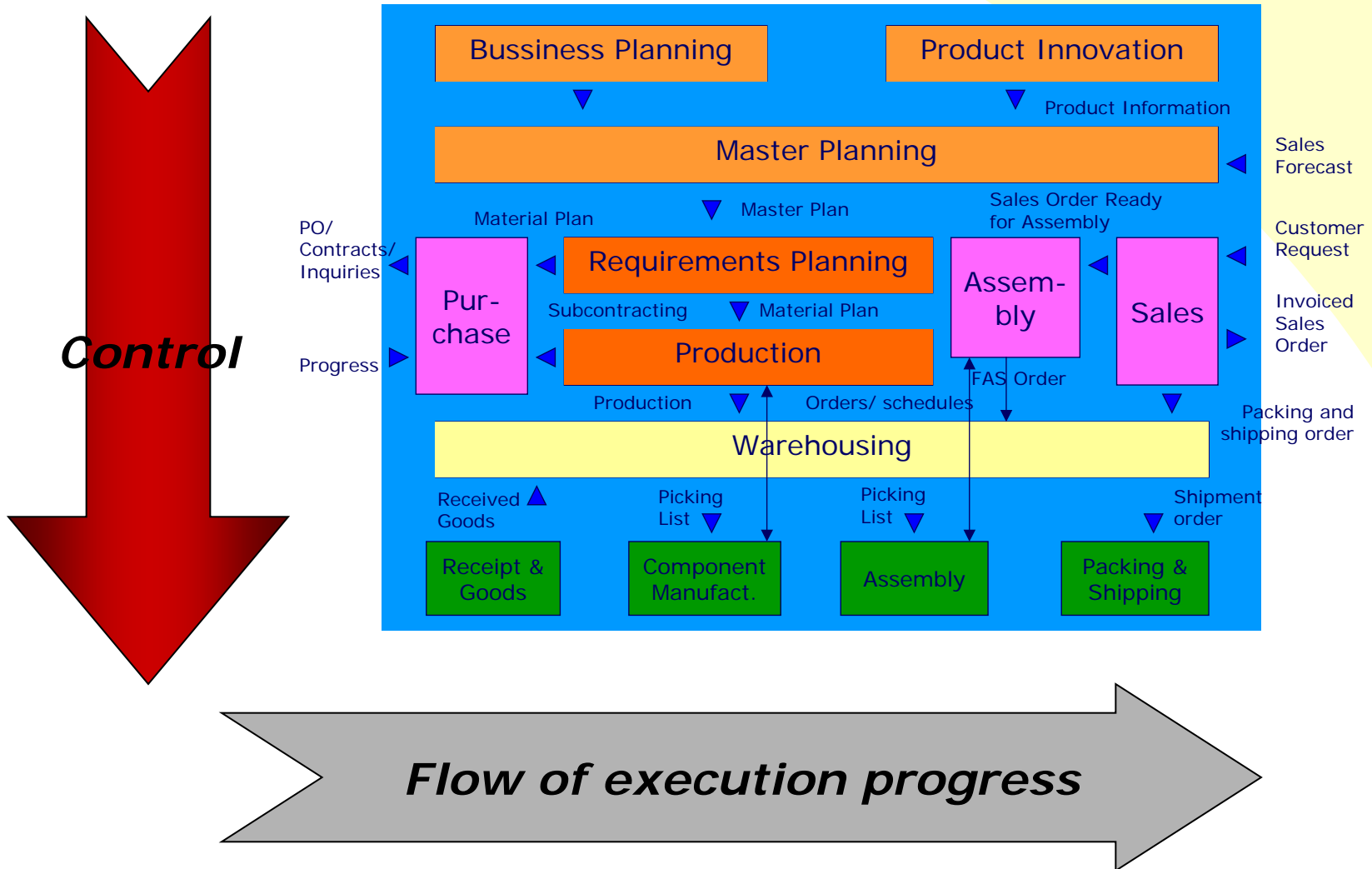


# Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture



# Positioning



# Hierarchical positioning of modules



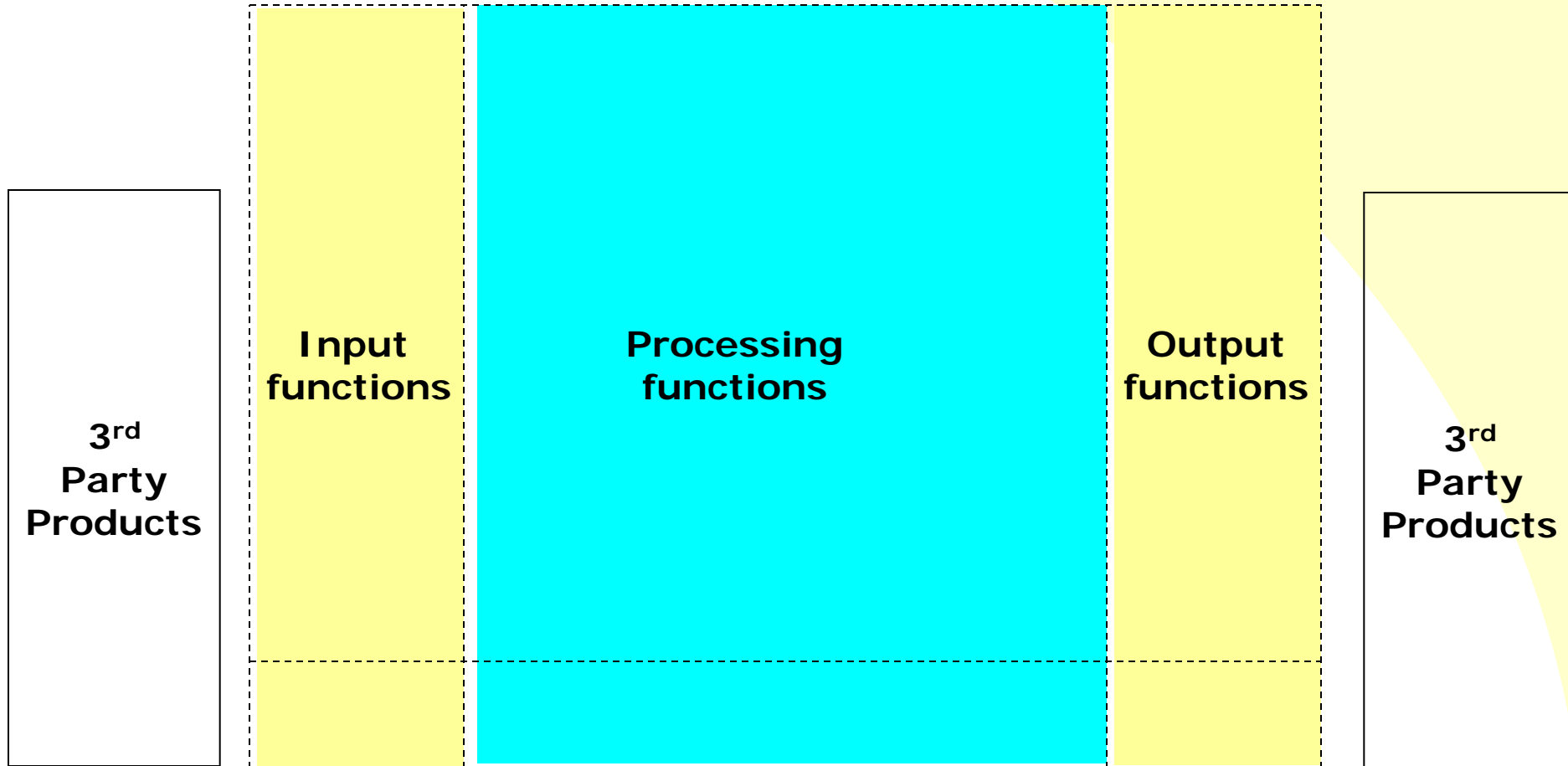
**Strategic functions (management)**

**Tactical functions (control)**

**Operational functions (execution)**

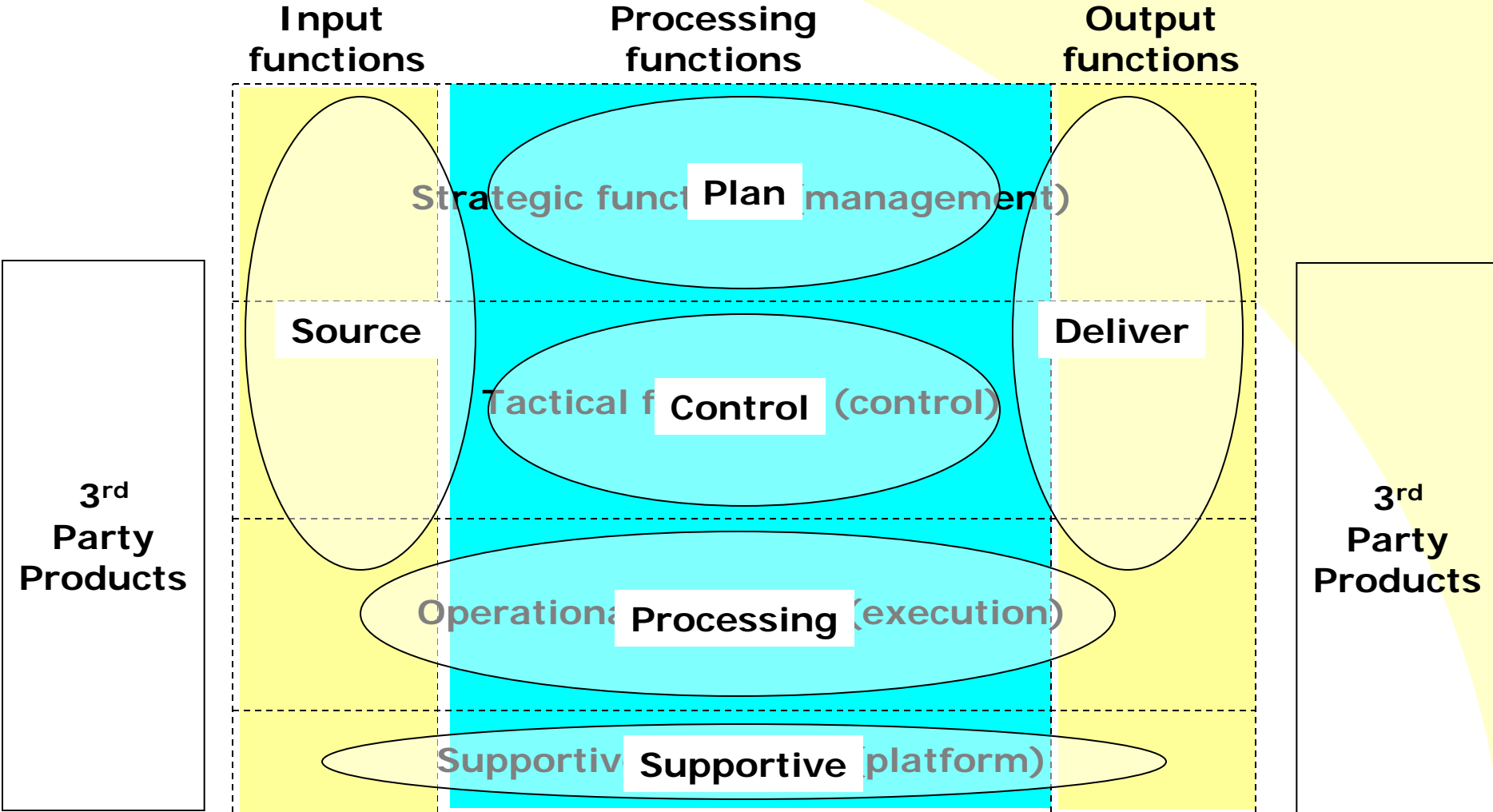
**Supportive functions (platform)**

# Sequential positioning of modules

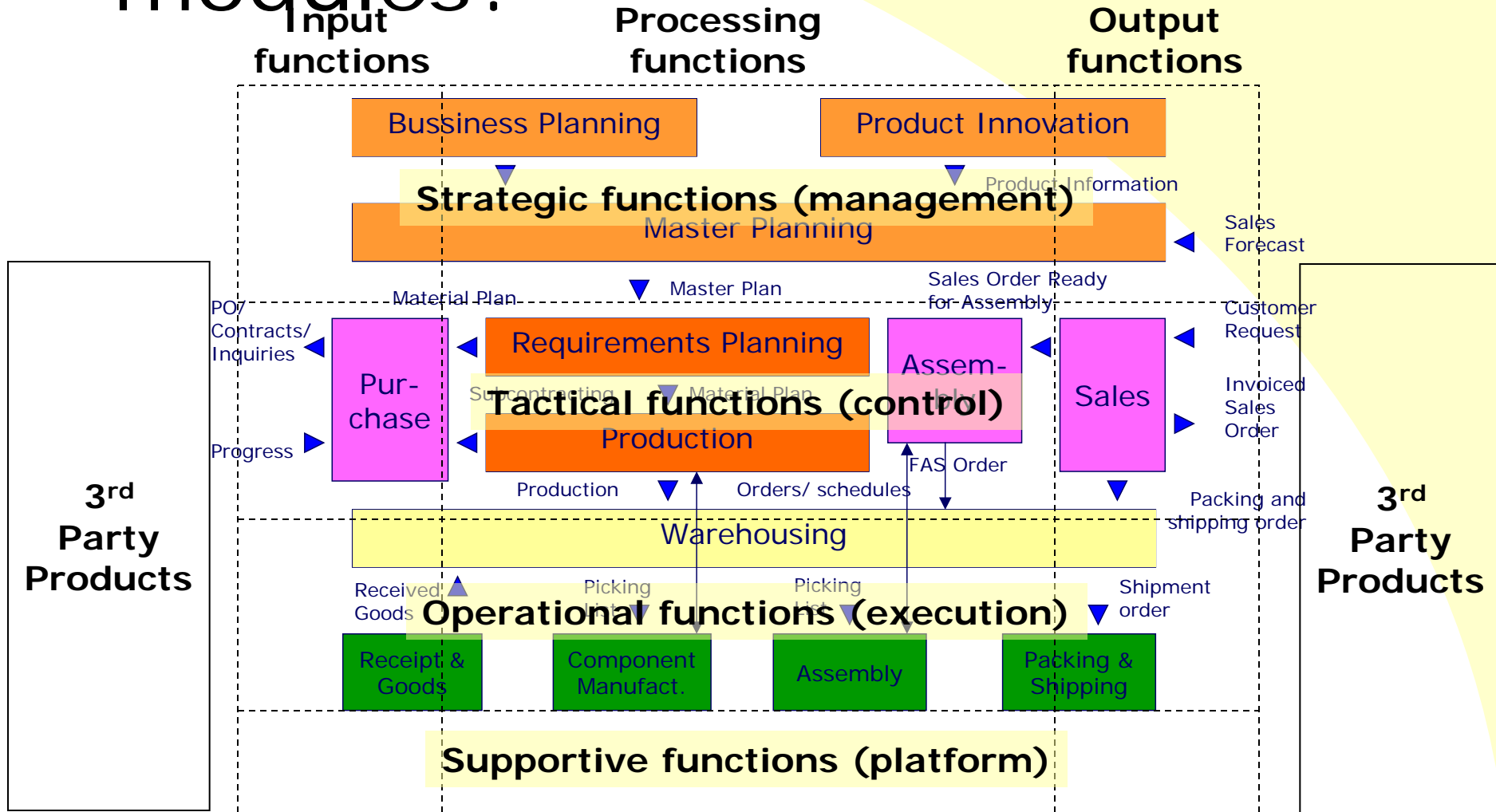




# Superpositioning



# How to position the modules?





# Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- Creating a Functional Architecture

# Reference Architectures



A Reference Architecture is a generic functional architecture for software products in a particular line of business

Purposes:

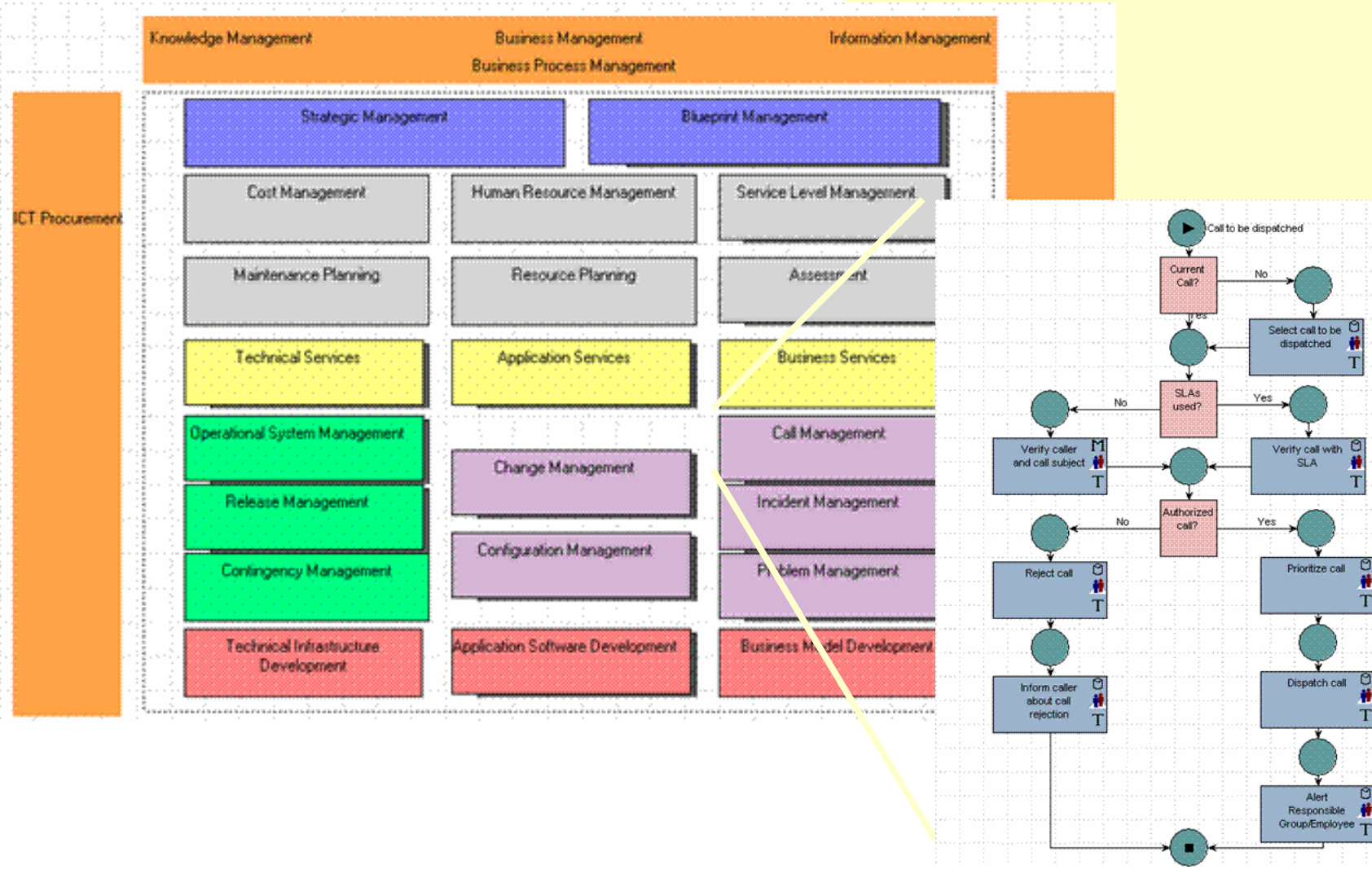
- Expressing **complete** functional coverage in line of business
- Indication of the **functional coverage** of a product
- Indication of the **technical quality** of a module
- Identifying market **growth** options and **roadmap**
- Positioning with **3rd party** products

Sources on the web, e.g.:

- Supply Chain Operations Reference (SCOR), see [scor.org](http://scor.org)

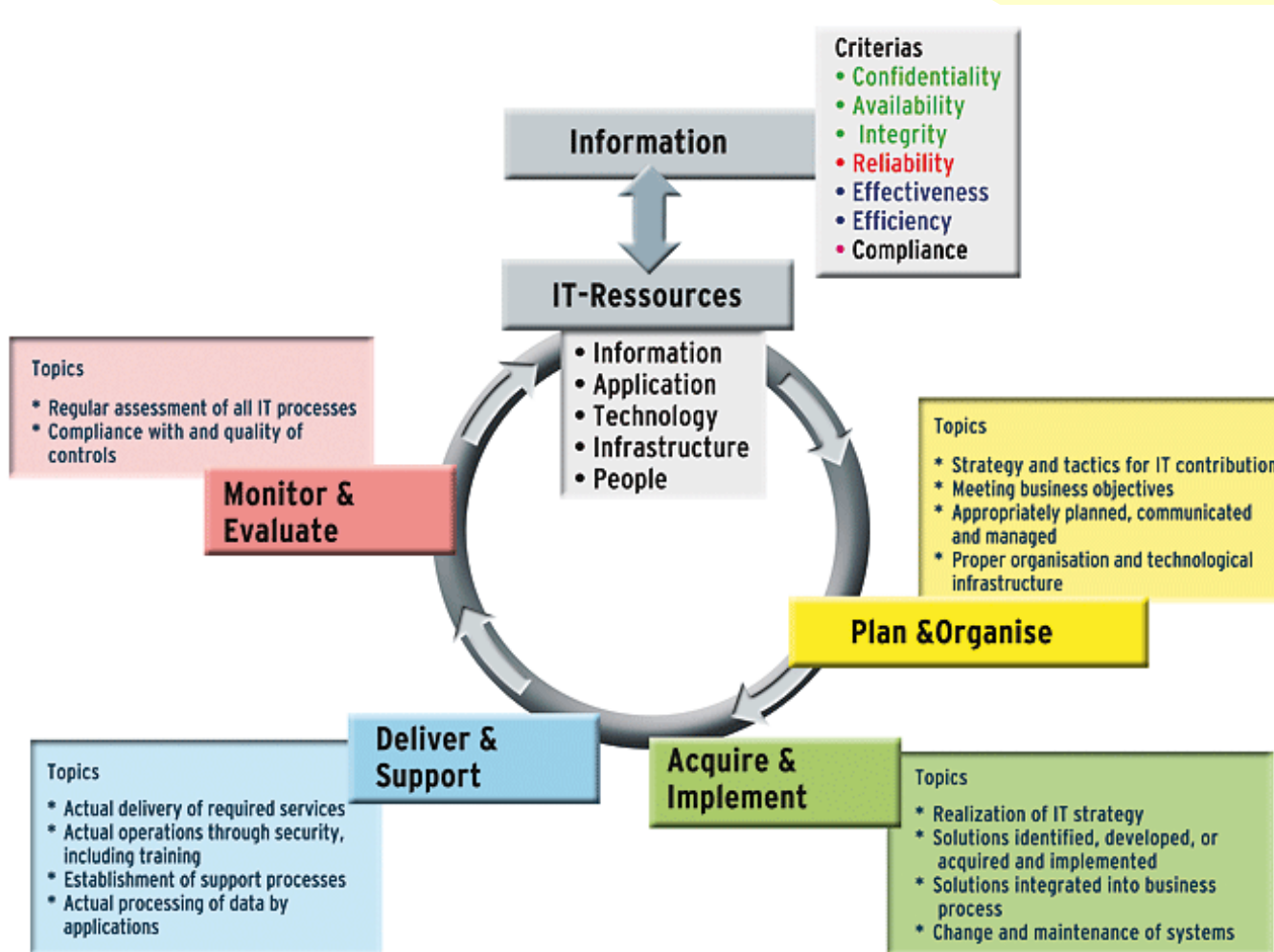


# ITIL models





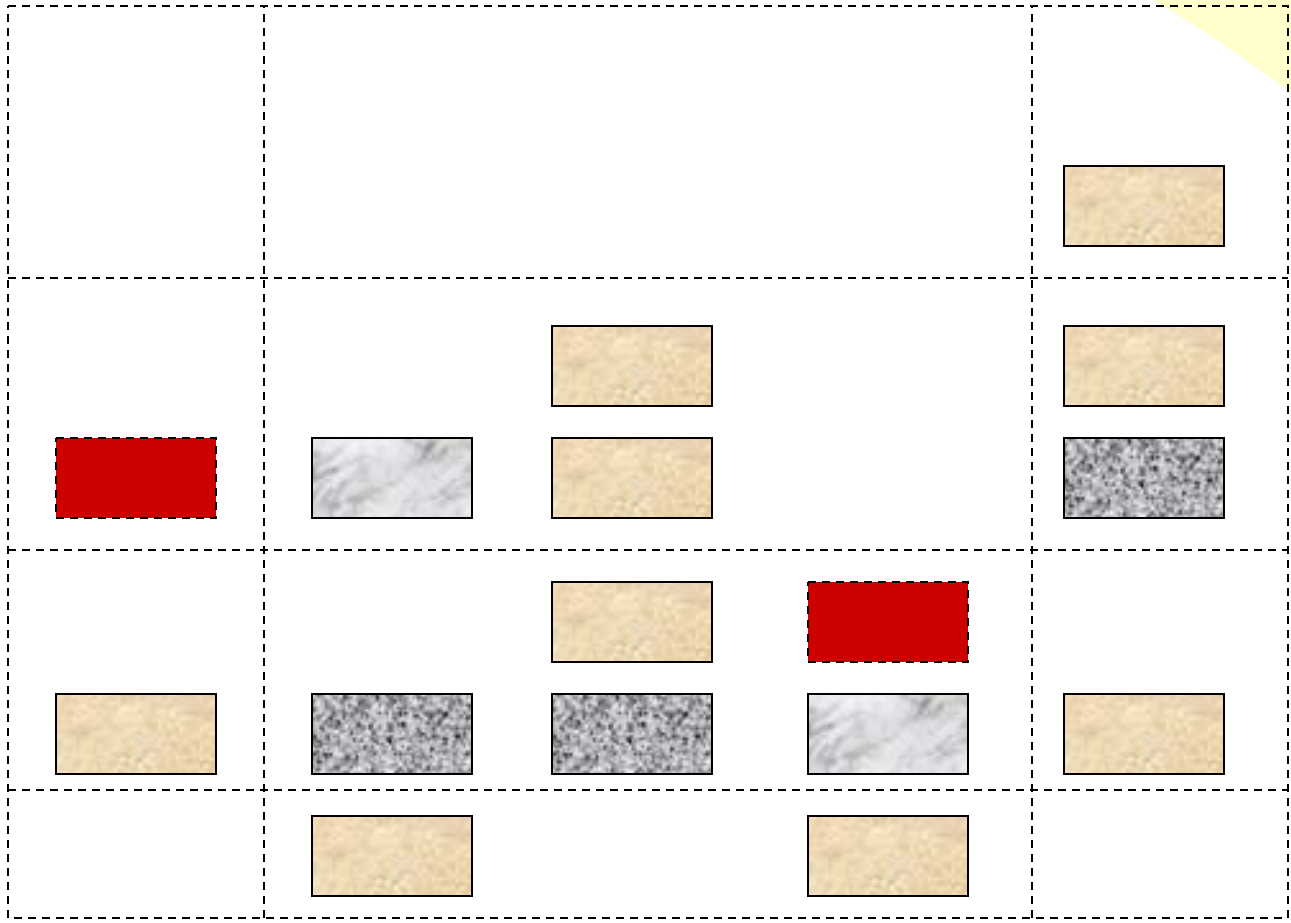
# Cobit:



Control Objectives for Information and related Technologies (COBIT)


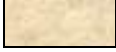




# Quality Indication on Reference Architecture

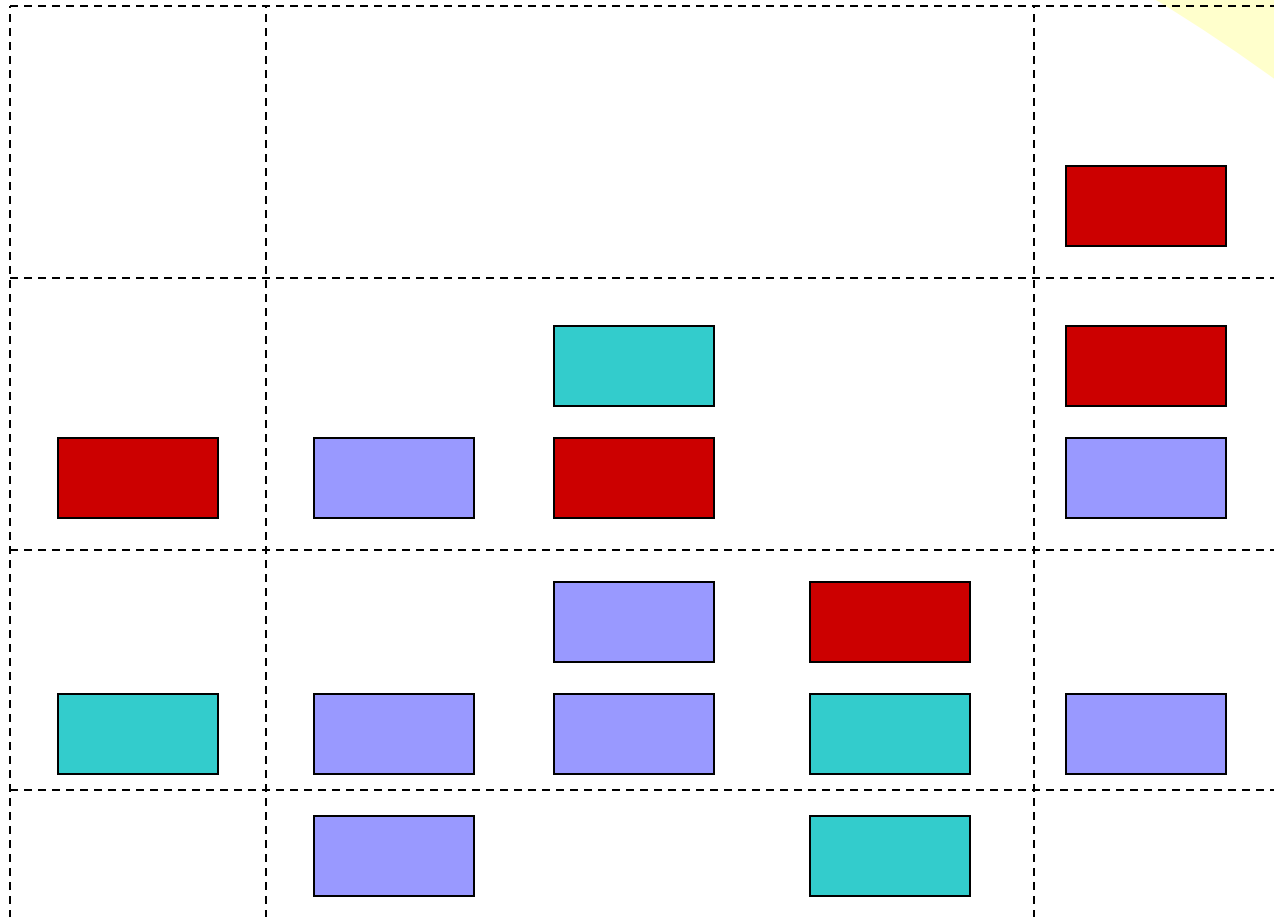


## *Technical Quality*

**Key:**




-  Differentiator
-  Good
-  Weak
-  Opportunity

# Variability Indication on Reference Architecture



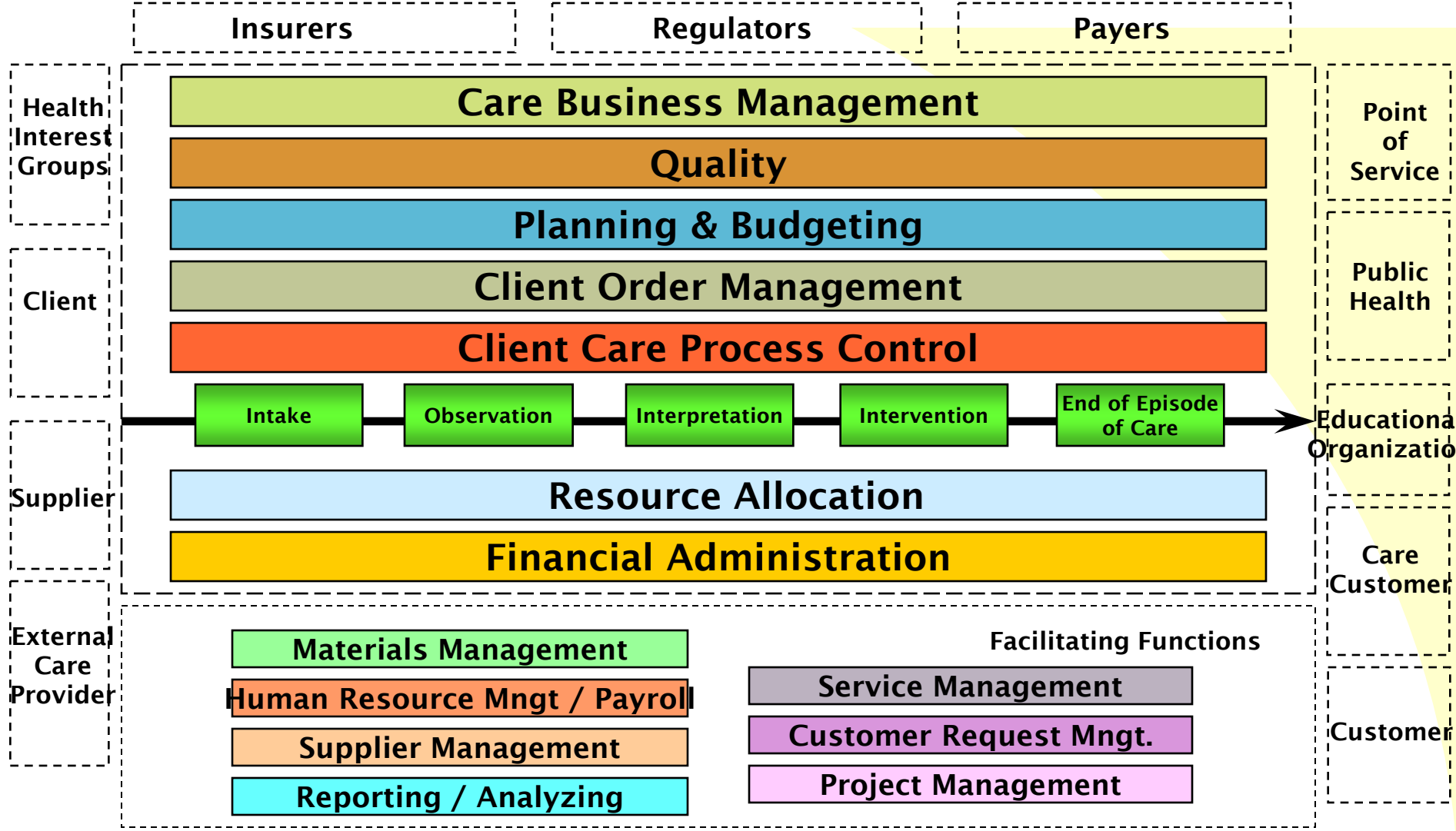
*Business Variability*

**Key:**

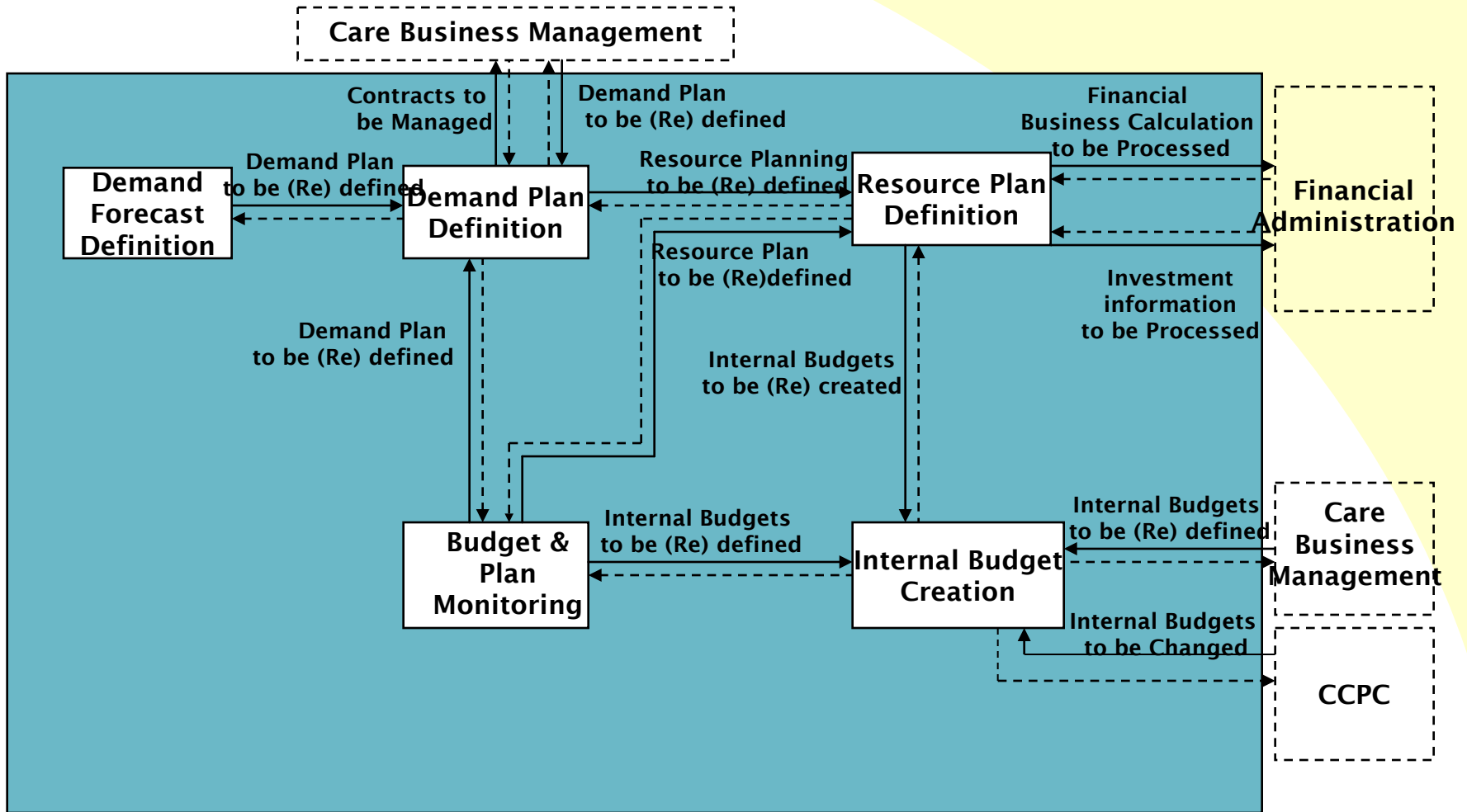
-  Real Estate
-  Facility Mgmt
-  Both



# Healthcare EFD



# Care Planning & Budgeting





# Contents

- What is architecture?
- Design structures in architecture
- Functional Architecture
  - Enterprise Function Diagrams
  - Decomposition
  - Positioning
- Reference Architectures
- **Creating a Functional Architecture**

# Creating a Functional Architecture



1. Determine scope
2. Define request - feedback flows
3. Model the primary function flow
4. Add control and monitoring functions
5. Specify external to/ from internal interactions



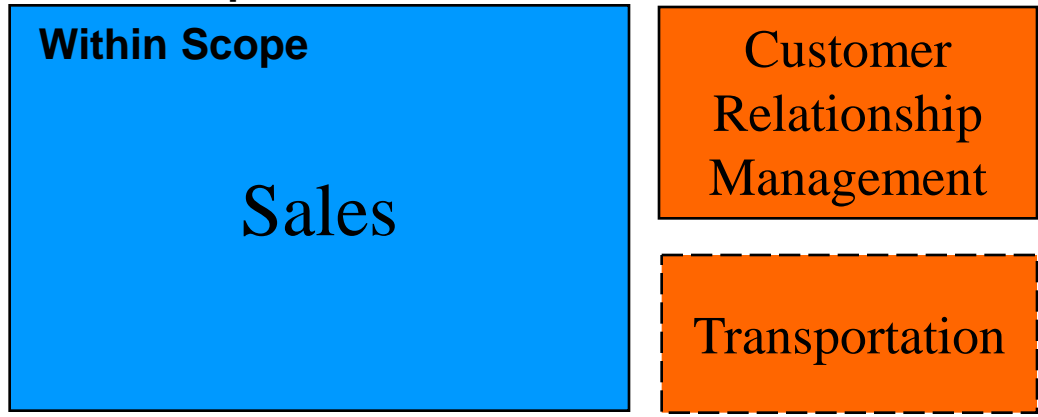
# Step 1: Determine scope

- Identify all external products in the usage environment
- Take the Product Context diagram as a starting point.
- Include future products to be interfaced with



# Example

Out of Scope



Suppose we are architecting a Sales product that will be interfaced with a CRM system and possibly with a Transportation product.

# Step 2: Define request - feedback flows



- Define all request-feedback loops
  - a. with the the external products
  - b. between functions and external products

Note: focus just on essential interactions; those interactions related to the main features of the product

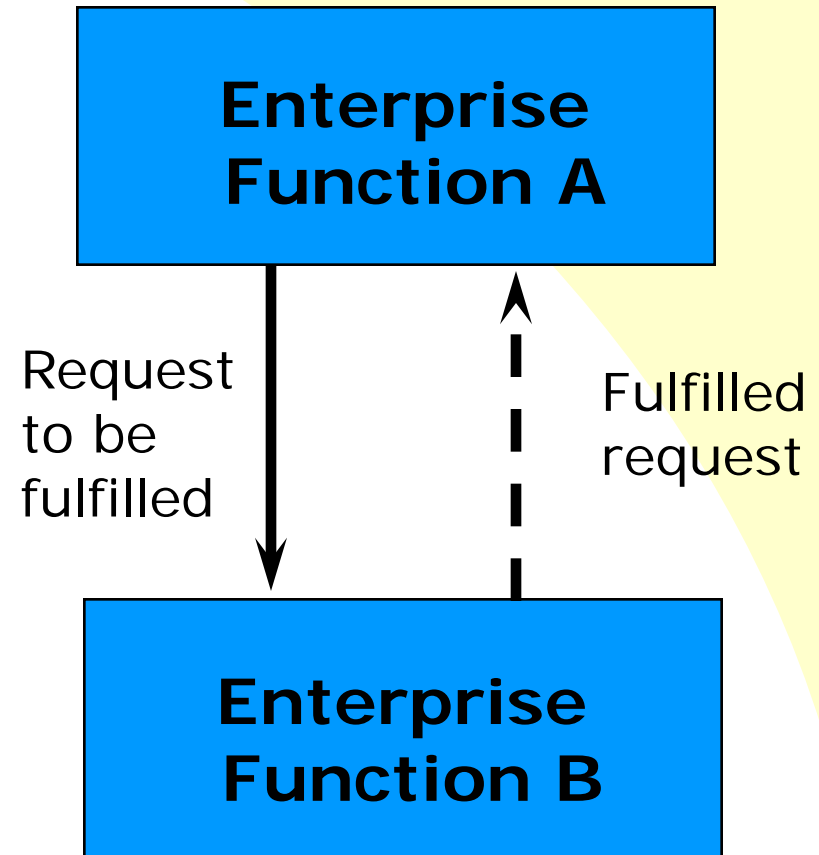


# Request - Feedback

Flows always appear in a pair:

- a combination of a Request and
- a return arrow that represents the Feedback of the result.

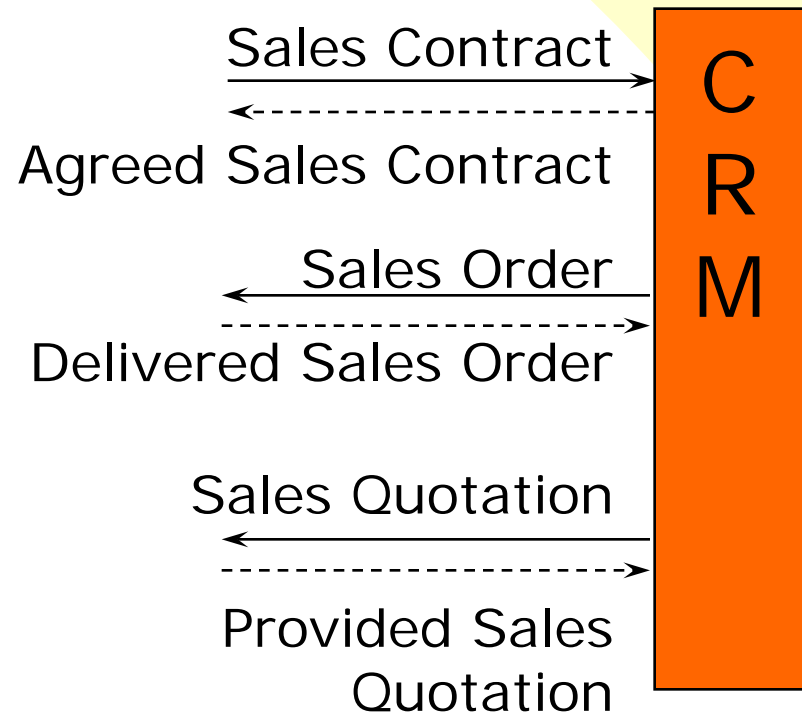
This construction is called a 'Request Feedback loop'.





# Example

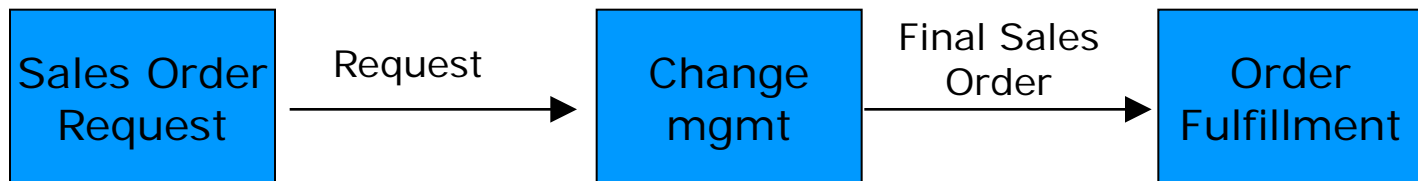
- Describe interaction between product and external products:
- E.g. with CRM



# Step 3: Model the primary function flow



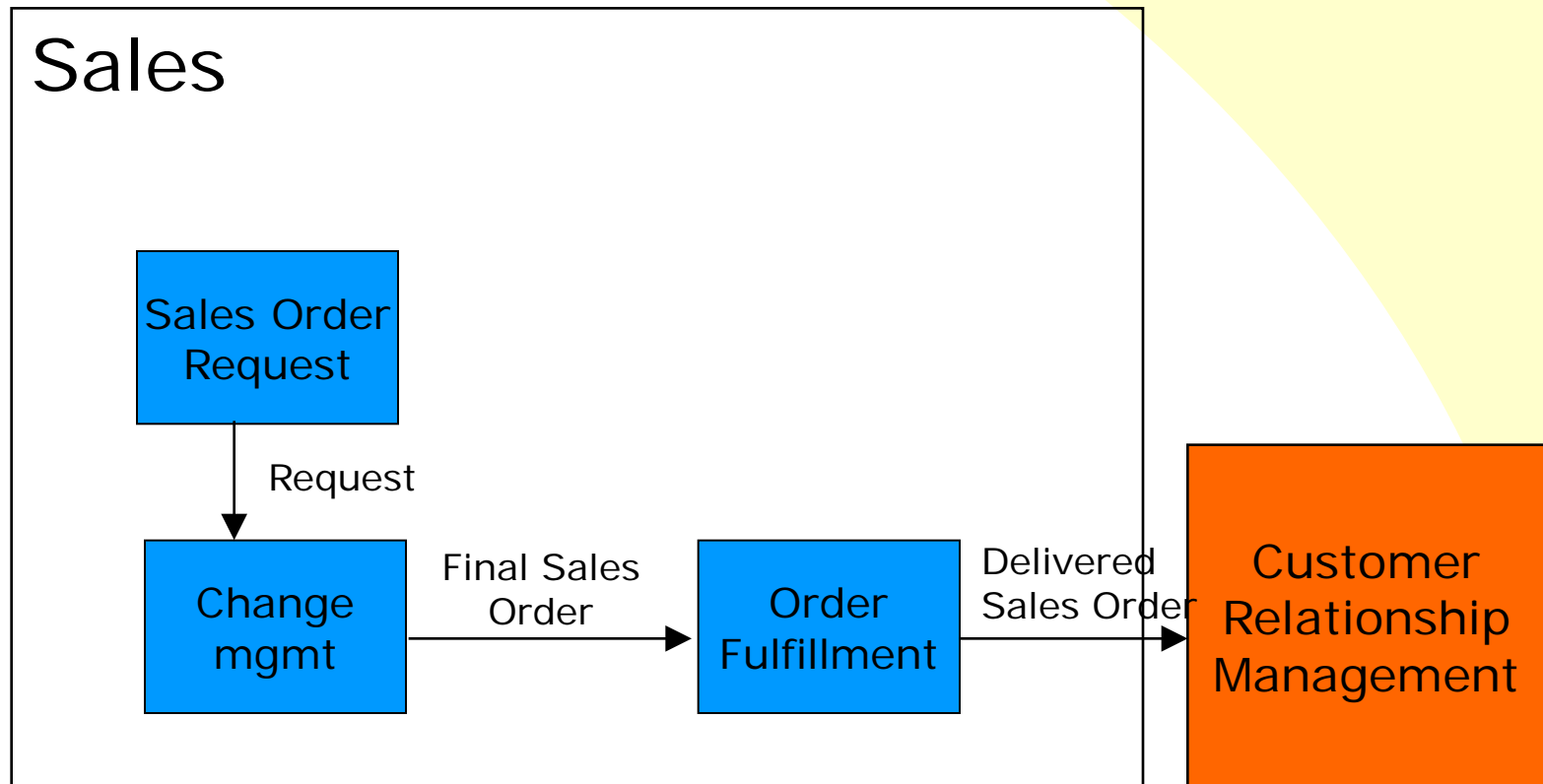
- The primary function flow (aka. primary process) is modeled by identifying the steps as function boxes. These functions are separated by flows that are usually information flows or waiting queues.





# Example

## Model primary function flow



# Step 4: Add control and planning functions

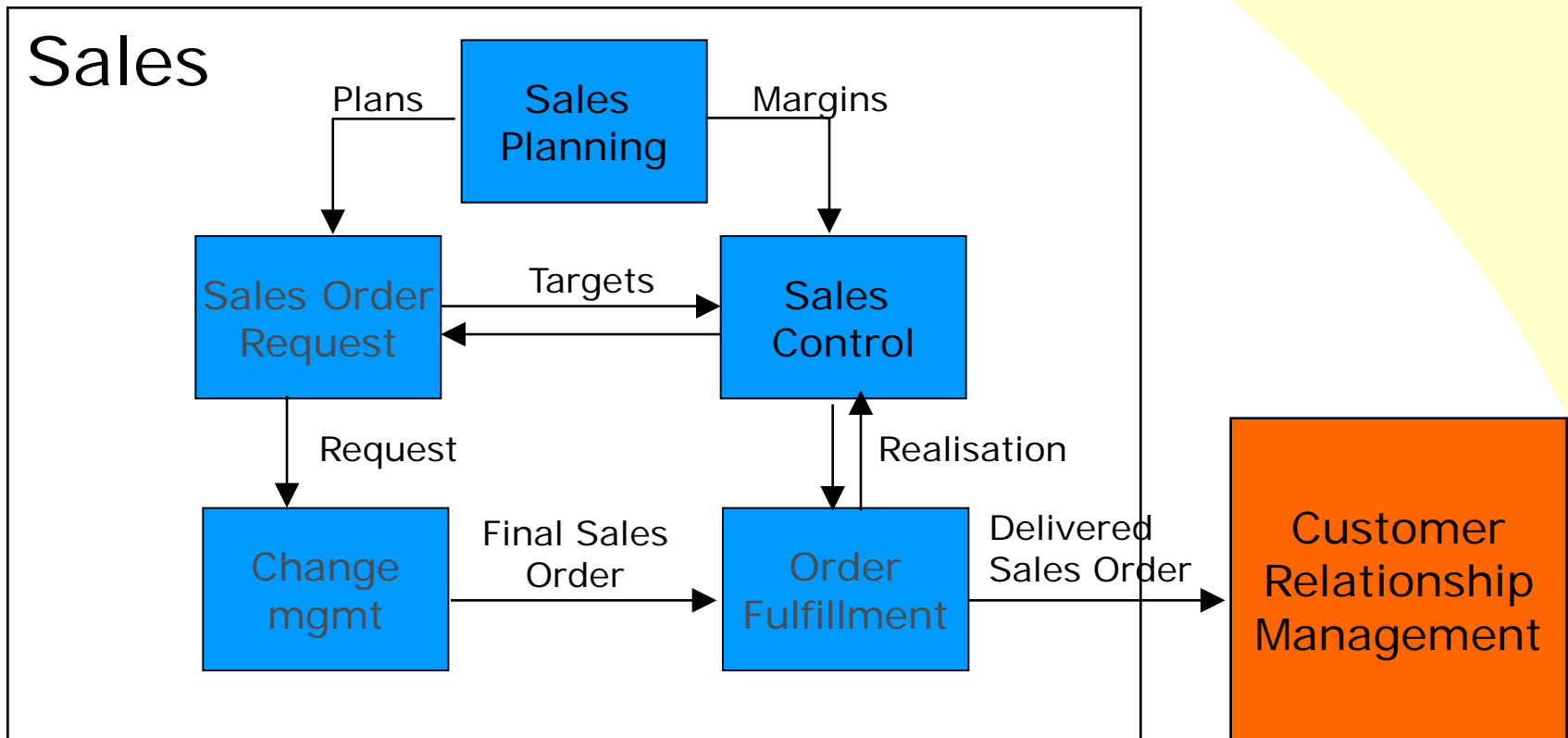


- The primary function flow is controlled by one or more enterprise functions
- The relation between every primary function and every control function is usually a request-feedback loop.
- On top of these functions the appropriate planning functions are added.



# Example

## Add control and planning





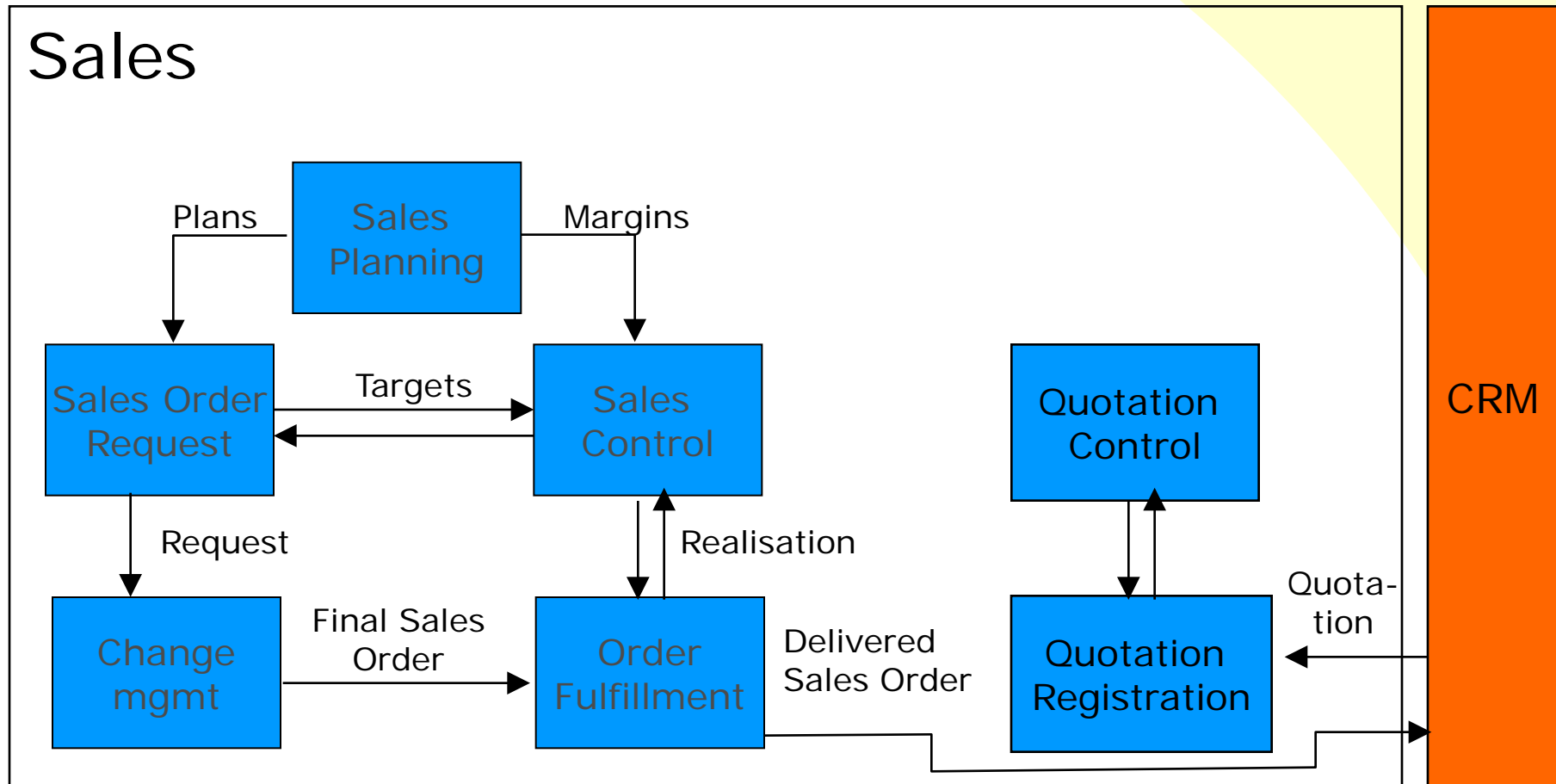
# Step 5: Specify interactions

- Specifying the information represented by the request flow from the external product  
e.g. CRM requests a sales order to be delivered.
- Follow the subsequent steps through the scope until one or more feedback flows; splitting and joining (bifurcations) are possible.
- Follow the process from the outside to inside the product.



# Example

## Specify interactions





# Concluding

- **Enterprise Function Diagrams** are a powerful modeling tool for a functional architecture of software products
- Assists in the **determination of modules**
- **Interactions** between the functions and the environment are shown
- Models are **recognized by non-specialists** without formal training
- Can be applied in **any type of business**:
  - Public sector: health care, governmental
  - Private sector: manufacturing, financial, services, food and beverage, project industries