Dr. Marcel Schlatter

IBM Distinguished Engineer

Member of the IBM Academy of Technology

University of Zurich   IBM

Developing a Solution's IT Architecture

Work product creation using a "top down", requirements driven approach
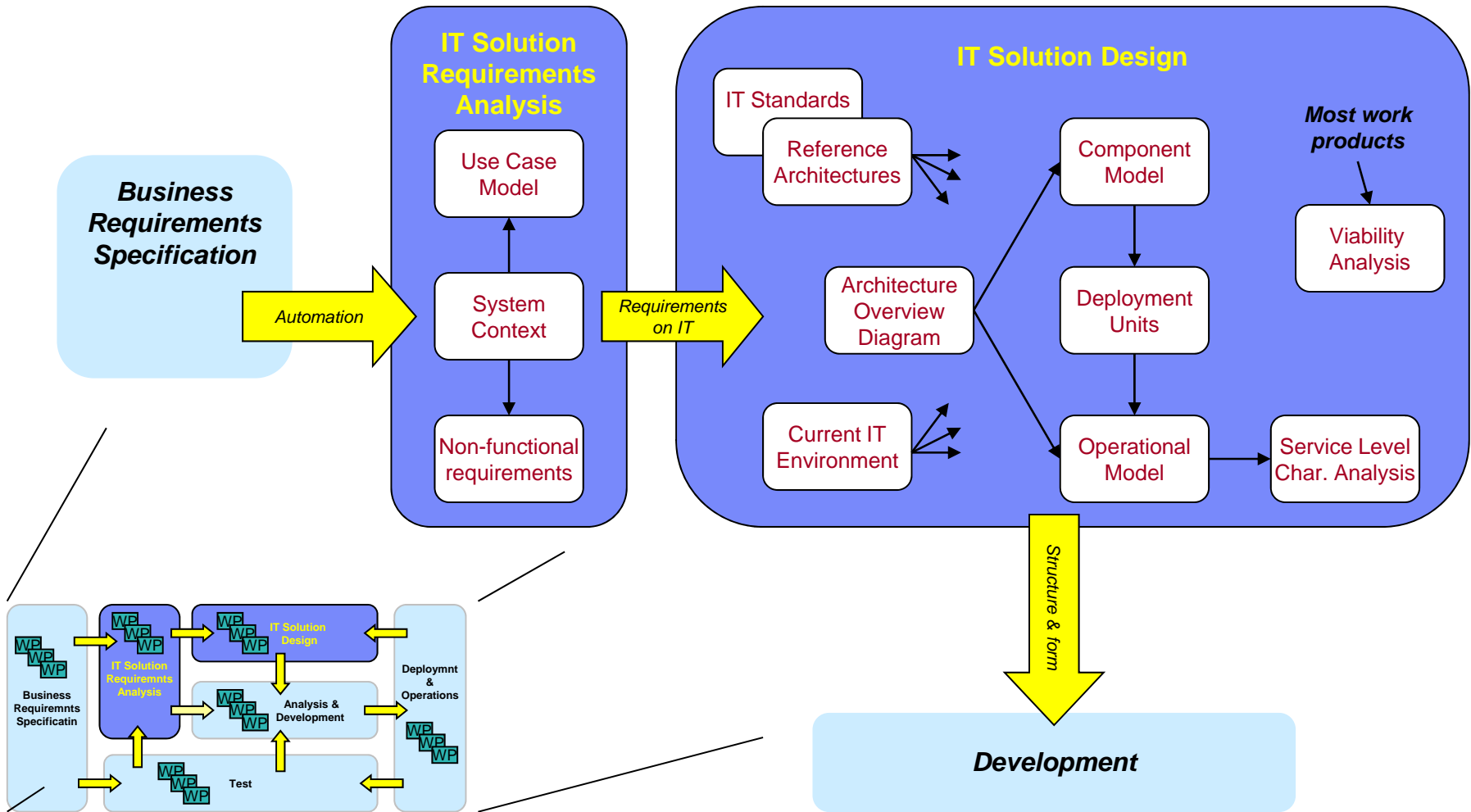
Separating concerns: organizing the requirements and design into distinct parts

Incrementally developing business requirements and their IT solution
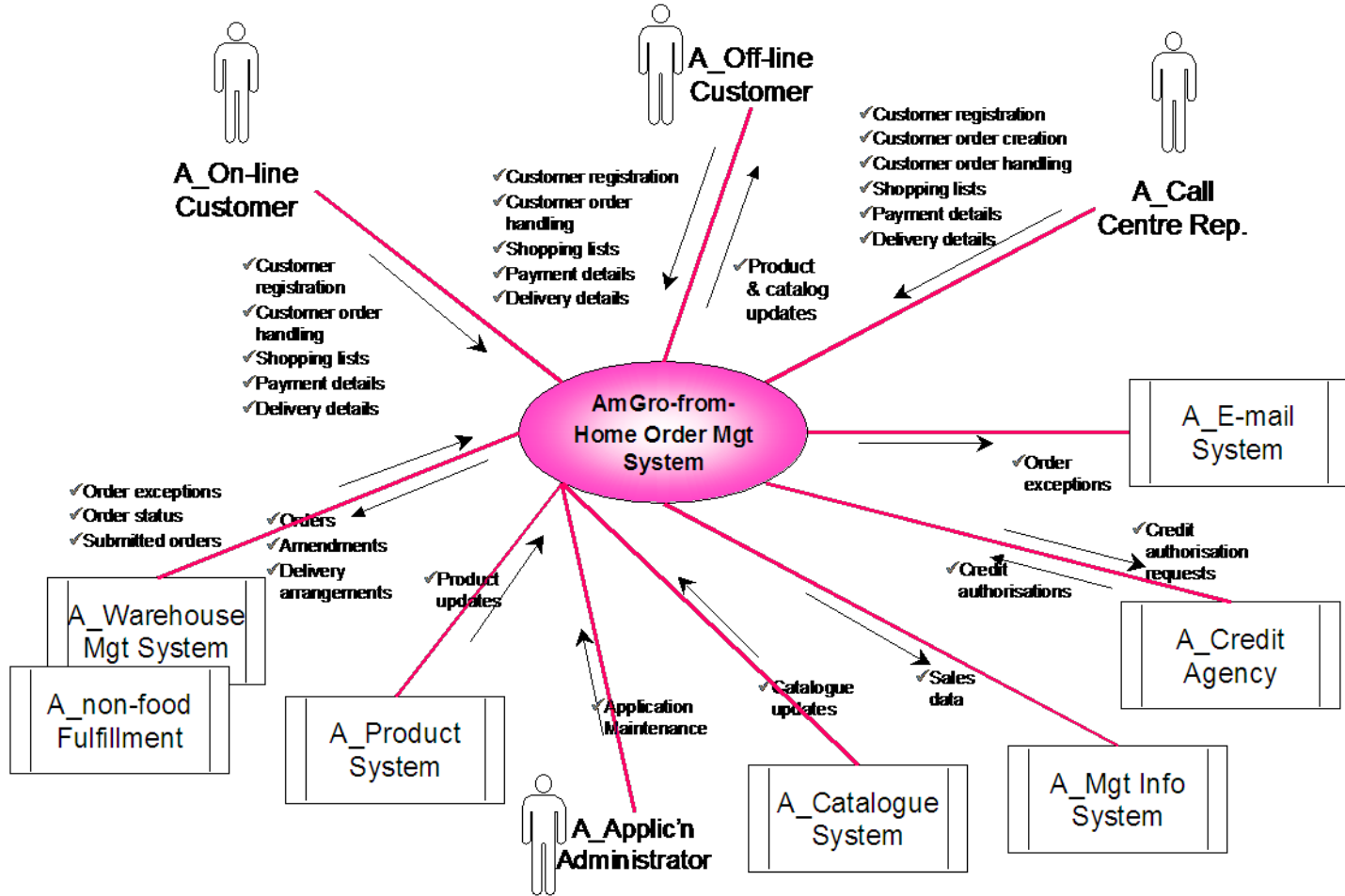
Defining and documenting the various aspects of the IT solution's requirements and design is achieved by using a set of **IT Architecture work products**, each focused on a specific view of the IT system
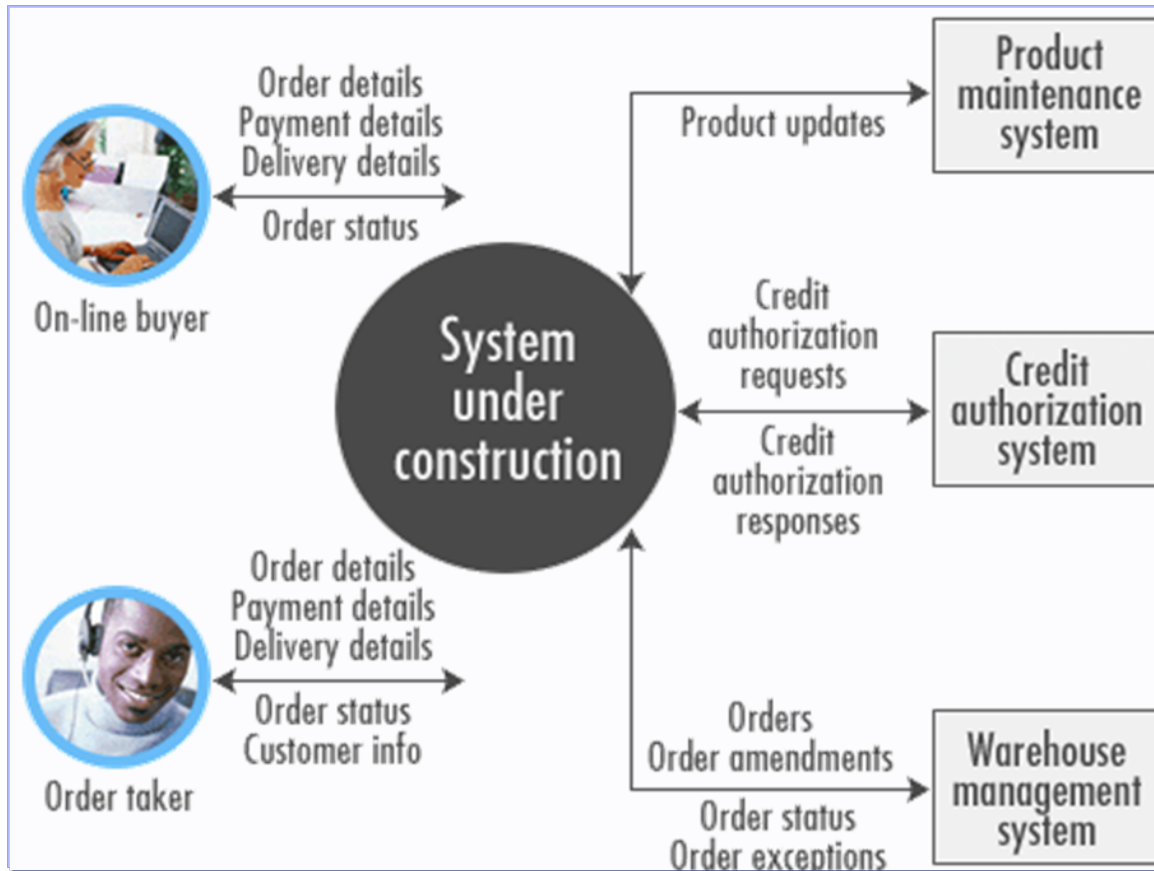
Separation of concerns

Business Requirements Specification

Automation →

IT Solution Requirements Analysis

- Use Case Model
- System Context
- Non-functional requirements

Requirements on IT →

IT Solution Design

- IT Standards
- Reference Architectures
- Architecture Overview Diagram
- Current IT Environment
- Component Model
- Deployment Units
- Operational Model
- Service Level Char. Analysis

Most work products → Viability Analysis

Structure & form ↓

Development

University of Zurich

IBM

The IT architect uses three core work products to document the business requirements their IT System will support…

IT Solution Requirements Analysis

Use Case Model

System Context

Non-functional requirements

UC-SEC-02 Logoff

HBA User

UC-SEC-01 Logon

UC-ACCNT-01 Manage Accounts

"Use Case Model" defines what the IT system must do for all its business (and other) users – it describes the system behaviour at the "automation boundary"

"System Context" defines the boundaries of the to be designed system, documenting the external users and other IT systems with which this system must interact

"Non functional requirements" describe the characteristics of the IT systems functionality, such as transaction response times and workload volumes

System Context

The System Context is essential to capturing the scope of the project
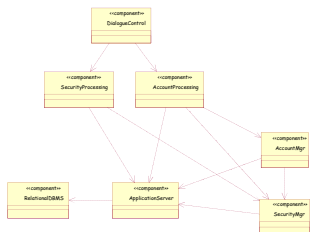

- **The System Context helps to:**
  - Clarify the environment in which the system has to operate
  - Put bounds on the system
  - Identify external interfaces (users or systems)

The IT architect uses four core work products to document and communicate their IT system's design
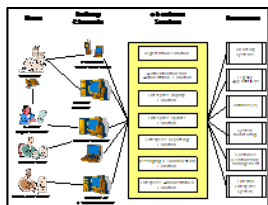
Application Architect - Software Engineering

"Component Model" describes the structure of an IT System in terms of its software components with their responsibilities, interfaces and relationships, and the way they collaborate to deliver the required functionality.
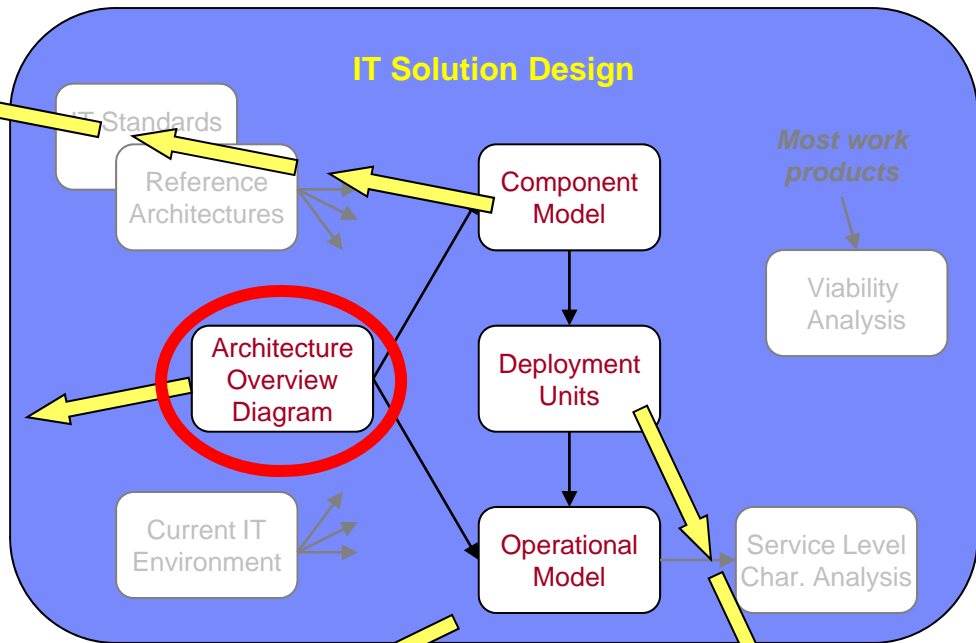
"Architecture Overview Diagram" provides a picture (not a model) of the whole IT system "on a page" as a means of communicating the salient points of the design. AODs are audience specific

IT Solution Design

- IT Standards
- Reference Architectures
- Component Model
- *Most work products*
- Viability Analysis
- Architecture Overview Diagram
- Deployment Units
- Current IT Environment
- Operational Model
- Service Level Char. Analysis

"Operational Model" defines the organisation of the IT system across locations, documenting the placement of the solution's components onto nodes connected across the organisation, in order to achieve the solution's operational NFRs
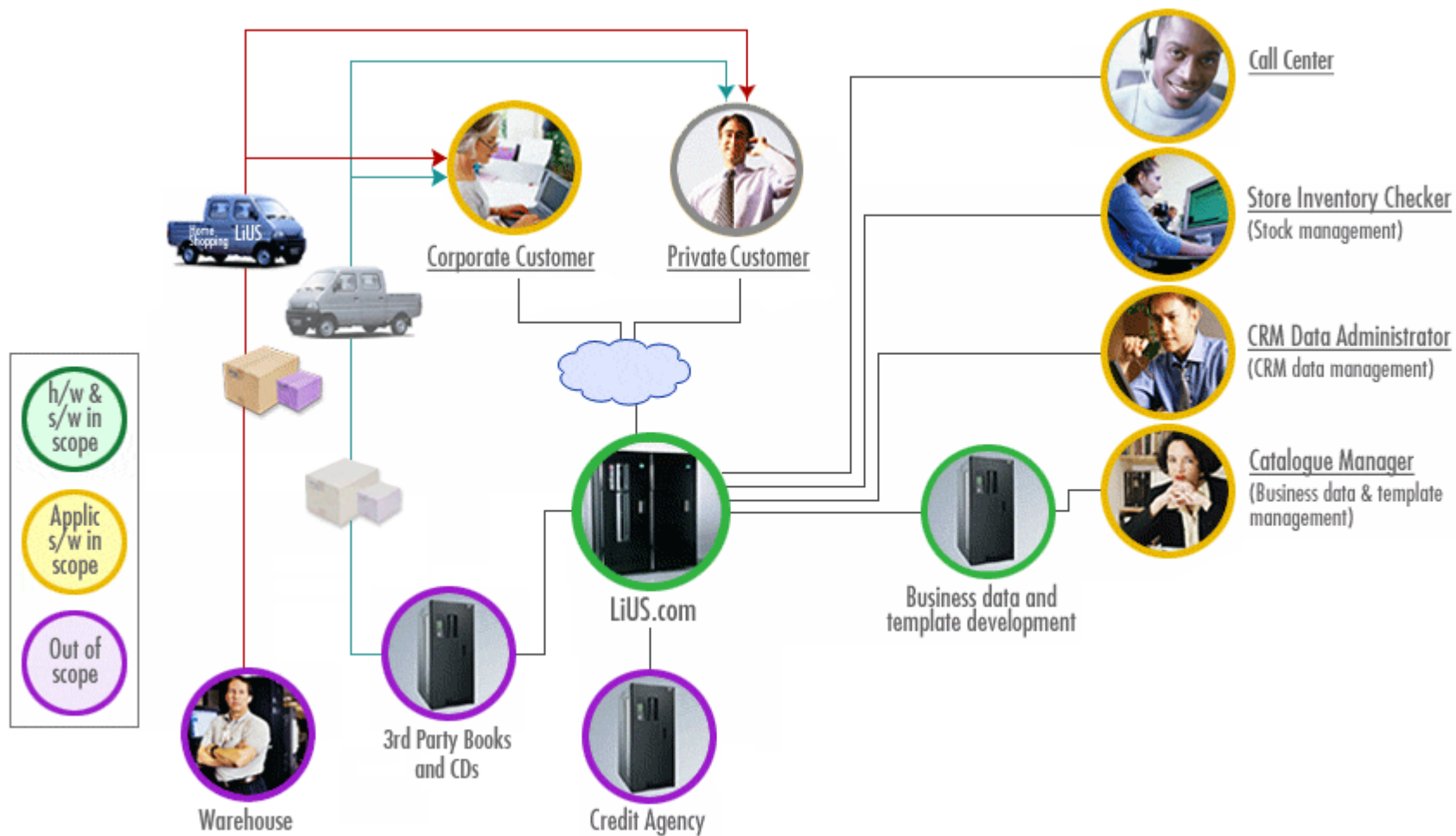
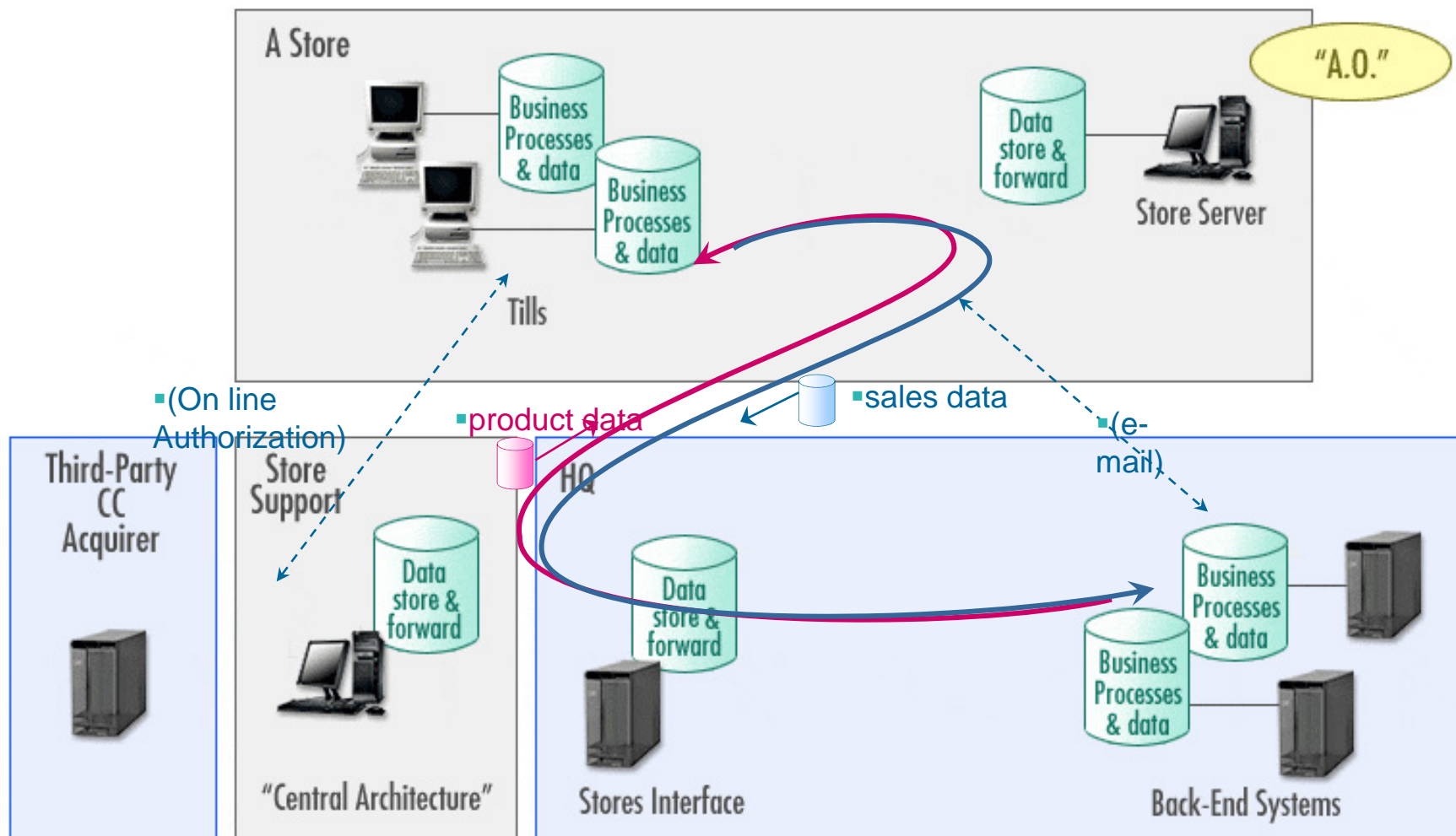**Infrastructure Architect - Systems Engineering**

"Deployment Units" represent various aspects of components, as a convenient means of documenting their non functional requirements, as well as their placement across the Operational Model
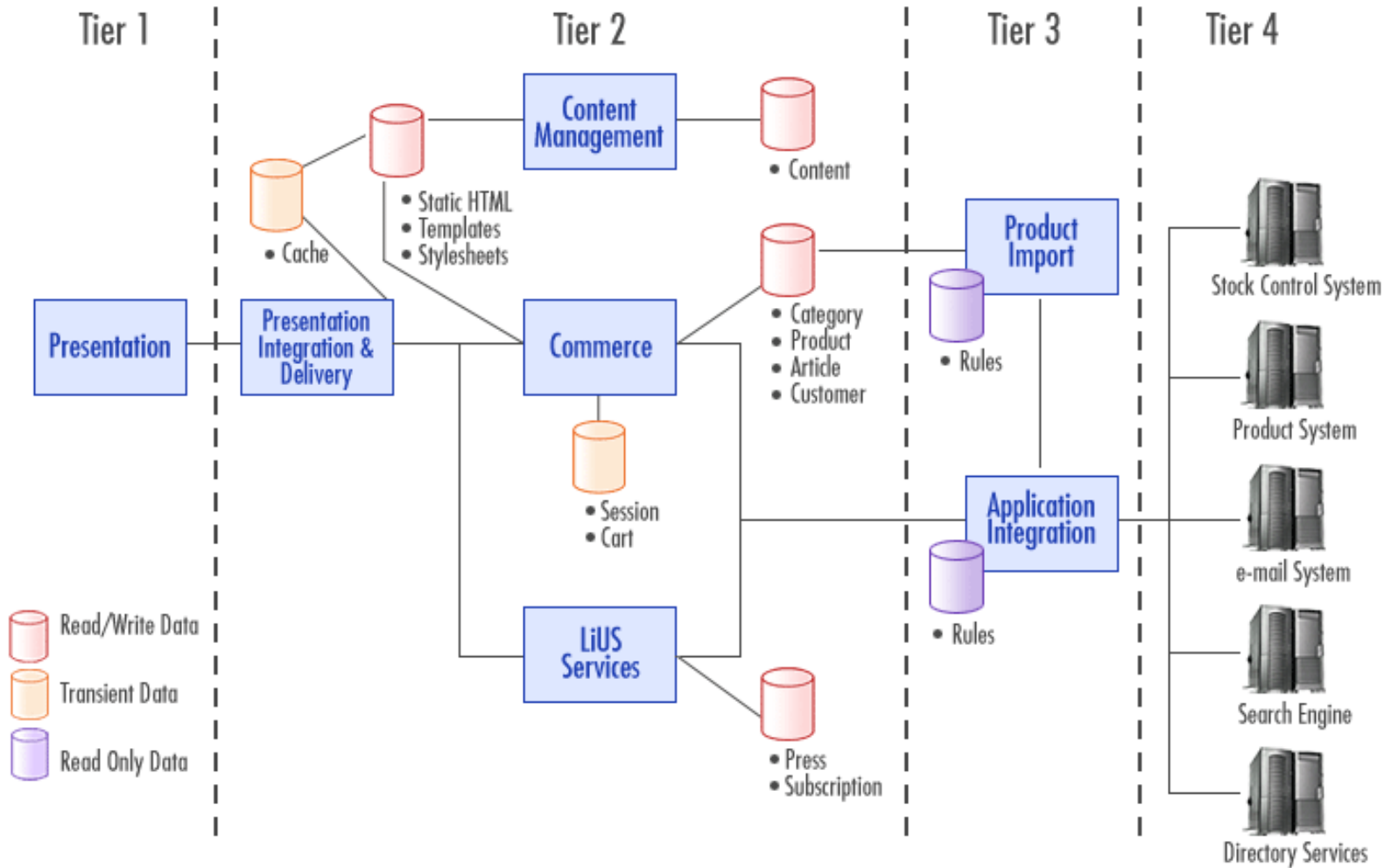
An Architecture Overview for Nontechnical Audiences

An Architecture Overview with Data Flow

An Architecture Overview showing the different tiers of a shopping system
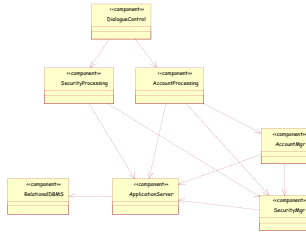
The IT architect uses four core work products to document and communicate their IT system's design

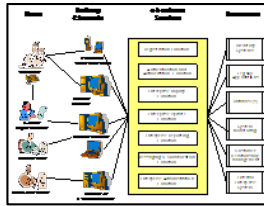**Application Architect - Software Engineering**

"Component Model" describes the structure of an IT System in terms of its software components with their responsibilities, interfaces and relationships, and the way they collaborate to deliver the required functionality.
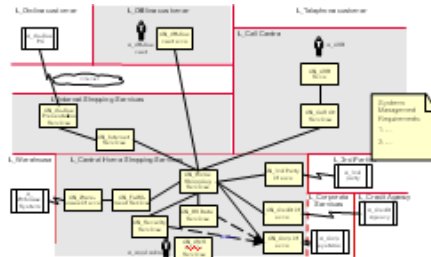
"Architecture Overview Diagram" provides a picture (not a model) of the whole IT system "on a page" as a means of communicating the salient points of the design. AODs are audience specific
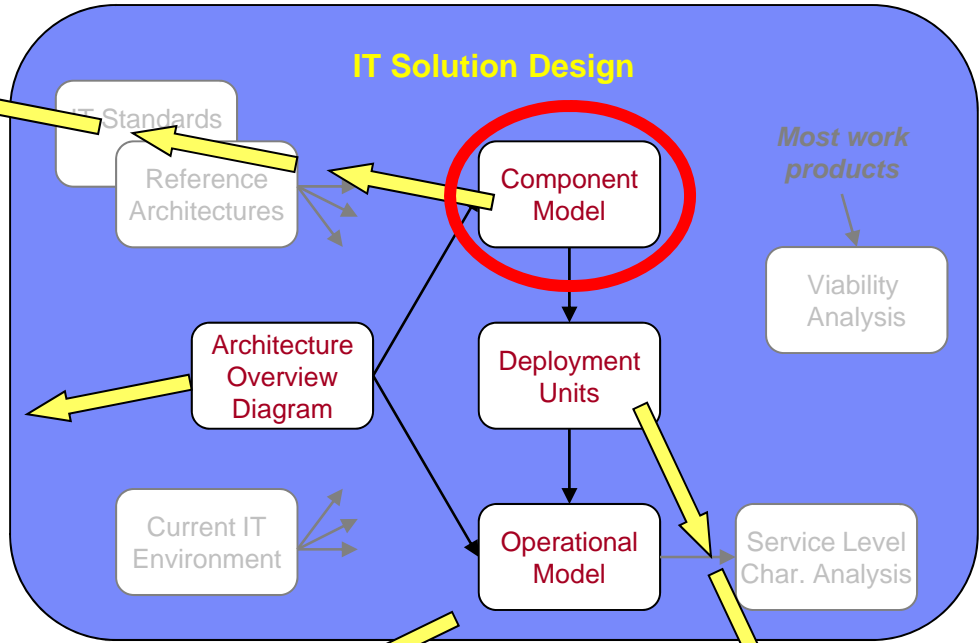
IT Solution Design

IT Standards

Reference Architectures

Component Model

Most work products

Viability Analysis

Architecture Overview Diagram

Deployment Units

Current IT Environment

Operational Model

Service Level Char. Analysis

"Operational Model" defines the organisation of the IT system across locations, documenting the placement of the solution's components onto nodes connected across the organisation, in order to achieve the solution's operational NFRs
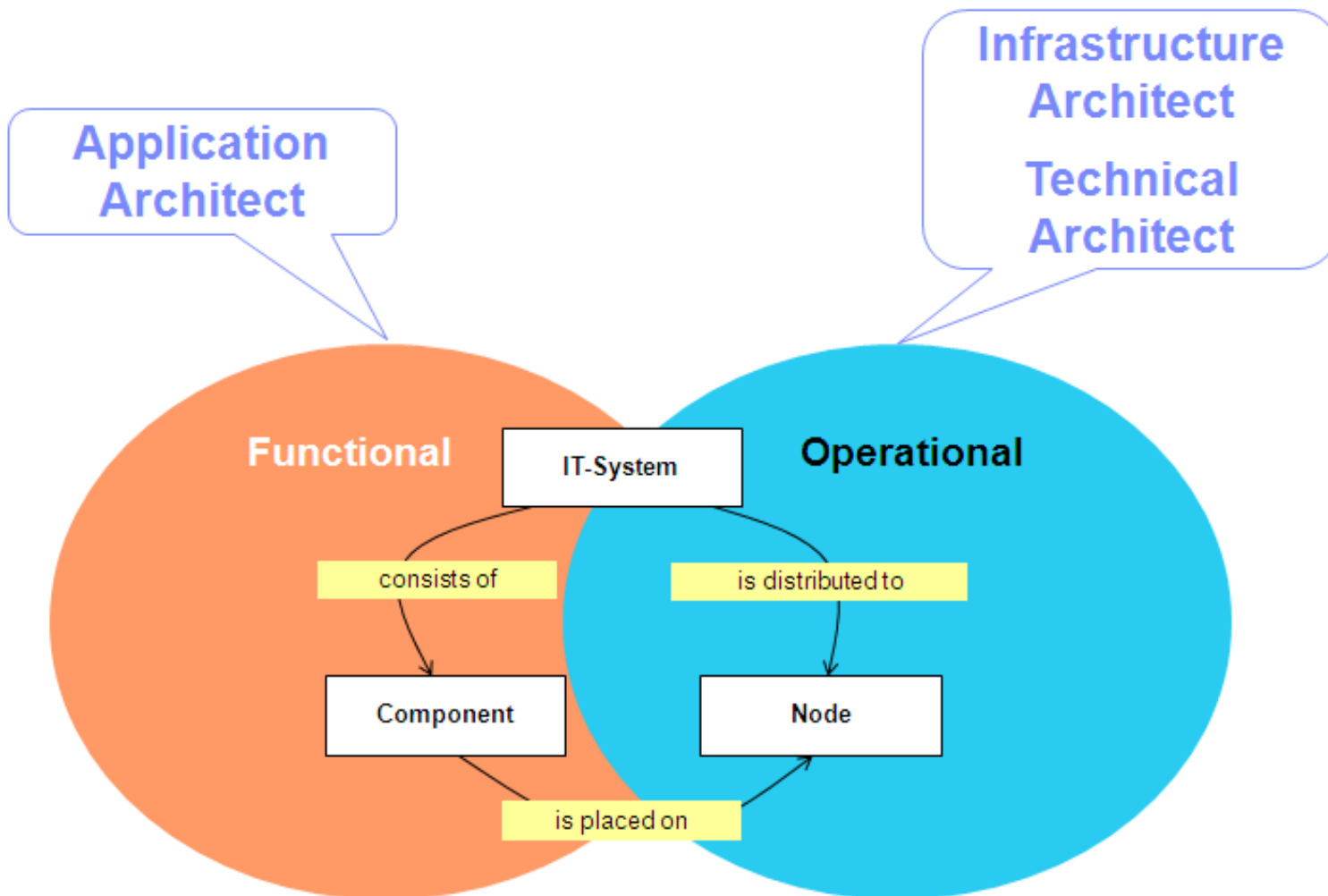
"Deployment Units" represent various aspects of components, as a convenient means of documenting their non functional requirements, as well as their placement across the Operational Model

Infrastructure Architect - Systems Engineering

The Component Modeling technique consists of three steps…

Component Identification

- Partition into subsystems and components and assign responsibilities
- Review architectural patterns, reference architectures, and reusable assets
- Structure ensuring loose coupling, high cohesion, and so on
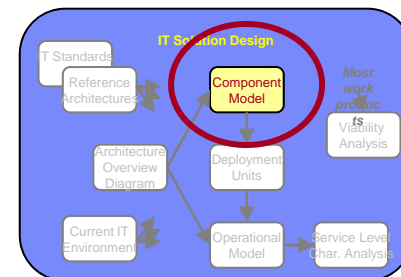
Component Specification

- Specify interfaces
- Specify operations and signatures
- Specify pre- and post-conditions

Component Transformation

- Identify products and packages
- Define implementation approach

…which are performed iteratively

University of Zurich

IBM
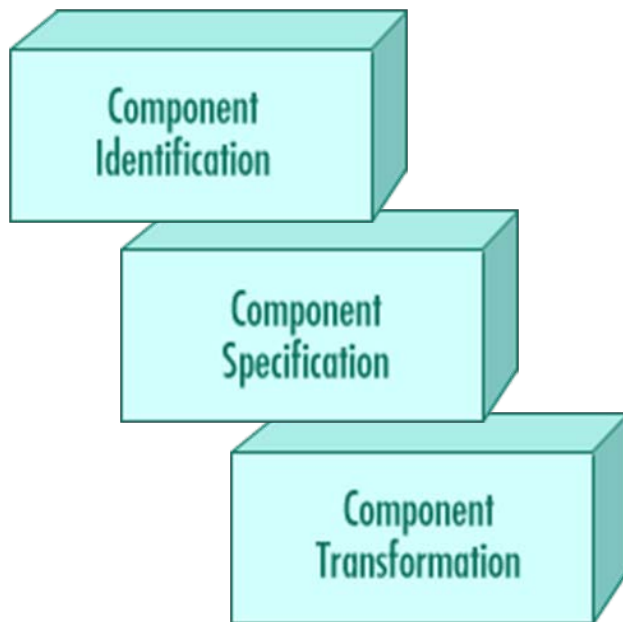
Each step is applied, to varying degrees, at different points in the delivery process
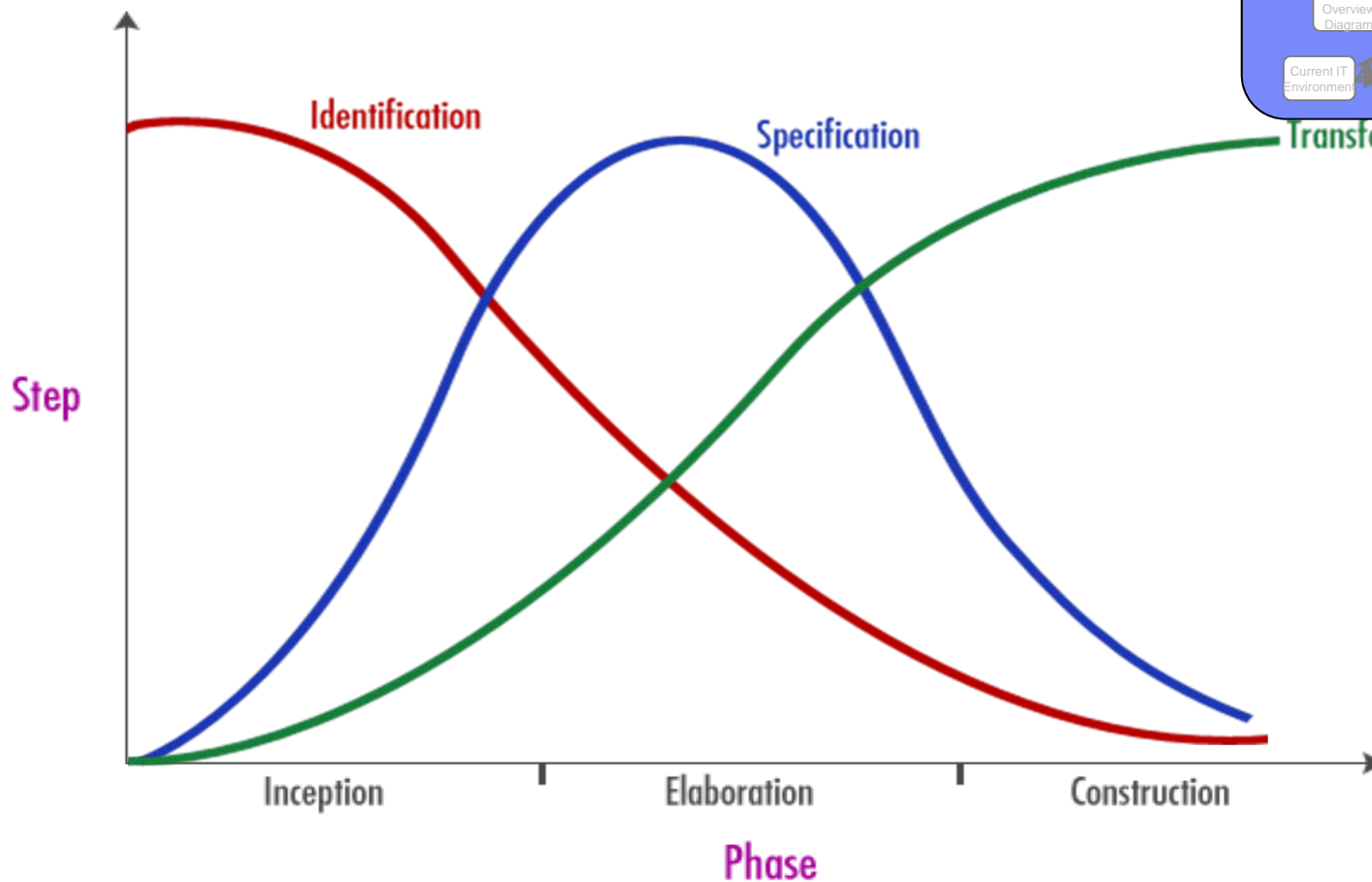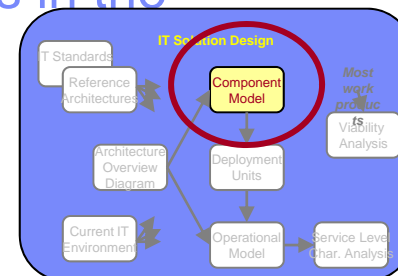
The Component Model
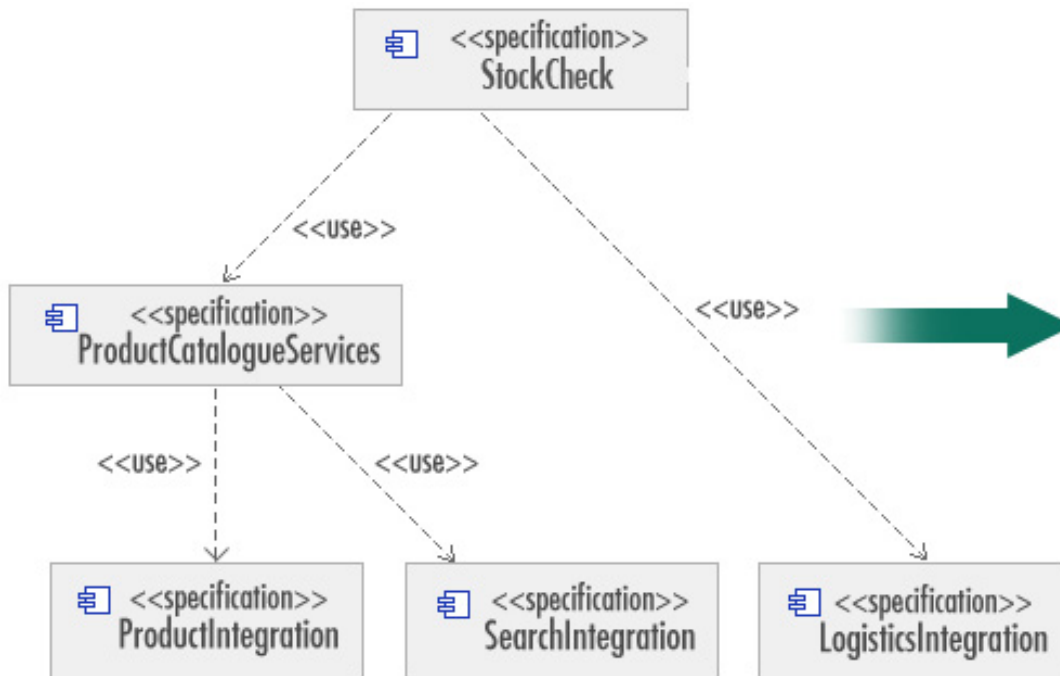
- **Bridge the gap between the requirements (the "what") and the solution (the "how")**

- **Visualize and help understand the system**

- **Specify the logical structure or behavior of the system**

- **Document decisions made**

- **Allow placement decisions to be made about where components will execute**

The Component Model is used as input into a number of activities
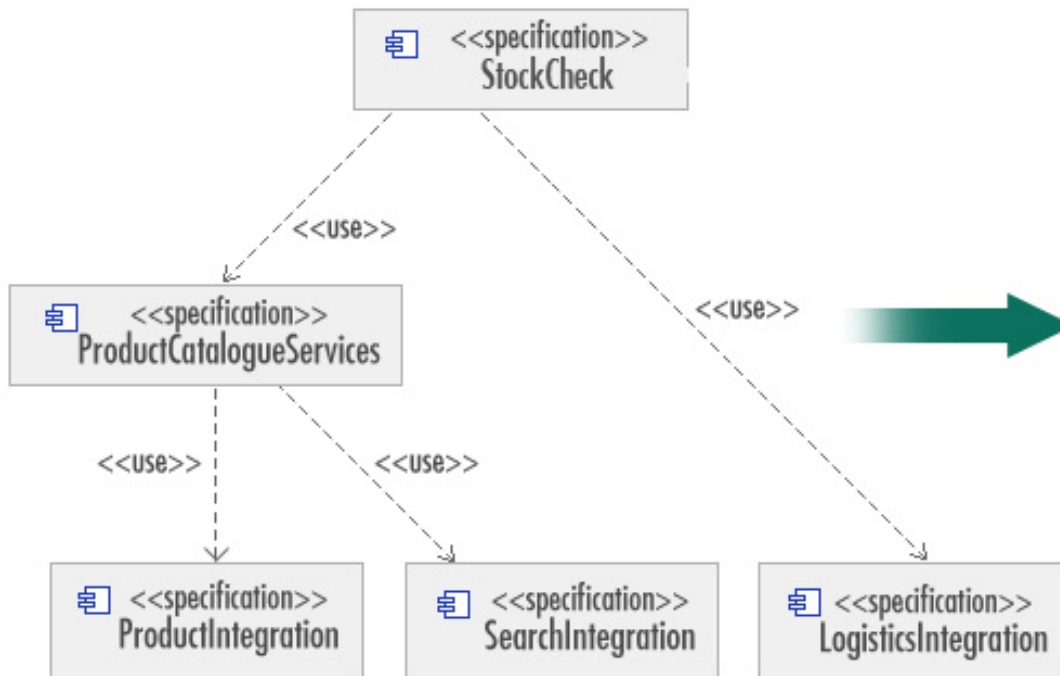
- **Work Allocation**
- **Version Control**
- **Design Strategy**
- **Reuse**
- **Testing**
- **Project Management**
- **Product/Package Selection**

Component Modeling often involves placing components into layers

- **Layering provides a logical partitioning of components into a number of sets (layers)**

- **Rules define relationships between layers**
  - Strict: Components only depend on components in the same layer or the one below
  - Non-Strict: Components may depend on components in any lower layer

- **Layering provides a way to restrict inter-component dependencies**

- **Well-layered systems are more loosely coupled and therefore more easily maintained**
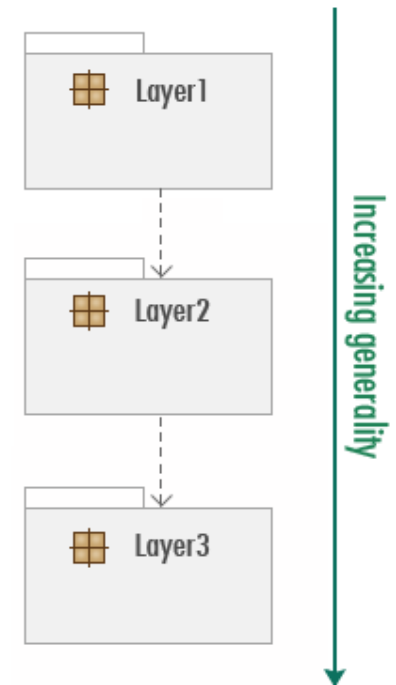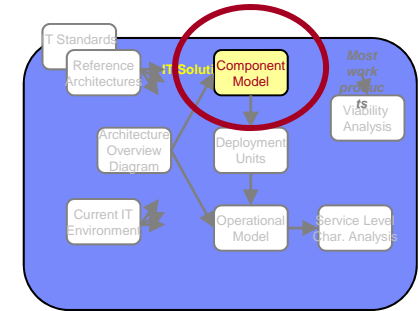
An example of layered architecture

- **The dialogue control layer handles user-system interactions and use case logic**

- **The business processing layer contains application-specific services that handle use case step logic and choreography**

- **The business services layer contains more general business components that may be used in several applications**

- **The middleware layer contains components such as interfaces to databases and platform-independent operating system services**

- **The system software layer contains components such as operating systems and databases**
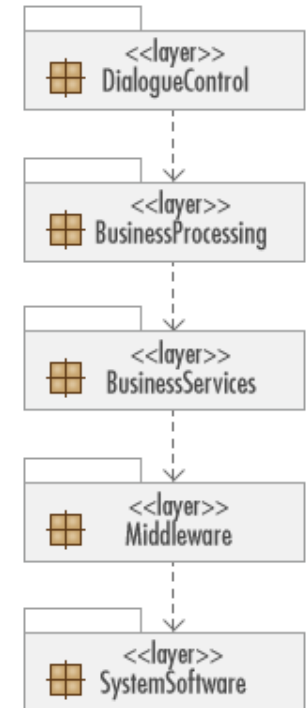
In a Multi-Tier System, each Tier can be layered independently

| -client | -applicationserver | -applicationserver | -databaseserver |

Client — ApplicationServer — DatabaseServer

Client:
- <<layer>> PresentationLogic
- <<layer>> Middleware
- <<layer>> SystemSoftware

ApplicationServer:
- <<layer>> DialogueControl
- <<layer>> BusinessProcessing
- <<layer>> BusinessServices
- <<layer>> Middleware
- <<layer>> SystemSoftware

DatabaseServer:
- <<layer>> Middleware
- <<layer>> SystemSoftware
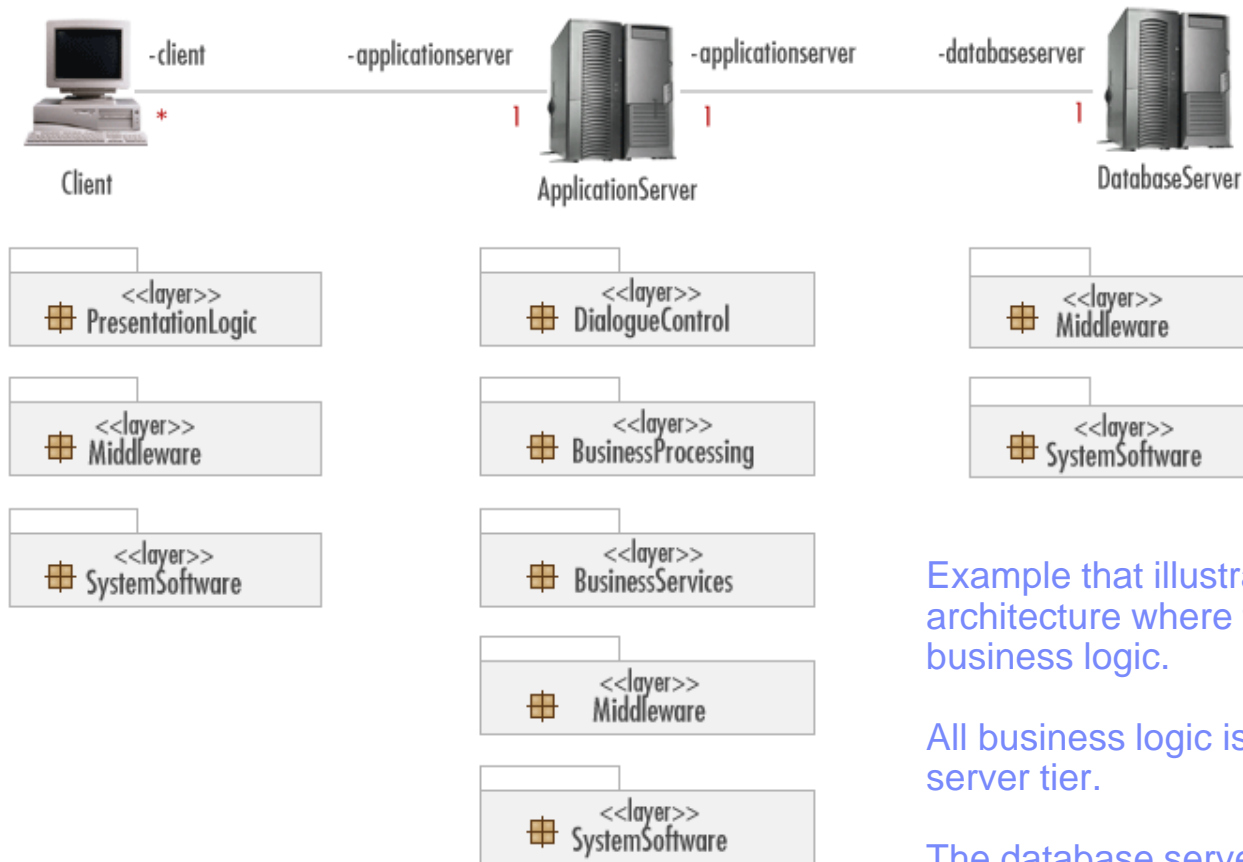
Example that illustrates a three-tier, thin-client architecture where the client contains no process or business logic.

All business logic is on the middle, application server tier.

The database server just contains middleware (that is, the database and communication software).

All tiers contain system software (such as an operating system)

The IT architect uses four core work products to document and communicate their IT system's design

Application Architect - Software Engineering

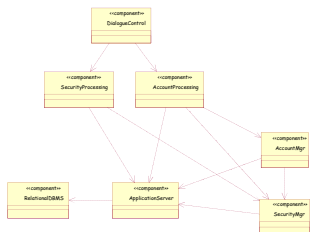**"Component Model"** describes the structure of an IT System in terms of its software components with their responsibilities, interfaces and relationships, and the way they collaborate to deliver the required functionality.

"Architecture Overview Diagram" provides a picture (not a model) of the whole IT system "on a page" as a means of communicating the salient points of the design. AODs are audience specific

IT Solution Design

- IT Standards
- Reference Architectures
- Component Model
- *Most work products*
- Viability Analysis
- Architecture Overview Diagram
- Deployment Units
- Current IT Environment
- Operational Model
- Service Level Char. Analysis

"Operational Model" defines the organisation of the IT system across locations, documenting the placement of the solution's components onto nodes connected across the organisation, in order to achieve the solution's operational NFRs

"Deployment Units" represent various aspects of components, as a convenient means of documenting their non functional requirements, as well as their placement across the Operational Model

Infrastructure Architect - Systems Engineering

University of Zurich

IBM

Interactions between functional and technical considerations
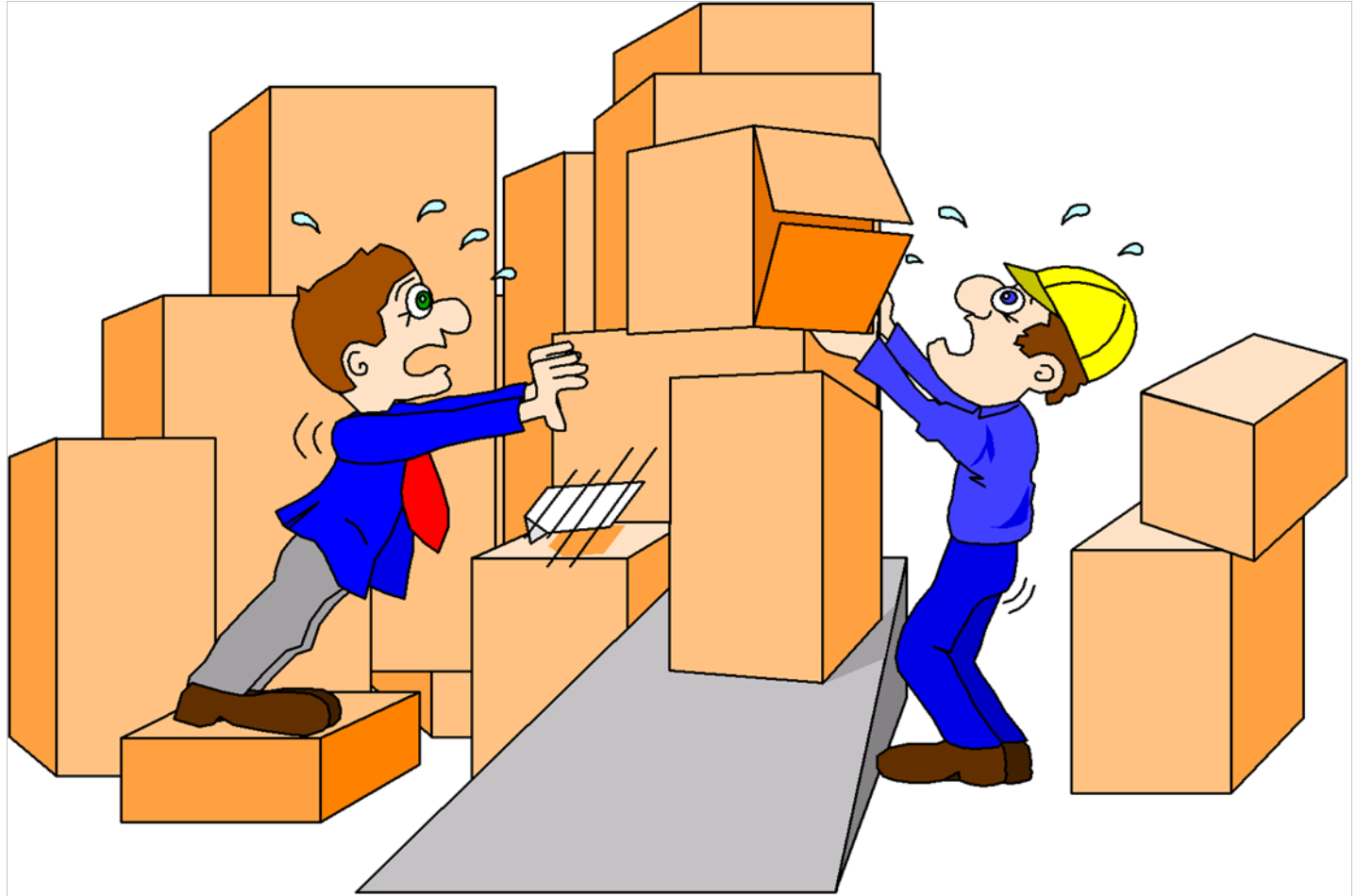
The Operational Model represents the system's "infrastructure architecture", using a variety of model elements

- The geographic structure of the locations and their borders, over which the IT system will be deployed and operated

- The placement of the system's nodes into these locations

- The deployment of the system's components across these nodes, using deployment units

- The connections between nodes

- The organisation of the system's elements into zones

- Sizing and other hardware specifications for all the computers, storage devices and network technologies

Operational Model
How do we decide where a system's components should go?

Let us consider a simple example: a "single component" system…

…Microsoft Word

Let us think about what we have to do, when "deploying" Word onto a very simple environment:

▪Where does "Word" go?
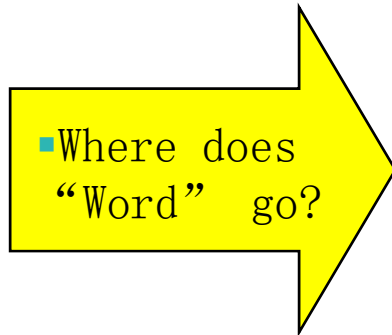
What is it we have to deploy? Where does "it" go? Let us sketch up some ideas…
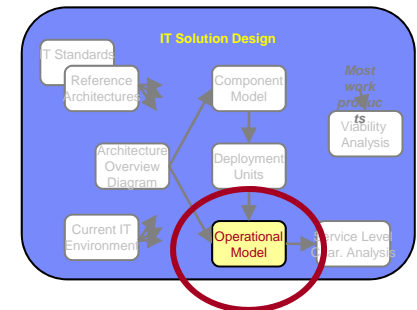
So, for our simple WORD example, we should first identify:

(1) That the Word component has the following deployment units:

DU	Description	Characteristics (e.g.)	
`P1_WYSIWYG_Display`	Microsoft Word desktop UI	Minimum screen size:	1024 x 768
`E1_Word`	Microsoft Word execution	Required operating memory:	512 MB
		Minimum CPU (equiv):	1Ghz
`D1_User_Documents`	Word documents being edited by the users	Typical document size:	5 MB
		Typical active documents:	100
		Some documents are critical to business operation	

(2) There will be two locations

– `L_Branch_Office`, which represents where Word users work
– `L_Central_site`, which represents a IT services data centre

(3) And two nodes

– `N1_Office_Workstation`, which represents a Microsoft Windows PC
– `N2_Central_Server`, which represents a Microsoft Windows Server

There are many ways of deploying a single component into a simple system…

Option 1

- **Presentation**
- **Execution**
- **Data**
- **Installation**

Word, running on XP, stand-alone

Option 2

- **Presentation**
- **Execution**
- **Data**
- **Installation**

- Word, running on XP, with remote file serving

Option 3

- **Presentation**
- **Data**

- **Execution**
- **Installation**

Word, running on Citrix, with local data

Option 4

- **Presentation**

- **Execution**
- **Data**
- **Installation**

Word, running on Citrix, with remote file serving

Option 1 - a local installation, with all DUs on the Office workstation…

L_Branch_Office

P1_WYSIWYG_Display

E1_Word

D1_User_Documents

▪N1_Office_ Workstation

☹ *Software update process must be provided and executed to 500 workstations per week*

☹ *Local backup and recovery of personal data*

▪N2_Central_Server

L_Central_Site
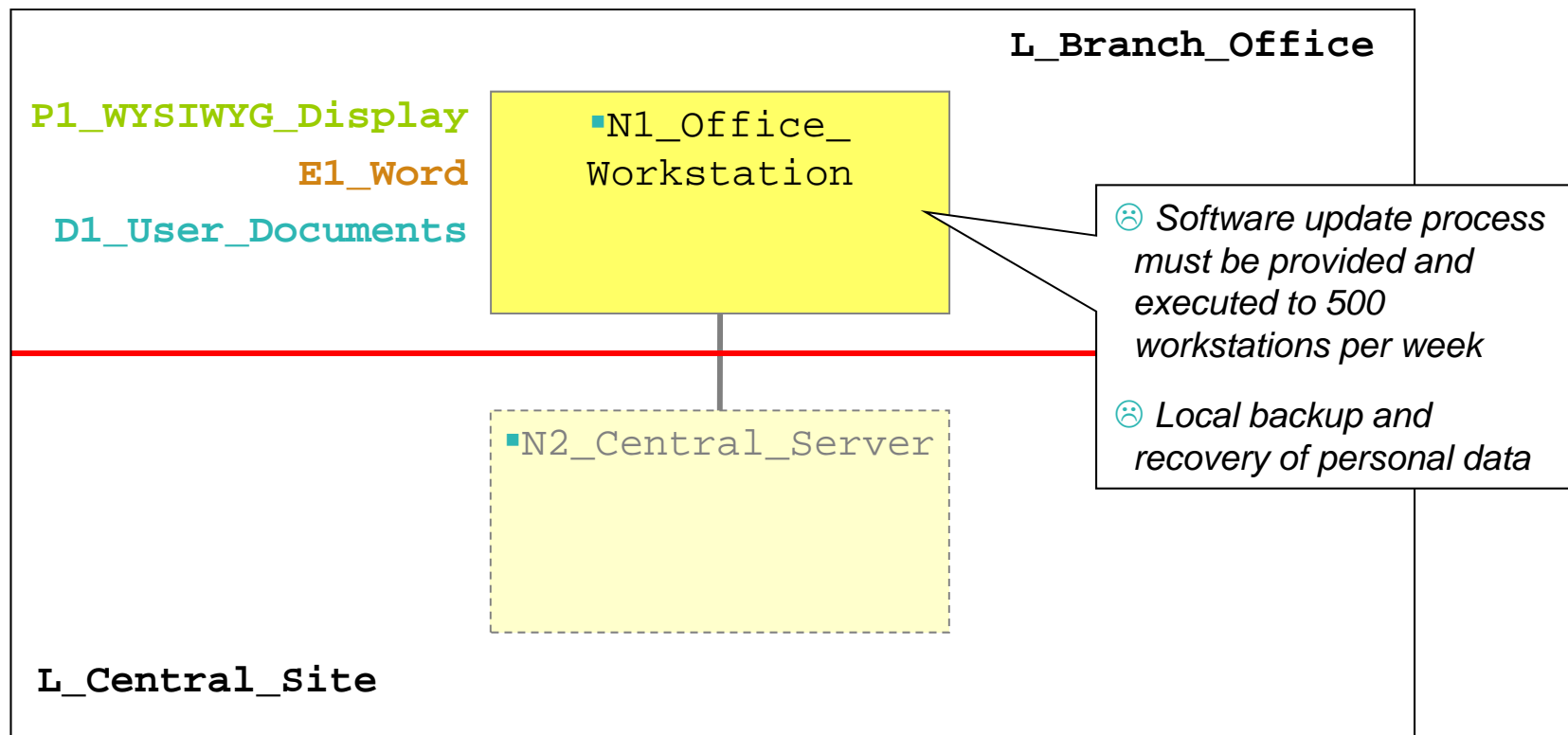
But this approach has many systems management issues:

– Software updates becomes an issue: e.g. regular security patches issued by Software vendor
– Backup and recovery the responsibility of the end user
– Number of users and number of branches

Reconsidering the placement decisions leads to…

Option 4: using a pattern based on a server-side installation with data served remotely

L_Branch_Office

P1_WYSIWYG_Display

▪N1_Office_
Workstation

▪N2_Central_Server

E1_Word

D1_User_Documents

☺ *Single update location –
guarantees that everyone in
the organisation is using the
most up-to-date Microsoft
Word image*

☺ *Central backup and recovery*

L_Central_Site
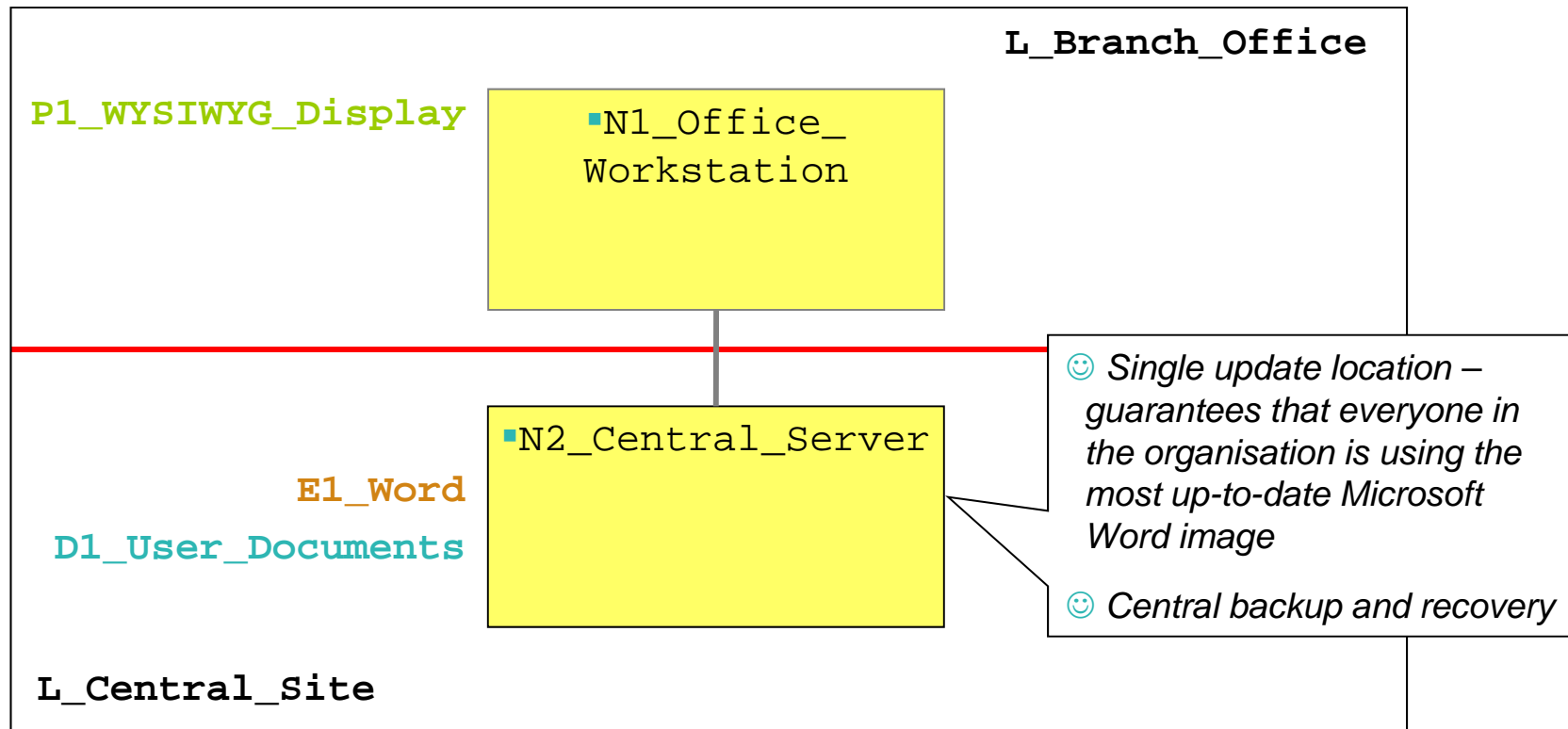
This is a much more manageable software update and data management regime.

However…

– …desktops may not have appropriate remote file server capability…
– …not all end users may be able to easily access central servers…
– …some workstations may be unable to support remote execution…

Operational Model -- Geographic Background

Logical Operational Model

ALN = Application Logical Node
D = Data Master ; d = Copy
E = Execution DU
P = Presentation DU

Call Center

ALN_ Customer_Service_Representative_PC

d_Catalog
E_Browse_and_Search_Catalog
P_Customer_Service_Representative

ALN_Call_Center_Services

d_Order
E_Order

Corporate Customer

ALN_Shopper_Services

d_Order
d_Catalog
E_Browse_and_Search_Catalog
E_Order_Client
P_PC

Internet Customer

ALN_Shopper_Services

P_Web

Data Center – e-Business Services

ALN_WWW_Shopping_Server

d_Catalog
d_Order
E_Search_and_Display_Catalog
E_Submit_Order

ALN_Order_Manager

E_Order_Server

Warehouse

ALN_Order_Fulfillment_Services

d_Order
E_Fulfillment
P_Pick

ALN_Shopping_Services

D_Order
D_Catalog
D_History
E_Credit_Client

Credit Agency

ALN_Credit_Services

D_Hot_Cards
E_Credit_Server

Data Center -- Backend

Physical Operational Model

IT Solution Design

IT Standards
Reference Architectures
Component Model
Architecture Overview Diagram
Deployment Units
Current IT Environment
Operational Model
Service Level Char. Analysis
Viability Analysis

A_CSR (30)

L_Call Centre (2)

N_CSR
RS/6000 43P
AIX 5L
(45)

100 E'net / HTTPS, MQ

PN_CSR services
IBM pSeries 620 F1
AIX 5L, WAS
(2)

Firewall
RS/6000 44P
Tivoli Secureway

512kbps / HTTP,HTTPS

Extranet

PN_Load Balancing
IBM pSeries 640 F1
Tivoli Edge Server
(2)   (2)

Router
Cisco

100 E'net / HTTP, HTTPS, MQ

PN_Order Mgt
IBM SP (P3-II)
AIX 5L, DB2, CICS

PN_WWW services
IBM SP (P3-II)
WAS Ap Svr

A_off line customer (1 in 3)

A_on line customer (2 in 3)

PN_Home Shopping
IBM Compatible PC
W95 or later
(1)

L_Private Customer (150,000)

56kbps / HTTP,HTTPS

Internet

Firewall
RS/6000 44P
Tivoli Secureway
(2)

Firewall
RS/6000 44P
Tivoli Secureway

PN_Security Svcs
IBM SP (P3-II)
Tivoli Secureway ND
(2)

L_Internet Services (2)

Intranet (3)

Router
Cisco

PN_Order fulfillment
IBM pSeries 620 F1
AIX 5L, WAS

512kbps / MQ

100 E'net / JDBC, MQ

PN_Security Svcs
IBM SP (P3-II)
Tivoli Secureway ND

PN_Credit Services
IBM xSeries 320
W2000, Credit S/w
(1)

Internet

X.25 / EDI

A_Credit Agency (2)

100 E'net / MQ

Router
Cisco

Intranet

Router
Cisco

L_Fulfillment Services (1)

PN_Picking Clerk
Lexmark
LaserJet XYZ
(5)

PN_Shopping Svcs
IBM pSeries 620 F1
AIX 5L, WAS

L_Central Services (1)

L_Credit Agency (1)

A_Picking Clerk
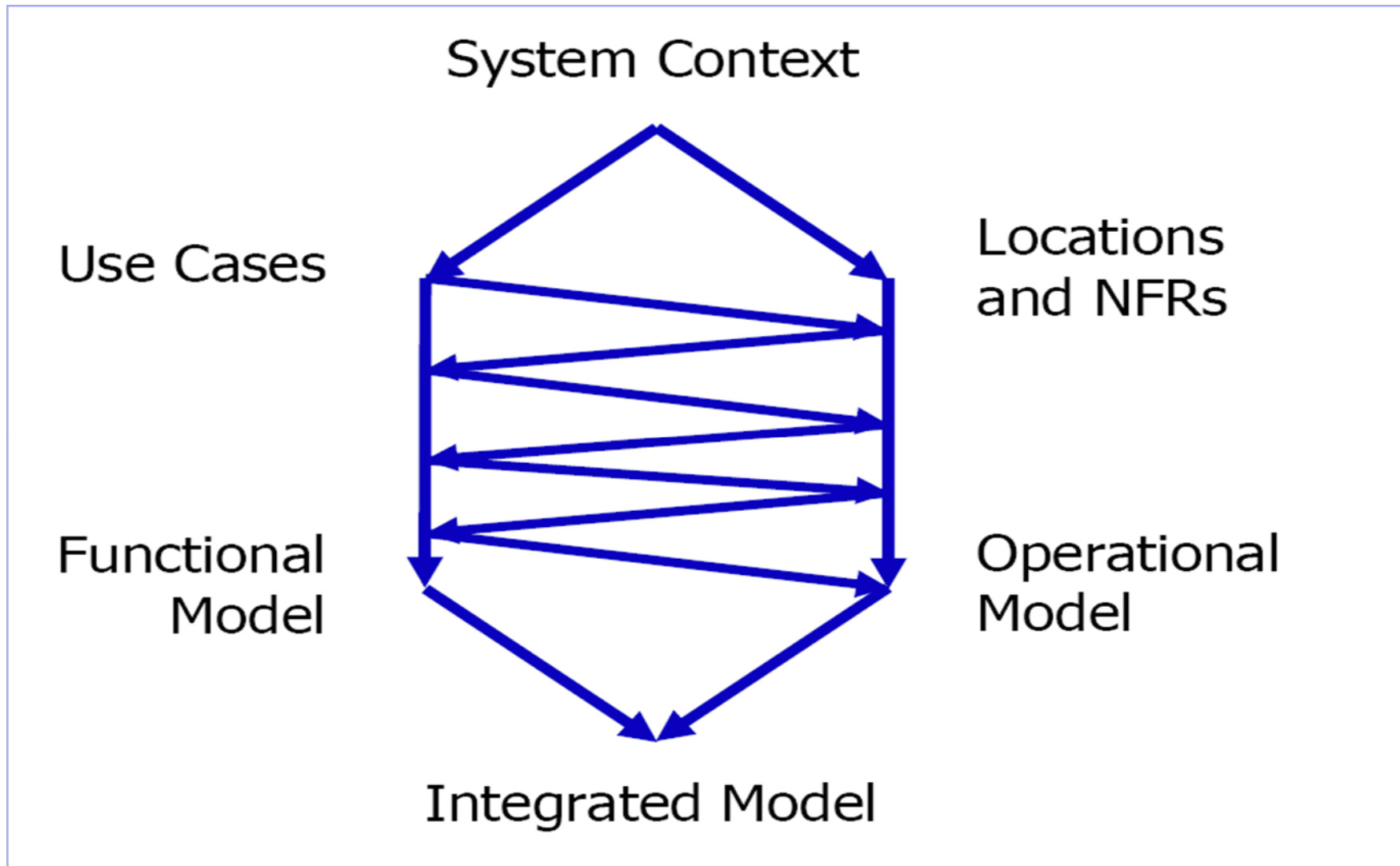(50)

Summary

…so that the project team (business and all parts of IT) can work more closely together, as well as helping ensure the project deals with difficulties very early in the lifecycle.

Phase Deliverable	Phase A	Phase B	Phase C
Business Requirements Specification	**Complete**	*Change control*	*Change control*
IT Solution Requirements Analysis		**Complete**	*Change control* **Bang!**
IT Solution Design			**Complete**

Phase Deliverable	Solution Outline	Macro Design	Micro Design
Business Requirements Specification	High Level, qualitative	Complete	*Change control*
IT Solution Requirements Analysis	Outline System **Bang!**	IT scope fully defined, key NFRs	Complete
IT Solution Design	Key identified	Outline solution defined	Complete

Catch "show-stopping problems" early in the project, enabling (if necessary) the project to be terminated at much less cost