
Solutioning Architectures

Method & Approach

12. August 2011

Kai Schwidder

© Zühlke 2011

Why a method / approach is helpful



■ **Goal:**

- Guides planning and delivery

■ **Objective: disciplined approach**

- Enables communication with teammates
- Break a large project into manageable 'chunks'
- Sync-Point where you left off with a customer
- Handover of knowledge between the customer and you



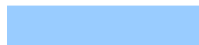
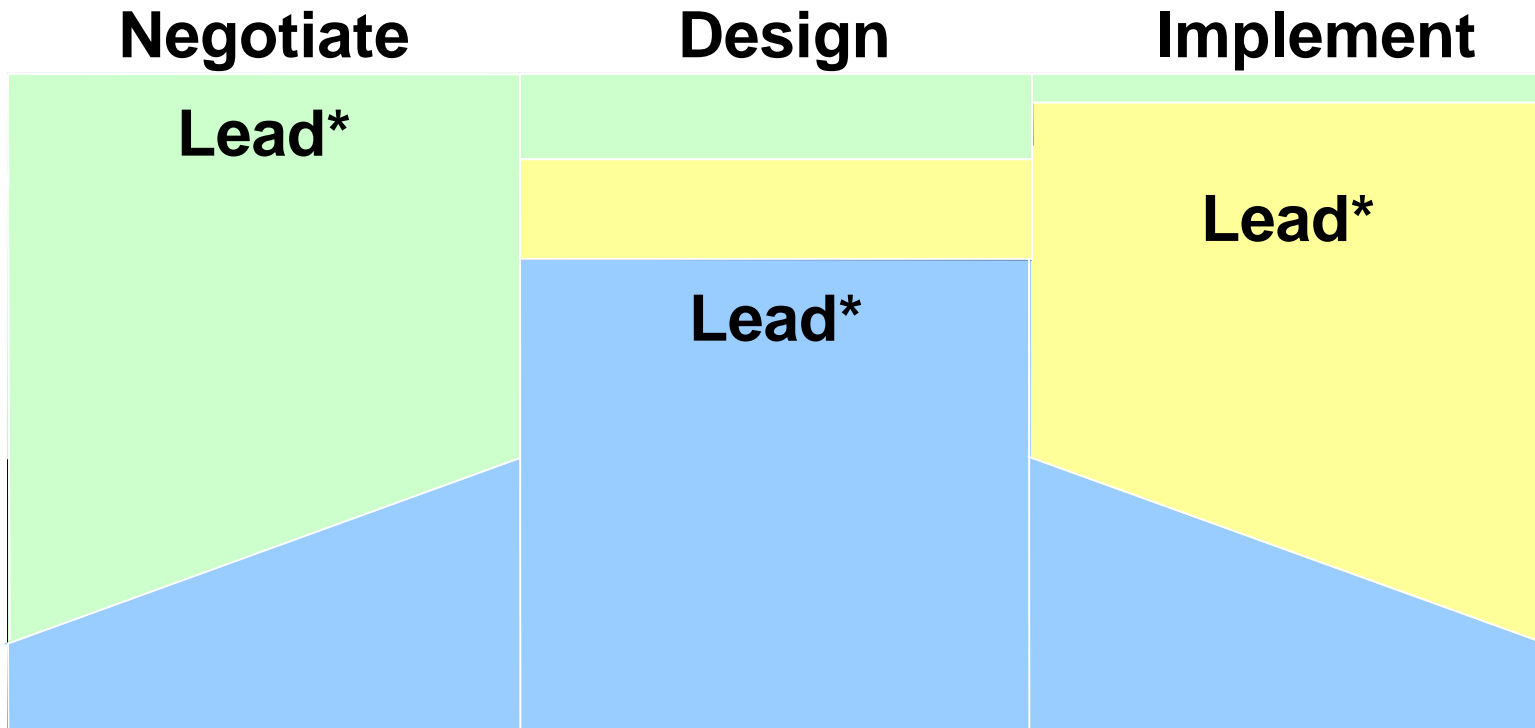
-
- Work products are comprehensive enough to capture input when you get it -- you rarely control timing.
 - Even when a method is not really needed, provides common vocabulary, approach to capturing notes, architectural way of thinking.
 - Outside-in approach is used, first with planning and then solution design.
 - Natural flow from System Context and Use Cases to Architecture Overview, Component and Operational Models.
 - Encourages the reuse of assets, from coarse-grained to detailed, business to IT, requirements to solution.

Philosophy of Solution Design – Part 2

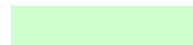


-
- Iterative approach with suggested iteration points.
 - Tasks are done as client situation allows, often in parallel.
 - Effective iteration through specific guidance on traceability of each decision back to requirements.
 - Iterative approach allows work products to evolve from general to more specific.
 - Each work product goes through multiple elaborations -- by you or others before and after the solution is "sold".
 - Provides just enough detail to assure the client that requirements are understood and solution will address them.

Solution Design Stages



Solution architect role

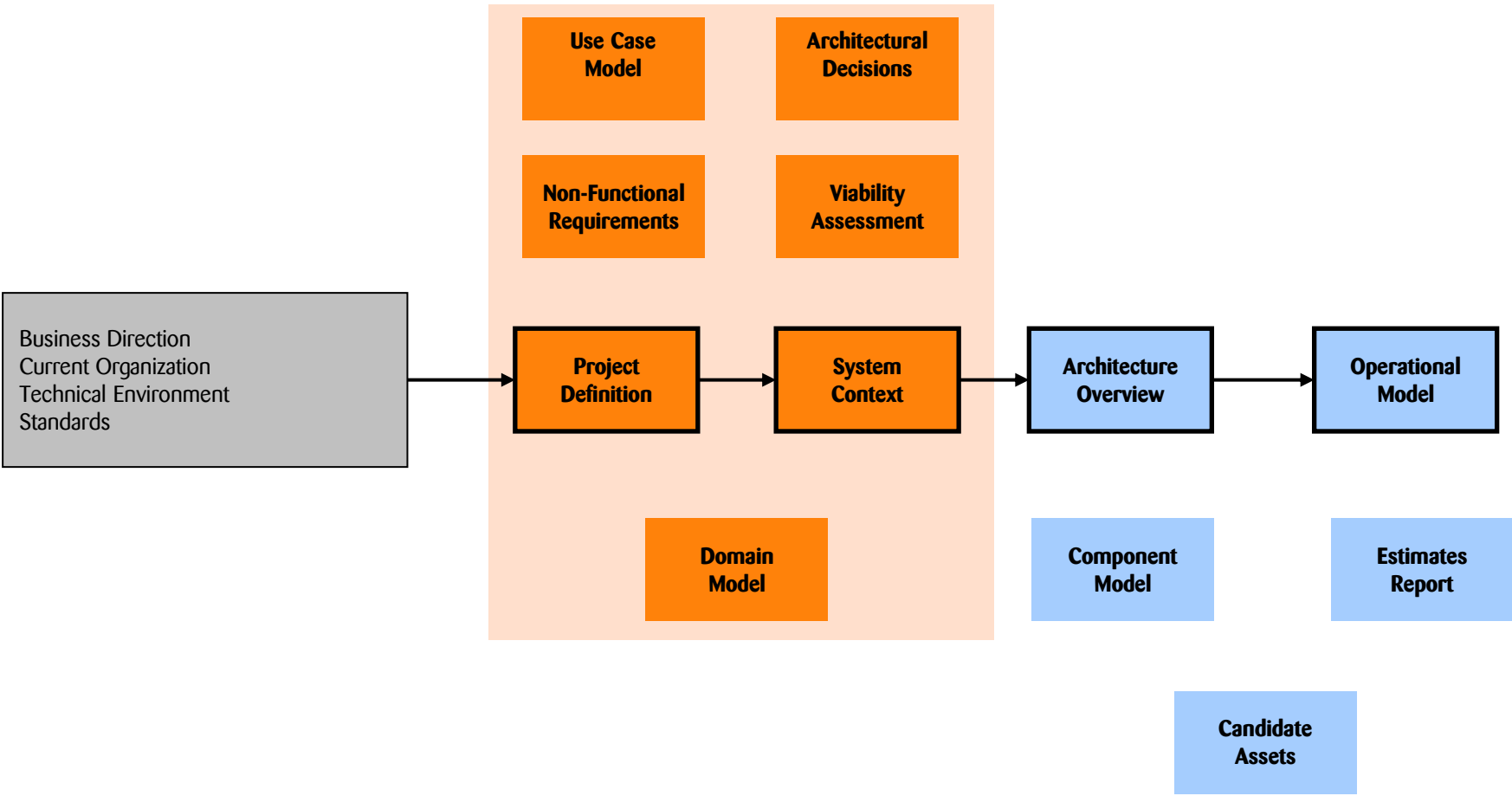


Sales role



Delivery architect role

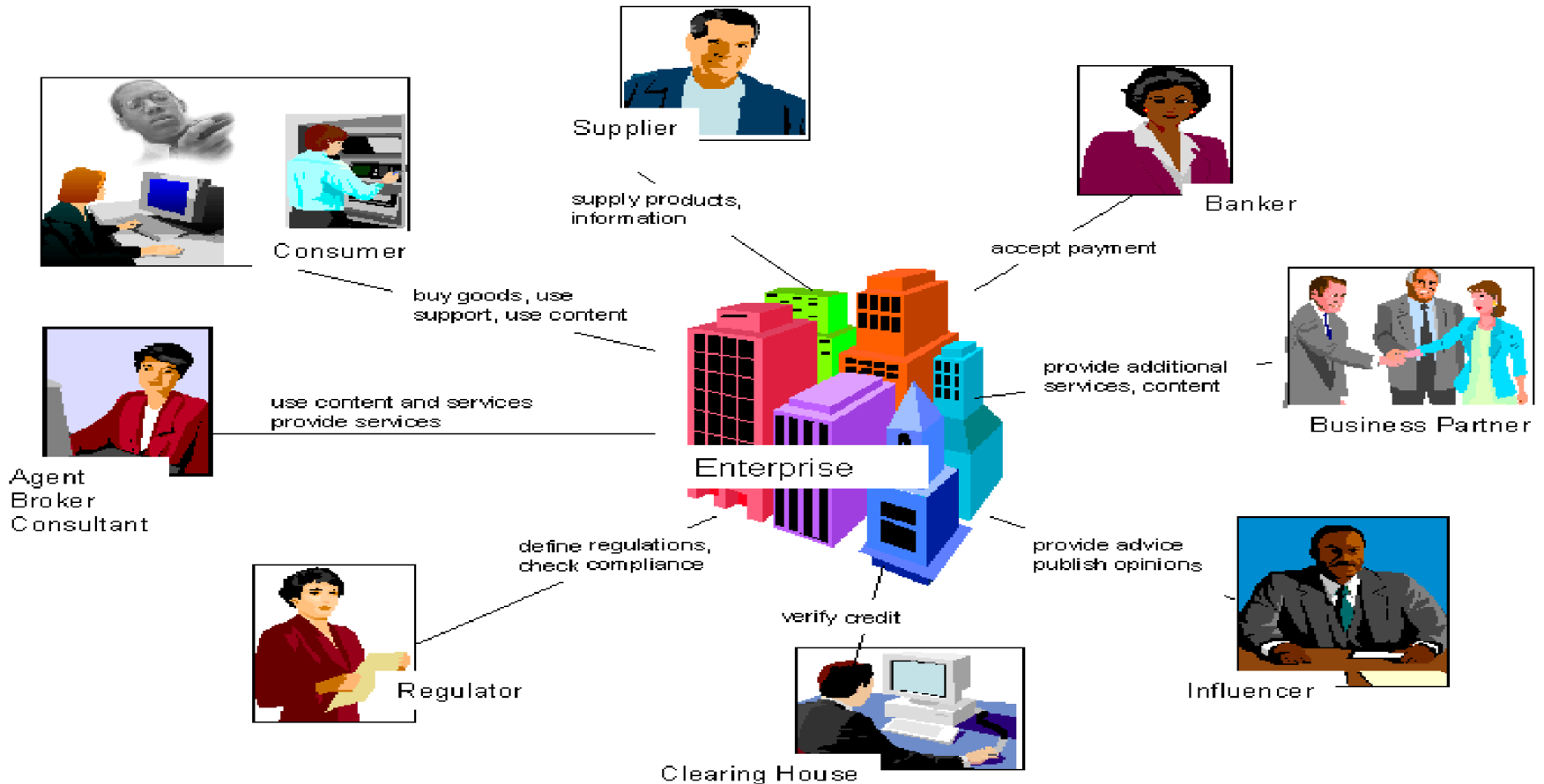
The critical path of a solution design and its work products



- Purpose :
 - Understanding of the client's Business Direction
 - To have an agreed basis for strategic decisions that will have to be made during the project.
 - To define the target for the assessment of the success of the project.
 - Discuss the state of their customer's business independent of I/T

- Description :
 - Understand the industry, the broad business climate, key industry initiatives of competitors and partners.
 - The Business Direction work product is a documentation of the Client's: Vision, Mission, Business Goals, Critical Success Factors. A diagram of the Business Context is often useful.
 - Scope narrows to this client's environment, their goals, issues, objectives and initiatives.

Example



■ Purpose :

- to communicate and gain agreement to the project goals and status.
This work product is defined by Team Solution Design and is required for every project.

■ Description :

- Answers to the questions: What, why, when, where, how and who?
- Provides a concrete starting point for solution design.
- Critical work product since most others used are dependent on it.
- Together with Architectural Decisions, it ties the other work products together.
- Project Definition and Viability Assessment are the primary mandatory work products for reviews.

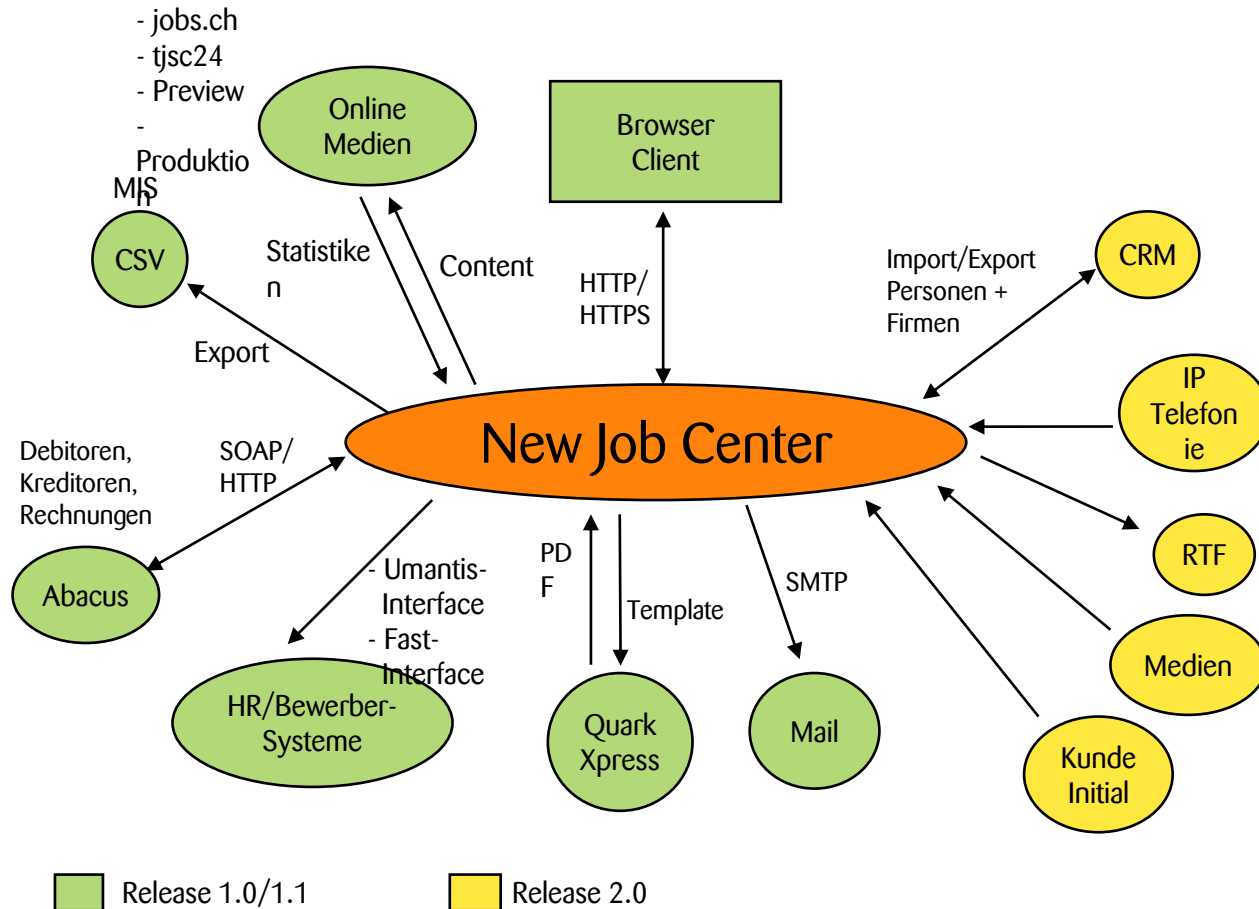
■ Purpose :

- Provide a basis for understanding the system to be proposed.
- Define how the proposed system will interoperate with other existing systems.
- Establish boundaries on the scope of the proposed system
- To clarify and confirm the environment in which the system has to operate.
- To provide the details at an adequate level to allow the creation of the relevant technical specification.

■ Description :

- The proposed system is treated as a black box with connections to other systems.
- Documents all connections between the proposed system and external systems/components.
- For each connection identify important attributes such as protocol, formats, frequency and volume.
- Users, external systems, batch inputs and outputs, and external devices.
- External events and data to which the system must respond.
- Events and data that the system generates that affect external entities.

Example : System Context



■ Purpose :

- The purpose of this task is to identify non-functional requirements that will affect the design and resulting performance of the system.
- To define requirements and constraints on the IT system.
- Provide a basis for early system sizing and estimates of cost and viability assessment of the proposed IT system.

■ Description :

- Identify service level requirements such as performance, capacity, volumes, availability, portability, maintainability, systems management and security.
- Identify system constraints imposed by the client with regard to cost, location, configuration, standards, vendor preferences and technology preferences.
- Documents the non-functional aspects of an IT system including examples such as:
- Performance, scalability, availability, maintainability, manageability, usability, accessibility, and data Integrity

-
- Availability
 - Backup & Recovery
 - Capacity Estimates and Planning
 - Configuration Management
 - Disaster Recovery
 - Environmental factors
 - Extensibility/Flexibility
 - Failure Management
 - Maintainability
 - Performance
 - Quality of Service
 - Reliability
 - Scalability
 - Security
 - Service Level Agreements
 - Standards
 - Systems Management

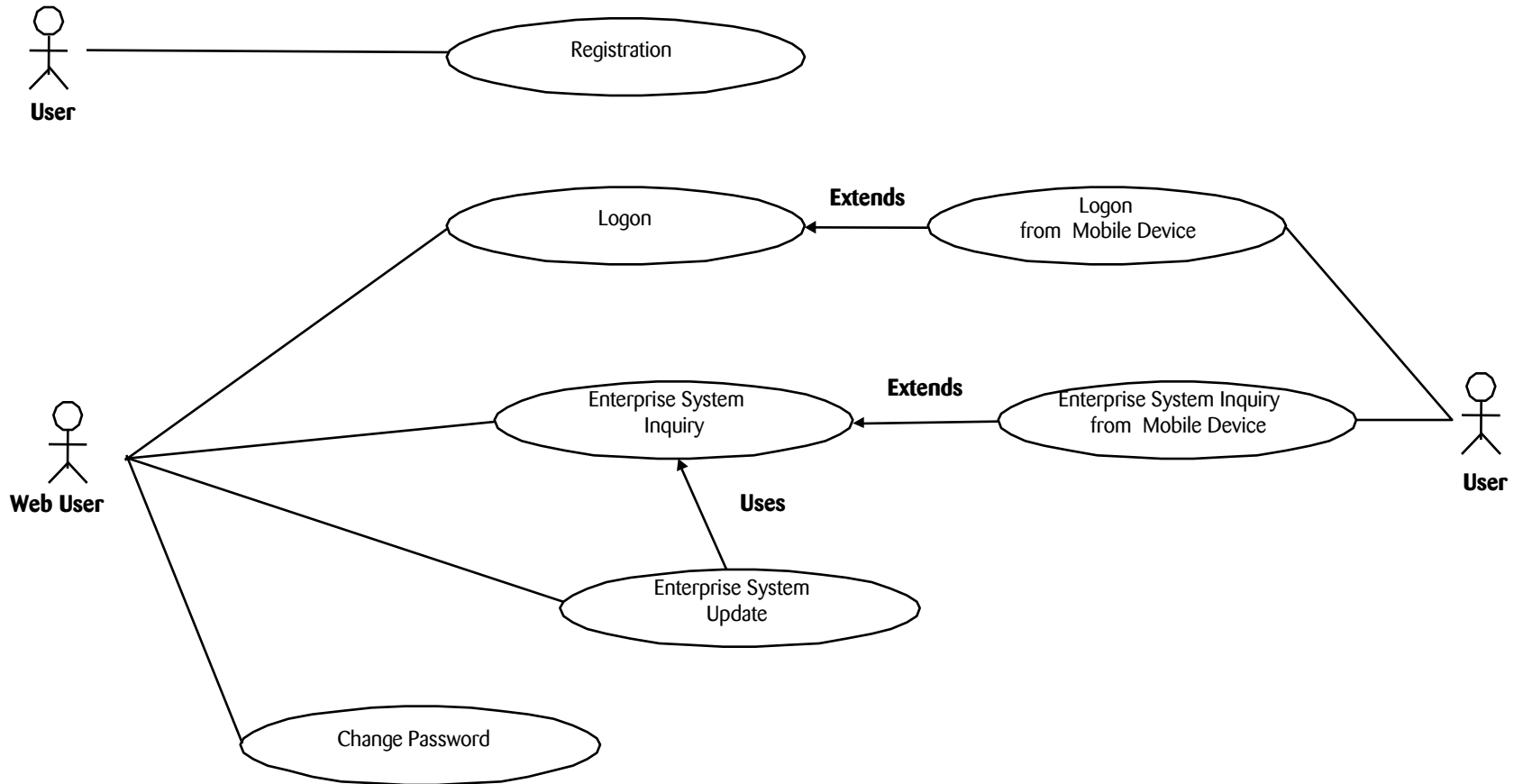
■ Purpose :

- Define a set of basic use cases that depict how the user will use the proposed system.
- Understand and document additional functional requirements
- Establish a small number of important scenarios that depict how the user will use the proposed system
- Provide a basis for planning a proof of concept and high level architectural walkthroughs

■ Description :

- Identify additional functional requirements when required for more complex solutions
- Develop initial use cases at a conceptual level.
- A set of use cases which illustrate primary usage scenarios and relationships of actors and use cases

Example : Use Case



■ Purpose :

- Identify and describe at a high level, data sources which are relevant to the proposed solution.
- Contribute to an understanding of architectural aspects which are necessary to use existing data sources
- Convey, graphically or textually, the scope of an enterprise, a desired capability or application from the point of the view of the data or information required to support the enterprise, application or capability.

■ Description :

- Document high existing level data sources that will be used as a part of the solution
- Document additional high level data sources that will be required for the solution
- Develop a conceptual data model of all relevant high level data sources
- Graphical as well as textual document of the major groupings of entities that are needed to support an enterprise, a capability or an application
- Also referred to as a Conceptual Data Model or Business Information Model

■ Purpose :

- Identify and document important architectural decisions where alternatives exist, choices are unclear and impact is likely significant.
- Provide a single place to find important architectural decisions
- Make explicit the rationale and justification of Architectural Decisions
- Avoid unnecessary reconsideration of the same issues

■ Description :

- Document architectural decisions regarding principles or policies.
- Document architectural decisions regarding elements of the architecture.
- Ensure the issue or problem is clearly stated, evaluate the options that are available, make the decision.
- An Architectural Decisions work product documents important decisions about any aspect of the architecture including the structure of the system, the provision and allocation of function, the contextual fitness of the system and adherence to standards.

Example : Architectural Decision



Kategorie	Nummer	Name	Kommentar
Schnittstellen	ARCH-01	CSS-Style Sheets	Für das Look&Feel werden CSS-Style Sheets verwendet
	ARCH-02	E-Mail Schnittstelle	SMTP wird als Protokoll verwendet
	ARCH-03	Umsysteme	Es wird SOAP mittels HTTP/HTTPS verwendet
	ARCH-04	Provider	Es werden über Standard-Schnittstellen Content-Elemente und Statistiken übermittelt.
NJC	ARCH-05	Lose Kopplung	Die Komponenten sind lose gekoppelt und ermöglichen den Ausbau des Systems
	ARCH-06	SOA	NJC folgt dem Service Orientierten Architektur Prinzip
	ARCH-07	Workflow	Die Status-Verwaltung der Aufträge wird durch ein Regelsystem abgebildet.
	ARCH-08	Optimistic Locking	Für die Daten-Operationen wird das Optimistic Locking verwendet. Parallele Updates von Daten müssen ggfs. vom Benutzer verwaltet werden.
	ARCH-09	Dienste sind "Stateless"	Alle Dienste werden "Stateless" implementiert. Somit kann die Skalierbarkeit in Zukunft gewährleistet werden.
	ARCH-10	Geschäftsregeln	Dienste können Geschäftsregeln ausführen. Hierdurch können Änderungen flexibel gepflegt werden (z.B. Preisfindung).
	ARCH-11	Datenbank	NJC basiert auf einem relationalen Datenbanksystem und verwendet SQL als Abfragesprache. Es wird MS-SQL eingesetzt.

■ Purpose :

- To qualify the opportunity: assess if the opportunity qualifies for further investment.
- To make an initial assessment of the viability of the solution ensuring that it lies within the "art of the possible".
- To identify unrealistic or challenging requirements as early as possible, and seek to re-negotiate them.
- Together with Project Definition, it is the primary vehicle for communication.
- Besides project risks, documents issues, assumptions and dependencies that might impact the proposal, implementation and delivery.

■ Description :

- Depending on the risk and complexity of the project, several formal peer reviews that might be required.
- Reviews include: Technical Delivery Assessment (Solution Assurance), Integrated Technical Review, Proposal Baseline Assessment and Project Management Review.

Example : Viability Assessment



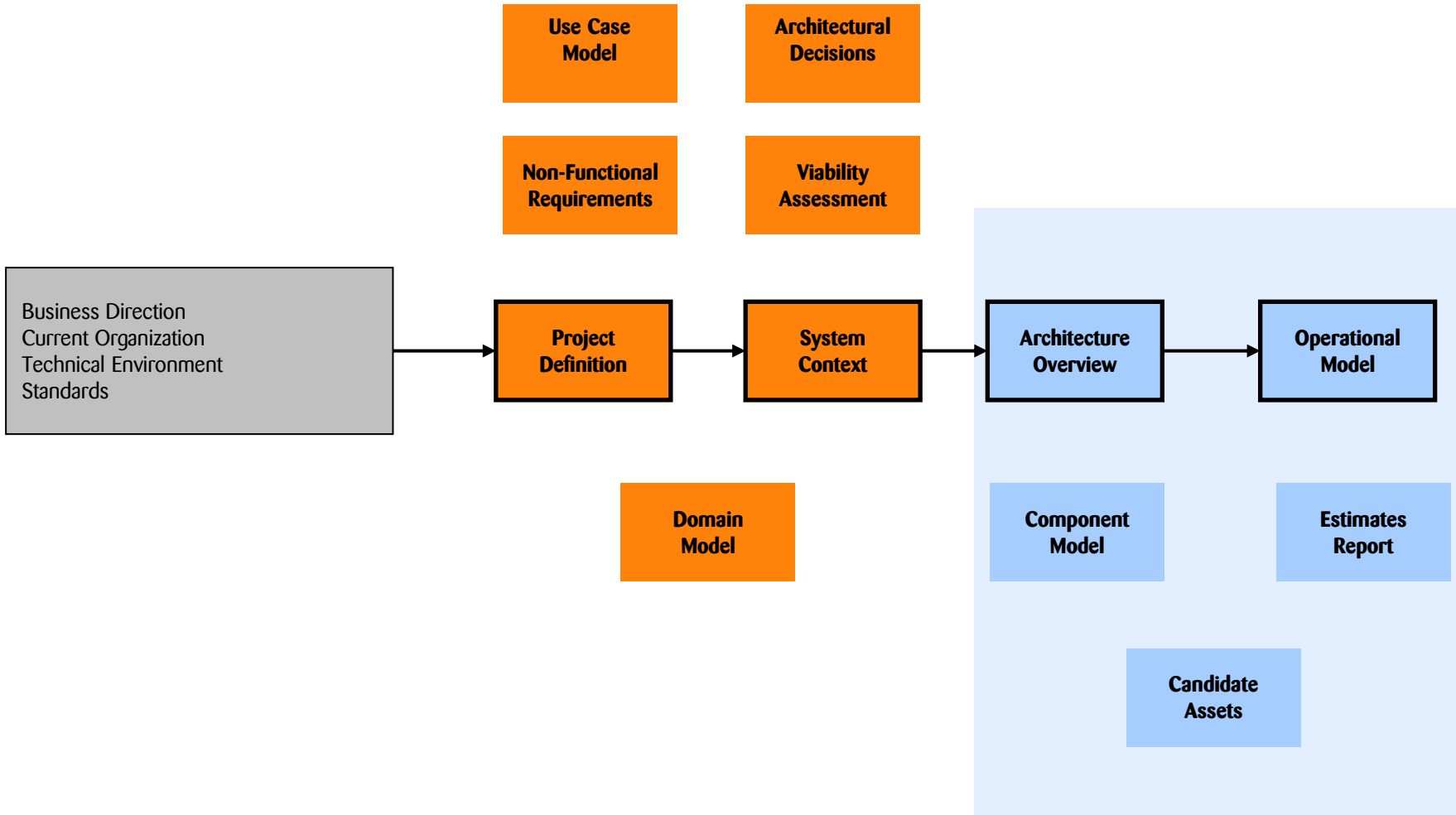
Risk ID	Finding / Risk Description	Probability (H/M/L)	Impact (H/M/L)	Effort / Cost	Contingency / Mitigation Recommendation	Person Responsible	Review Date
R01							
R02							
R03							

Assumption ID	Finding / Assumption Description	Confidence Level (H/M/L)	Impact (H/M/L)	Assumption Identified By	Review Date	Closed Date
A01						
A02						
A03						

Issue ID	Finding / Issue Description	Priority (H/M/L)	Raised By / Date	Issue Responsibility	Review Date	Action: Closed or Risk / Change Reference
I01						
I02						
I03						

Dependency ID	Finding / Dependency Description	Effect on Plan	Required By Date	Owner	Associated Risk ID	Closed Date
D01						
D02						
D03						

The critical path of a solution design and its work products



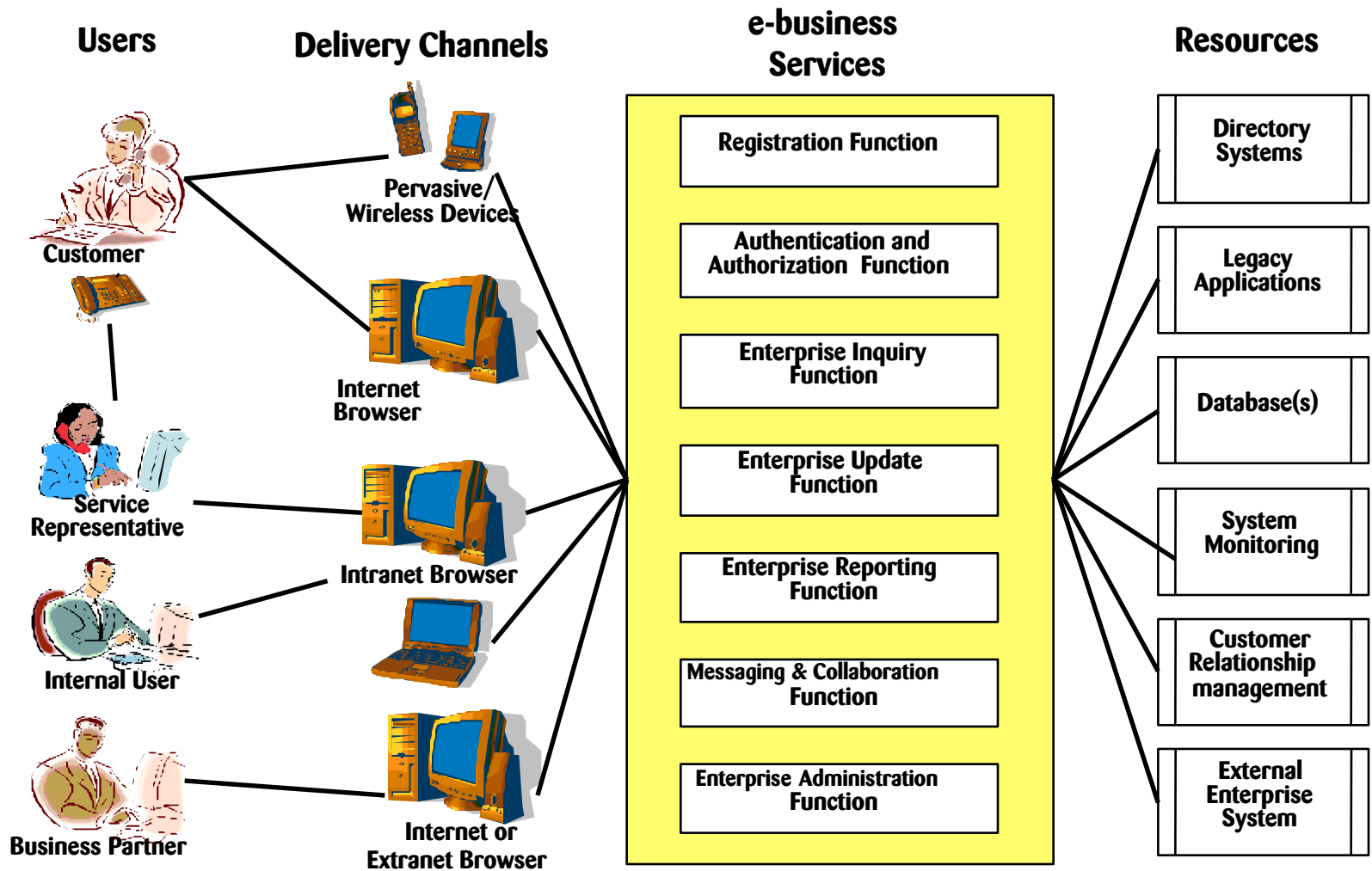
■ Purpose :

- Develop a high level abstraction of the proposed system or application.
- Provide a basis for discussion, explanation and evolution of the architecture.
- Provide a high-level shared vision of the architecture and scope of the proposed IT system
- Explore and evaluate alternative architectural options

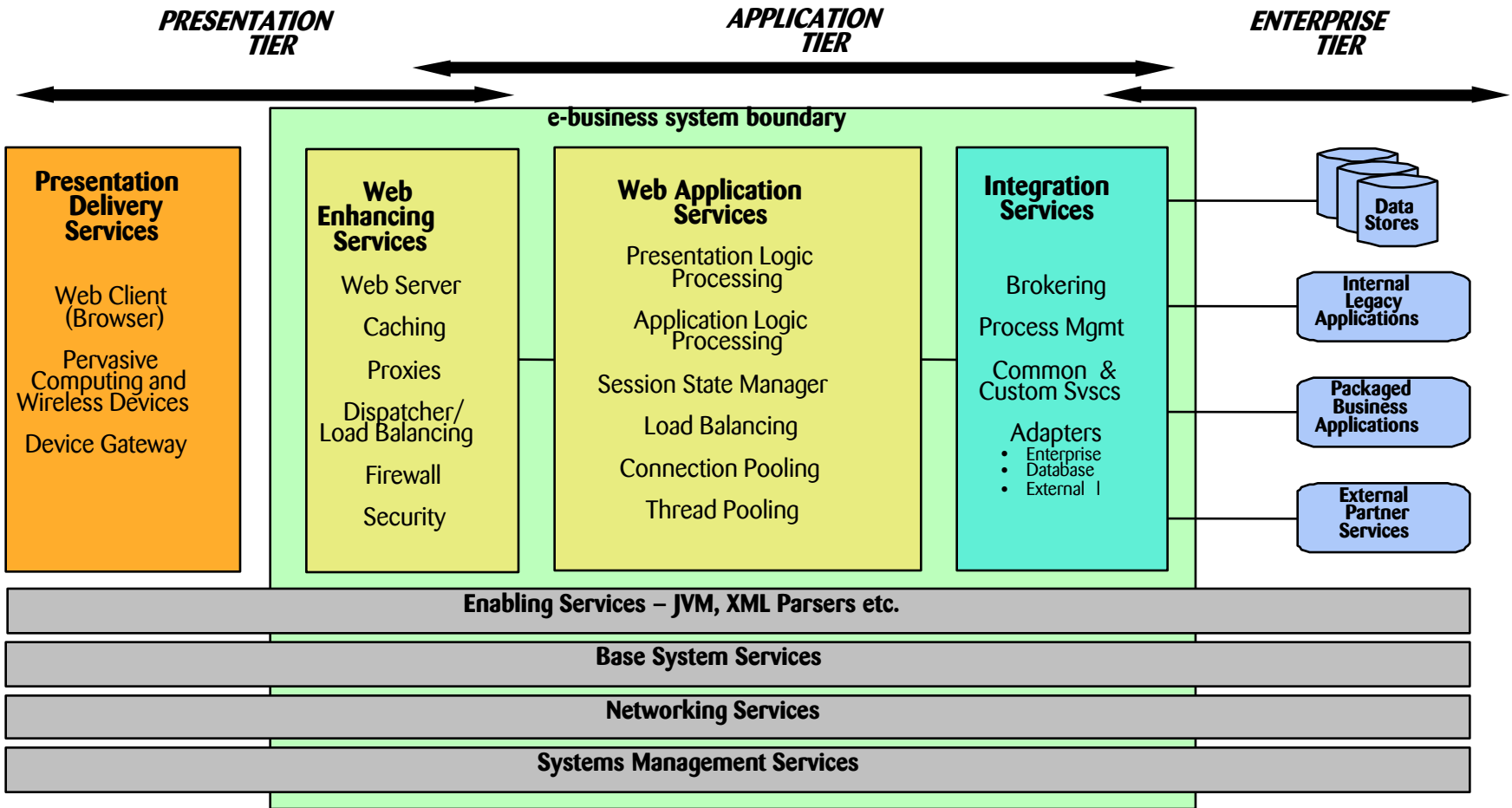
■ Description :

- What non-functional requirements exist?
- What system constraints or preferences exist? * What are the goals for this project?
- What application requirements have been defined?
- What are the system or application boundaries?
- What systems or subsystems are known to be a part of this system or application?
- The Architecture Overview work product is a schematic diagram that represents the governing ideas and candidate building blocks of an IT system or enterprise architecture, typically including candidate subsystems, components, nodes, connections, data stores, users and external systems.

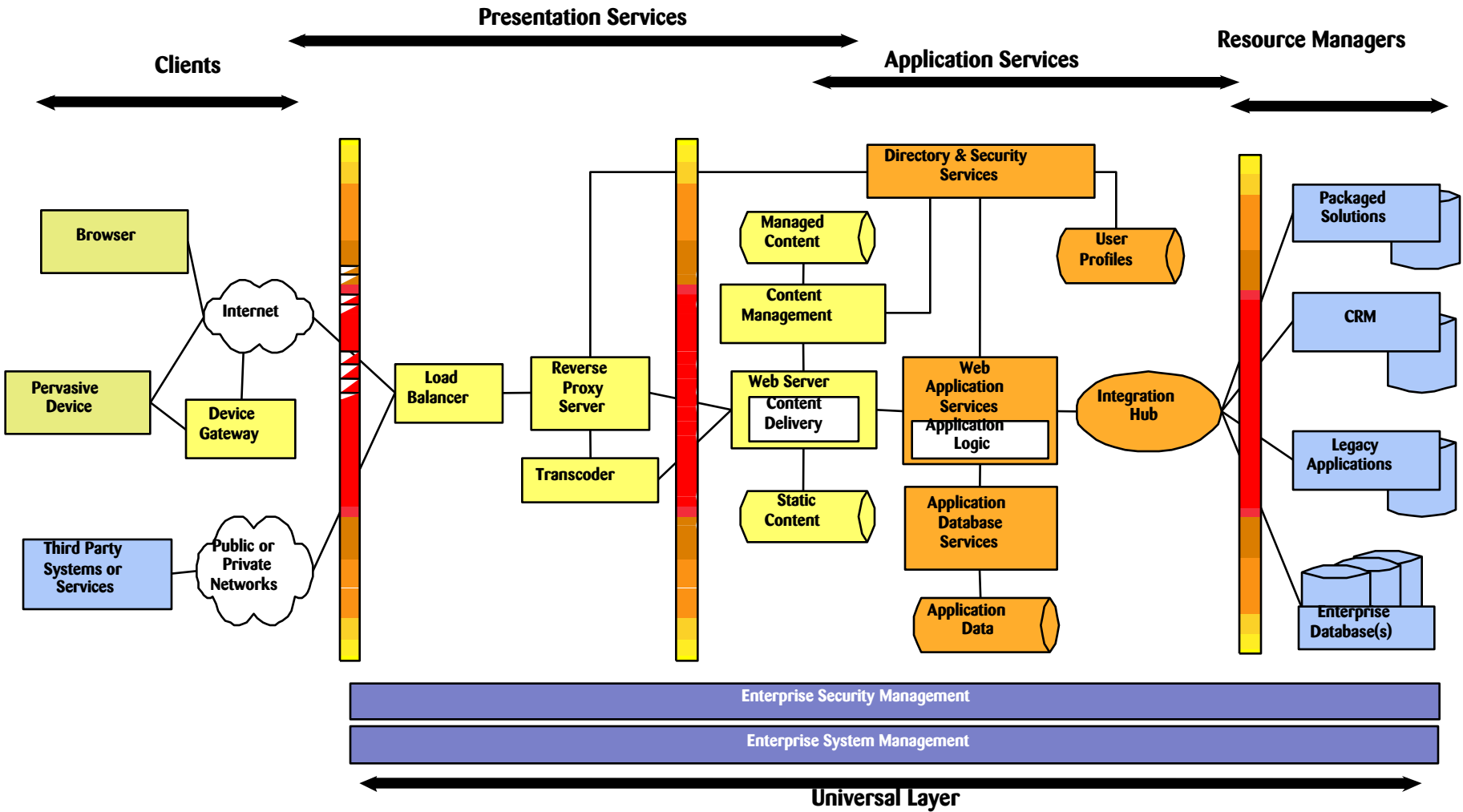
Example : Architecture Overview



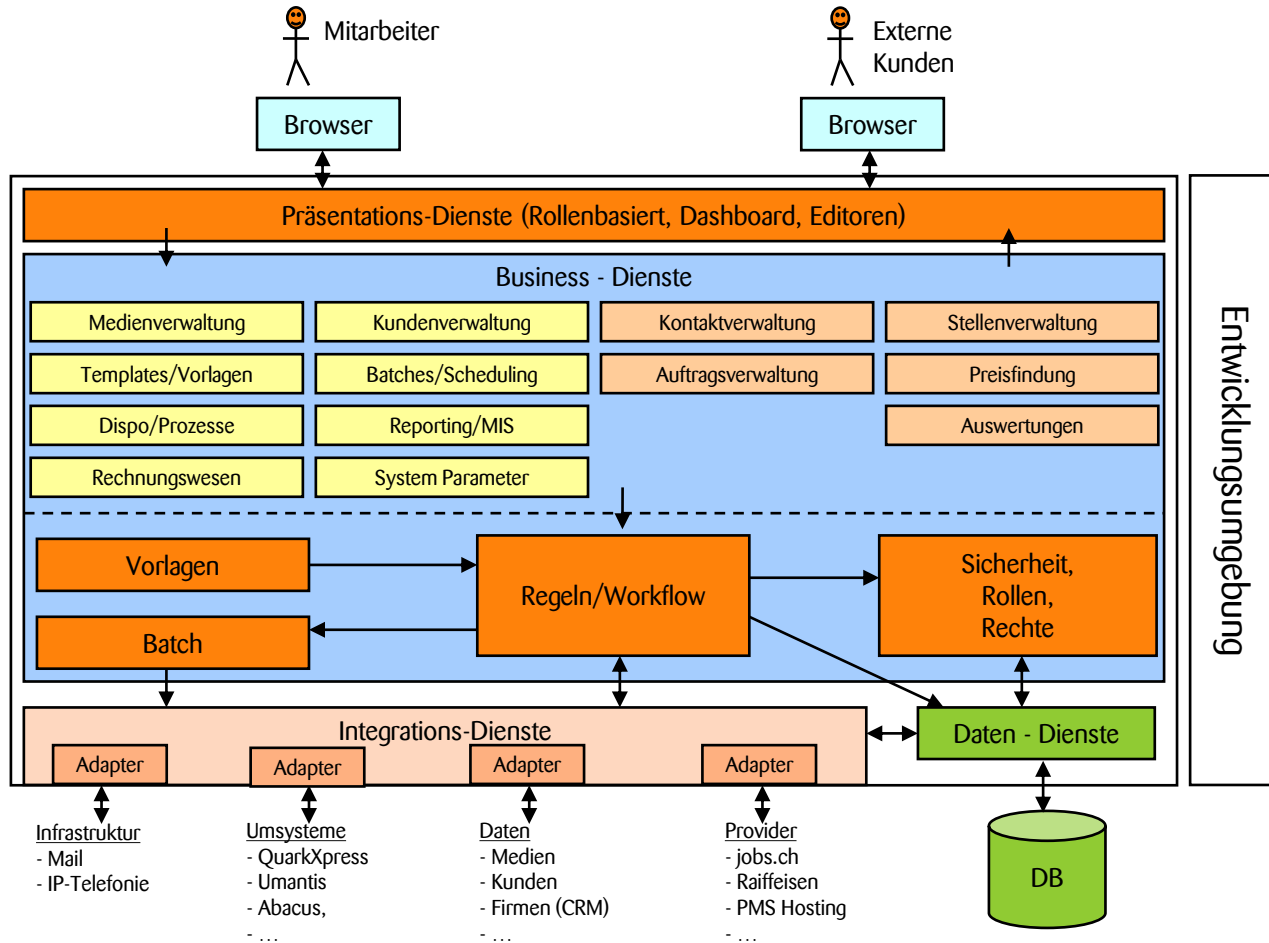
Example : Architecture Overview



Example : Architecture Overview



Example : Architecture Overview



■ Purpose :

- Identify architectural assets that might be relevant for the project.
- Analyze the fit and gap between assets and project requirements.
- Decide whether to base areas of the solution on available assets.

■ Description :

- Identify the need.
- Understand the project scope and the general functionality required. Find assets and analyze fit/gap.

■ Key Considerations :

- If applicable, hold a Solution Optimization Workshop to maximize the attractiveness of the solution and minimize the cost.

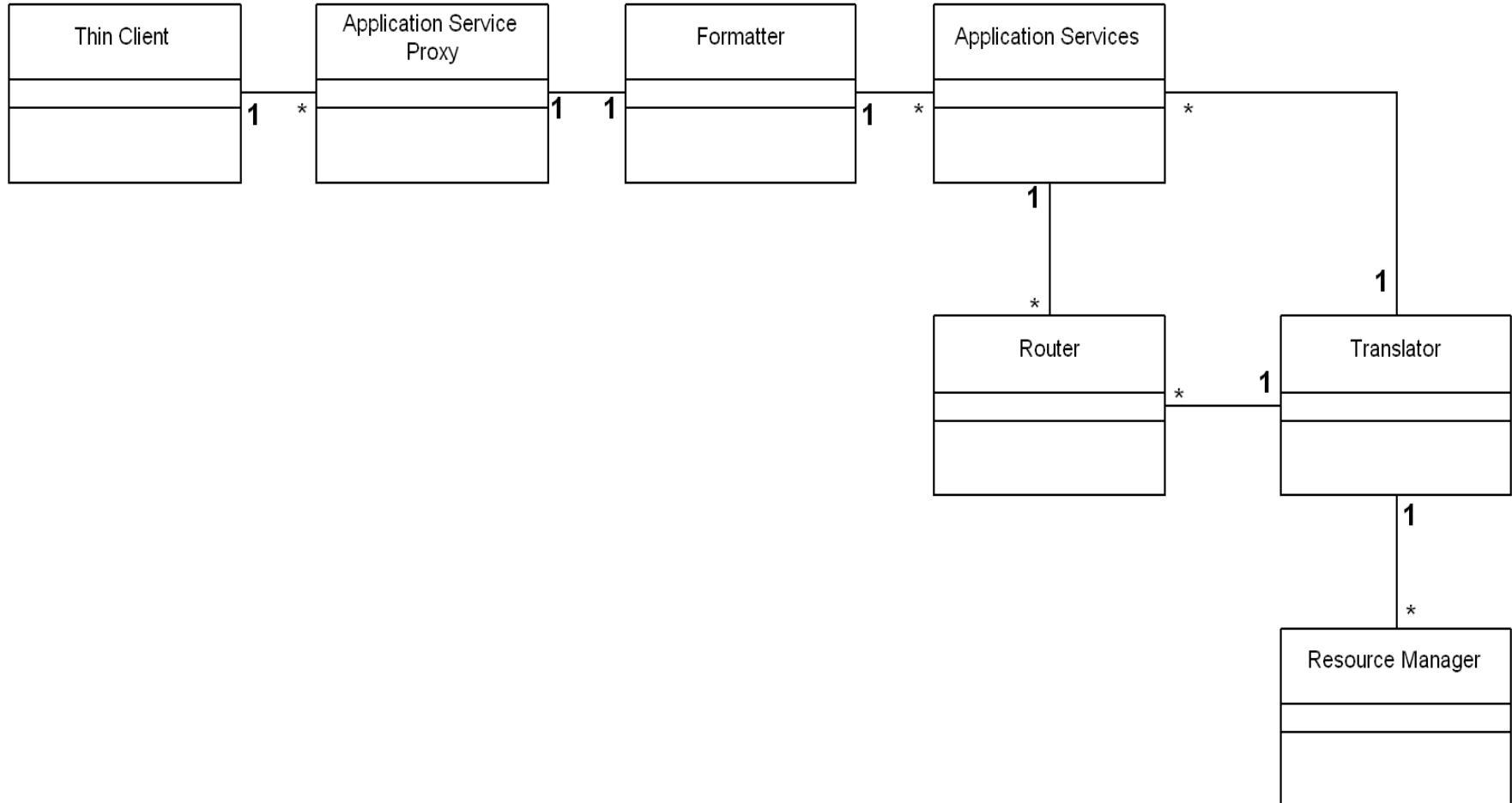
■ Purpose :

- Describe the high-level structure of the system
- Describe precisely the responsibilities, relationships boundaries and interactions of components
- Document how application/technical parts of the system are related
- Help define and document the structure of a particular IT System.
- Document the recurring interactions and dependencies between particular sets of components.

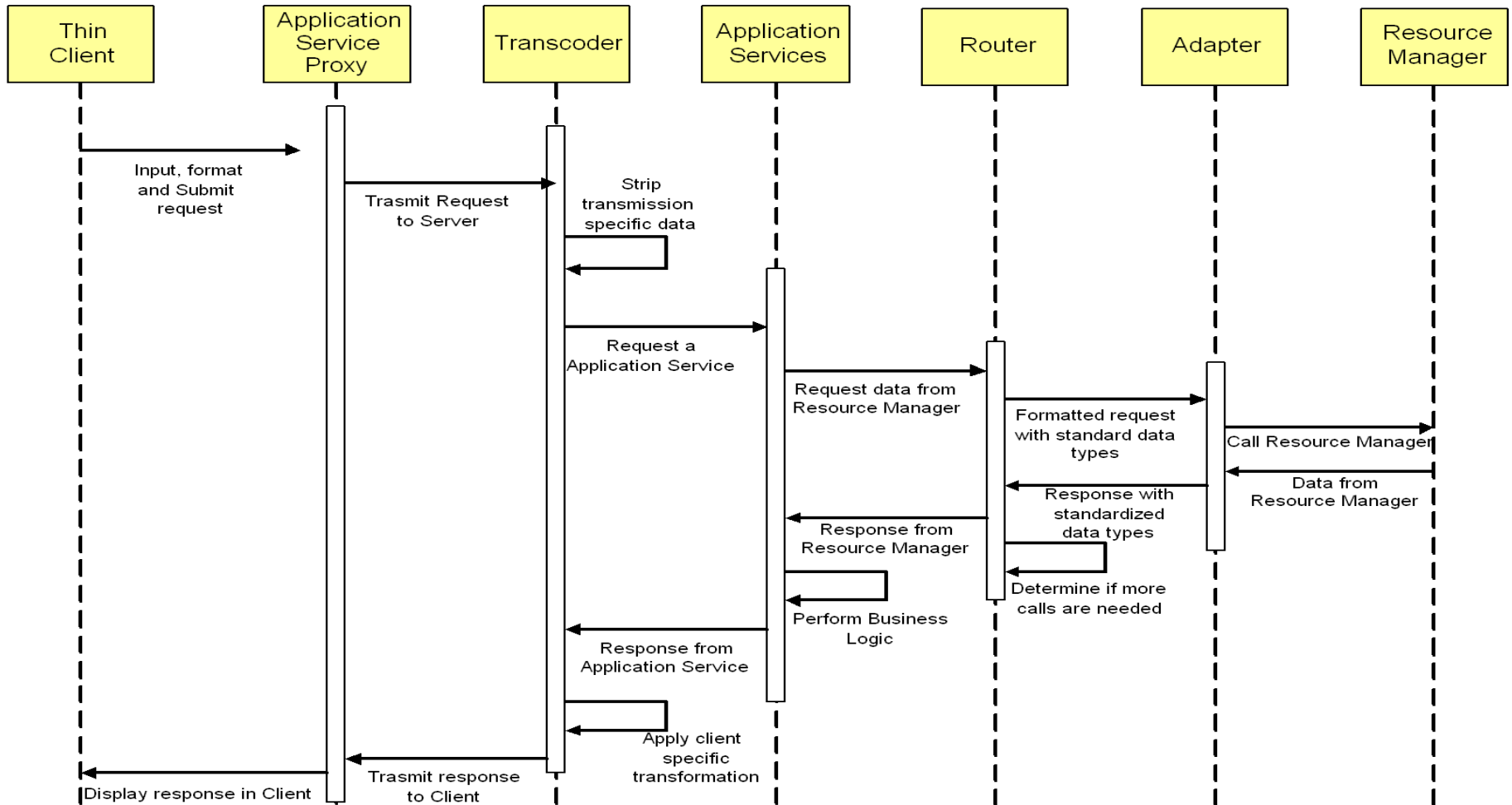
■ Description :

- This task will help identify important information about components for the proposed system by creating a component model.
- The component model work product describes the structure of an IT System in terms of its software components with their responsibilities, interfaces, (static) relationships, and the way they collaborate to deliver the required functionality.
- Components depict the major subsystems and boundaries (interfaces) of the overall system.
- Components are described by responsibilities, required service levels, performance, capacity and availability.

Example : Component Model



Example : Component Model



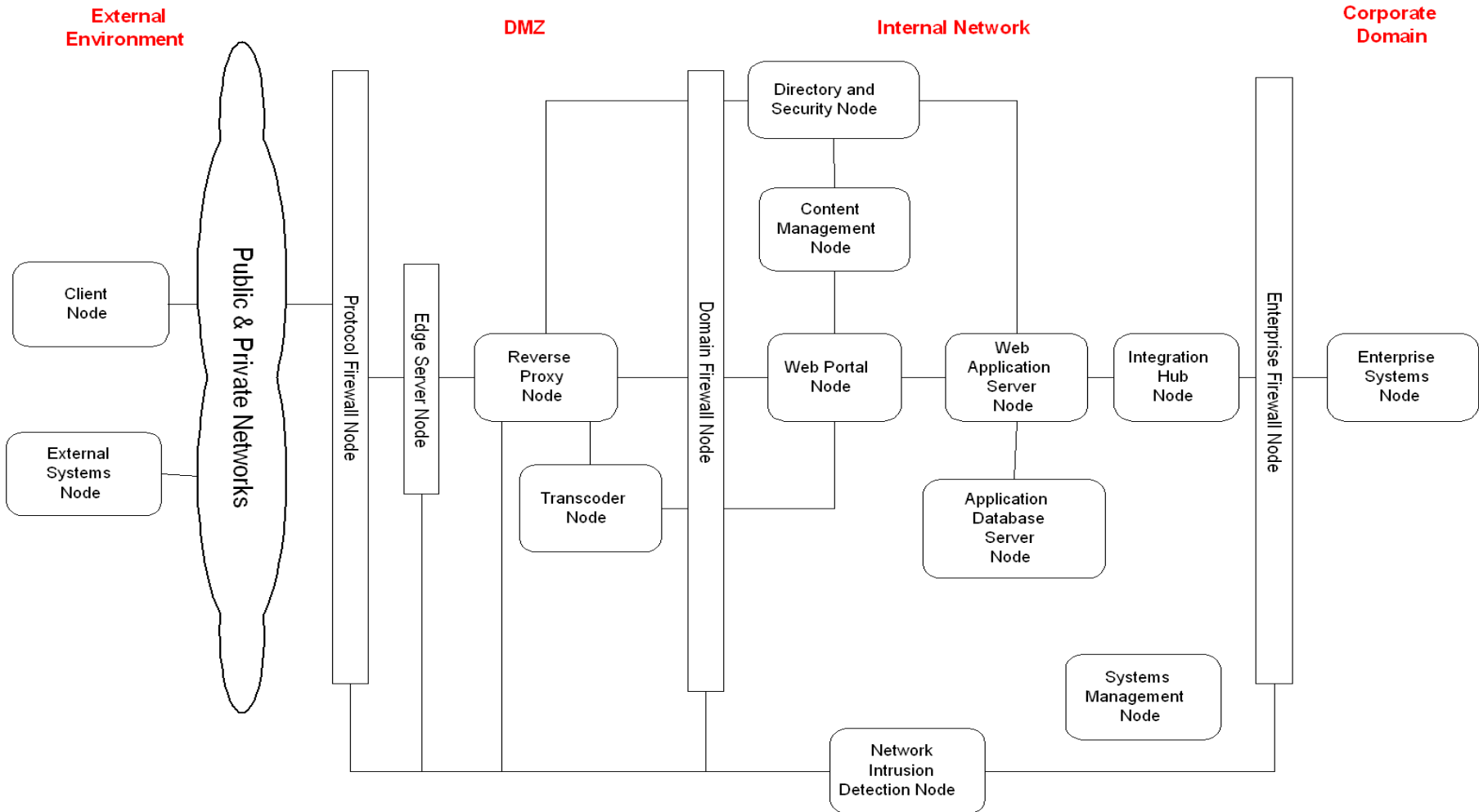
■ Purpose :

- Develop a preliminary view of the system for use in understanding customer requirements and general architectural approaches.
- Gain understanding of major elements of the IT system to be built such as primary system nodes, connections, locations, major components (including technical infrastructure) and existing assets to be reused.
- Illustrate major elements of IT system to be built such as primary system nodes, connections.
- Provide a mechanism for early discussion regarding functional characteristics of the system by enabling basic design walkthroughs.

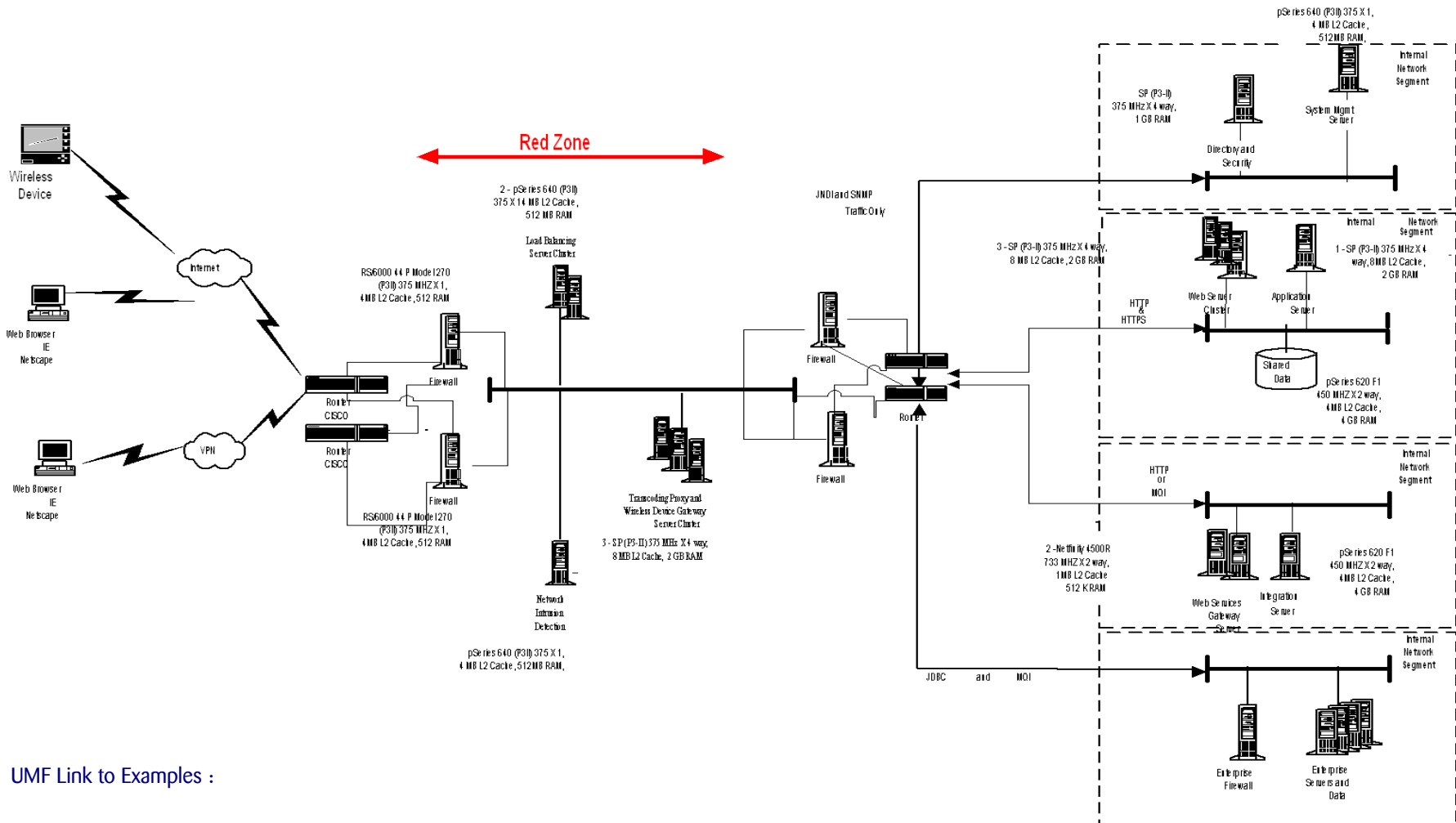
■ Description :

- Develop a System Topology diagram (one or more) that shows the topology and geographic distribution of the system,
- Develop a set of node descriptions which include applicable nonfunctional requirements, and an inventory of components (grouped as a deployment unit) that will be deployed on this node.
- The Operational Model may include a system topology diagram, a set of node descriptions, an inventory of components (grouped as a deployment unit) and a description of connections arranged as a table or matrix.

Example : Operational Model

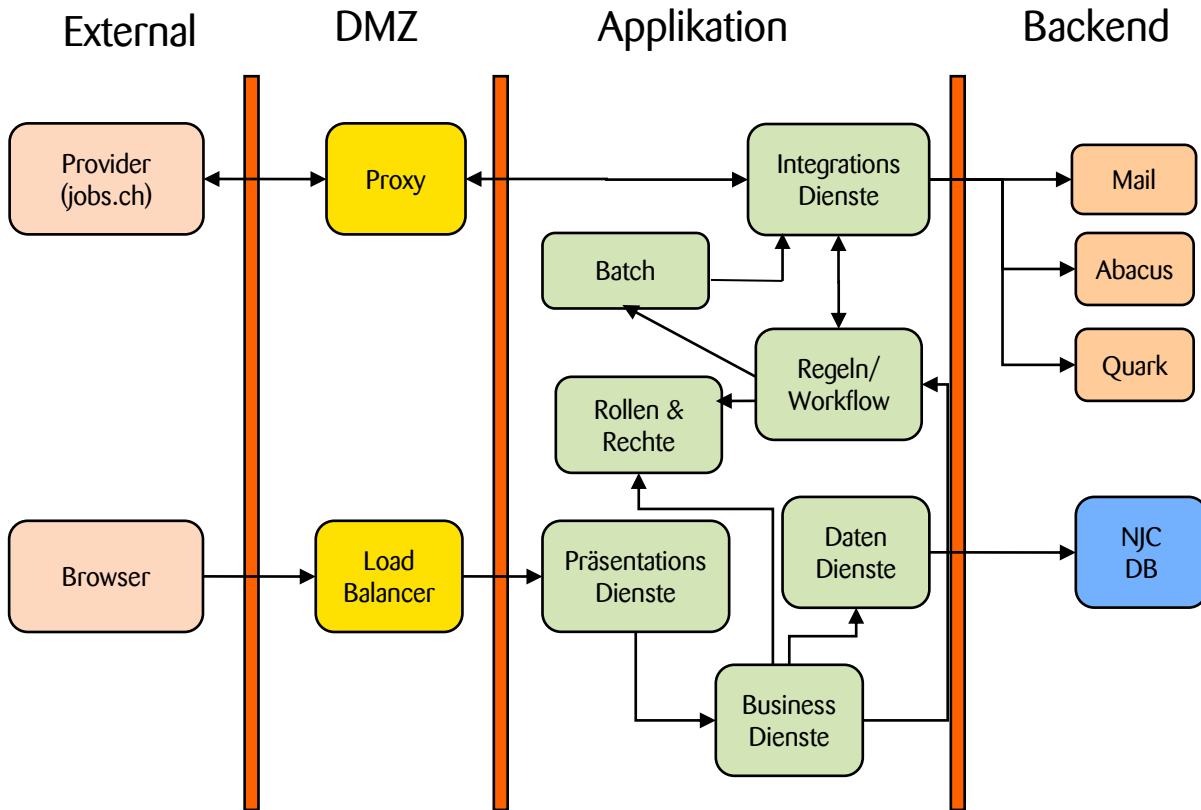


Example : Physical Operational Model



UMF Link to Examples :

Example : Operational Model



■ Purpose :

- Develop estimates for hardware, software and technical and non-technical effort for the purpose of developing a solution proposal for the client.
- Provide a historical record of the estimates and approaches with traceability and accountability on a project
- Provide a basis for comparisons on future estimates and projects
- Provide key inputs to improving overall estimating processes

■ Description :

- Evaluate type of estimates required and select estimating approach
- Estimate project schedule and resource costs
- Estimate hardware and software costs
- Information relevant to the estimate, including a description of the project scope and solution
- A description of each approach used to estimate the project including estimation tools used,
- The resulting size, effort, schedule, and resources required to perform the technical work
- The estimated costs and schedule to deliver this project

The critical path of a solution design and its work products

