What is IT Systems Architecture ?

Introduction and Overview

Dr. Marcel Schlatter
IBM Distinguished Engineer
Member of the IBM Academy of Technology
marcel.schlatter@ch.ibm.com

Architecture is a term that lots of people try to define

There is not just one way to state a system's architecture

Common elements of most attempts to define architecture, in the context of IT and other systems:

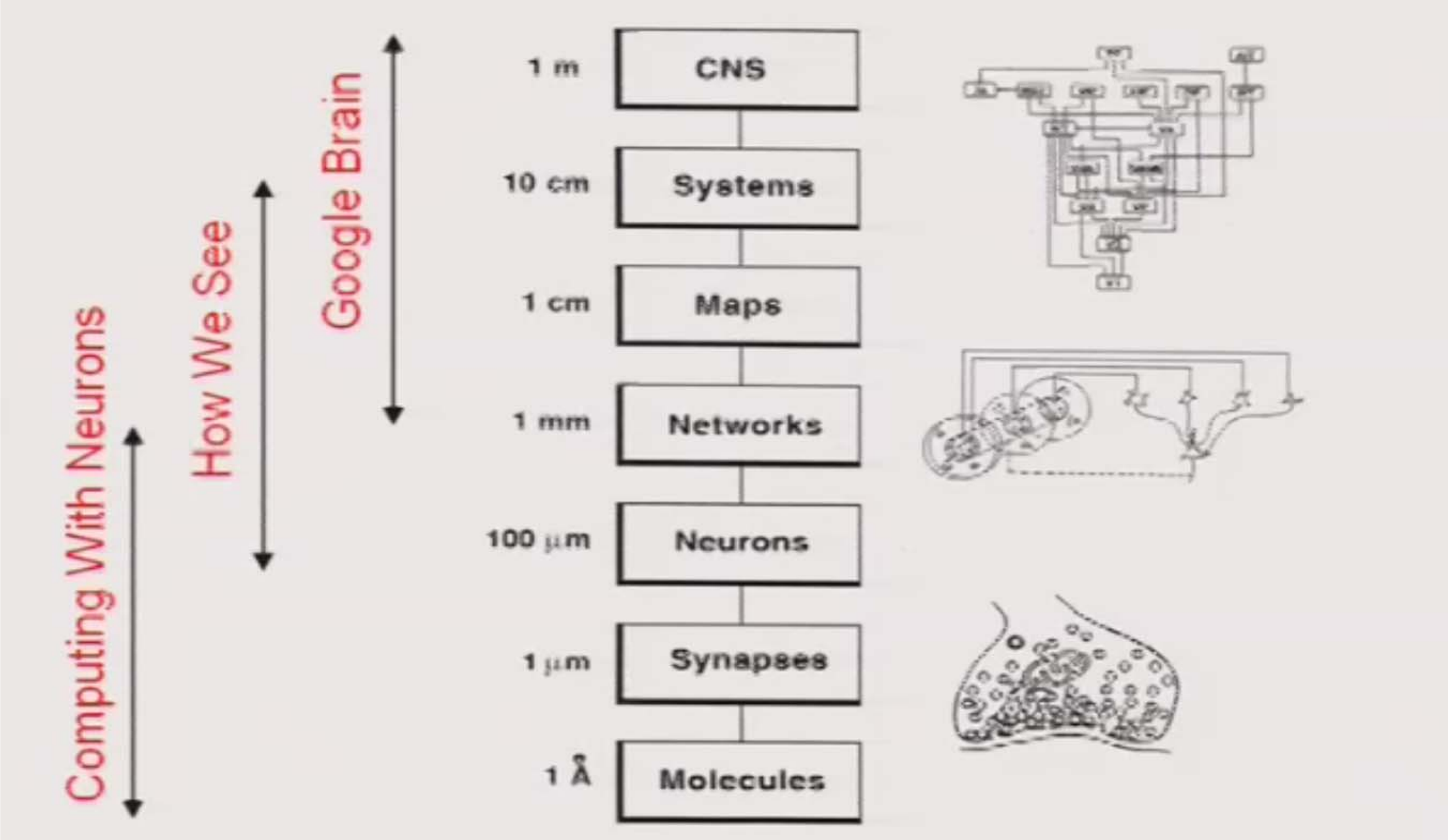
- Breakdown of a system into its parts
- The relationship between the parts (static and dynamic)
- Decisions about the design of a system that are hard to change

Architectures can be implied, apparent, or explicitly planned

- Implied architecture
  - of abstract things such as music or mathematics
- Apparent architecture
  - of natural things, such as geological formations or the structure of biological cells
- Explicitly planned architecture
  - of human-made things such as software, computers, enterprises, and databases, in addition to buildings.

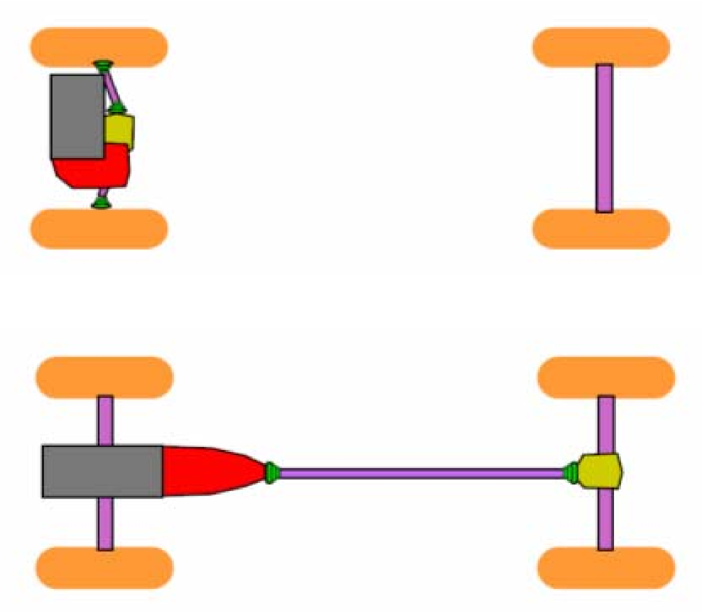
- Etymology:
  - Latin: architectus
  - Greek: arkhitekton (αρχιτεκτων) = master builder
    - From arkhi (αρχι) = chief + tekton (τεκτων) = builder, carpenter
    - archon: one of the nine chief magistrates of ancient Athens, 1659, from Gk. arkhon "ruler"

Apparent (or not so apparent) architecture of natural things
(Pauli Lectures 2008, ETH, Prof. Terrence J. Sejnowski)



In every usage, an architecture, whether implied, apparent or explicitly planned, may be seen as:

- A *subjective* mapping from one of many possible human perspectives
  - to the elements or components of some kind of structure or system,
  - which preserves the relationships among the elements or components.



IT Architecture: Definitions and Characteristics

- ANSI/IEEE Std 1471-2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems
  - The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution
- Rechtin, The Art of Systems Architecting
  - The structure (in terms of components, connections, and constraints) of a product, process, or element.
  - Architecture is what architects produce: The set of information that defines a system's **value**, **cost**, and **risk** for the purposes of the systems sponsor.
- Must address
  - **Function** and **quality**, including **aesthetics** for the user (client / customer)
  - **Feasibility** and **cost** for the builder

Three main dimensions of IT Architecture

- **Models**
  - Representations of the architecture that are meaningful to one or more stakeholders (architectural views)
- **Frameworks** **(Meta-Level)**
  - Establish terms and concepts for **architectural thinking**
  - Processes, activities, and guidelines to help solve IT architecture problems
  - Key techniques (methods and tools) used by IT architects
  - Specifications of how to describe architectures
  - Reference Architectures
  - Patterns
- **Skills**
  - Roles and responsibilities of the IT architect
  - Different types of architects

Three main dimensions of IT Architecture

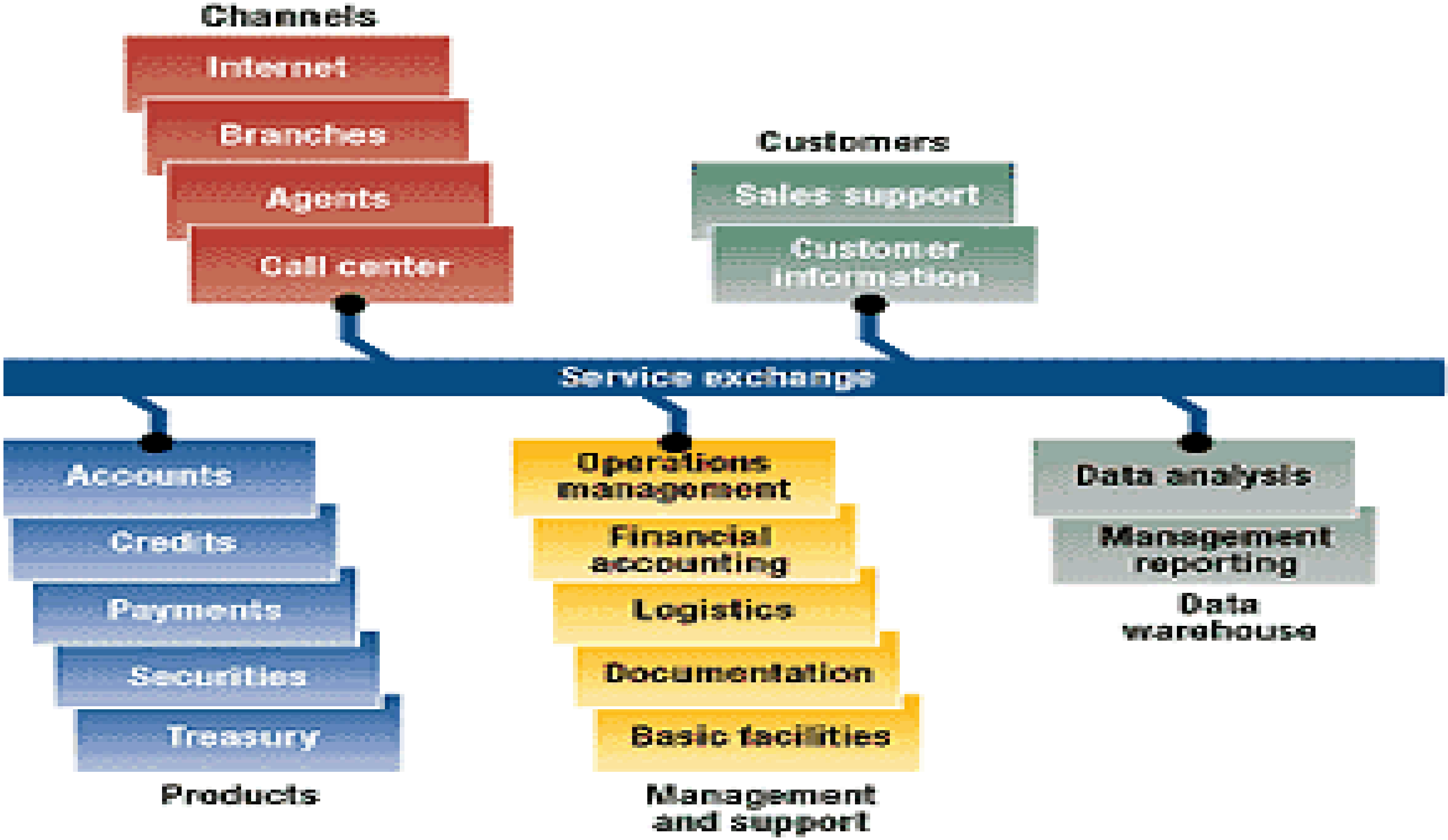
- **Models**
  - Representations of the architecture that are meaningful to one or more stakeholders (architectural views)
- **Frameworks (Meta-Level)**
  - Establish terms and concepts for architectural thinking
  - Processes, activities, and guidelines to help solve IT architecture problems
  - Key techniques (methods and tools) used by IT architects
  - Specifications of how to describe architectures
  - Reference Architectures
  - Patterns
- **Skills**
  - Roles and responsibilities of the IT architect
  - Different types of architects

IT Architecture Models: A set of Architectural Views

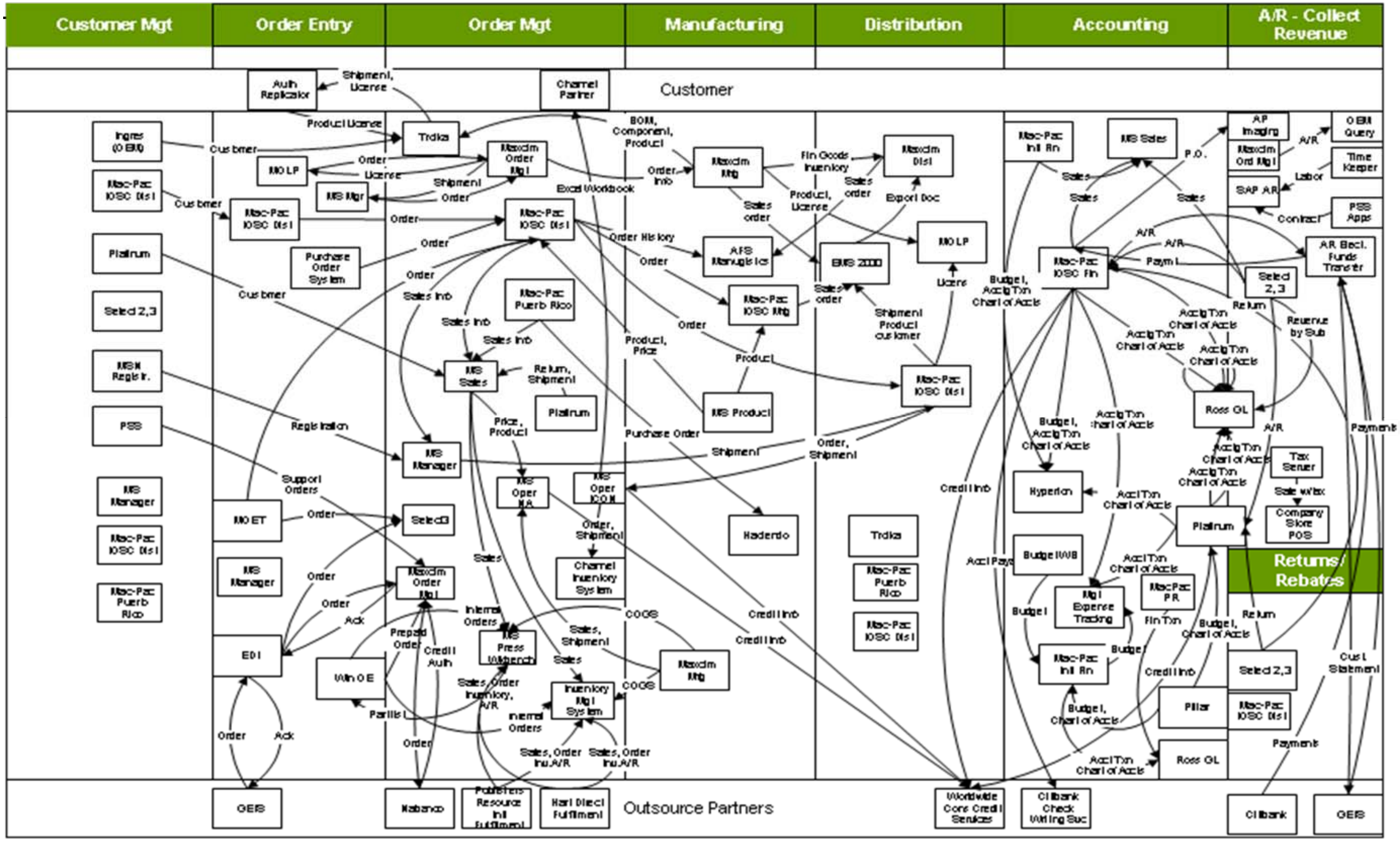
- Enable the architecture to be communicated to, and understood by, all the stakeholders
- Enable stakeholders to verify that the system will address their concerns
- Examples

Scope description:	Planner’s view
Model of the business:	Owner’s view
Information system model:	Designer's view
Technology model:	Builder's view
Detailed blueprints:	Subcontractor’s view

Example:
The Scope View of a Bank's IT Architecture



Designer's View

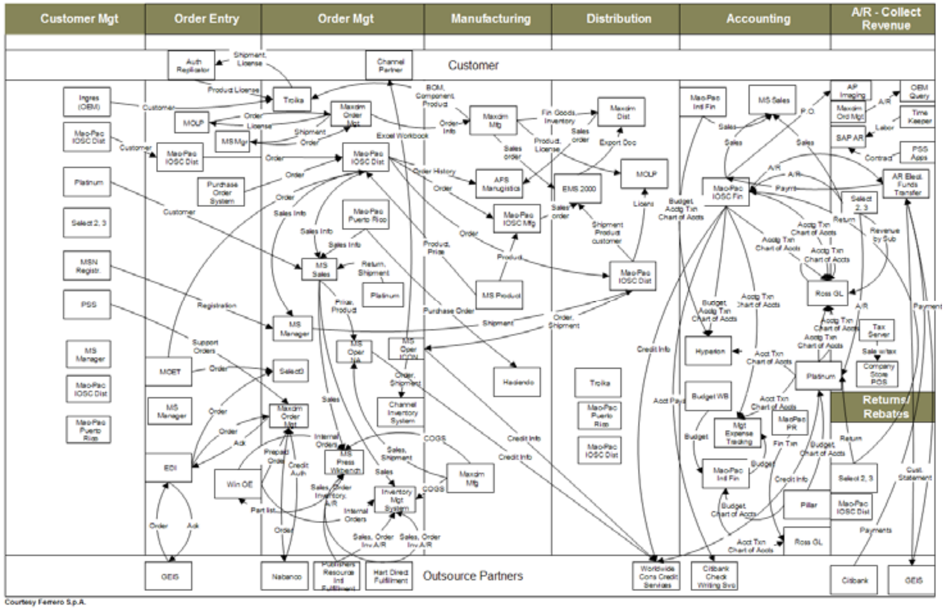


Courtesy Ferrero S.p.A.

Why is IT Architecture important?
Business flexibility depends on IT flexibility

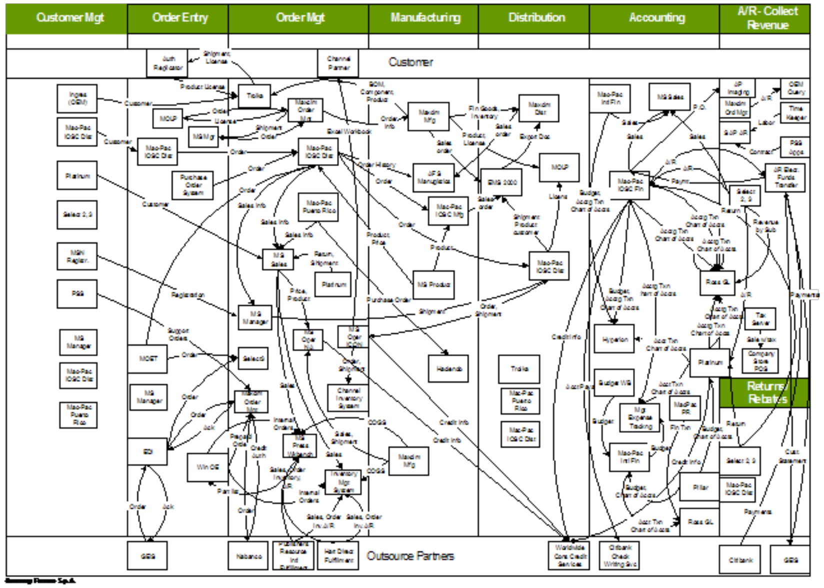
“Today’s IT architectures, arcane as they may be, are the biggest roadblocks most companies face when making strategic moves.”

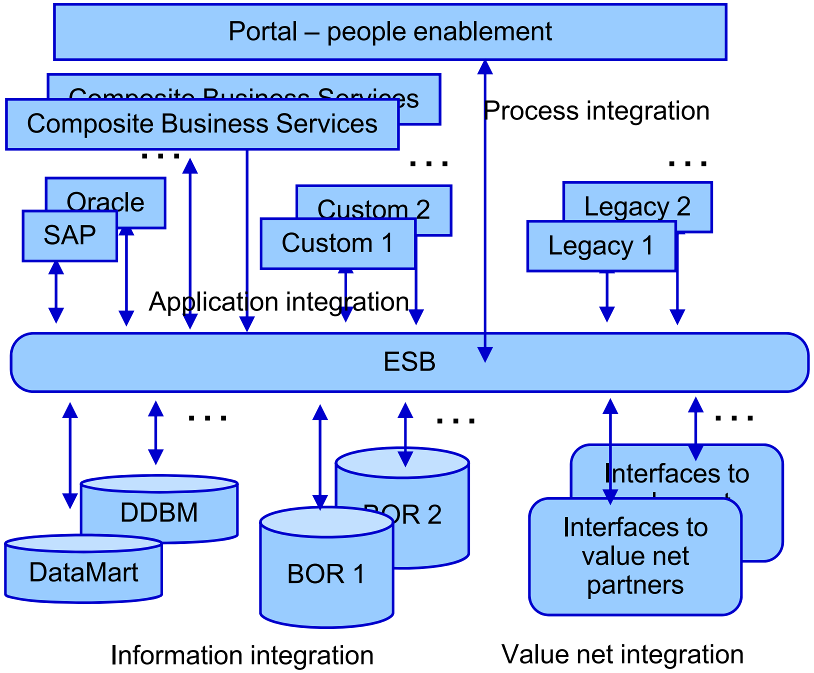
McKinsey
“Flexible IT, Better Strategy”



Courtesy Ferrero S.p.A.

From traditional to SOA-based integration





Traditional Integration

- Point to point or adaptor based interfaces
- Expensive to develop & maintain
- Complex, inflexible and expensive to change
- Duplicate development efforts across different software versions
- Not conducive to asset reuse
- Difficult to enforce enterprise governance standards

SOA Integration

- Service Interface
- Reduced Complexity and Cost
- Flexible and easy to change
- Reuse across different software versions
- Conducive to asset reuse
- Support enterprise governance standards

Three main dimensions of IT Architecture

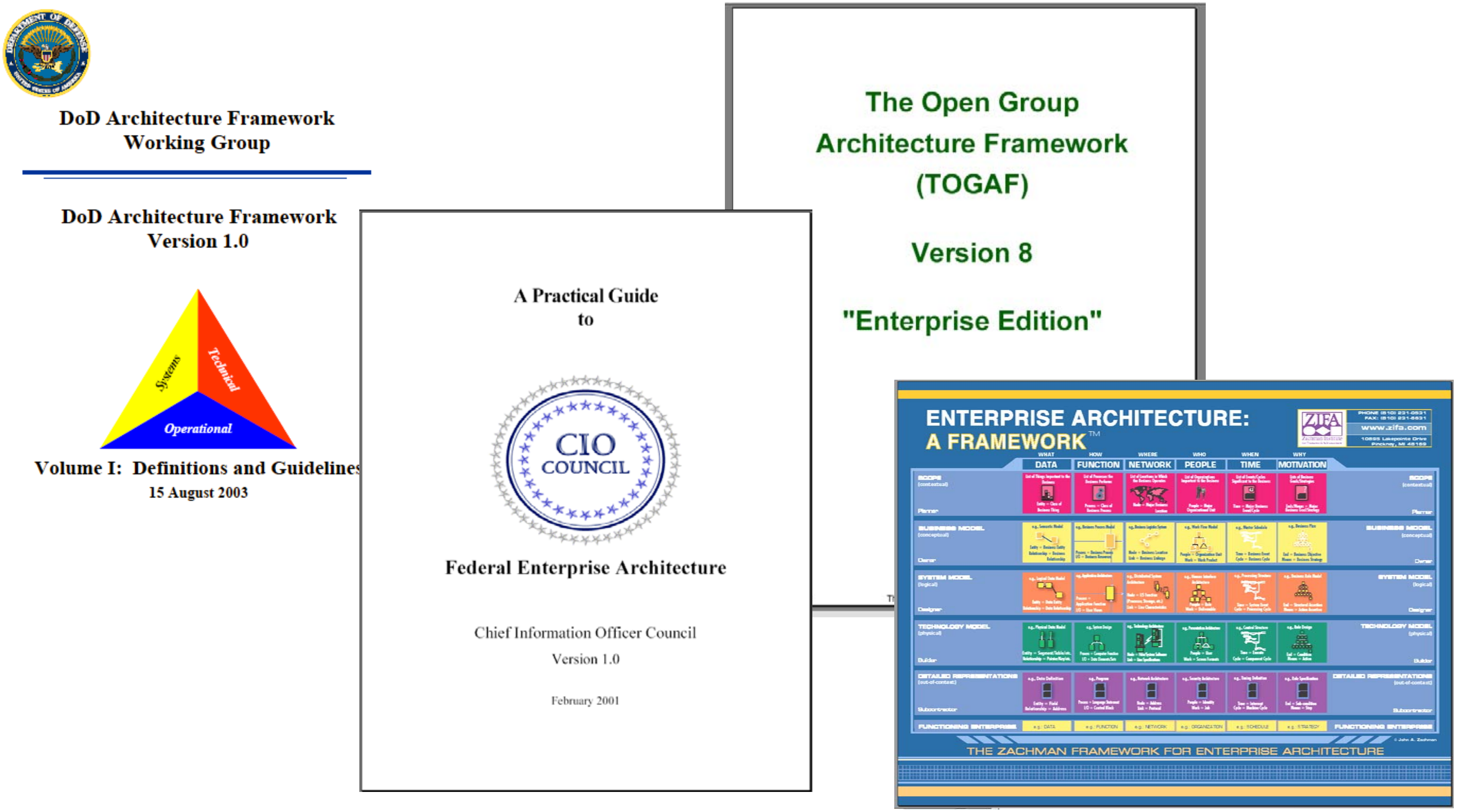
- Models
  - Representations of the architecture that are meaningful to one or more stakeholders (architectural views)
- **Frameworks (Meta-Level)**
  - Establish terms and concepts for **architectural thinking**
  - Processes, activities, and guidelines to help solve IT architecture problems
  - Key techniques (methods and tools) used by IT architects
  - Specifications of how to describe architectures
  - Reference Architectures
  - Patterns
- Skills
  - Roles and responsibilities of the IT architect
  - Different types of architects

What is an Architecture Framework?

- An **architecture framework** is a toolkit which can be used for developing a broad range of different architectures.
- It should describe a method for designing an information system in terms of a set of building blocks, and for showing how the building blocks fit together.
- It should contain a set of tools and provide a common vocabulary.
- It should also include a list of recommended standards and compliant products that can be used to implement the building blocks.

TOGAF™

Examples of Architectural Frameworks



DoD Architecture Framework
Working Group

DoD Architecture Framework
Version 1.0

Volume I: Definitions and Guidelines
15 August 2003

A Practical Guide
to

Federal Enterprise Architecture

Chief Information Officer Council
Version 1.0
February 2001

The Open Group
Architecture Framework
(TOGAF)

Version 8

"Enterprise Edition"

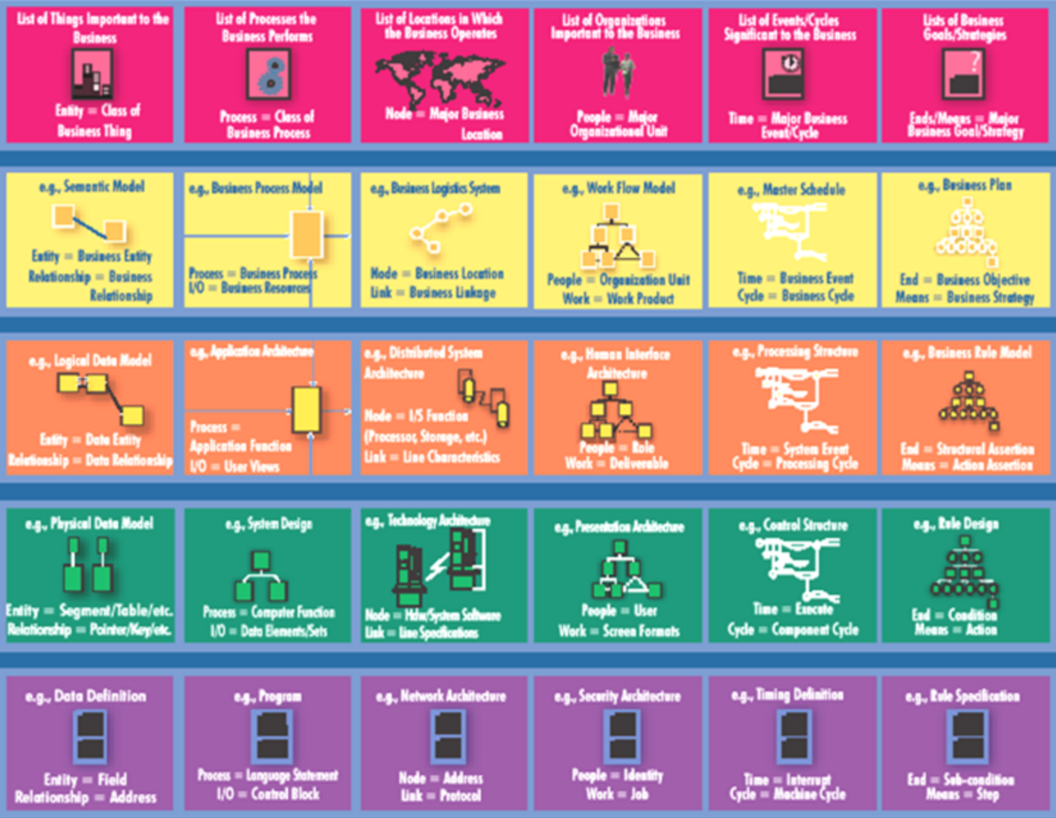
ENTERPRISE ARCHITECTURE:
A FRAMEWORK™

THE ZACHMAN FRAMEWORK FOR ENTERPRISE ARCHITECTURE

ENTERPRISE ARCHITECTURE:
A FRAMEWORK™

ZIFA
Zachman Institute
for Framework Advancement

PHONE (810) 231-0531
FAX: (810) 231-6631
www.zifa.com
10895 Lakepointe Drive
Pinckney, MI 48169

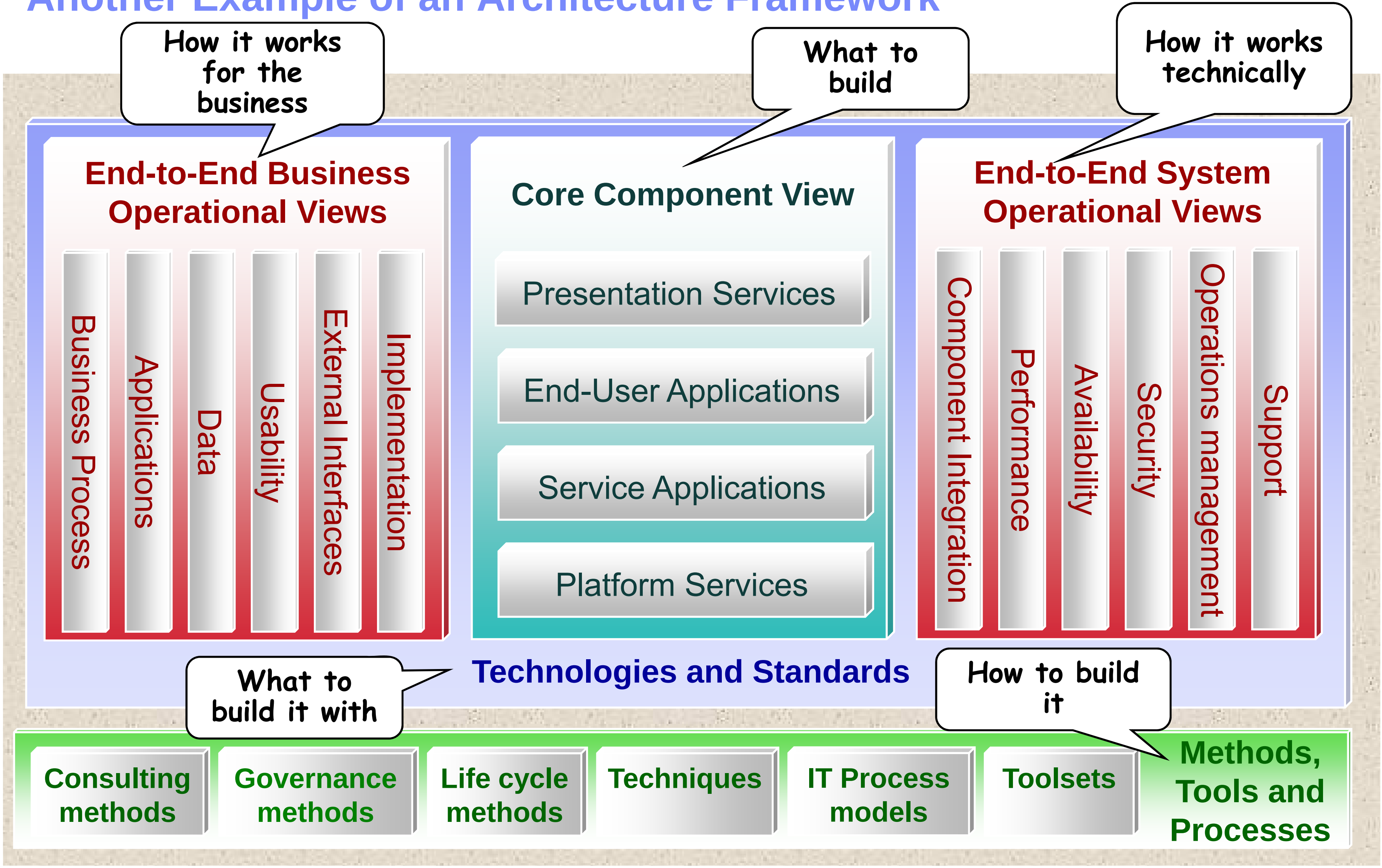


	WHAT	HOW	WHERE	WHO	WHEN	WHY	
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	
SCOPE (contextual) Planner	List of Things Important to the Business Entity = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in Which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizational Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	Lists of Business Goals/Strategies Ends/Means = Major Business Goal/Strategy	SCOPE (contextual) Planner
BUSINESS MODEL (conceptual) Owner	e.g., Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g., Business Process Model Process = Business Process I/O = Business Resources	e.g., Business Logistics System Node = Business Location Link = Business Linkage	e.g., Work Flow Model People = Organization Unit Work = Work Product	e.g., Master Schedule Time = Business Event Cycle = Business Cycle	e.g., Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (conceptual) Owner
SYSTEM MODEL (logical) Designer	e.g., Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture Process = Application Function I/O = User Views	e.g., Distributed System Architecture Node = I/S Function (Processor, Storage, etc.) Link = Line Characteristics	e.g., Human Interface Architecture People = Role Work = Deliverable	e.g., Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion	SYSTEM MODEL (logical) Designer
TECHNOLOGY MODEL (physical) Builder	e.g., Physical Data Model Entity = Segment/Table/etc. Relationship = Pointer/Key/etc.	e.g., System Design Process = Computer Function I/O = Data Elements/Sets	e.g., Technology Architecture Node = Hdw/System Software Link = Line Specifications	e.g., Presentation Architecture People = User Work = Screen Formats	e.g., Control Structure Time = Execute Cycle = Component Cycle	e.g., Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (physical) Builder
DETAILED REPRESENTATIONS (out-of-context) Subcontractor	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language Statement I/O = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Security Architecture People = Identity Work = Job	e.g., Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification End = Sub-condition Means = Step	DETAILED REPRESENTATIONS (out-of-context) Subcontractor
FUNCTIONING ENTERPRISE	e.g.: DATA	e.g.: FUNCTION	e.g.: NETWORK	e.g.: ORGANIZATION	e.g.: SCHEDULE	e.g.: STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman

THE ZACHMAN FRAMEWORK FOR ENTERPRISE ARCHITECTURE

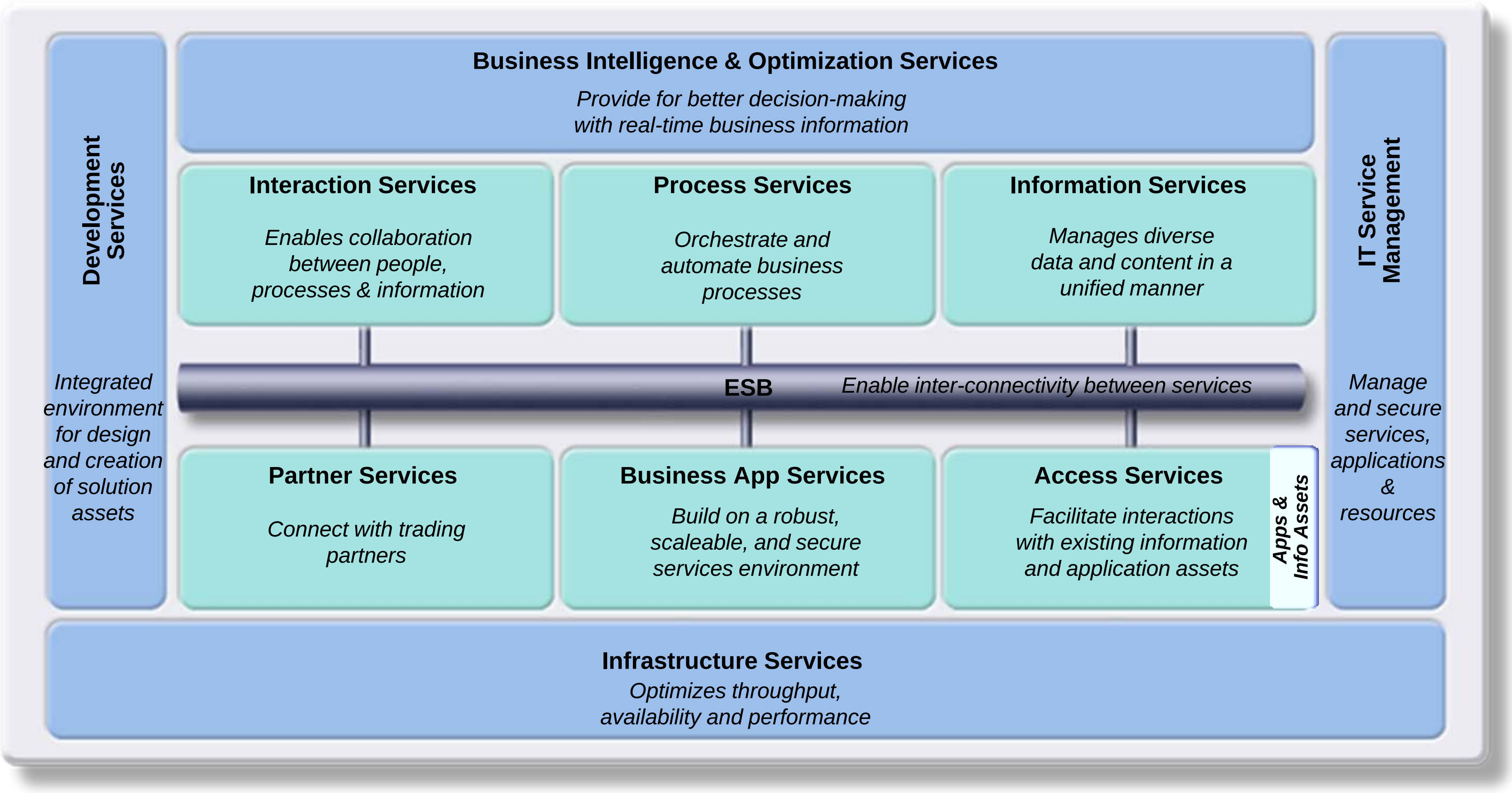
Another Example of an Architecture Framework



Frameworks help establish terms and concepts for architectural thinking

- Guide development of a Solution's IT Architecture
  - What subjects should an architect / an architecture cover
- Guide development of a broad range of different architectures, and solutions
  - **How to structure a system**
  - How to describe an architecture
  - Architectural Description Standard(s)

SOA Reference Architecture
State-of-the art framework for structuring IT systems



Development Services

Integrated environment for design and creation of solution assets

Business Intelligence & Optimization Services

Provide for better decision-making with real-time business information

Interaction Services

Enables collaboration between people, processes & information

Process Services

Orchestrate and automate business processes

Information Services

Manages diverse data and content in a unified manner

ESB *Enable inter-connectivity between services*

Partner Services

Connect with trading partners

Business App Services

Build on a robust, scaleable, and secure services environment

Access Services

Facilitate interactions with existing information and application assets

Apps & Info Assets

IT Service Management

Manage and secure services, applications & resources

Infrastructure Services

Optimizes throughput, availability and performance

Purpose and Role of Frameworks

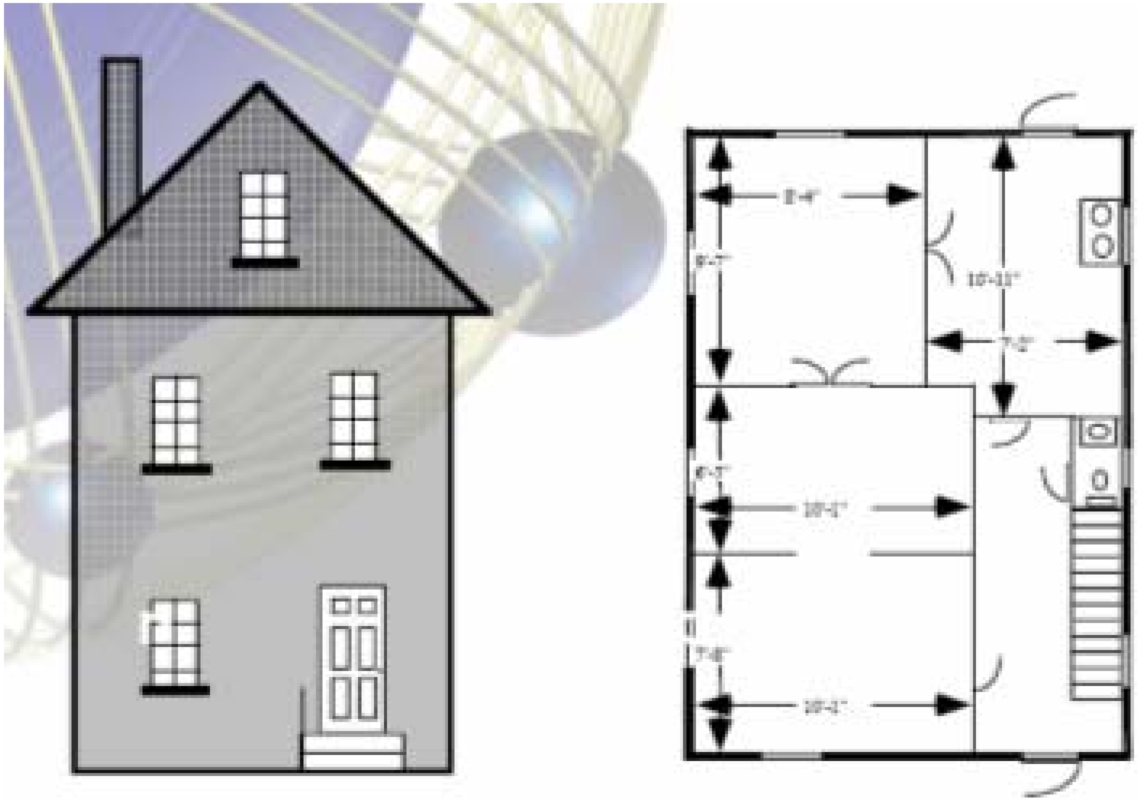
- To establish terms and concepts for **architectural thinking**
  - What subjects should an architect / an architecture cover

- To guide the development of a broad range of different architectures, and solutions
  - How to structure a system
  - **How to describe an architecture**
  - **Architectural Description Standard(s)**

Architectural Description

- A collection of **Workproducts** to document an architecture
- Addressed to one or more **Stakeholders** to answer their **Concerns** about the system
- Organized into one or more **Views** of the system
- Each **View** addresses one or more **Concerns** of the **Stakeholders**
- A **View** is a way of looking at an architecture
- A **View** is what you see when you look at the architecture from a particular **Viewpoint**

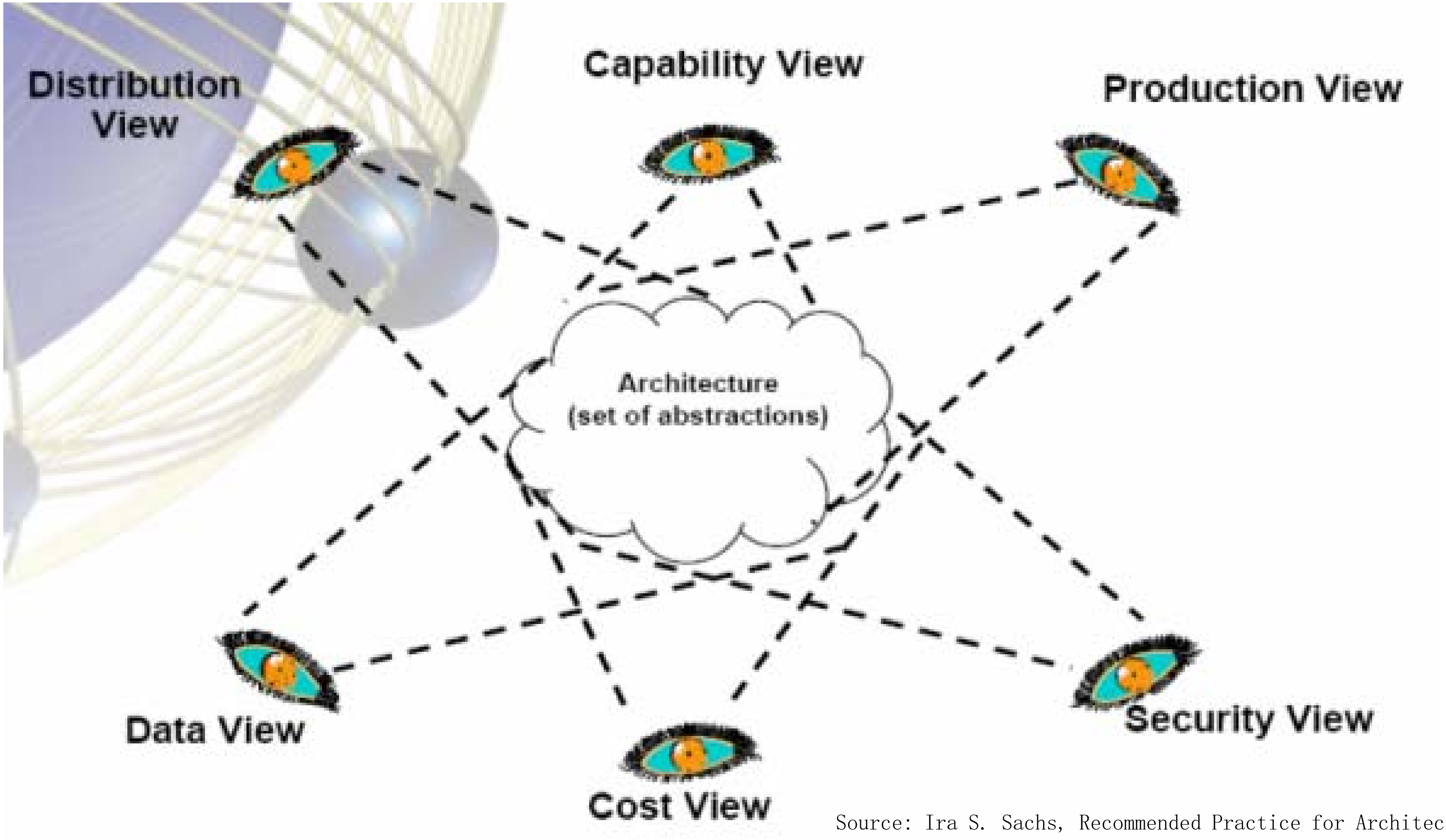
Multiple views and models



Bill of Materials	
2x4x8	250
2x6x8	150
4x4x10	10
Siding	1500 sq ft
Shingles	500 sq ft
8d nails	20 lb
6d nails	10 lb

Source: Ira S. Sachs, Recommended Practice for Architectural Description, IEEE Standard P1471

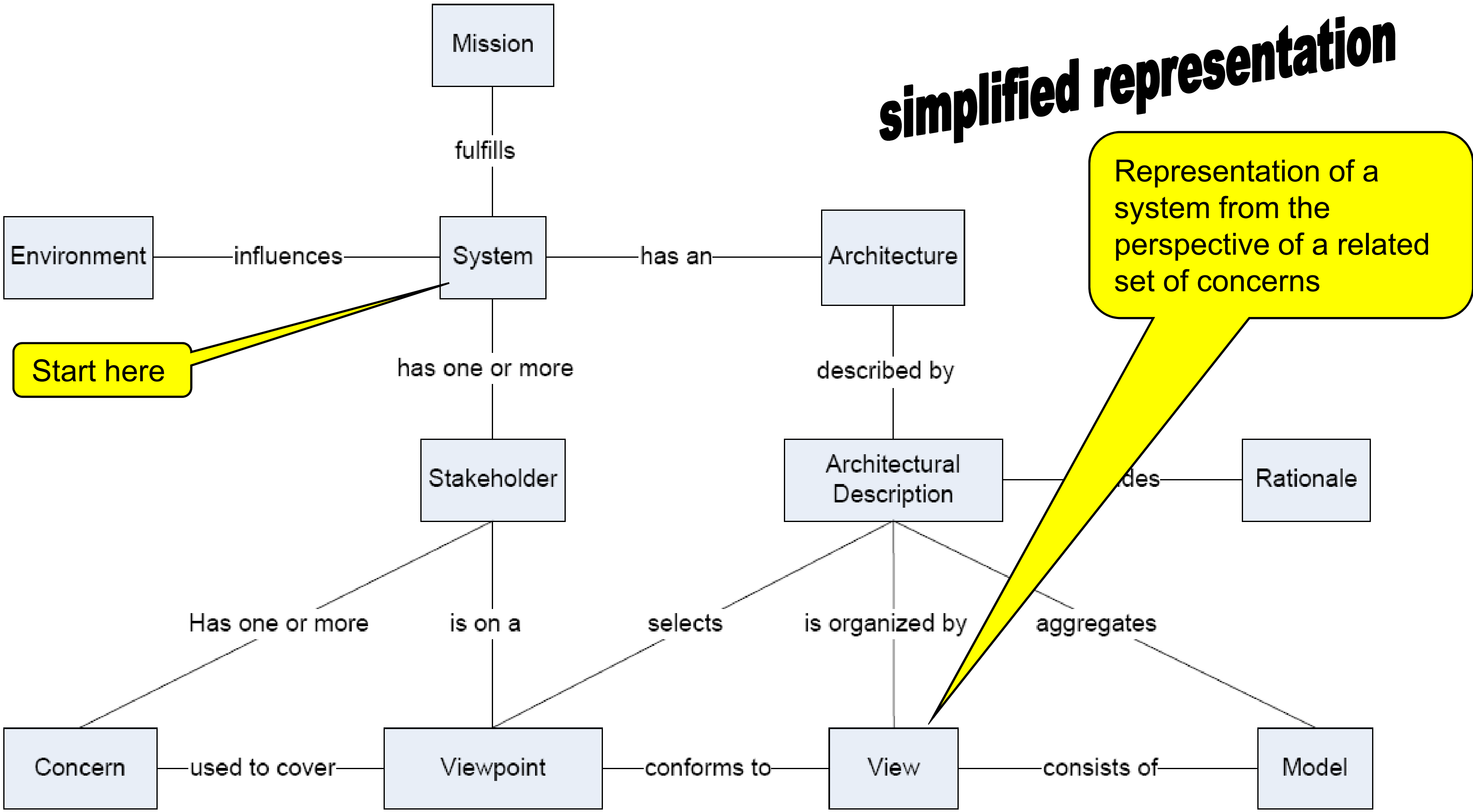
Sample Views



Source: Ira S. Sachs, Recommended Practice for Architectural Description, IEEE Standard P1471

Conceptual Framework of the IEEE Standard P1471

Recommended Practice for Architectural Description of Software-Intensive Systems



Mission

fulfills

Environment

influences

System

has an

Architecture

has one or more

described by

Stakeholder

Architectural Description

des

Rationale

Has one or more

is on a

selects

is organized by

aggregates

Concern

used to cover

Viewpoint

conforms to

View

consists of

Model

simplified representation

Start here

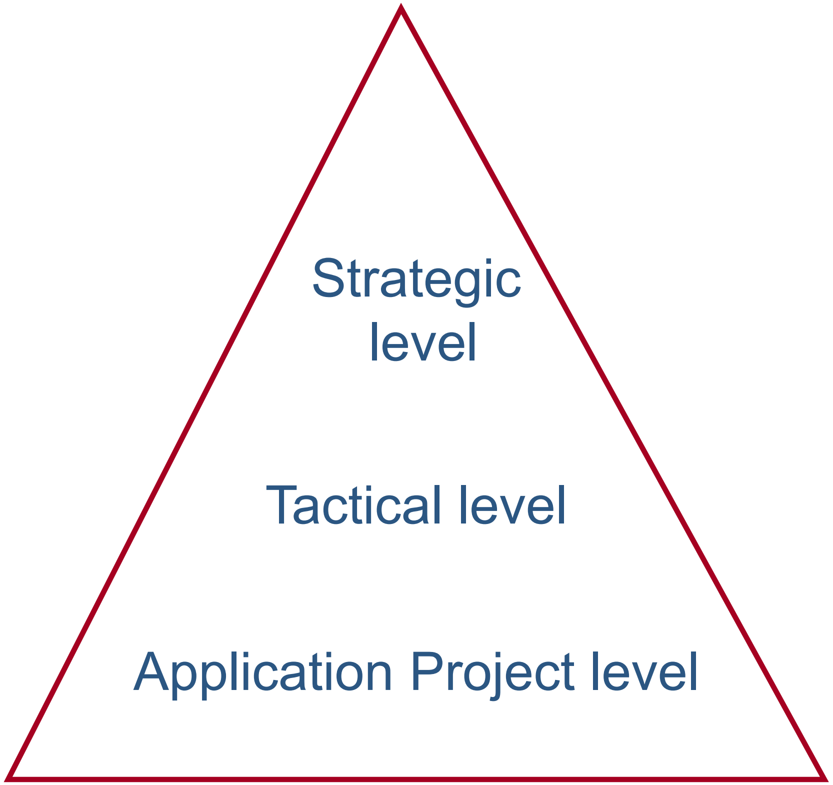
Representation of a system from the perspective of a related set of concerns

Three main dimensions of IT Architecture

- Models
  - Representations of the architecture that are meaningful to one or more stakeholders (architectural views)
- Frameworks (Meta-Level)
  - Processes, activities, and guidelines to help solve IT architecture problems
  - Key techniques (methods and tools) used by IT architects
  - Specifications of how to describe architectures
  - Reference Architectures
  - Patterns
- Skills
  - Roles and responsibilities of the IT architect
  - Different types of architects

Key skills of successful IT Architects

- Dealing with complexity
- Designing and maintaining system integrity
  - Agreeing policies/standards
  - Sticking to policies/standards
  - Evolving policies/standards
  - Making exceptions
- Managing risks
- Managing costs
  - Efficiency, re-use
  - Solution Optimization


Strategic level

Tactical level

Application Project level

Roles and responsibilities of the IT architect job
Key techniques (methods and tools) used by IT architects

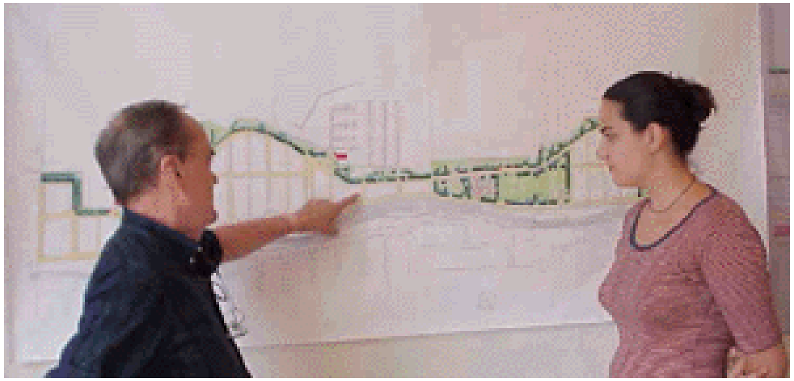
- Problem Solving
  - Abstract the problem, divide and conquer
- Communication
  - Understand client's needs, explain choices and justify solutions
- Team Work
  - Engage and work together with team members both from your company as well as other companies towards reaching the best solution
- Technical
  - Ensure the best technological setting is used in the solution
- Managerial
  - Ensure you get things done properly

Compare with the Construction Architect and/or Town Planner

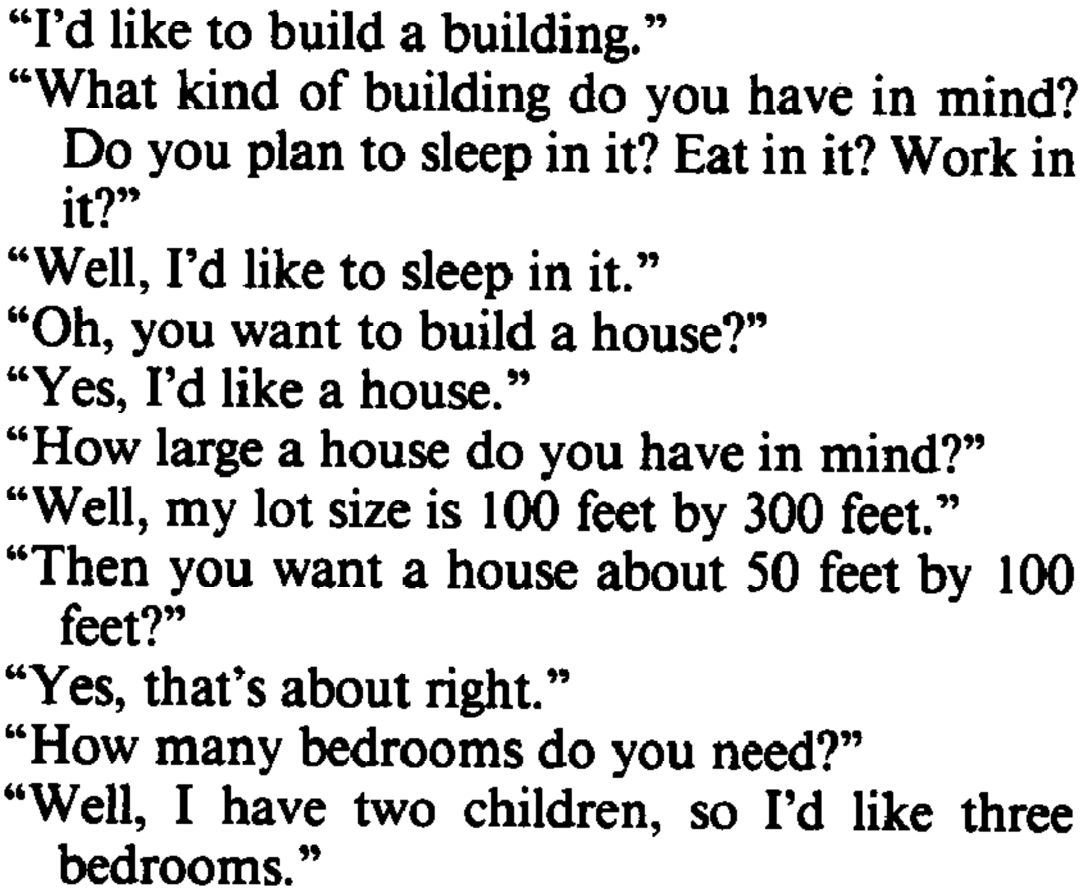
- Familiar with **technology** and **business**
  - The latest or most economical / reliable **materials and methods**
  - **Requirements, architectural styles, patterns and solutions**

- Mediator
  - **Translates a vision into a plan** that a builder can use to construct the building
  - **Balances conflicting tensions** between the different aspects and multiple levels of technical design, drawing on business, technical, project management and inter-personal concepts and skills.

- Dealing with people
  - **Listening** and negotiating
  - **Representing** the client's interests
  - **Supervising** the build



An example of a conversation between the architect and the prospective owner.



"I'd like to build a building."

"What kind of building do you have in mind? Do you plan to sleep in it? Eat in it? Work in it?"

"Well, I'd like to sleep in it."

"Oh, you want to build a house?"

"Yes, I'd like a house."

"How large a house do you have in mind?"

"Well, my lot size is 100 feet by 300 feet."

"Then you want a house about 50 feet by 100 feet?"

"Yes, that's about right."

"How many bedrooms do you need?"

"Well, I have two children, so I'd like three bedrooms."

Each question serves to pose a

constraint (the lot size)

or to identify a

requirement (number of bedrooms)

in order to establish the conditions within which any design will take place.

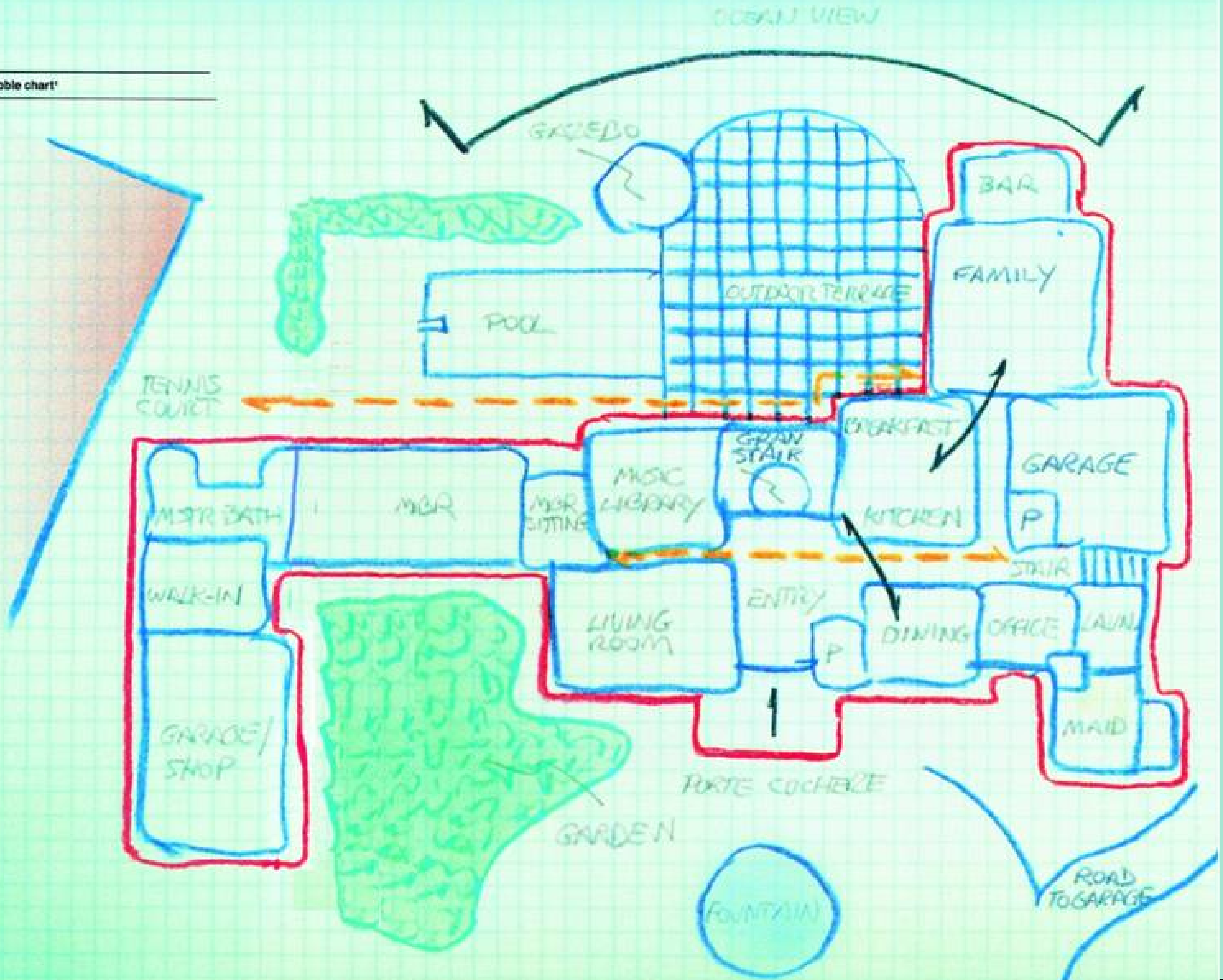


Figure 1 Architect's bubble chart[1]