What is IT 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

What is architecturally significant can change over a system's lifetime

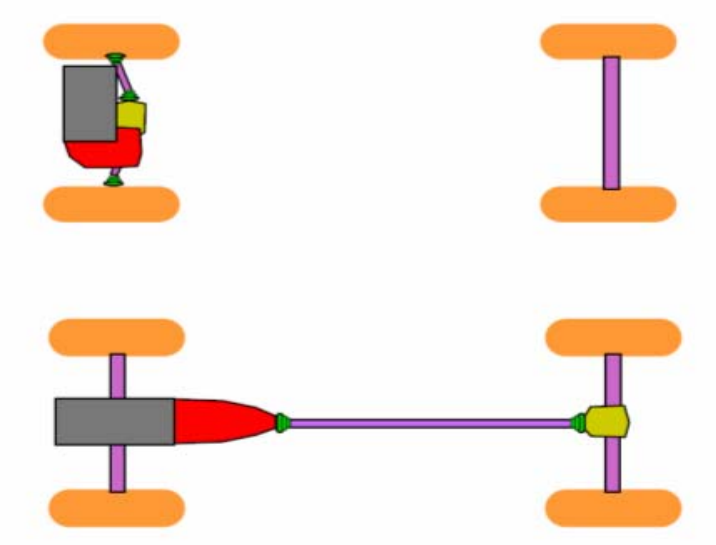
- Common elements of most attempts to define architecture, in the context of IT and other systems:
  - Breakdown of a system into its parts
  - The relationship between the parts (static and dynamic)
  - Decisions about the design of a system that are hard to change

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.

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

- A *subjective* *mapping* from a human perspective
  - to the elements or components of some kind of structure or system,
  - which preserves the relationships among the elements or components.



Abstraction

- One thing that represents several real things equally well (E.W. Dijkstra)
- The process of generalization by reducing the information content of a concept or an observable phenomenon
  - Typically in order to retain only information which is relevant for a particular purpose.

Use the Simplest Model, But Not Too Simple

Architecture – Key Definitions (Wikipedia)

- Art and science of designing buildings and structures.
  - Designing = sketching, outlining, planning
- A wider definition would include the design of the total built environment, from the macrolevel of town planning, urban design, and landscape architecture to the microlevel of creating furniture.
- Must address:
  - **Function** and **aesthetics** for the user
  - **Feasibility** and **cost** for the builder
- 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,"

IT Architecture – Key Definitions

- ANSI/IEEE Std 1471-2000:
  - 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
- Perry and Garlan:
  - The structure of the components of a system, their interrelationships, and principles and guidelines governing their design and evolution over time.
- 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.

In the context of this lecture series, architecture is the art and discipline of creating a plan of any complex object or system

The IT Architecture Discipline has three main dimensions:

- **Models**
  - Representations of the overall architecture that are meaningful to one or more stakeholders (architectural views)
- **Frameworks**
  - Processes, activities, guidelines to help solve IT architecture problems
  - Key techniques (methods and tools) used by IT architects
  - Specifications of how to describe architectures
  - List of recommended techniques/standards and compliant products that can be used to implement the building blocks.
- **Skills**
  - Roles and responsibilities of the IT architect job

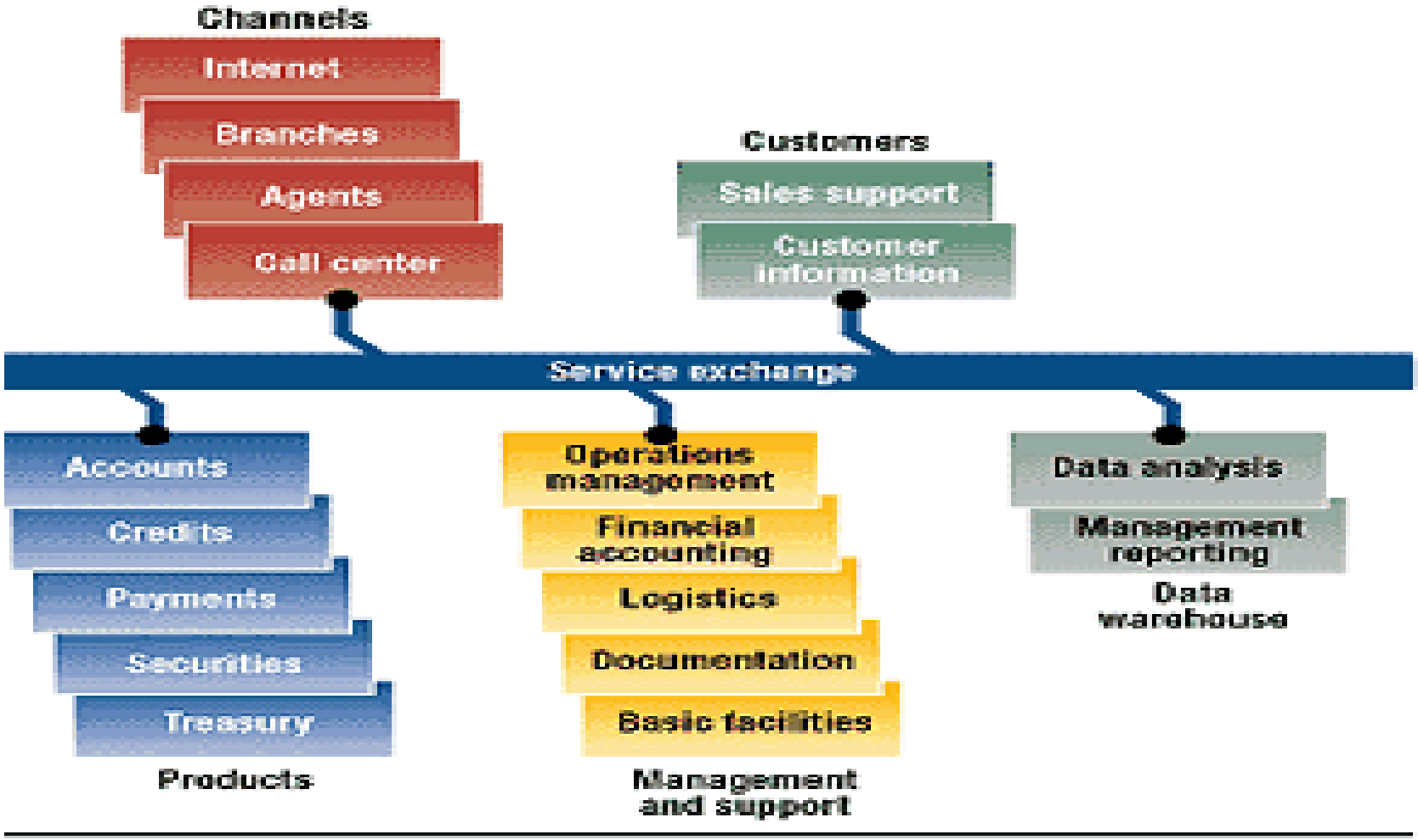
IT Architecture Models: Representing Architectural Views

- Representations of the overall architecture that are meaningful to one or more **stakeholders** in the system.

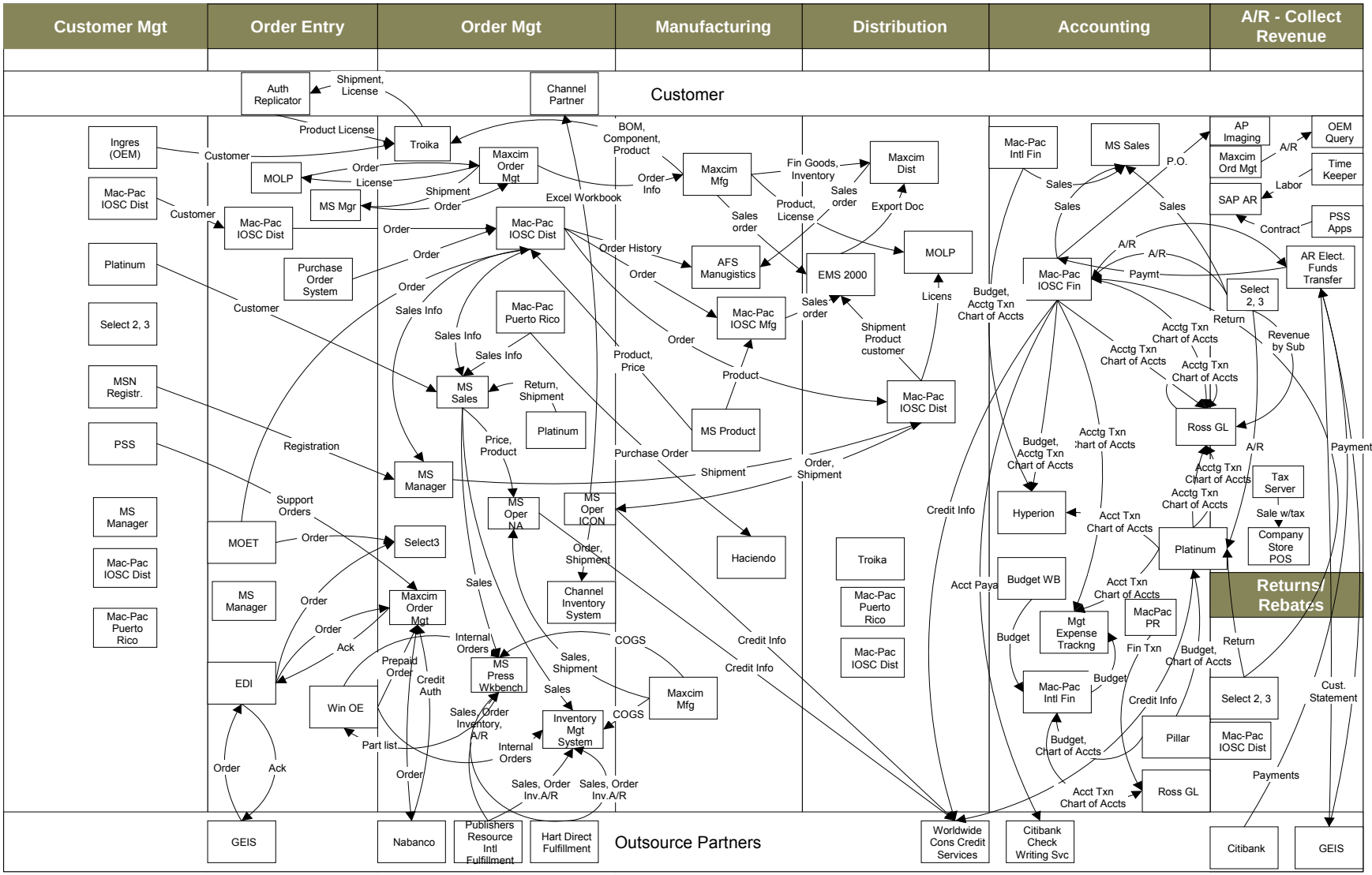
- The architect chooses and develops a set of **views** that will
  - Enable the architecture to be **communicated** to, and **understood** by, all the stakeholders
  - Enable them to **verify** that the system will address their concerns (→ "Testing")

- Examples:
  - Scope description: Bird's Eye view
  - Model of the business: Customer's view
  - Model of the information system: Designer's view
  - Technology model: Builder's view
  - Detailed blueprints: e.g. Program Code

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



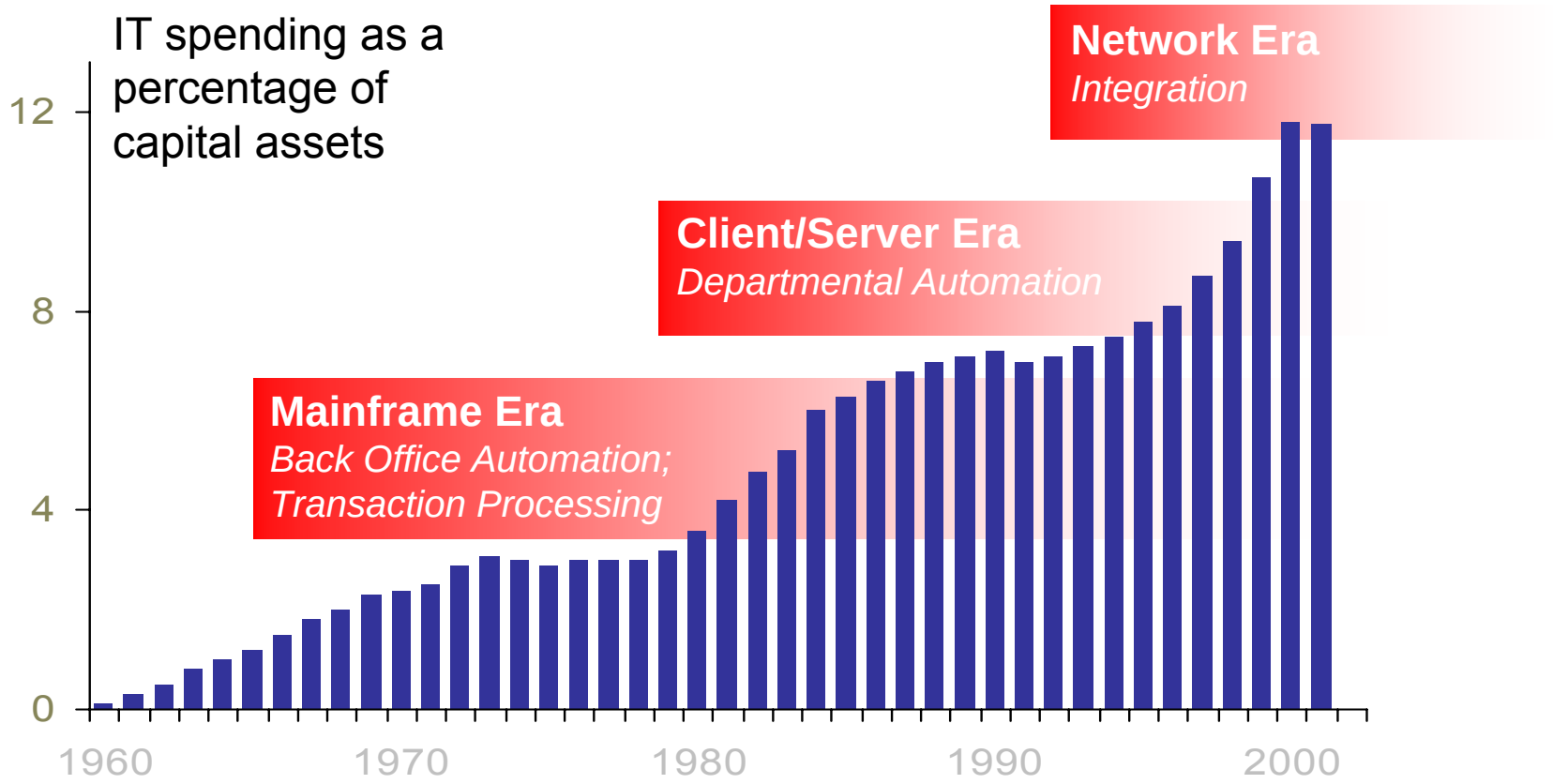
Designer‘s View



Courtesy Ferrero S.p.A.

Architectural Evolution in the IT Industry: Eras of Computing

Business system computing models

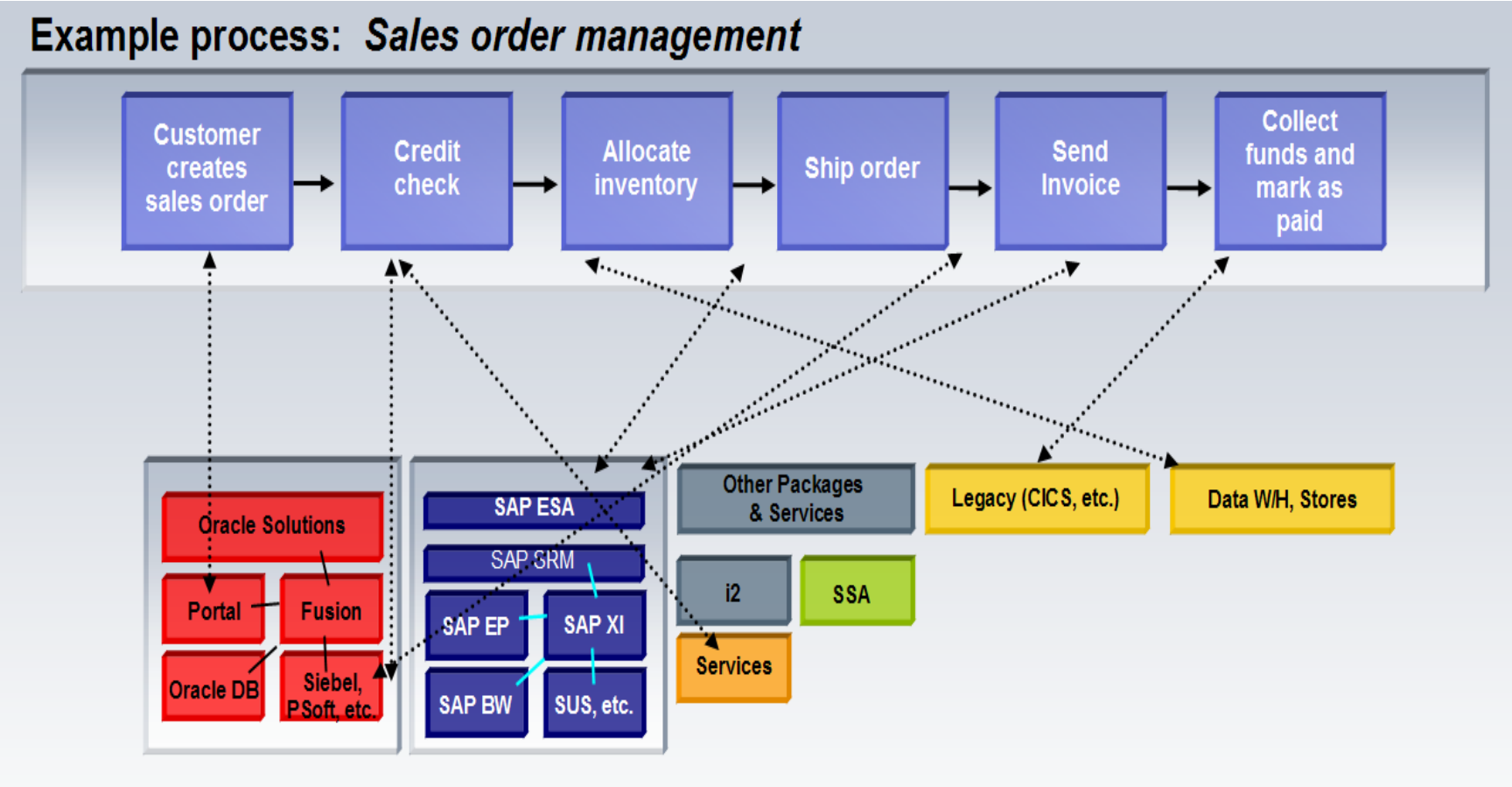


Source: U.S. Department of Commerce, Sept. 2000
2000/2001/1H02 Estimates

Eras of Computing – Key Concepts

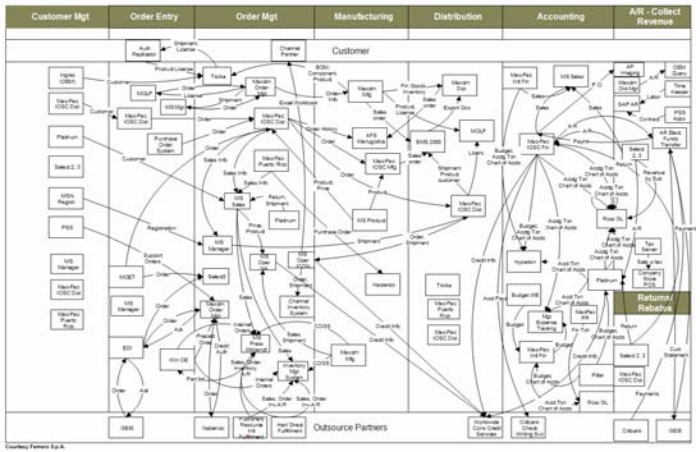
- The previous chart shows IT spending as a percentage of capital assets over the past 40 years. The data comes from the U.S. Commerce Department. During this period there were three major computing eras: the mainframe era, the client/server era, and the networked or e-business era. And in each era, IT investments addressed very specific business issues:
- The mainframe era was about solving back office problems — inventory control, payables, payroll systems, etc. It was built around automating those behind-the-scenes, centralized processes to be more efficient and effective. And mainframe computing existed in glass-house, highly controlled environments.
- Then along came the personal computer with its “killer applications,” the spreadsheet and the word processor. It democratized computing, and helped improve personal and departmental productivity. Computing moved out of the glass house and onto everyone’s desktops. It empowered individuals and small groups, but also created a big problem for enterprises – how to integrate all of these islands of computing. As a result, a new computing architecture emerged called “client/server computing.” It attempted to integrate all of these systems together, but it was very complicated and costly.
- Then along came the Internet with the advent of standards for connectivity and transactions. It suddenly gave enterprises the means to conduct transactions and integrate processes inside and outside an enterprise. IBM came to define this as “e-business.” And so today, we find ourselves in the middle of another major shift in the industry. And as in the other eras, the computing model necessary to support this is quite different from the previous one -- and it is driving a change in the industry landscape today.
- If you look at the data, you will see that the highest percentage of spending over these eras has been in the e-business era. This is because IT has become a more integral part of business strategy for most enterprises. It is seen as a way to drive down cost through efficiencies and to gain share through innovative business practices, such as new ways of managing customers, selling, interacting with suppliers, etc.
- By the way, none of these computing models described in the previous chart will ever go away. For example, the mainframe has been growing revenue now for the past nine or ten quarters. This industry doesn't stop and then recreate; it evolves. But clearly the infrastructure, the underpinning model that makes all this work, is different in the networked era.

Designer's View:
Processes span multiple applications, databases, and services.
How do you architect the solution? To serve the business?
To scale? To enable flexibility?



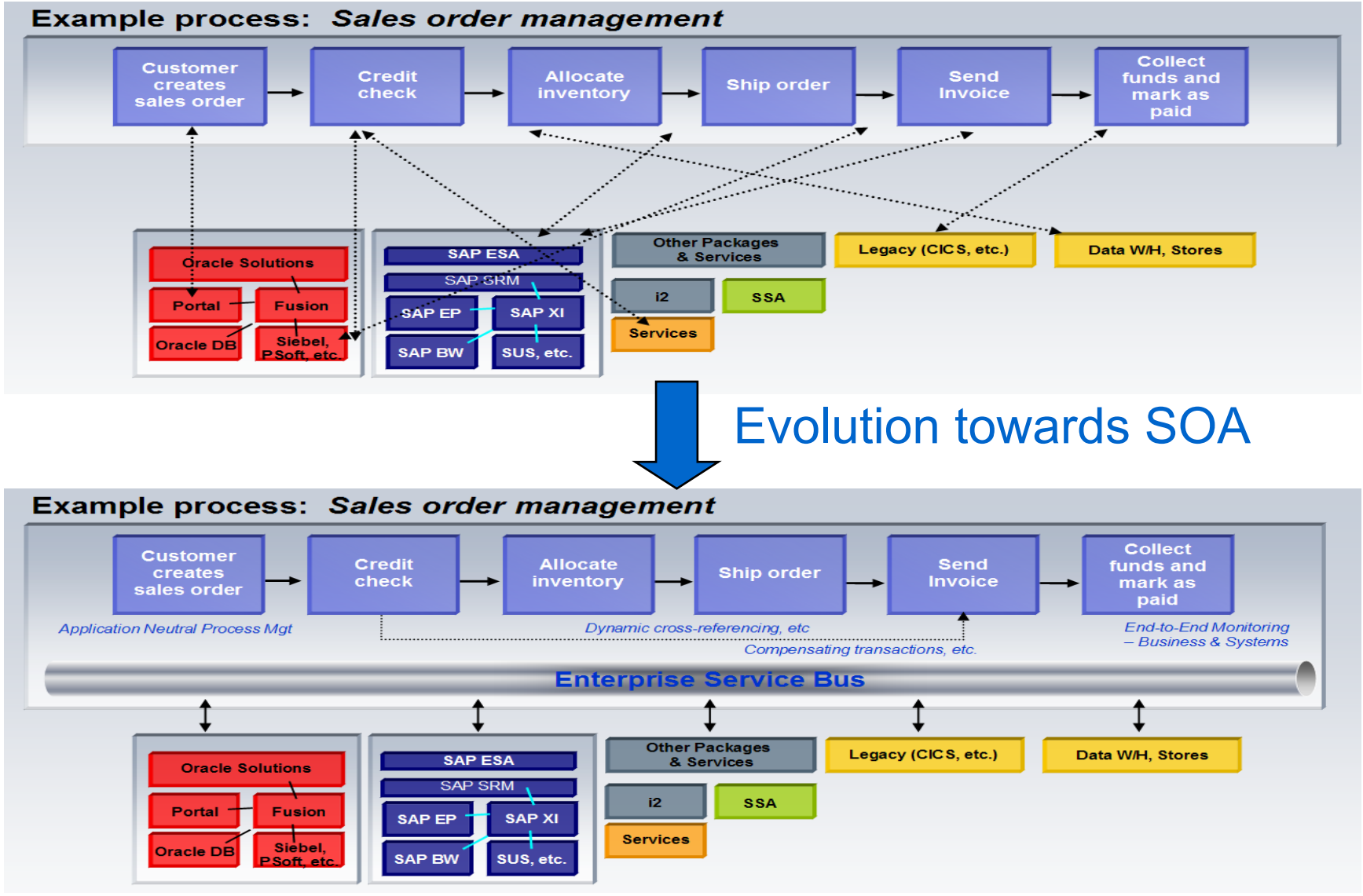
Why is IT Architecture important?
Business flexibility depends on IT flexibility

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

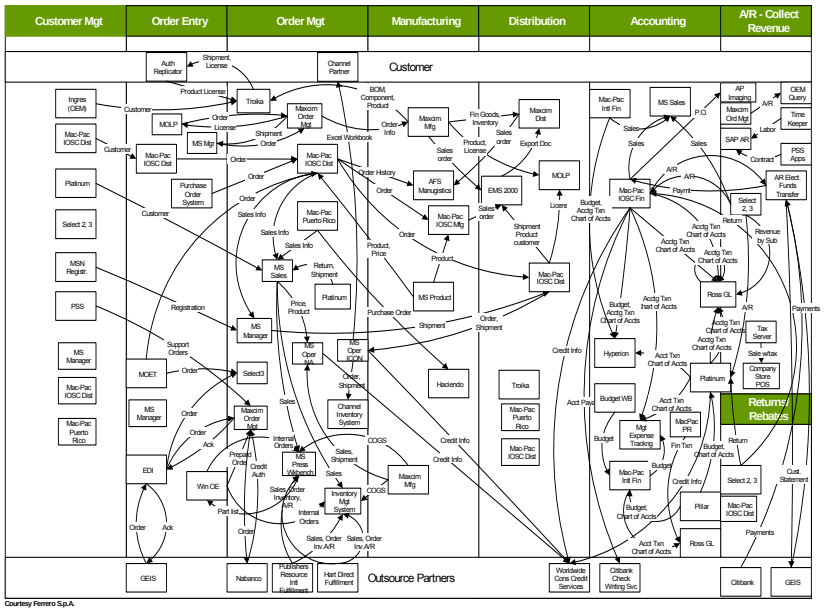


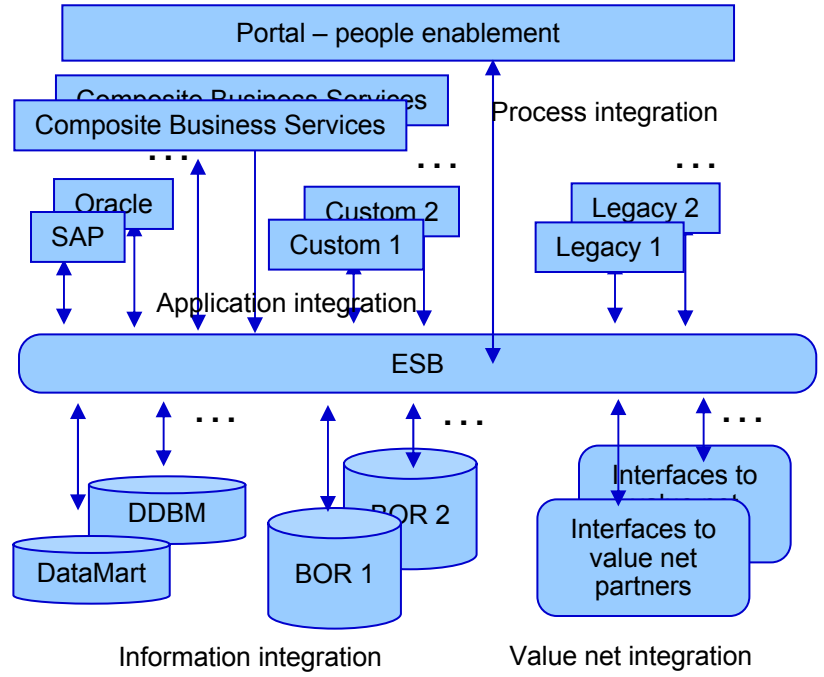
McKinsey
“Flexible IT, Better Strategy”

Latest Developments in Architecture Evolution: SOA (Service Oriented Architecture), Enterprise Service Bus



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

Latest Developments in Architecture Evolution: SOA (Service Oriented Architecture)

- Architecture that uses open standards to represent software assets as services
- Standard way of representing and interacting with software assets
- Individual software assets become building blocks that can be used in developing other applications
- Used internally to create new apps out of existing components
- Used externally to integrate with apps outside organizational boundaries

Dimensions of the IT Architecture Discipline

- **Models**
  - Representations of the overall architecture that are meaningful to one or more stakeholders (architectural views)
- **Frameworks**
  - Processes, activities, guidelines and tools to help solve IT architecture problems
  - Key techniques (methods and tools) used by IT architects
  - Specifications of how to describe architectures
  - List of recommended techniques/standards and compliant products that can be used to implement the building blocks.
- **Skills**
  - Roles and responsibilities of the IT architect job

Role of Frameworks

- To establish terms and concepts for architectural thinking
- To provide a means to talk about Architectural Descriptions in the context of
  - Stakeholder concerns
  - Life cycle
- To guide the development of a broad range of different architectures
  - What subjects should an architect cover
  - How to describe an architecture
    →Architectural Description Standard(s)

Examples of Architectural Frameworks

DoD Architecture Framework Working Group

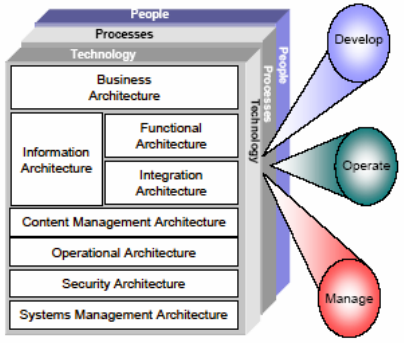
DoD Architecture Framework Version 1.0

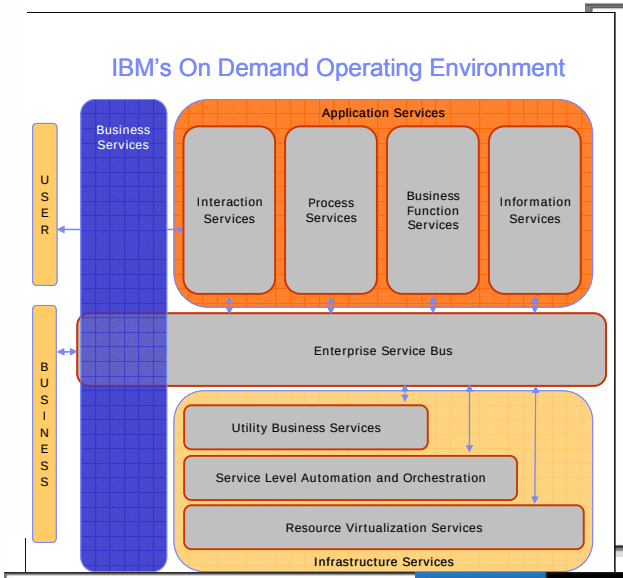
Volume I: Definitions and Guidelines

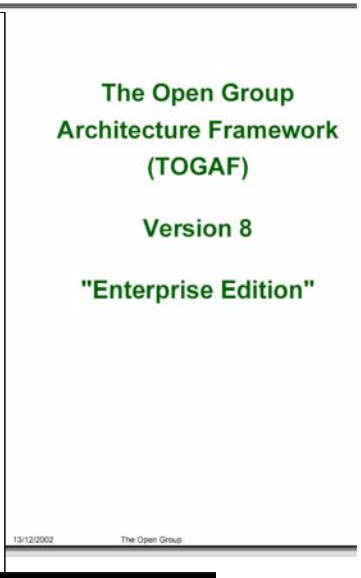
15 August 2003

Architecture Blueprint

- An e-business architecture blueprint consists of 8 collaborating models that represent the overall solution.
- The architecture models breakdown the complexity of a system, to support parallel development, incremental construction and isolated verification of parts.
- These models may vary in relative importance to one another based on the solution they support.
- Any custom built architecture or reference architecture should include all of the 8 models described in this blueprint.

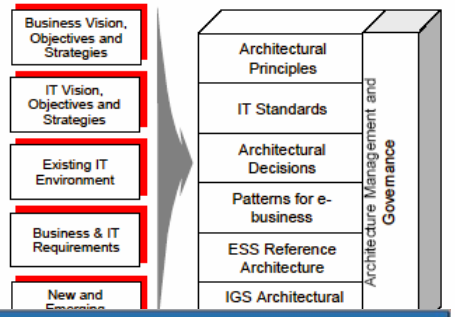


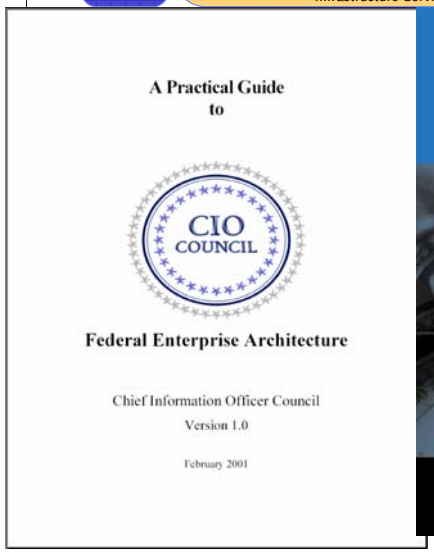


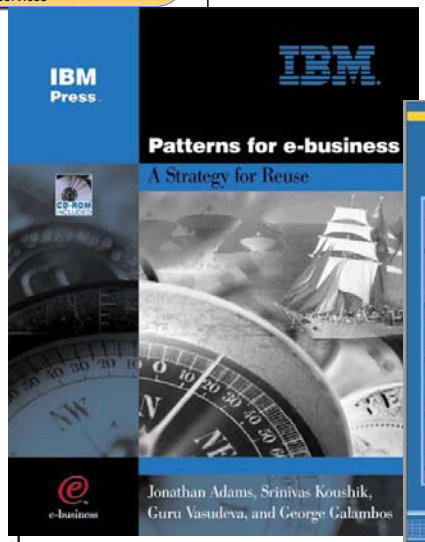


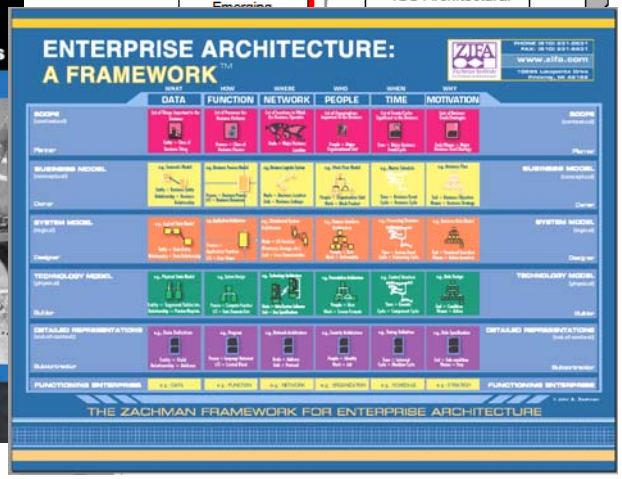
Adaptive Blueprint for Enterprise Architecture

- The Adaptive Blueprint for Enterprise Architecture (ABEA) is much like a city plan in that it defines an infrastructure that will meet current and future needs of a diverse user population and will adapt to changing business requirements and technology capabilities.
- Development of an enterprise architecture is influenced by several business and technical factors.
- The foundations of the Adaptive Blueprint for Enterprise Architecture are based on robust, proven architectural assets from IBM.
- The power & elegance of ABEA comes from the process used to leverage IBM assets in defining an architecture that stands the test of time.

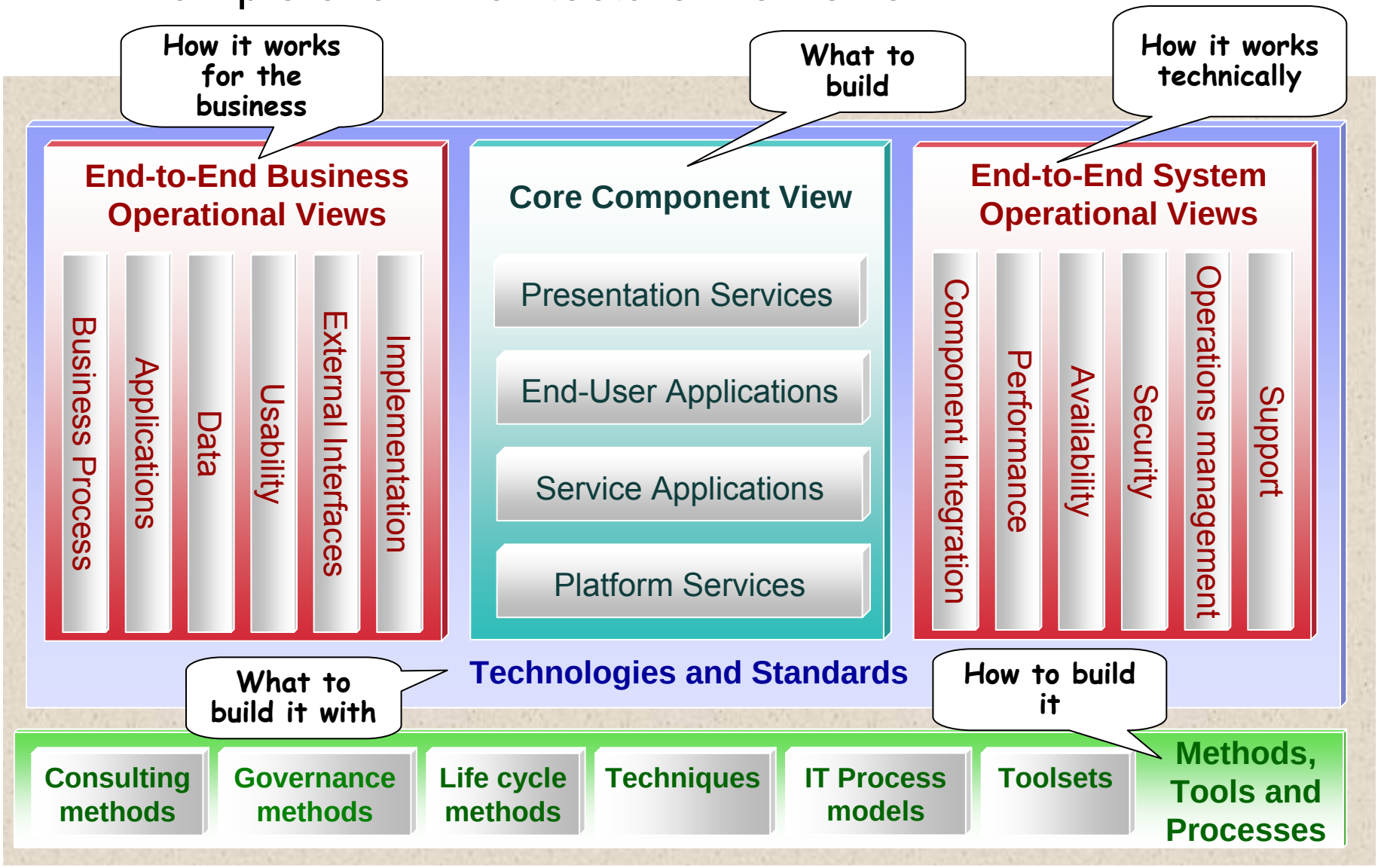




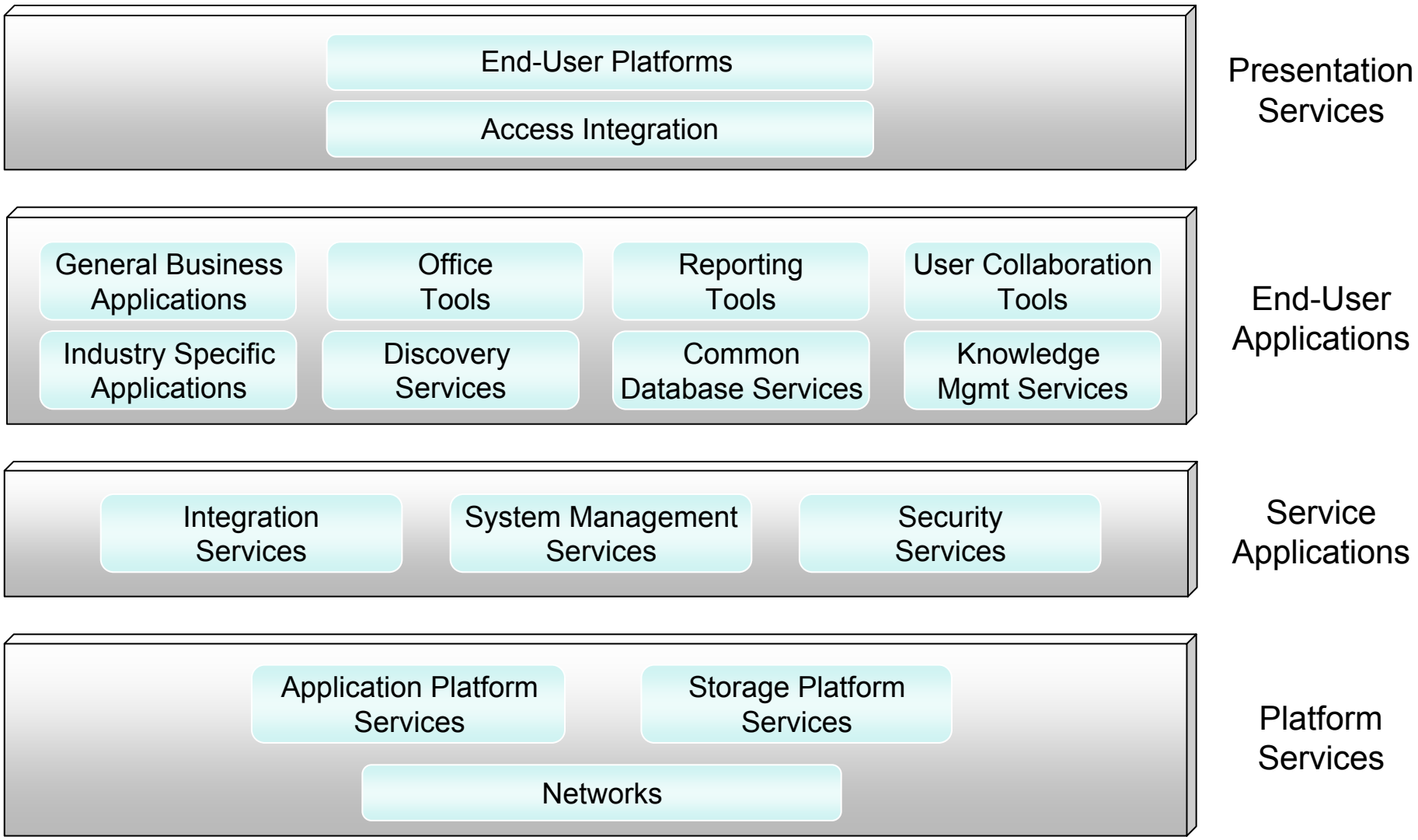




An Example of an Architecture Framework



What to build: Common Core Components – Component Groups



How it works technically: End-to-end Operational System Views

Component Integration
- Computing models
  e.g. Web Architecture, Windows rich client, .NET, or server centric
- Integration style
  e.g. batch, real time, message based
- Purpose of integration
  e.g. user to user, user to data, application to application

Performance
- Performance and response times
- Scalability and capacity

Availability
- Reliability, resilience, redundancy
- Monitoring, restart / recovery

Security
- Overall system security view and policy
- Threat analysis and containment

Operations Management
- System management
- Change management
- Configuration management
- Business continuity and disaster recovery

Support
- Help Desk arrangements and integration
- 2nd and 3rd level support

How it works for the business: End-to-End Operational Business Views

Business Process View
- End-to-end business processes and narratives
- Use cases

Applications Architecture
- Vertical integration: N-Tier view
- Horizontal integration: Business process view

Data Architecture
- Data models and data standards
- Data ownership, data quality
- Data copies and data management

Usability
- Usability and consistency of applications
- Visibility of integration points to users

External Interfaces
- Interfaces to other systems, external to the system architecture under study
- Business to Business interfaces

Implementation
- Education and training
- Management of Change
- Roll out arrangements – eg big bang or phased

How to do the job: Methods and Processes - Examples

Consulting methods
- Enterprise Architecture (EA)
- Management of Change
- Component Business Modelling (CBM)

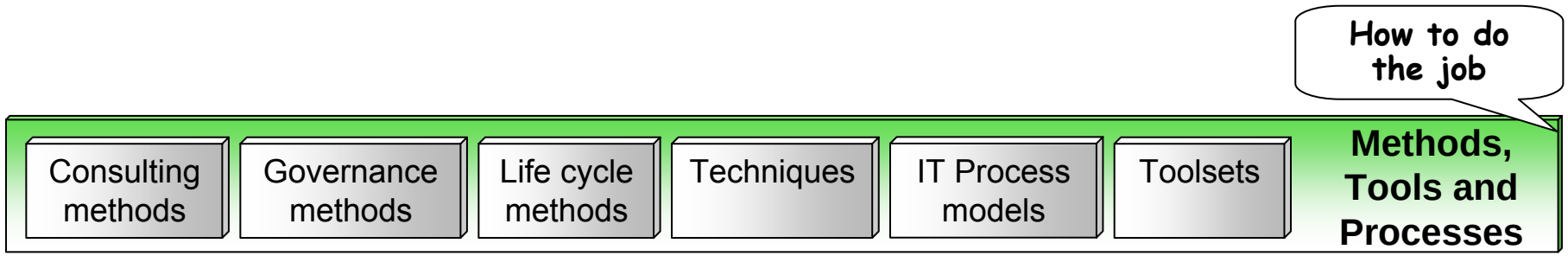
Governance methods
- Programme / Project management
- DA and SE&A
- Change, risk, issue management
- RAC

Life cycle methods
- GS Method, SLF
- DSDM, RAD & time boxing

Techniques
- JAD, Simulation,
- Process modelling, Data modelling

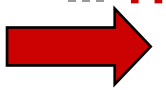
IT Process models
- CMMi, ITIL, EOP

Toolset Support
- MS Project
- Engagement Support Environment - ESE
- Lifecycle Estimation and Duration - LEAD
- Rational Requisite Pro, IDE/XDE, Rose, TestManager, Robot, Test RT, ClearQuest
- HyPerformix, Relex
- GS Risk



Role of Frameworks

- To establish terms and concepts for architectural thinking
- To provide a means to talk about Architectural Descriptions in the context of
  - Stakeholder concerns
  - Life cycle
- To guide the development of a broad range of different architectures
  - What subjects should an architect cover
  - **How to describe an architecture**

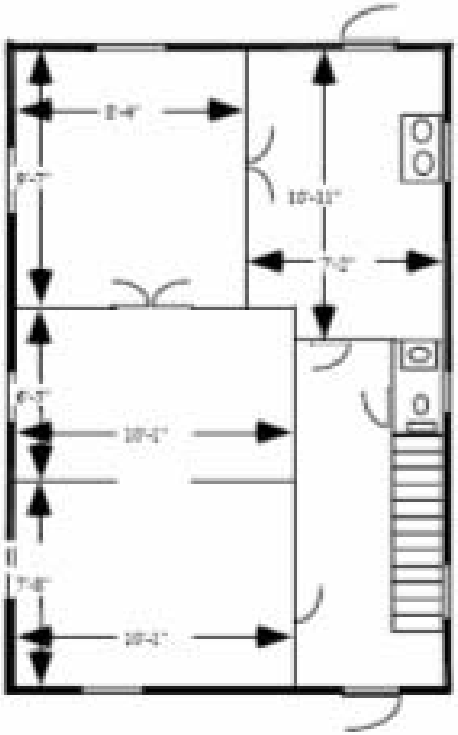


Architectural Description Standard(s)

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**

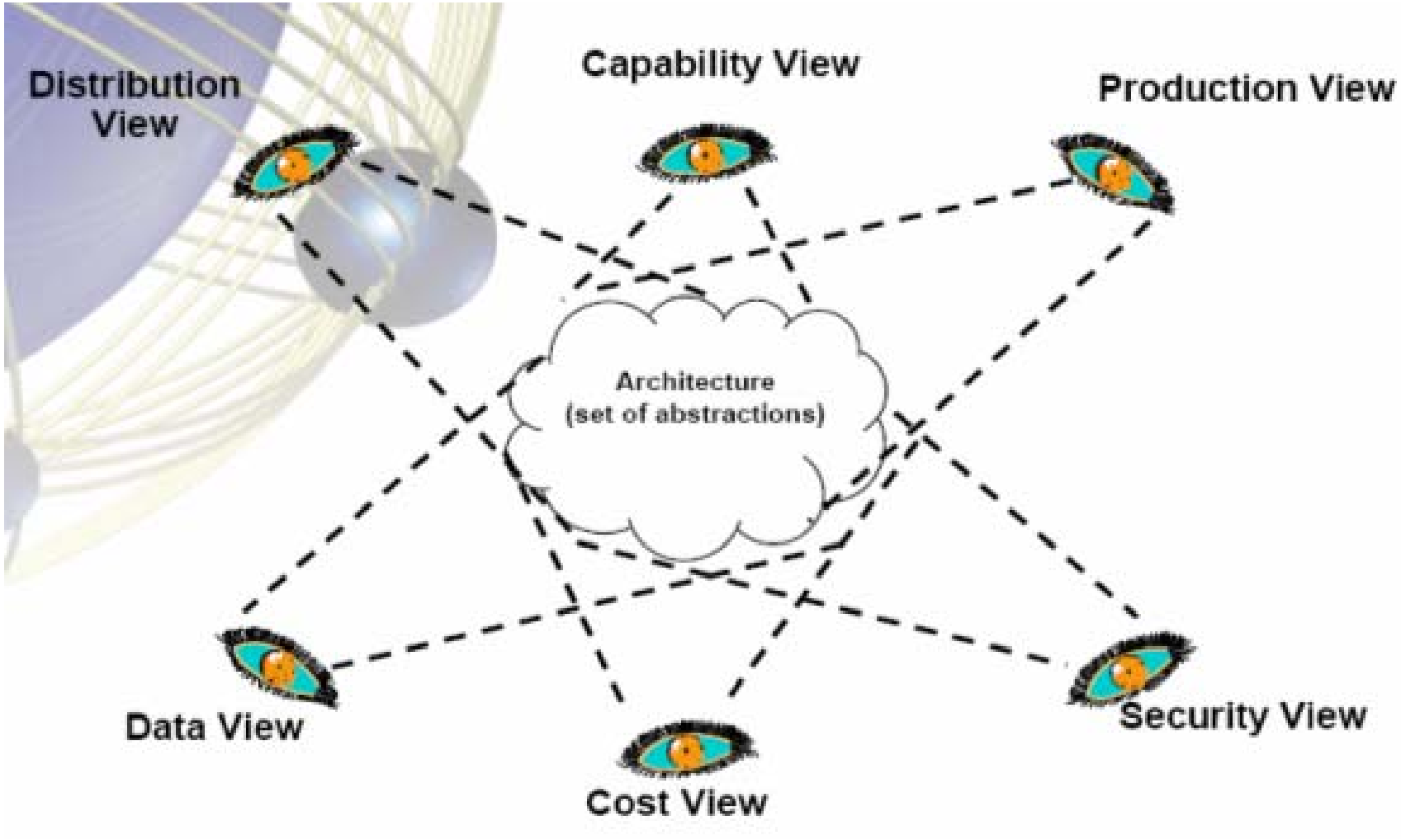
Multiple views and models



- 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

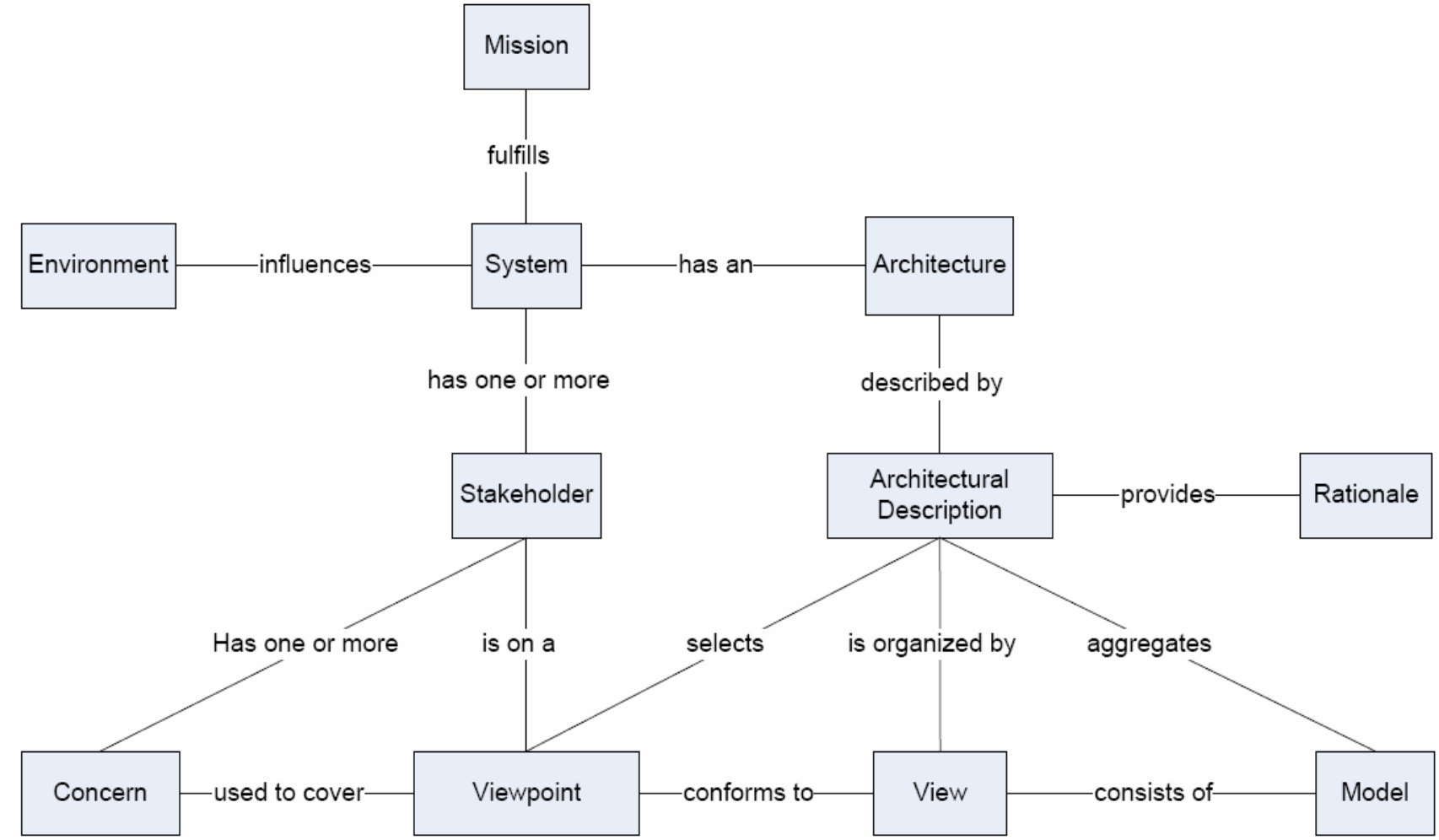
Sample Views



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



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

Viewpoints

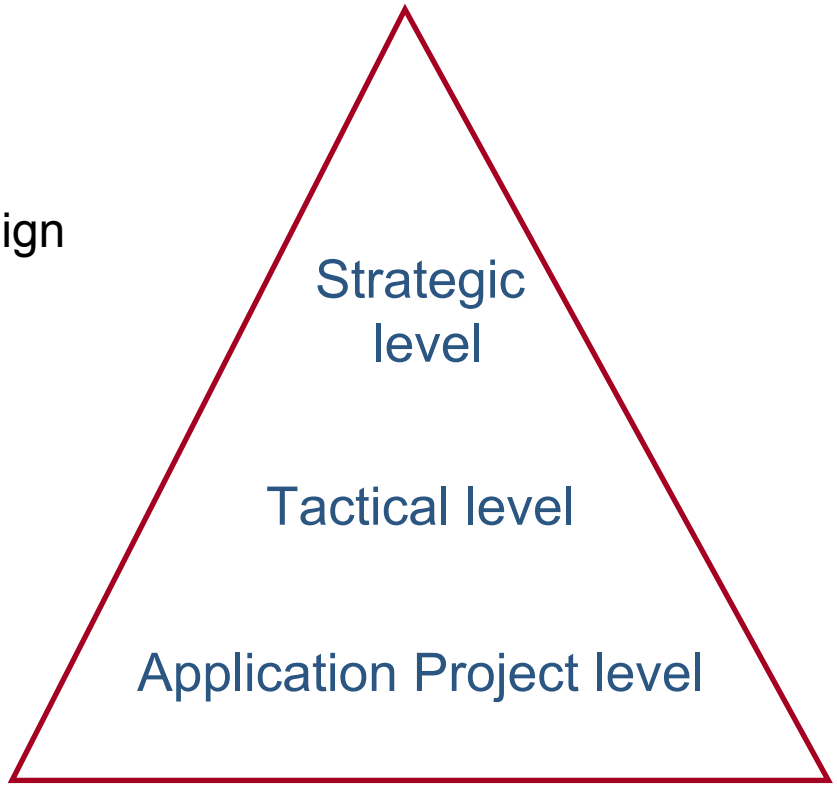
Specified by	Example
Viewpoint name	Physical data storage viewpoint
Stakeholders addressed by the viewpoint	System developers; technical architects; IT investment analyst
Concerns addressed by the viewpoint	Where is data, how will it be distributed, how much will it cost?
Vocabulary, language, modeling techniques or analytical methods used in constructing a view from this viewpoint	Data normalization / de-normalization, UML, business rules

Dimensions of the IT Architecture Discipline

- **Models**
  - Representations of the overall architecture that are meaningful to one or more stakeholders (architectural views)
- **Frameworks**
  - Processes, activities, guidelines and tools to help solve IT architecture problems
  - Specifications of how to describe architectures
  - Answers to the question: “What is IT Architecture?”
- **Skills**
  - Roles and responsibilities of the IT architect job
  - Key techniques (methods and tools) used by IT architects

The necessity of the IT Architecture role

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



Skills: 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

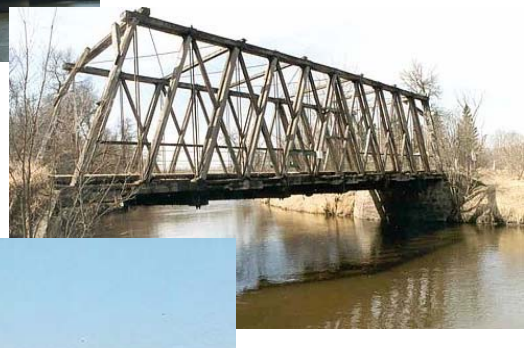
Compare with the Construction Architect and/or Town Planner

- 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**, negotiating
  - **Represents** the client's interests
  - **Supervising** the build

Compare with the Construction Architect and/or Town Planner

- For example, if a client wants a bridge built, a civil engineer can ask the user a number of question about the bridge's function, traffic loads, setting, and environmental factors.
- Based on the answers, the engineer can identify the most appropriate **architectural style** of the bridge: suspension, cantilever, arch, truss, etc.
- This intermediate abstraction then enables the engineer to capitalize on codified principles and experience to specify, analyze, plan, and monitor the construction of this style of bridge, with high levels of efficiency and confidence.

(Shaw and Garlan 1996)



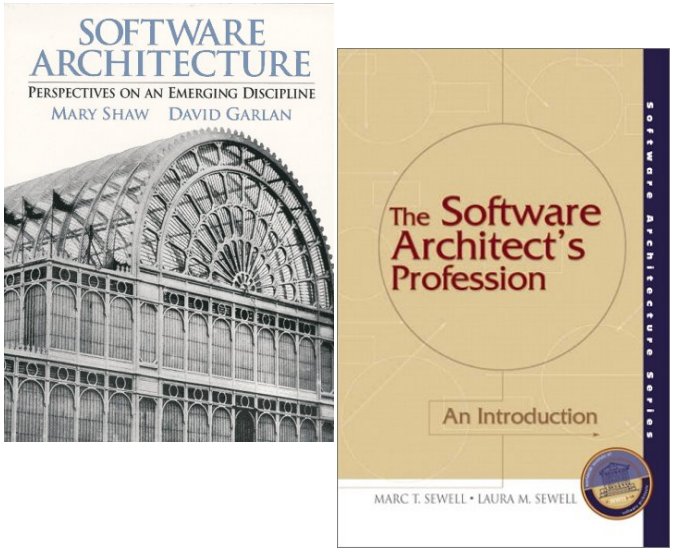
A word of caution

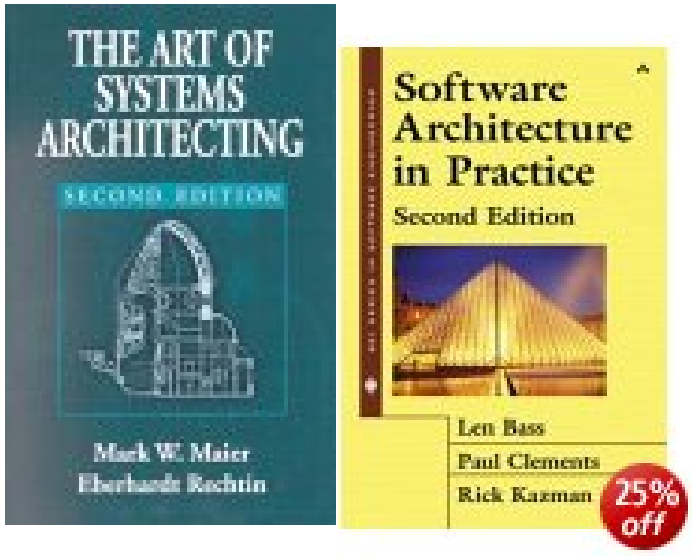
- *Analogies between buildings and software systems should not be taken too far, as they break down fairly quickly. Rather, they help us understand that the viewer's perspective is important and that structure can have different meanings depending on the motivation for examining it. A precise definition of software architecture is not nearly as important as what investigating the concept allows us to do.*
- *People often make analogies to other uses of the word architecture, about which they have some intuition. They commonly associate architecture with physical structure (buildings, streets, hardware) and physical arrangement. A building architect must design a building that provides accessibility, aesthetics, light, maintainability, and so on. A software architect must design a system that provides concurrency, portability, modifiability, usability, security, and the like, and that reflects consideration of the tradeoffs among these needs.*

Software Architecture in Practice, Second Editionby Len Bass, Paul Clements, Rick Kazman

Books

- Be selective; no single best source





Reading

- Hand-outs (available for download from ....)
  - Intro.ppt
  - What is Architecture.ppt
- Preparation for lecture next week (available for download from ....)
  - What do IT Architects do all day?
  - L. Cherbakov, G. Galambos, R.Harishankar, S. Kalyana, G. Rackham: Impact of service orientation at the business level, IBM Systems Journal, 2005
  - R. Schulte, Architecture and Planning for Modern Application Styles, Gartner Strategic Analysis Report, 1997
  - R. Schulte, Clarifying the Terms "Event-Driven" and "Service-Oriented Architecture", Gartner Research, 2005
  - J.A. Zachman, A framework for information systems architecture, IBM Systems Journal, 1987
- Other Suggested Reading (available for download from ....)
  - IEEE Recommended Practice for Architectural Description of Software-Intensive Systems: IEEE Standard 1471
  - M. Maier, D. Emery, R. Hilliard: Software Architecture - Introducing IEEE Standard 1471
  - Rechtin, Appendix A: Heuristics for systems level architecting
  - Rechtin, The art of systems architecting, IEEE Spectrum 1992
  - R. Youngs et al., A standard for architecture description, IBM Systems Journal, 1999
  - M. R. McBride, The software architect, Comm. ACM, May 2007
  - David E. Emery, Architectural Frameworks: Defining the contents of architectural desriptions
  - THE INFLUENCE OF ARCHITECTURE IN ENGINEERING SYSTEMS, Engineering Systems Monograph, MIT esd, 2004.

Extended Reading List

- Alexander, Christopher (1979) , The Timeless Way of Building New York: Oxford University Press
- Applegate, Lynda.M. (1995) Designing and managing the information age IT architecture, Harvard Business School Publishing Teaching case 9-196-005
- Applegate, Lynda.M. (1995b) Managing in an Information Age: IT challenges and Opportunities, Harvard Business School Publishing Teaching case 9-196-004
- Bass, L., Clements, P and Kazman, R. (2003) Software Architecture in Practice (2-ed), Reading, Massachusetts: Addison-Wesley / Pearson.
- Dewang, Arjan (2003) Face-off. Computer Bulletin Jan 03 26-27
- Earl MJ (1989) , Management Strategies for information Technologies, Prentice Hall Ch5, 95-116
- Ince, DC & Hekmatpar, S (March 1988) An Approach to automated software design based on product metrics. Software Engineering Jnl: 53-56
- Lloyd PTL, Galambos GM (1999) Technical Reference Architectures. IBM Systems Journal 38 (1) 51-75.
- Ross JW (2003) Creating a Strategic IT Architecture Competency: Learning in stages. MIS Quarterly Executive (2) 1 March 2003: 31-43
- Rozanski, N (2005) The qualities that define an IT Architect, Computing 24 Nov 2005 p22.
- Sauer C and Willcocks LP (2002) The evolution of the Organizational Architect, Sloan Management Review 43(3) 41-49
- Seger, K and Stoddard, DB (1993) Managing Information: the IY Architecture, Harvard Business School Publishing: teaching case 9-193-059.
- Sewell MT, Sewell LM (2002) The Software Architect's Profession: An Introduction. Upper Saddle River NJ: Prentice Hall PTR.
- Shaw, M. and Garlan, D. (2003) Software Architecture: perspectives on an emerging discipline. New Jersey: Prentice Hall.
- Wexelblat, R.L. and Srinivasan, N. (1999) Planning for information technology in a federated organization. Information and Management, 35 (5), 3 May 1999, 265-282
- Willcocks, L; Feeny, D and Islei, G (1997) Managing IT as a Strategic Resource, McGraw-Hill
- Yadav (Jan1990), IEE transactions on software engineering