Software (Systems) Architecture Foundations

Lecture #5

Larger Context

Alar Raabe

Recap of Last Lecture

It is not about bits, bytes and protocols, but profits, losses and margins

Lou Gerstner

- (Software) Quality
  - degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value (ability of system to meet customer or user needs, expectations, or requirements)
- (Software) Quality Attribute
  - a measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders
- Economic Value of (Software) Architecture
  - Value of Architecture = cost of realization of risks compared to cost of architecture
  - Value of Architecture Description = cost of performing activities without architecture description compared to cost of documenting architecture
- (Software) Architecture creates choices/options, which have value – **designing and building an architecture is an investment activity**
  - Architecture Investment is a real option
    - provides an opportunity (right, but not an obligation) to make a decision in the future
    - might be exercised multiple times (different from financial option)

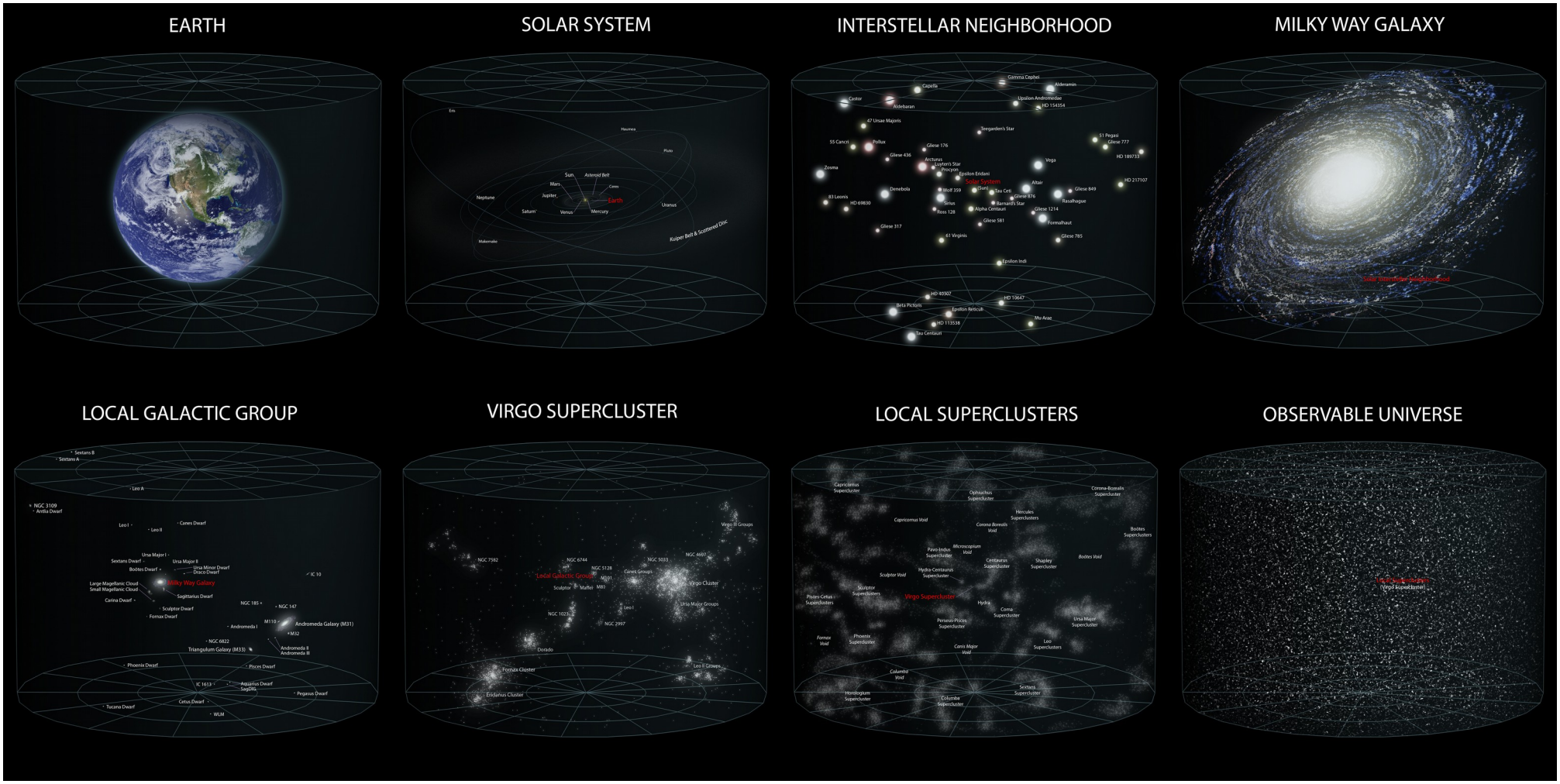
Content

Strategy without tactics is the slowest route to victory
Tactics without strategy is the noise before defeat

Sun Tzu

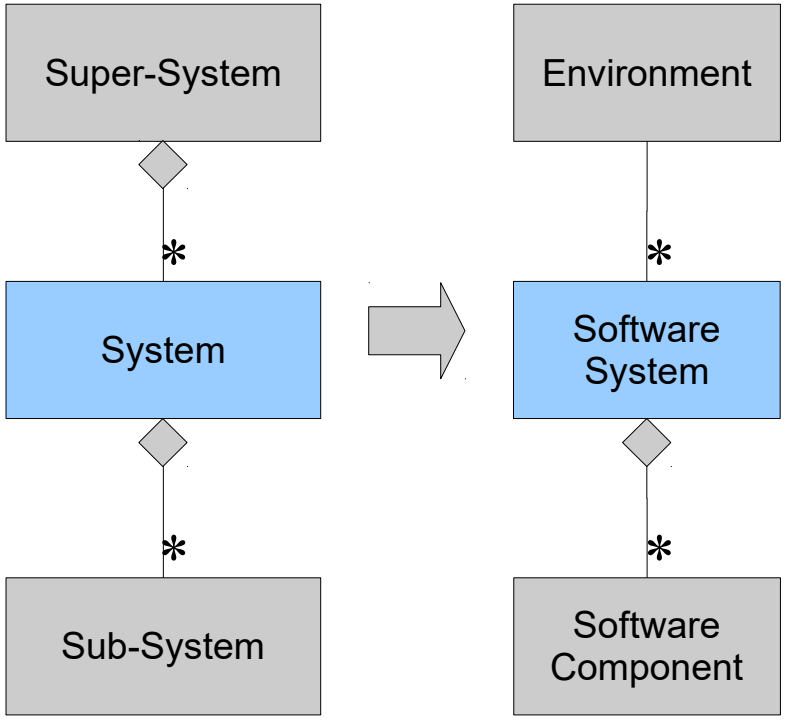
- Larger context
  - Hierarchy of Systems
  - System of Systems
- Enterprise Architecture
  - Need for Larger Context and Structured Approach
  - Some Approaches to Enterprise Architecture
  - Standard for Enterprise Architecture – TOGAF
  - Reference Architectures
  - Language for Enterprise Architecture – ArchiMate
- Conclusions

Big → Bigger → Biggest

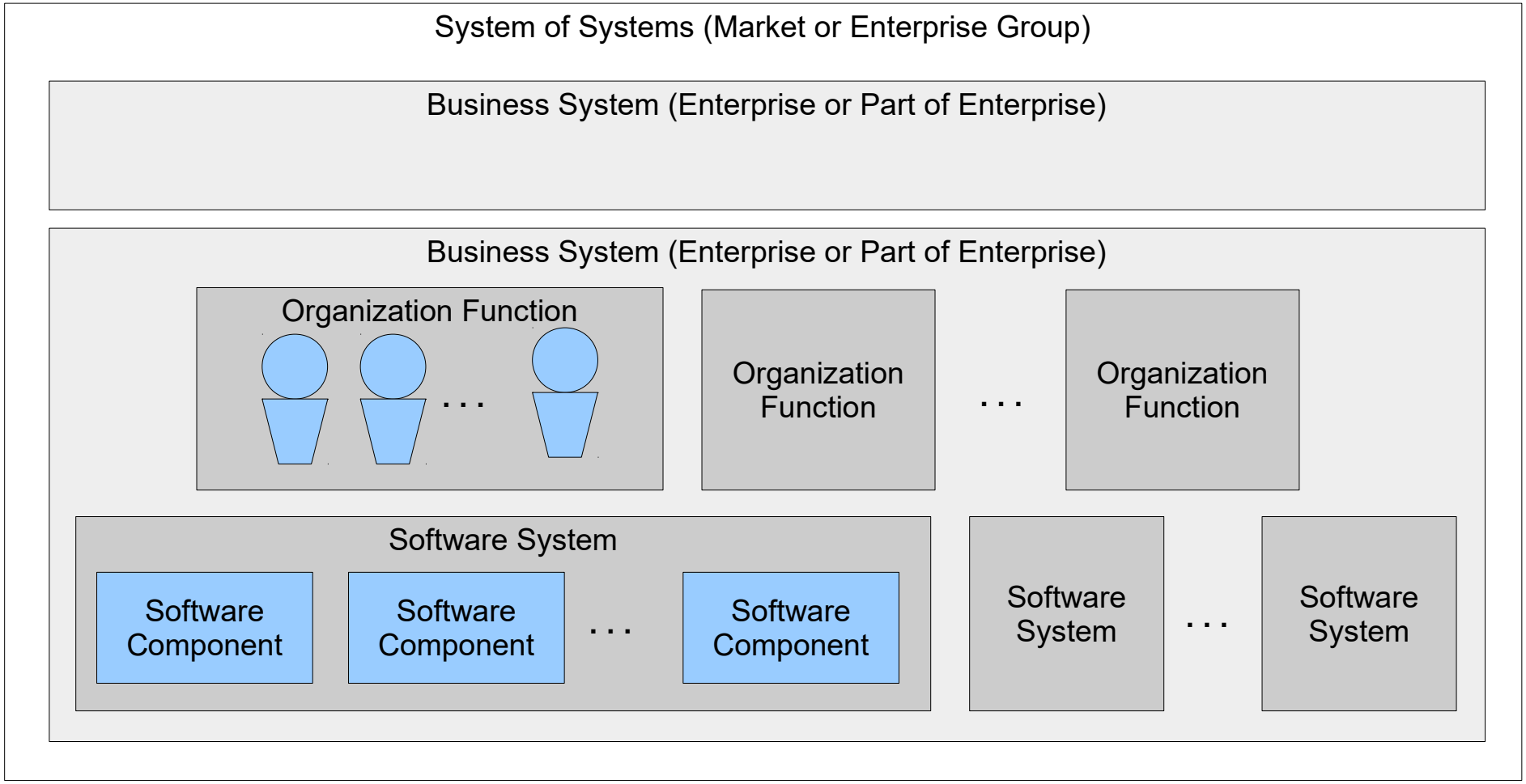


Hierarchy of Systems

- One’s Element/Component is another’s System
- One’s System is another’s Environment (or Context)
- One’s Design is another’s Architecture



Hierarchy of Systems in Enterprise



Systems of Systems

CMU SEI

- System of Systems – a collection of task-oriented or dedicated systems that pool their resources and capabilities together to create a new, more complex system which offers more functionality and performance than simply the sum of the constituent systems

- Kinds of Systems of Systems

  - Directed
    - objectives, centralized management, funding, and authority for the overall SoS are in place
    - systems are subordinated to the SoS

  - Acknowledged
    - objectives, centralized management, funding, and authority in place
    - however, systems retain their own management, funding, and authority in parallel with the SoS

  - Collaborative
    - there are no overall objectives, centralized management, authority, responsibility, or funding at the SoS level
    - systems voluntarily work together to address shared or common interests

  - Virtual
    - like collaborative, but systems don't know about each other

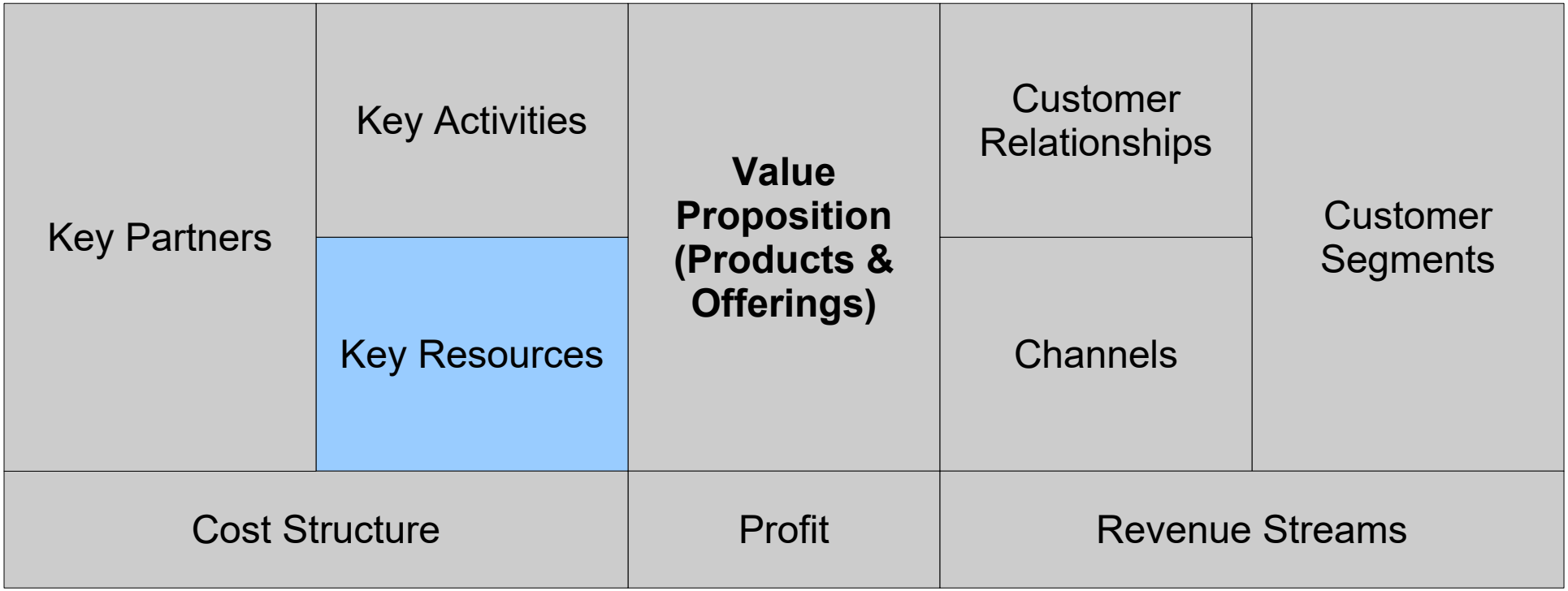
Properties of Systems of Systems

CMU SEI

- Operational Independence of the Components
  - If the system of systems is disassembled into its component systems, the component systems must be able to usefully operate independently – that is, the components fulfill customer-operator purposes on their own
- Managerial Independence of the Components
  - The component systems not only can operate independently, they do operate independently
  - The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system of systems
- Evolutionary Development
  - The system of systems does not appear fully formed, its development and existence is evolutionary with functions and purposes added, removed, and modified with experience
- Emergent Behavior (also negative!)
  - The system of systems performs functions and carries out purposes that do not reside in any component system – these behaviors are emergent properties of the entire system of systems and cannot be localized to any component system
  - The principal purposes of the systems of systems are fulfilled by these (emergent) behaviors
- Geographic Distribution
  - The geographic extent of the component systems is large (the components can readily exchange only information and not substantial quantities of mass or energy)

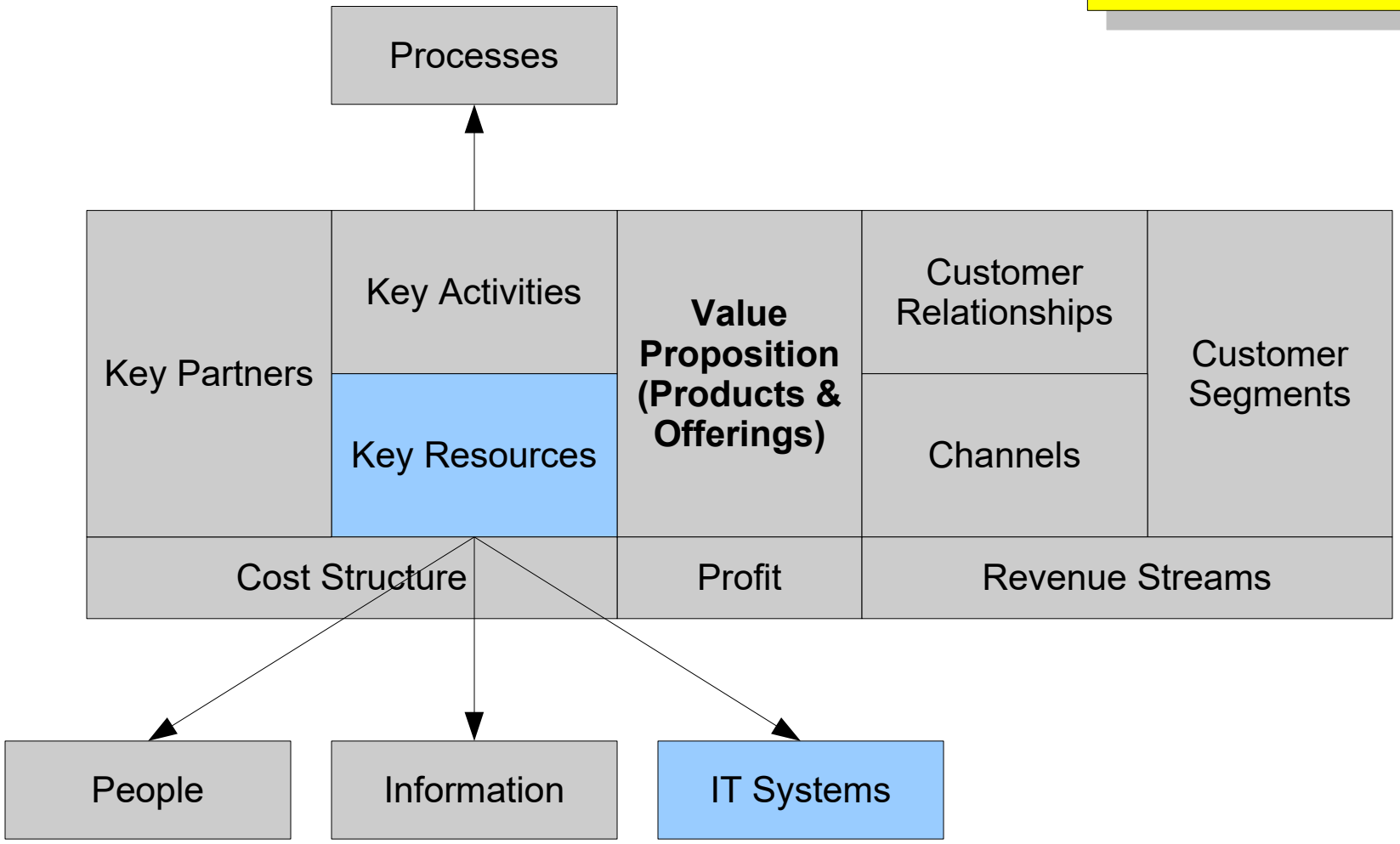
Conceptual Model of Business

Business Model Canvas
A. Osterwalder



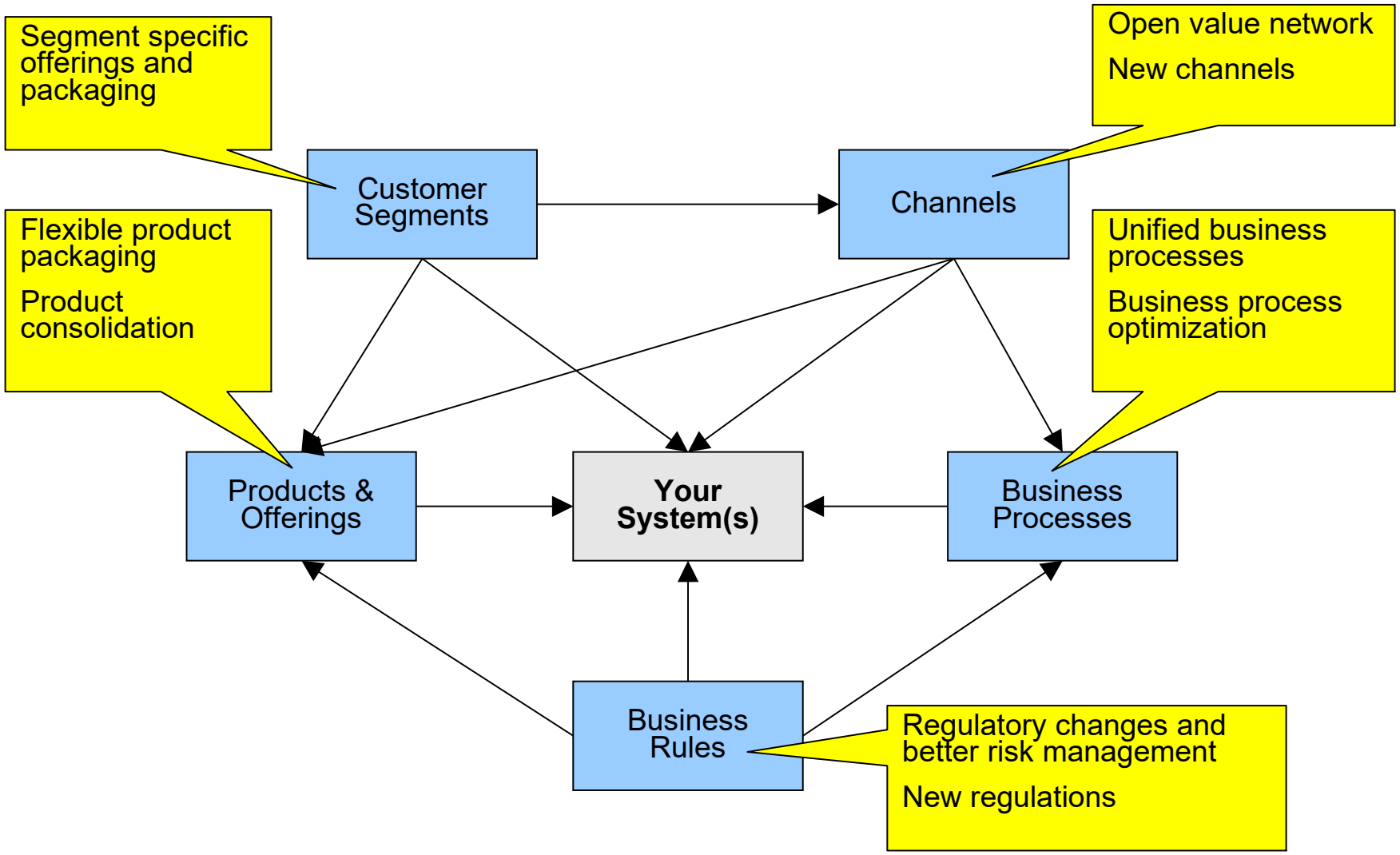
Business Model → Context of IT Systems

Business Model Canvas
A. Osterwalder



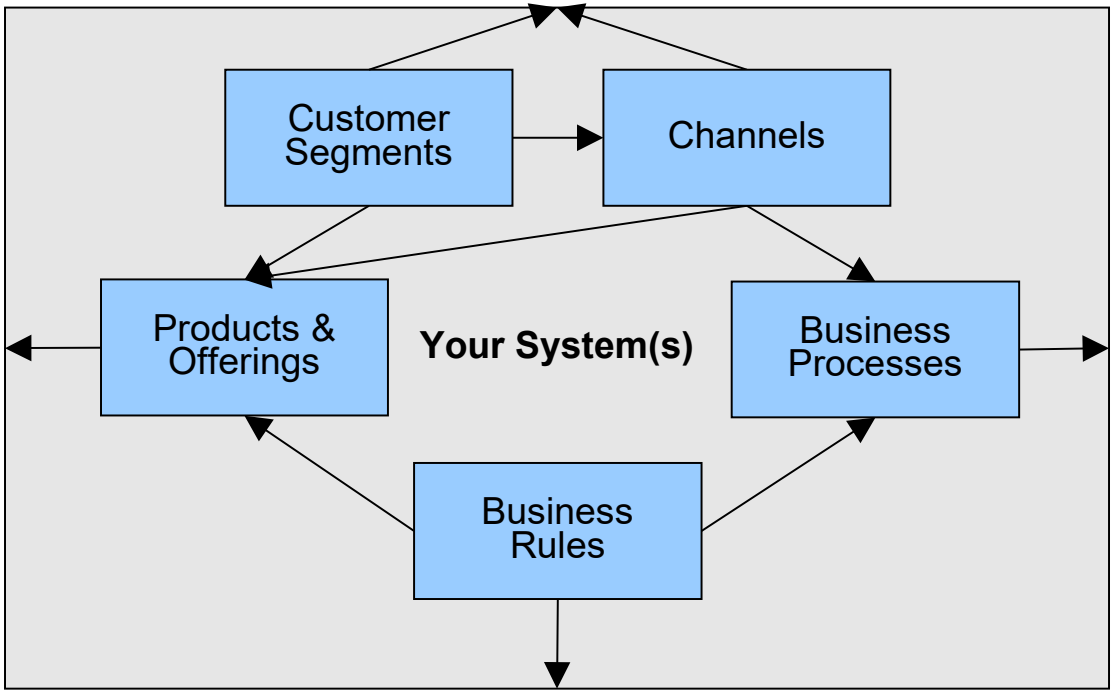
Example

Effects of Business Model on the IT Systems (Ask What changes in business and how often?)



Example

Effects of Business Model on the IT Systems
(Plan for business change – make business explicit!)



General Principles for Architecting Systems of Systems

M. W. Maier, 1998
H. Sillitto, 2010

- Design systems so that they can deliver value if they are incomplete
  - There should be several ‘stable intermediate forms’ so that a partial system works and can do useful things
- Be realistic about what can be controlled – “chain of command” must evolve to a “web of shared interest”
  - The best performance from a SoS may be achieved when there is an individual or group that exerts control on the overall system and its constituents, however, attempts to over control the SoS are likely to lead to resistance from the individual system owners
- Focus on the system interfaces
  - To build a successful system of systems you have to design interfaces so that the system elements can interoperate – these interfaces must not be too restrictive so that the system elements can evolve and continue to be useful participants in the SoS
- Provide collaboration incentives
  - When the system elements are independently owned and managed, it is important that there are incentives for each system owner to continue to participate in the system (these may be financial, access/sharing or community incentives)

- Design a SoS as nodes and web architecture
  - Nodes include data, software, hardware, infrastructure (technical components), organizational policies, people and processes
  - Web provides communications (incl. a mechanism for informal and formal social communications between the people)
- Specify behavior as services exchanged between nodes
- Understand and manage system vulnerabilities
  - It is critically important to try to understand vulnerabilities and design the system to be resilient to such failure

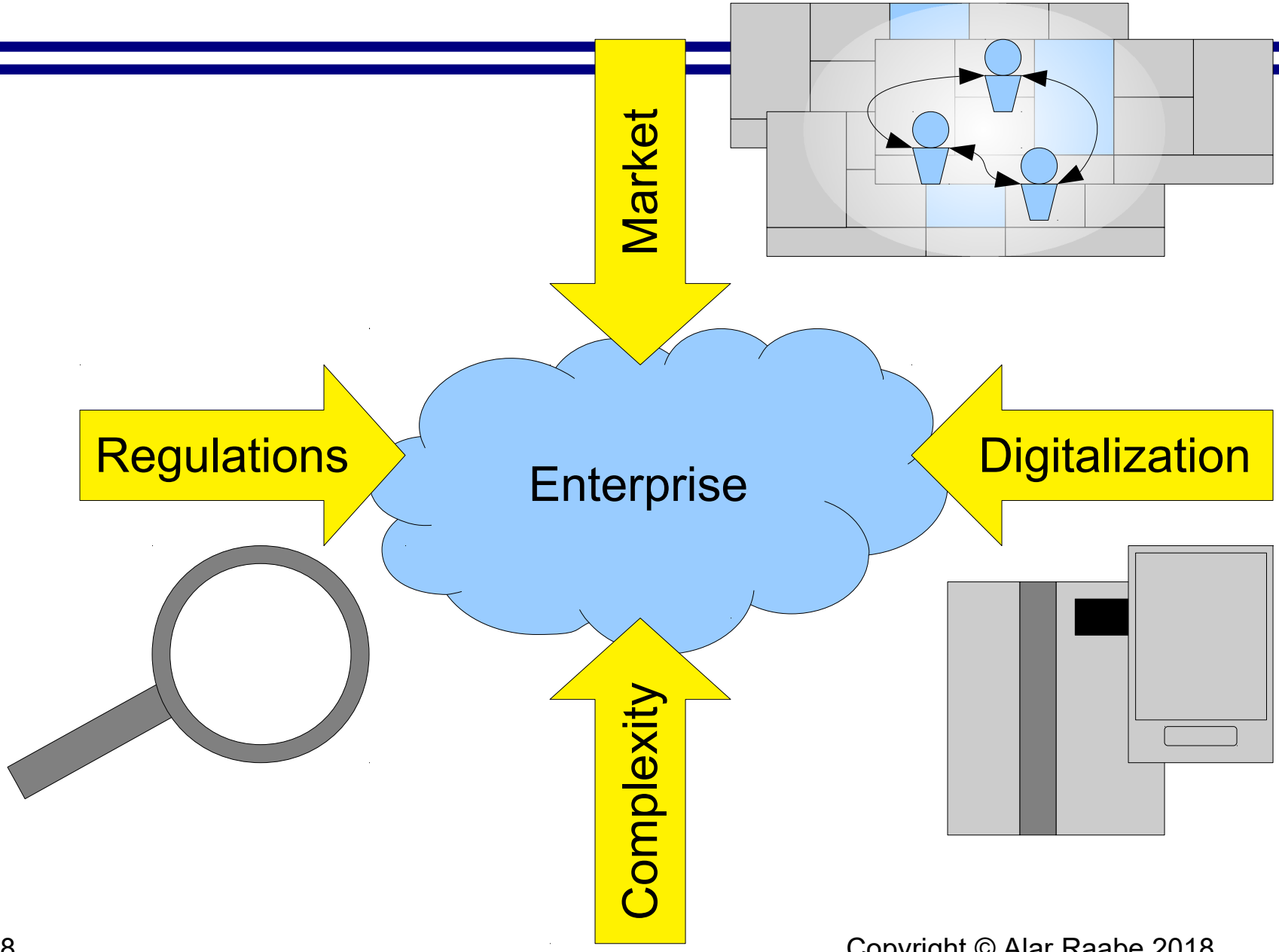
Content

Strategy without tactics is the slowest route to victory
Tactics without strategy is the noise before defeat

Sun Tzu

- Larger context
  - Hierarchy of Systems
  - System of Systems
- Enterprise Architecture
  - Need for Larger Context and Structured Approach
  - Some Approaches to Enterprise Architecture
  - Standard for Enterprise Architecture – TOGAF
  - Reference Architectures
  - Language for Enterprise Architecture – ArchiMate
- Conclusions

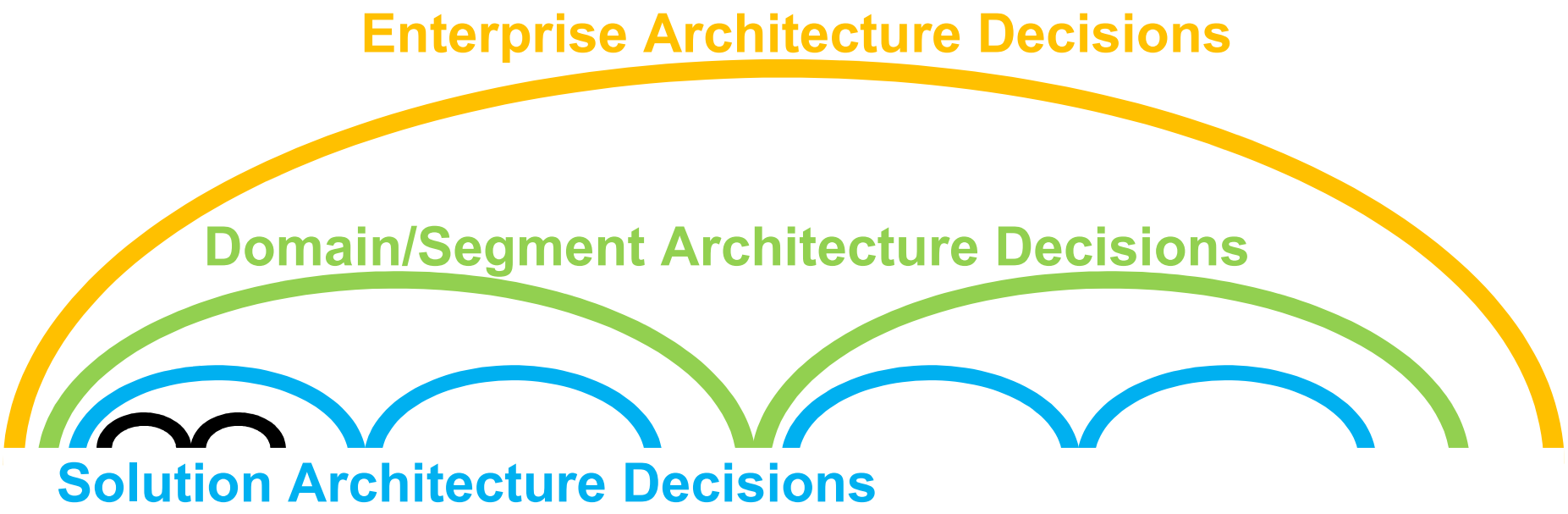
Need for Larger Context and Structured Approach



Different Architecture Levels – Decision Scopes

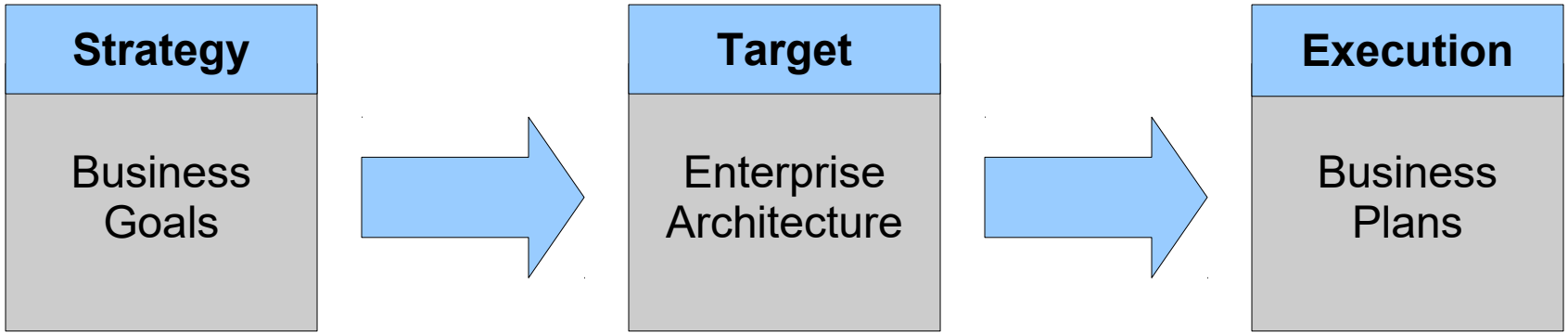
> The significant problems we face cannot be solved by the same level of thinking that created them
>
> Jeanne W. Ross

- Enterprise Architecture – **a holistic view** on whole enterprise, a description of the enterprise that provides a common understanding



Enterprise Architecture

- Architecture Levels
  - Software Architecture → Building Architecture → Design
  - Enterprise Architecture → City Planning → Governance
- Enterprise Architecture as a strategic planning tool – a bridge between Strategy and Execution



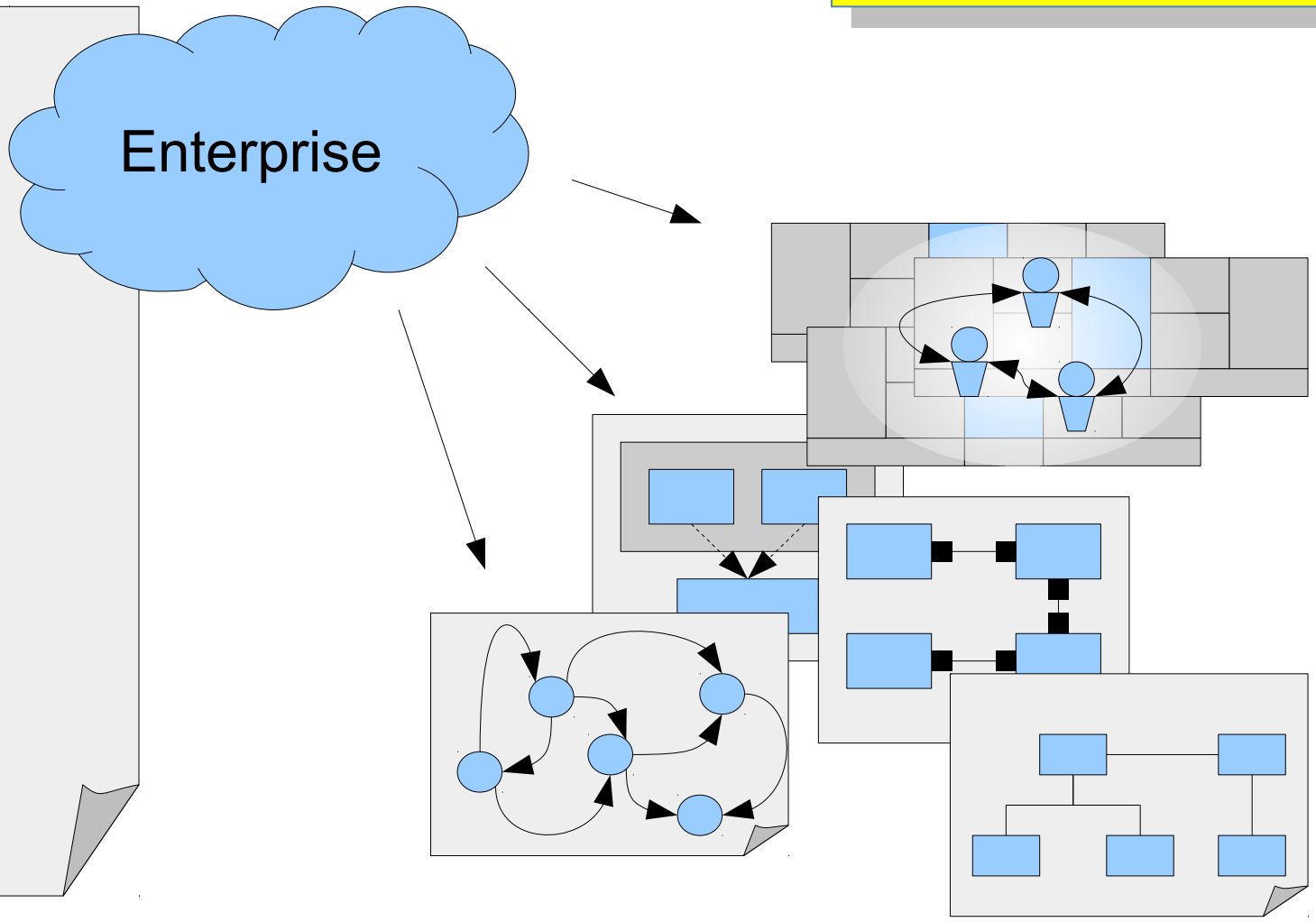
Enterprise as very Complex System
→ requires more Viewpoints

Enterprise is a typical System of Systems

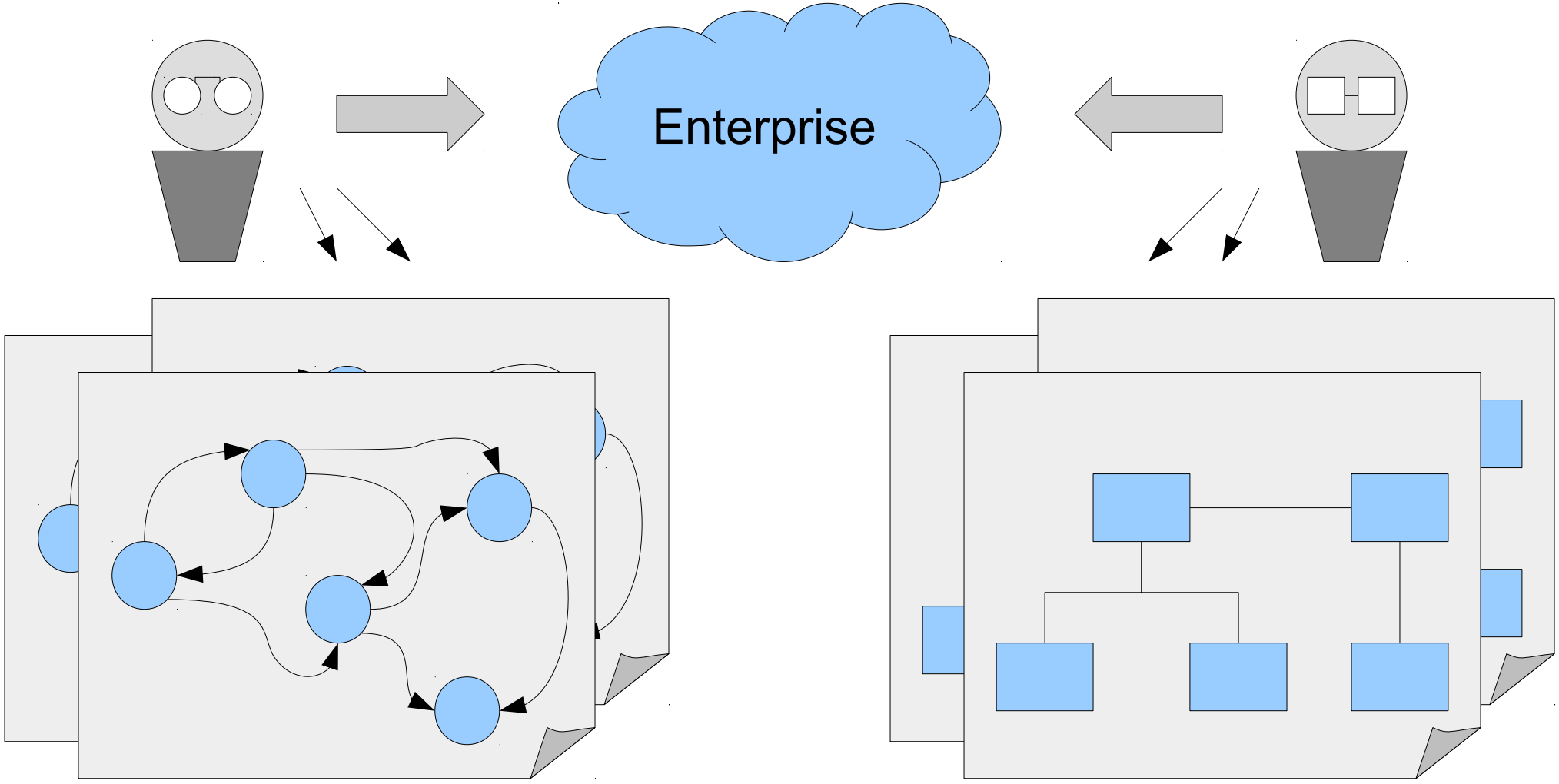
Many

- Business Models
- Customer Segments
- Channels
- Products/Services
- Business Processes
- Organizational Units
- IT Systems

- Connections
- (Conflicting) Interests
- … etc. ...



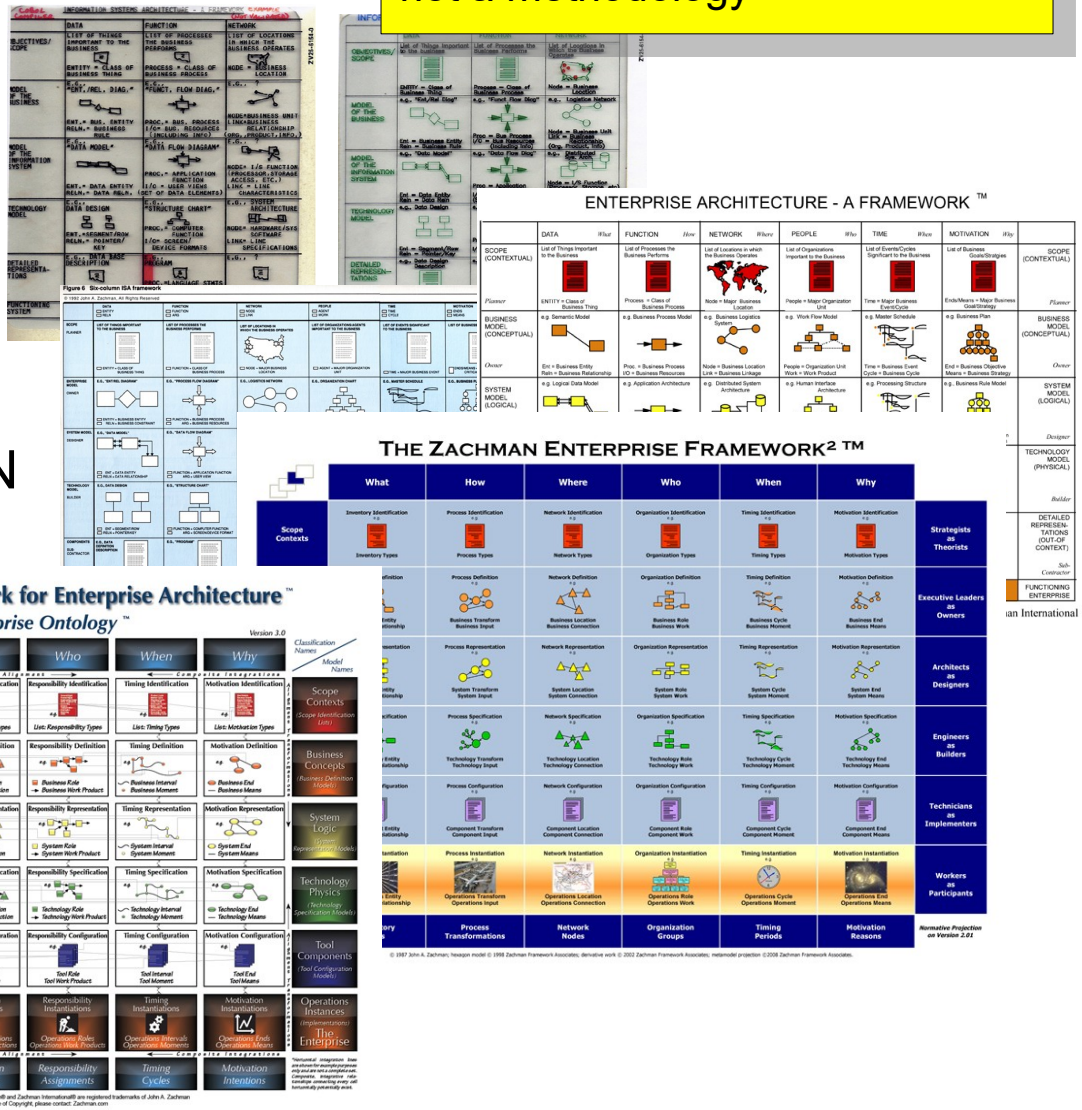
Enterprise Architecture Frameworks – ways to Look at Enterprise



Zachman Framework (1984 –)

Classification scheme (ontology), not a methodology

- 1984
  DATA – FUNCTION – NETWORK
- 1992 – 2004
  + PEOPLE – TIME – MOTIVATION
- 2011
  + ENTERPRISE



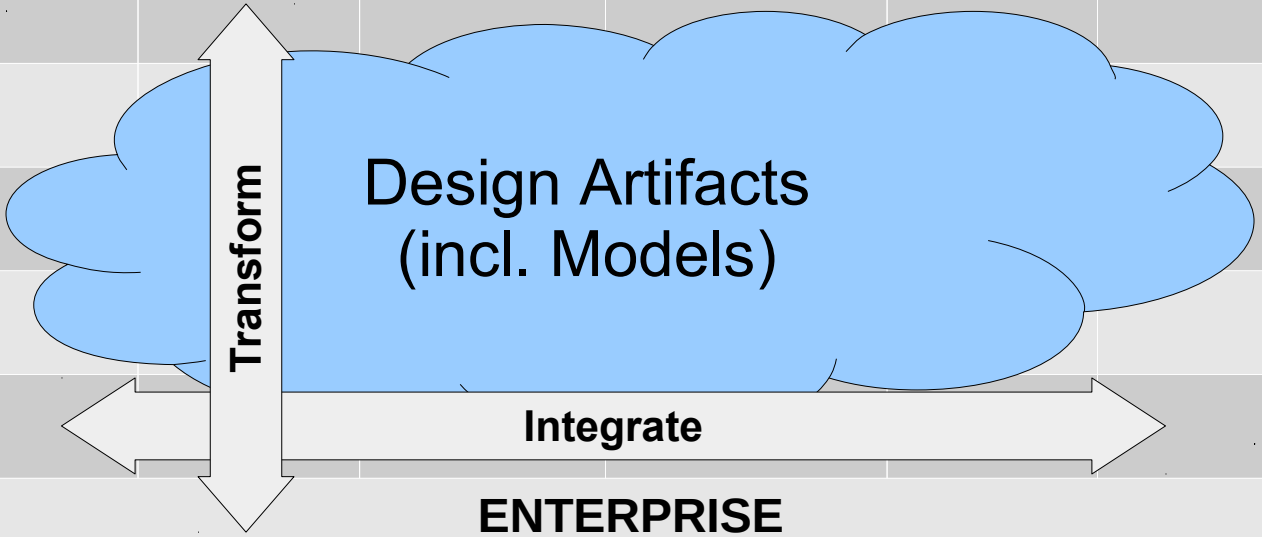
Pictures © J.A.Zachman, Zachman International

Zachman Framework

Classification scheme (ontology), not a methodology

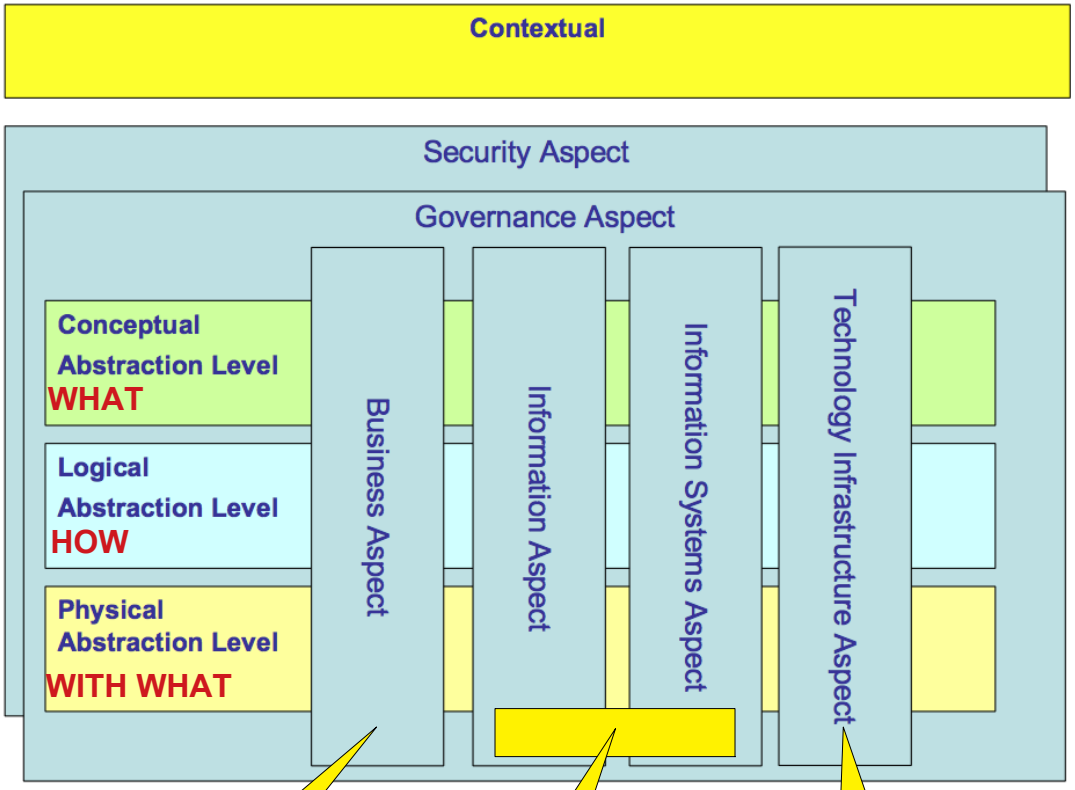
- A generic **classification scheme** (ontology or meta-model) for design artifacts, that is, descriptive representations of any complex object – to enable focused concentration on selected aspects of an object without losing a sense of the contextual, or holistic, perspective

Perspective	What	How	Where	Who	When	Why	
Planner (Executives)							Scope Context
Owner (Managers)							(Business) Concepts
Designer (Architects)							System Logic
Builder (Engineers)							Technology & Physics
Implementer (Technicians)							Tools & Components
Operator (Users)							Product & Operations
	Material (Inventory)	Process (Flows)	Geometry (Networks)	Instructions (Assignments)	Timing (Periods)	Objectives (Motivation)	

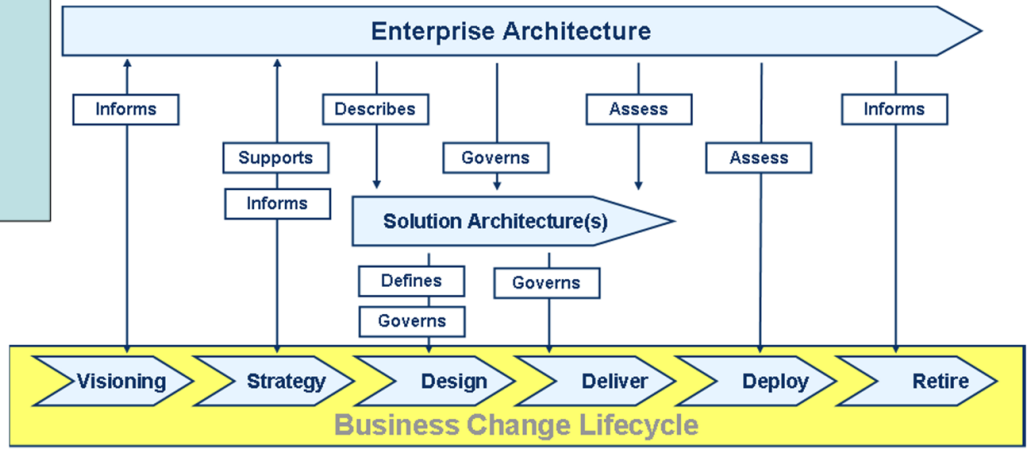


Capgemini Integrated Architecture Framework (IAF)

Support for Business Change,
Framework for Solution Architectures



- A comprehensive and coherent view across Business, Information, Systems and Technology, to deliver Business Change which may also be supported and enabled by IT
- This holistic view of the business through the use of architecture as the way that organizations think about and organize their business



Enterprise Architecture is a key participant in the business change lifecycle
Solution Architecture provides the detailed architecture definition to support the design and delivery of specific projects

Environment (Super-System)

System

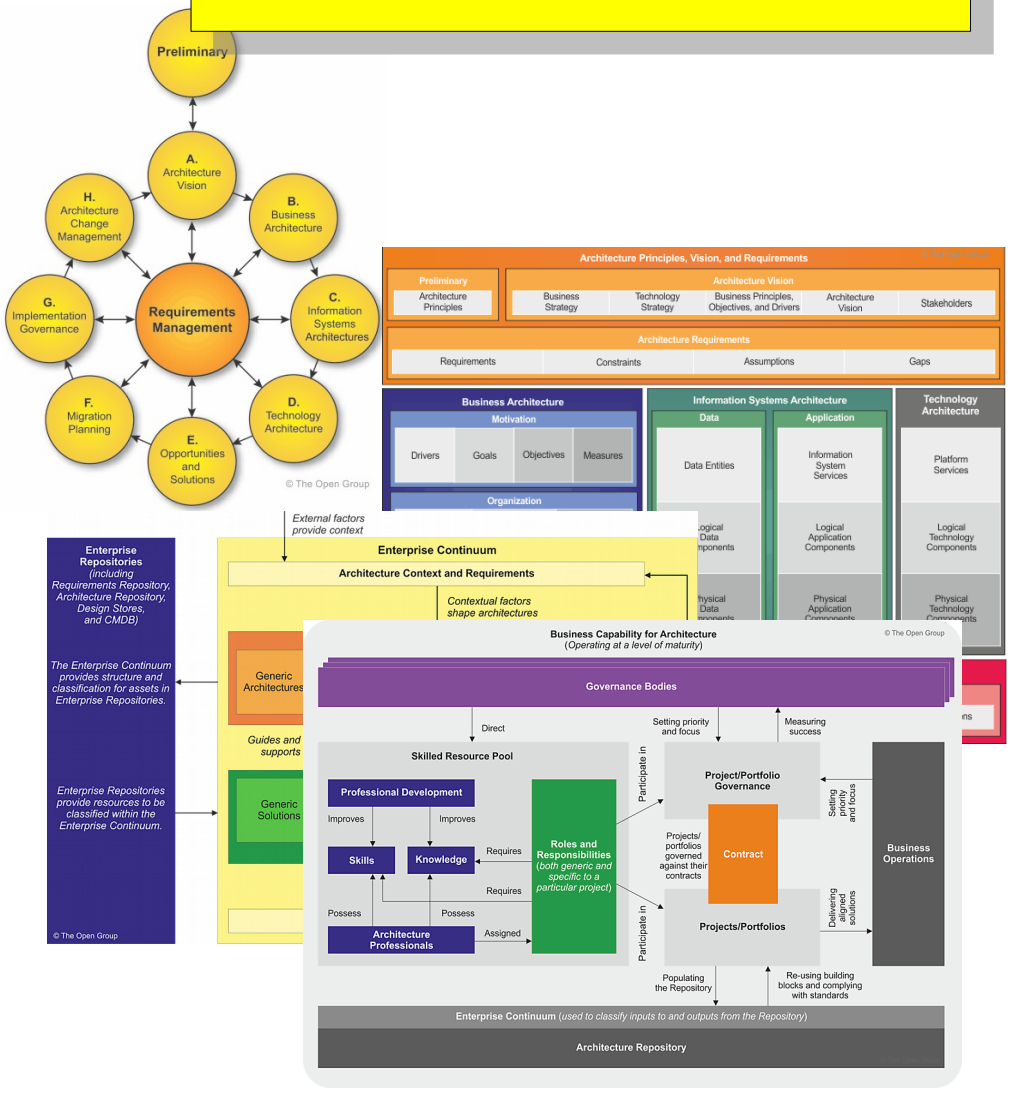
Environment (Infra-System)

Pictures © Capgemini

The Open Group Architecture Framework (TOGAF)

Framework for Enterprise Architecture

- Architecture Development Method (ADM)
  - the core of the TOGAF framework, describes a step-by-step approach to developing an Enterprise Architecture
- ADM Guidelines & Techniques
  - a collection of guidelines and techniques available for use in applying the TOGAF approach
- Architecture Content Framework
  - a structured metamodel for architectural artifacts, the use of re-usable Architecture Building Blocks (ABBs), and an overview of typical architecture deliverables
- Enterprise Continuum & Tools
  - discusses appropriate taxonomies and tools to categorize and store the outputs of architecture activity within an enterprise
- Architecture Capability Framework
  - discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise



Pictures © Open Group

The Open Group Architecture Framework (TOGAF)
Benefits of Enterprise Architecture

Benefits from better discipline and structure!

- More effective and efficient business operations
  - **lower** business operation and change management **costs**; **agility of the organization**; sharing of business capabilities; **improved business productivity**
- More effective and efficient Digital Transformation and IT operations
  - **lower** software development, support, and maintenance **costs**; increased portability of applications; improved interoperability and easier system and network management; **improved ability to address critical enterprise-wide issues** like security; easier upgrade and exchange of system components
- Better return on existing investment, reduced risk for future investment
  - **reduced complexity** in the business and IT; **maximum return on investment** in existing business and IT infrastructure; reduced risk overall in new investments and their cost of ownership
- Faster, simpler, and cheaper procurement
  - buying **decisions are simpler**; the ability to procure heterogeneous, multi-vendor open systems; the ability to secure more economic capabilities

MIT Sloan CISR
Benefits of Enterprise Architecture

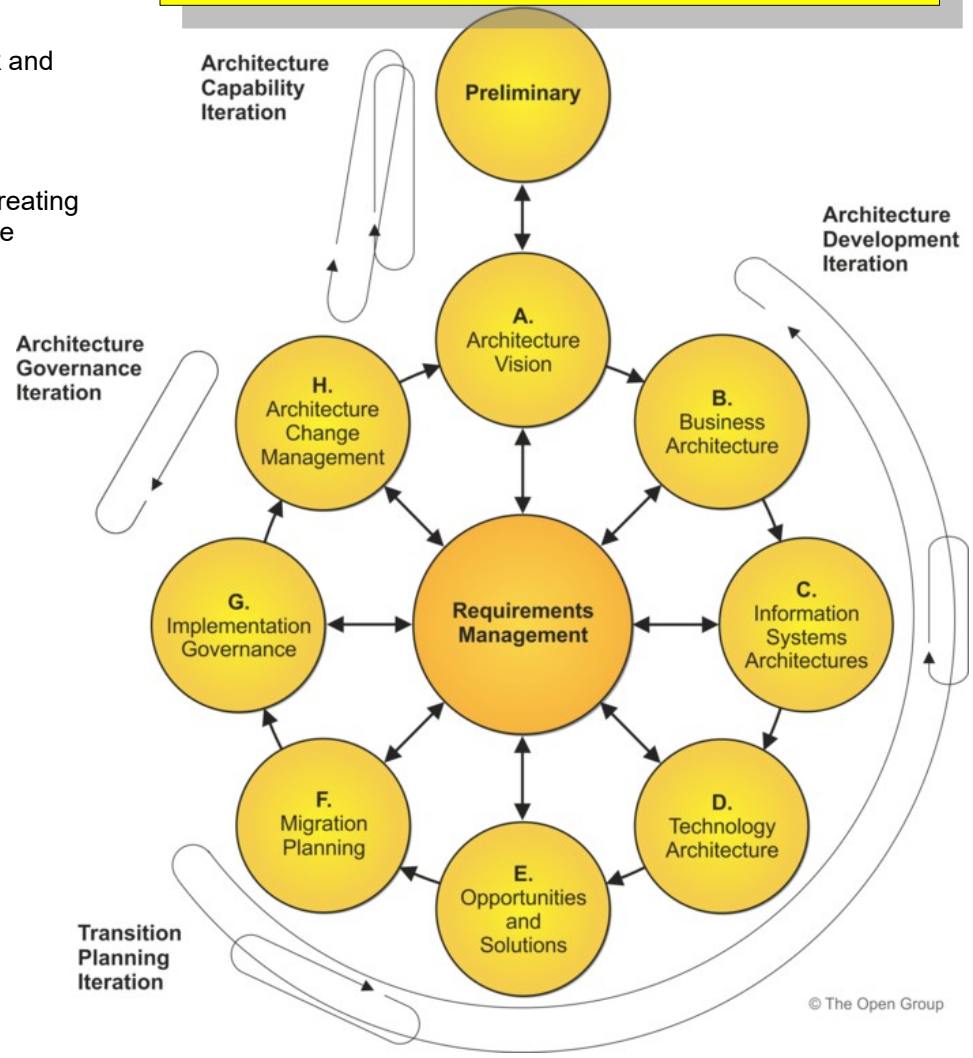
Benefits from better discipline and structure!

- Business related benefits
  - Strategic Business Impacts
    - **Operational excellence** – low cost, with efficient, reliable and predictable operations
    - **Customer intimacy** – extraordinary customer service, based on deep customer knowledge
    - **Product leadership** – first to market with innovative products and services, due rapid processes (e.g. market leader)
    - **Strategic agility** – ability to respond rapidly to competitor initiatives and new market opportunities
  - Managerial satisfaction with IT (the confidence of non-IT executives in the IT unit ability to provide value)
  - Shared Business Platforms
    - Data and process standardization when combined with integrating technologies results greater data sharing (accessibility of data) and integrated process standards (providing reliability and predictability)
- Technology related benefits
  - Reduced IT Costs – a non-value-adding variations in technologies are eliminated, reducing operations and maintenance costs
  - Better IT Responsiveness – with fewer technology choices less time is spent making technology decisions or addressing unexpected technical problems
  - Better Risk Management – through a more manageable IT environment
    - Reduced business risk (systems up and running as needed to support the business)
    - Improved regulatory compliance (accurate data to respond to government requirements is accessible)
    - Increased disaster tolerance (backup and recovery minimize business losses)
    - Reduced security breaches

The Open Group Architecture Framework (TOGAF) Architecture Development Method (ADM)

Framework for Enterprise Architecture

- The Preliminary Phase
  - creation of an Architecture Capability, customization of the TOGAF framework and definition of Architecture Principles
- Phase A: Architecture Vision
  - scoping the architecture development initiative, identifying the stakeholders, creating the Architecture Vision, and obtaining approval to proceed with the architecture development
- Phase B: Business Architecture
- Phase C: Information Systems Architectures
- Phase D: Technology Architecture
- Phase E: Opportunities & Solutions
  - implementation planning for the architecture defined in the previous phases
- Phase F: Migration Planning
  - a detailed plan for moving from the Baseline to the Target Architectures
- Phase G: Implementation Governance
  - architectural oversight of the implementation
- Phase H: Architecture Change Management
  - procedures for managing change to the new architecture
- Requirements Management
  - the process of managing architecture requirements throughout the ADM

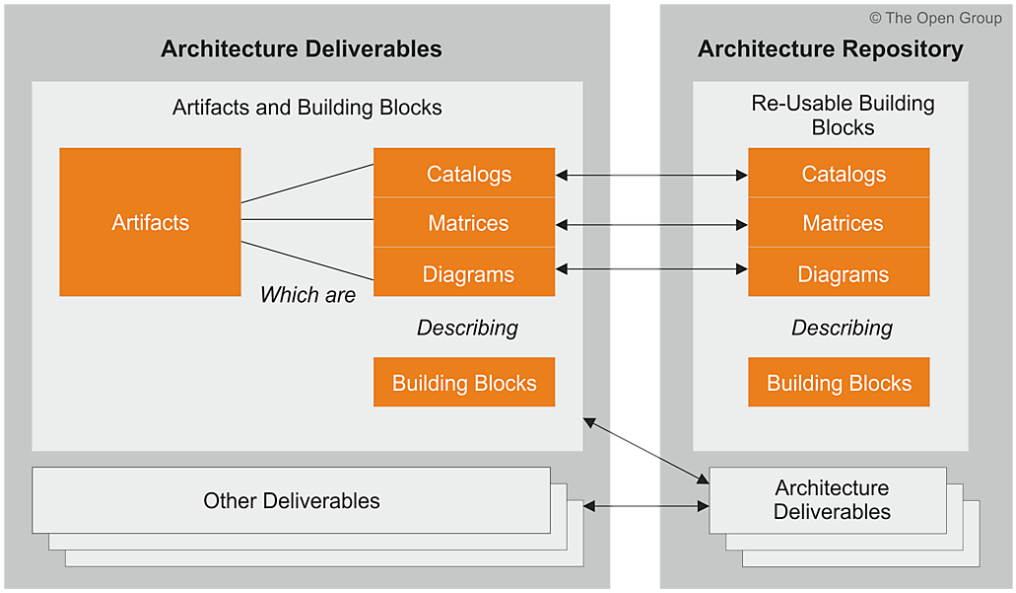


Pictures © Open Group

The Open Group Architecture Framework (TOGAF)
Architecture Content Framework

Framework for Enterprise Architecture

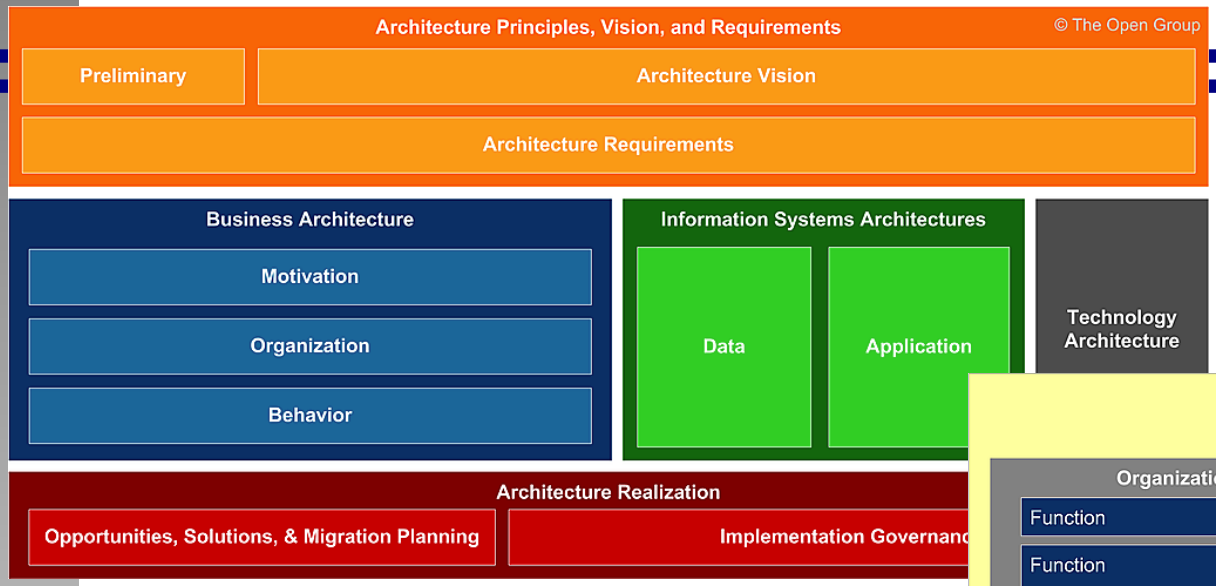
- Deliverable
  - a work product that is contractually specified and
  - in turn formally reviewed, agreed, and signed off
  - by the stakeholders
- Artifact
  - an architectural work product that describes
  - an aspect of the architecture
    - catalogs (lists of things)
    - matrices (showing relationships between things)
    - diagrams (pictures of things)



- Building block – a (potentially re-usable) component of enterprise capability that can be combined with other building blocks to deliver architectures and solutions
  - Architecture Building Blocks (ABBs) – describe required capability (e.g. a customer services capability)
  - Solution Building Blocks (SBBs) – components that will be used to implement the required capability (e.g. a network)

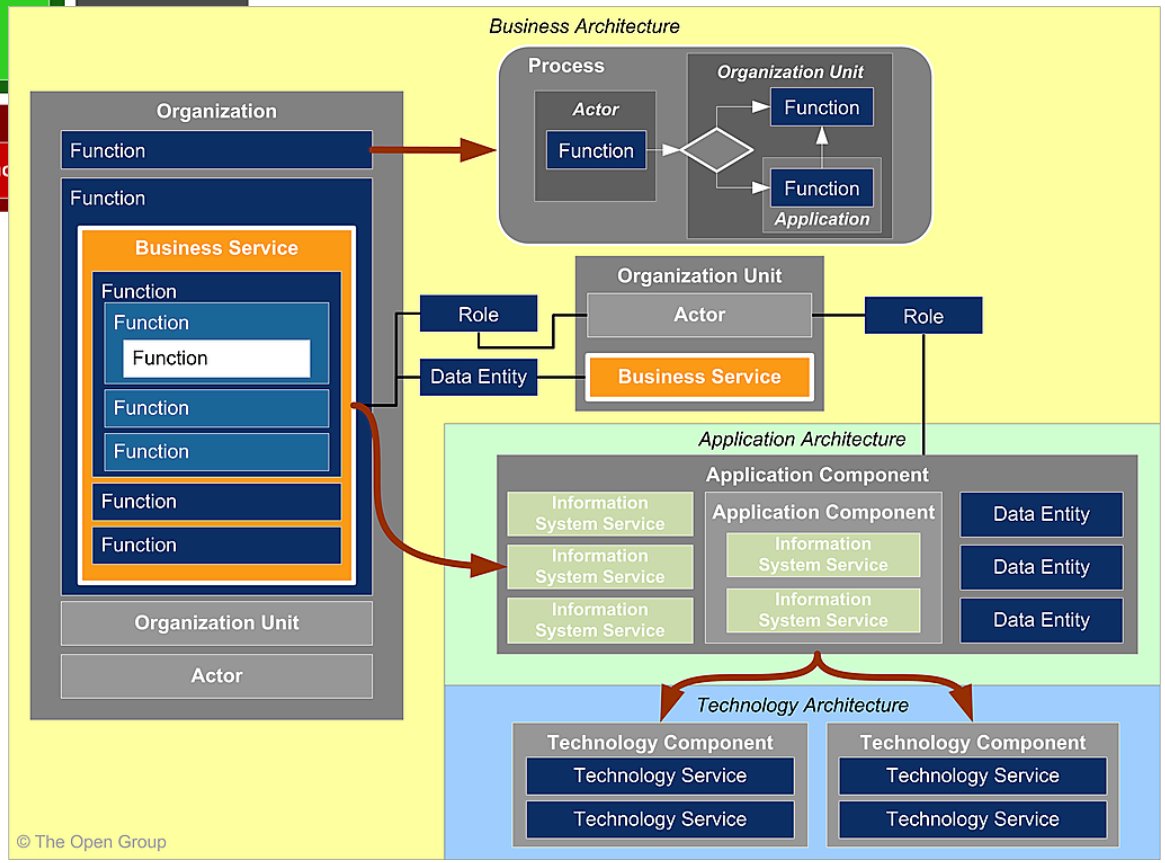
Pictures © Open Group

The Open Group Architecture Framework (TOGAF)
Architecture Content Framework



Framework for Enterprise Architecture

- The content meta-model defines a set of entities that allow architectural concepts to be captured, stored, filtered, queried, and represented in a way that supports consistency, completeness, and traceability

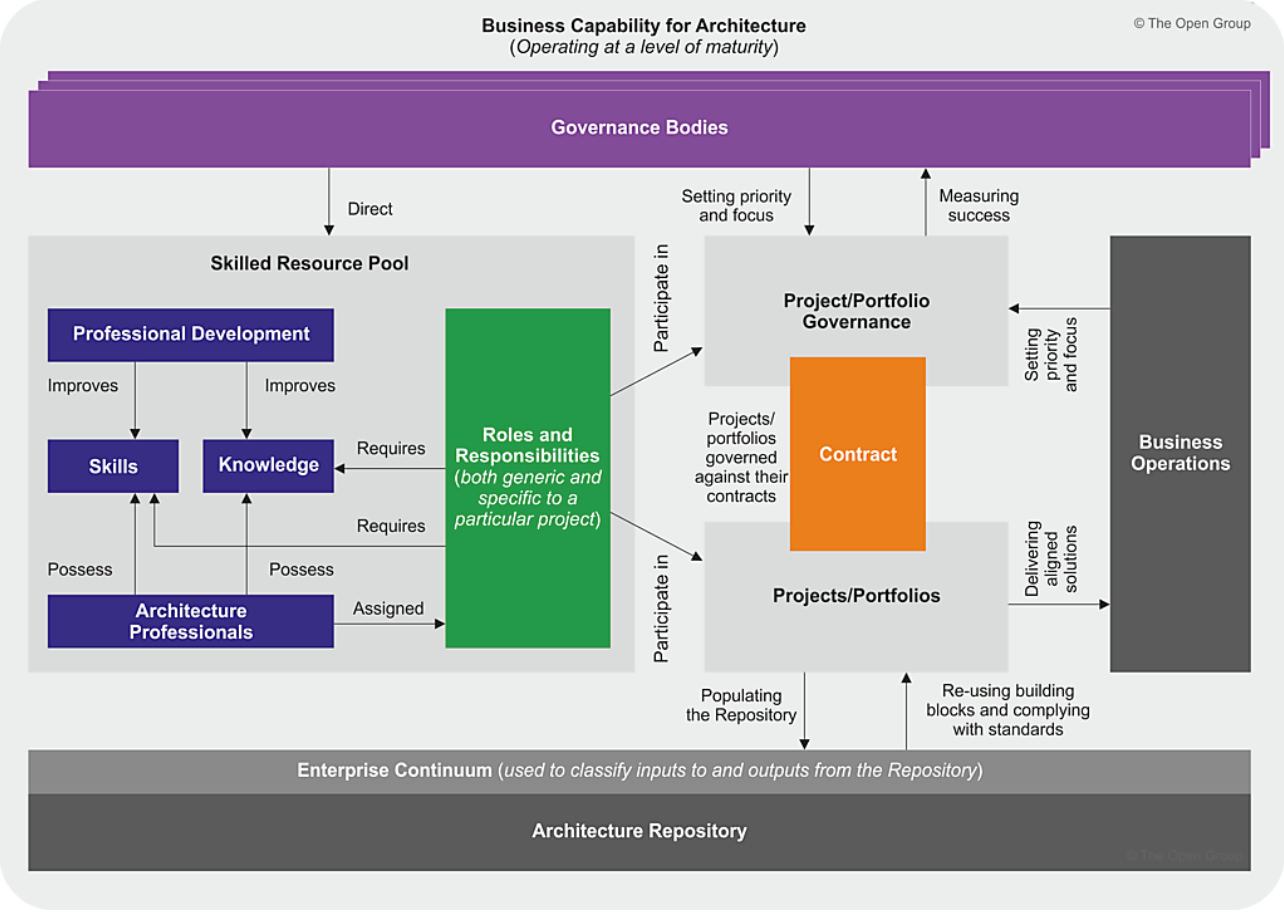


Pictures © Open Group

The Open Group Architecture Framework (TOGAF) Architecture Capability

Framework for Enterprise Architecture

- A set of reference materials for how to establish and operate architecture function (Architecture Capability) within an enterprise
  - appropriate organization structures
  - maturity models
  - processes (governance)
  - roles (bodies)
  - responsibilities
  - skills

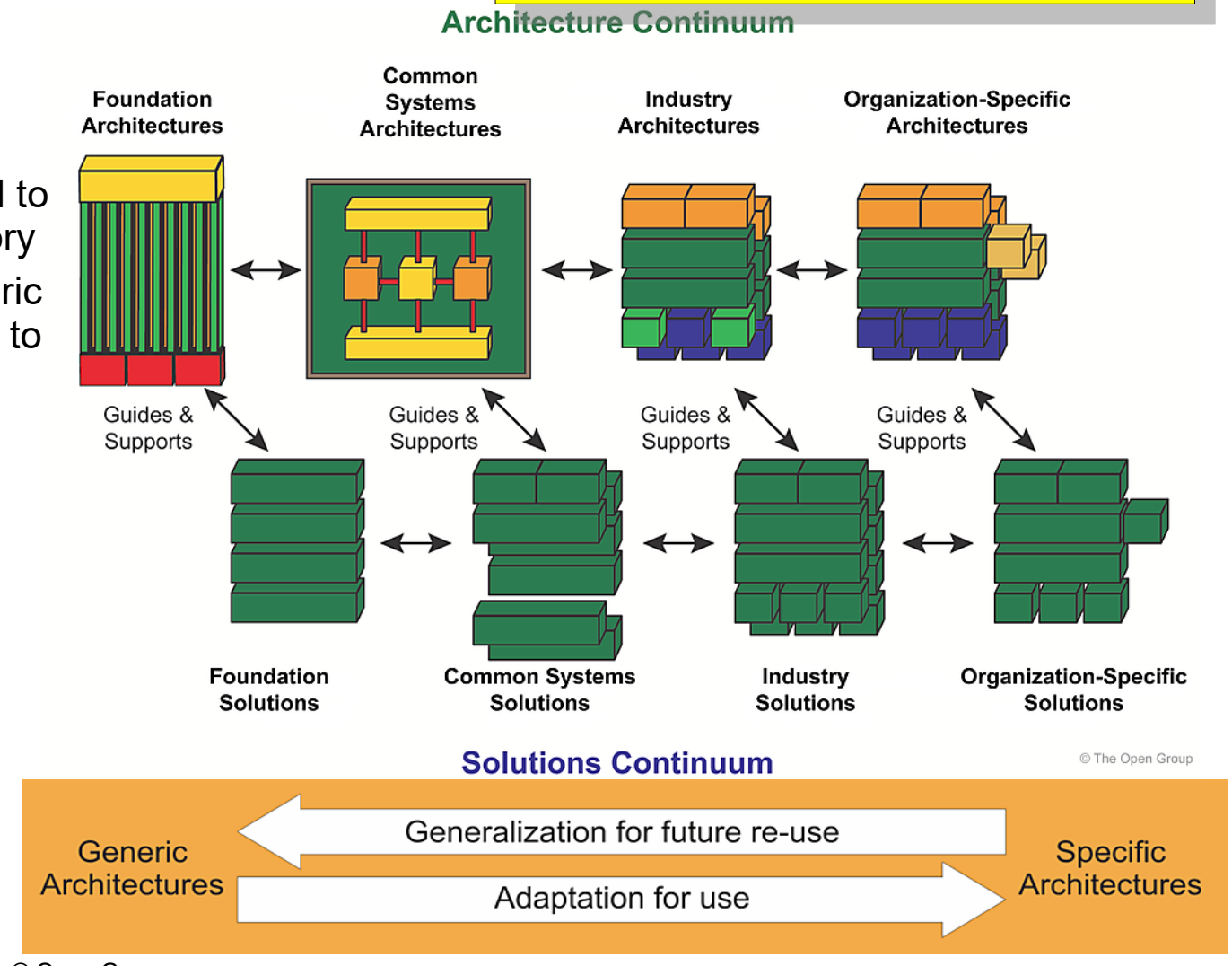


Pictures © Open Group

The Open Group Architecture Framework (TOGAF) Enterprise (Architecture) Continuum

Framework for Enterprise Architecture

- Methods for classifying architecture and solution artifacts
  - both internal and external to the Architecture Repository
  - as they evolve from generic Foundation Architectures to Organization-Specific Architectures



Pictures © Open Group

(Industry) Reference Architectures

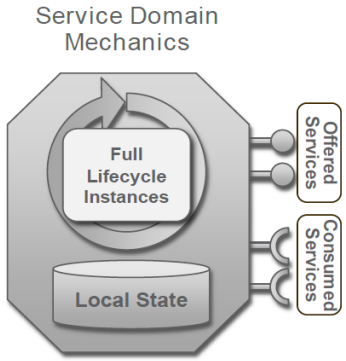
- Reference architectures
  - are standardized architectures that provide a frame of reference for a particular domain, sector or field of interest
  - provide a common vocabulary, reusable designs and industry best practices
  - are not solution architectures, i.e. they are not implemented directly

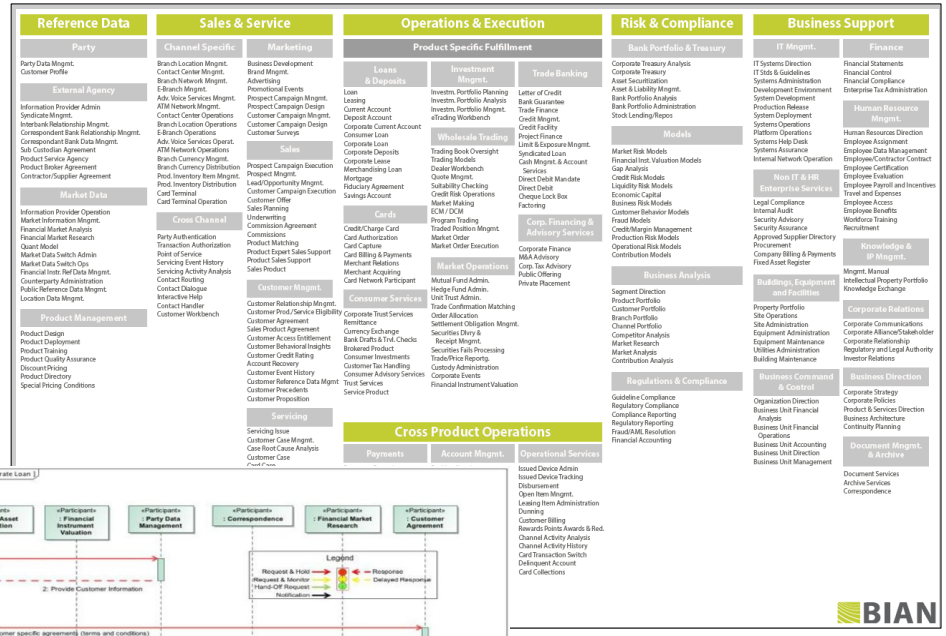
- Examples
  - Banking Industry Architecture Network (BIAN)
    - Service Landscape, Asset Categories, Business Scenarios, APIs, ...
  - Association for Cooperative Operations Research and Development (ACORD) – for Insurance Services
    - Business Glossary, Information Model, Data Model, Capability Model, Component Model, Process Fwk., Product Fwk.
  - TM Forum Frameworx (eTOM, SID, TAM, Open APIs) – for Communication Services
  - IBM Industry Models – for Banking and Financial Markets, Insurance, Healthcare, Telecommunications, Energy and Utilities, Retail
    - Glossaries, Object Models, Process Models, Service Models, Business Intelligence (Data Warehouse) Models, ...
  - Microsoft Industry Reference Architecture for Banking (MIRA-B)

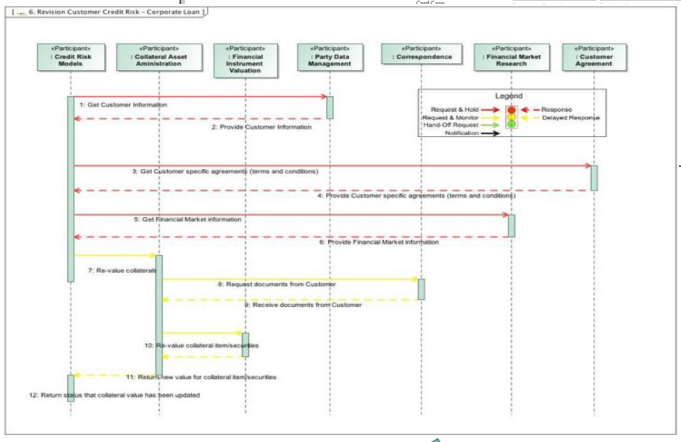
Example

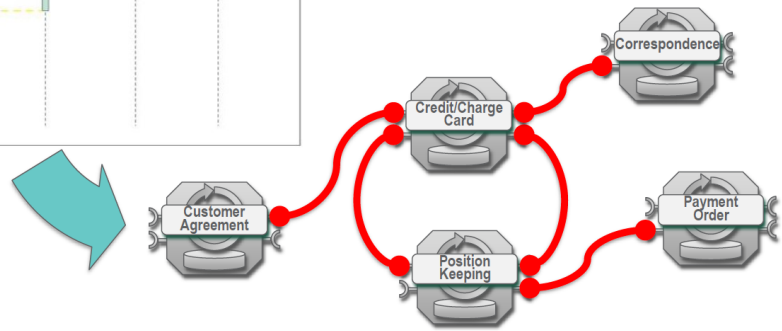
Banking Industry Architecture Network (BIAN)

- **Service Landscape**
  - **Service Domains**
- Building Blocks
  - Asset Types
  - Functional Patterns
  - ...
- Scenarios
- Wireframes
- APIs
- ...





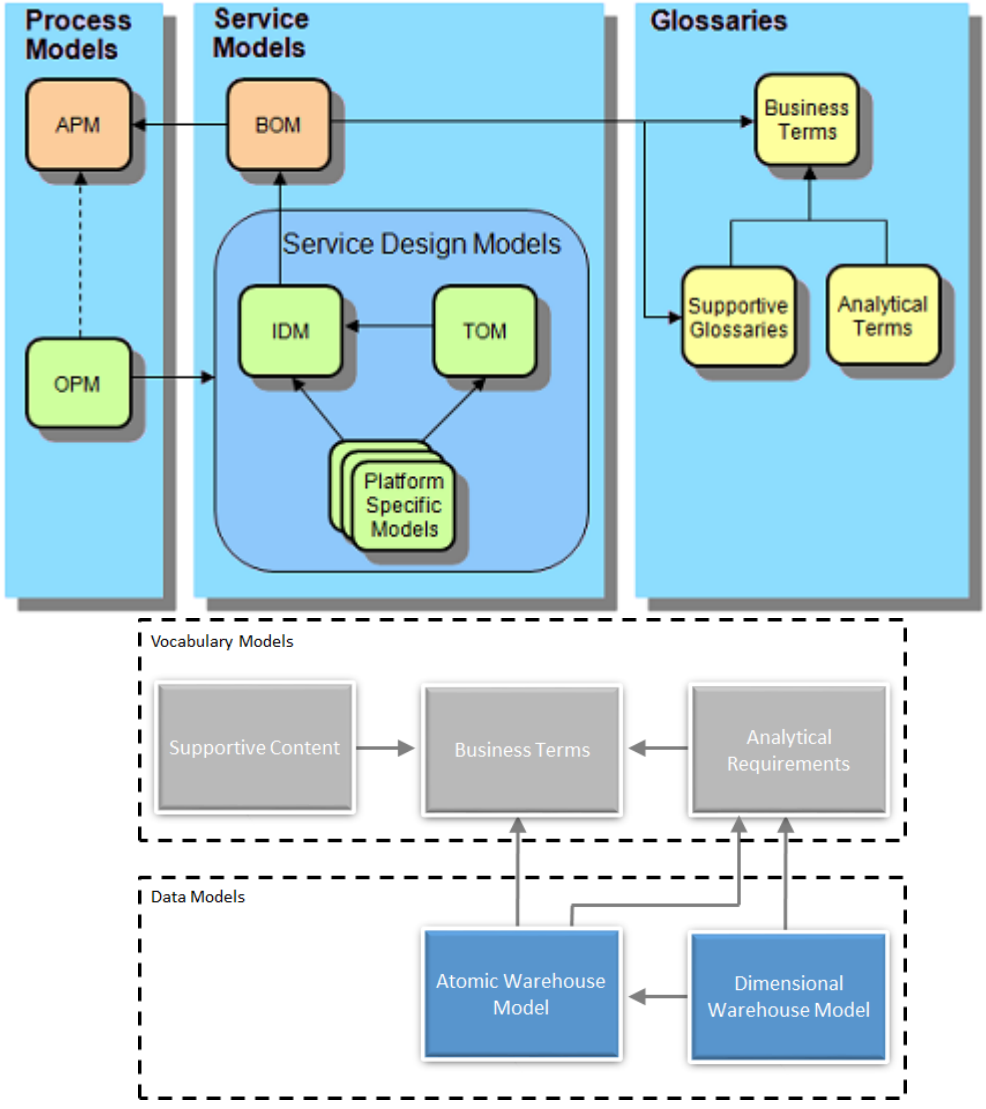




Example

IBM Industry Models for Banking and Financial Markets

- Foundation
  - Business Glossaries (Business Terms)
    - Supportive Content (regulations, standards)
    - Analytical Requirements
  - Business Object Model
- Process Models
  - Analysis Process Models
  - Orchestration Process Models
- Service Models
  - Interface Design Models
  - Transfer Object Models
  - Platform Specific Models (WSDL, REST, ...)
- Business Intelligence Models
  - Atomic Warehouse Model
  - Dimensional Warehouse Model

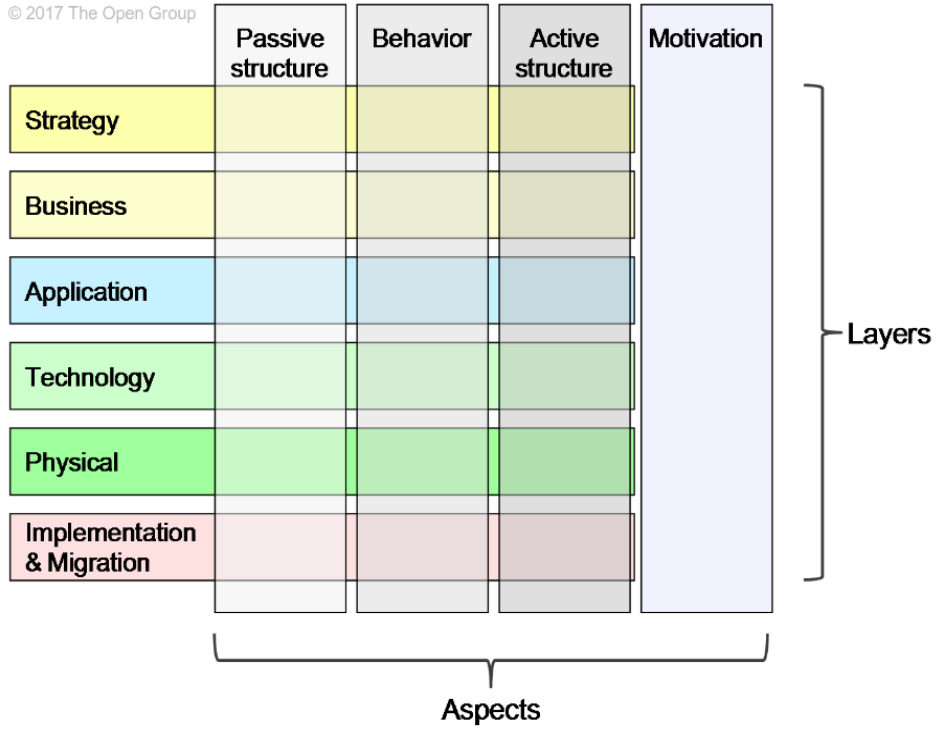


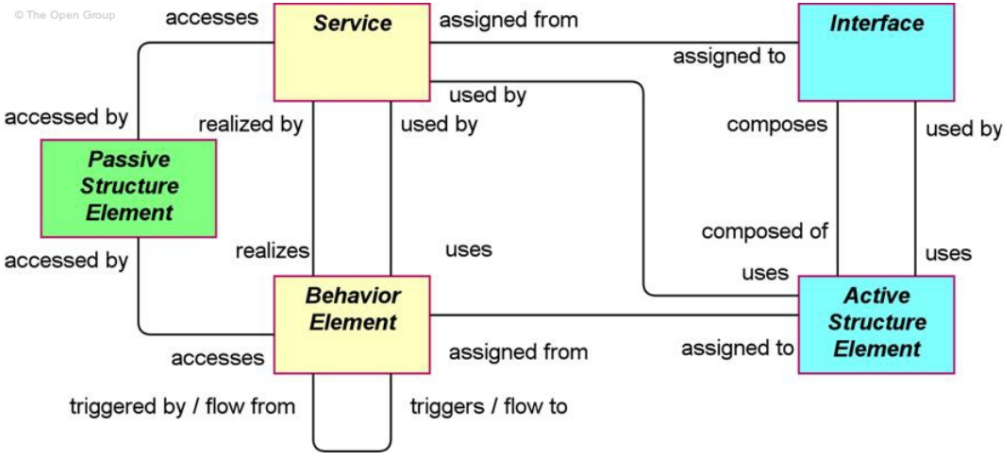
Using (Industry) Reference Architectures/Models

- As a coherent and structured set of knowledge about a certain domain
  - Learn the domain that has been described
  - Use it as a start for a common body of knowledge
- As an example/inspiration for creating own models
  - Lear the techniques used in modeling
  - Use similar structures in your models
- As a tool-box for taking pieces for own models
  - Use the fragments of models in your own developments
- As an input for automating the software development
  - Use the models as is and extend the models to customize for task at hand
  - Use the provided tooling to automate software development

ArchiMate – Language for Enterprise Architecture

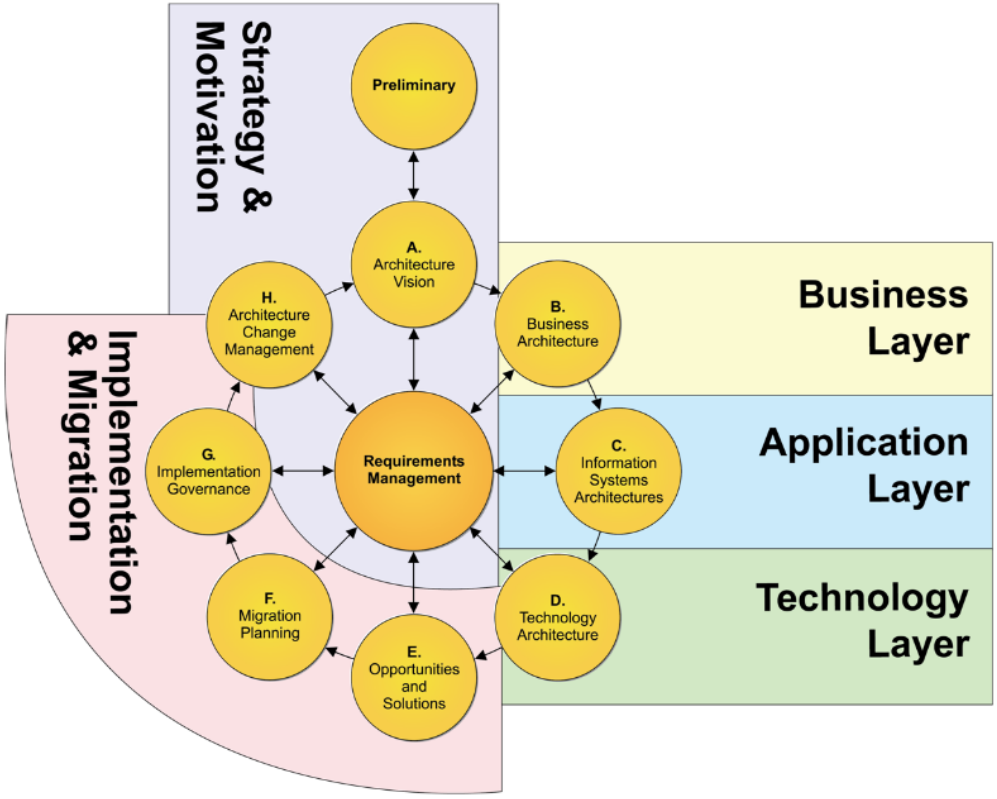
- Layers – levels at which an enterprise can be modeled
- Aspects
  - *Active Structure* – the structural elements that display actual behavior (i.e., the "subjects" of activity)
  - *Behavior* – the behavior (processes, functions, events, and services) performed by the active structural elements
  - *Passive Structure* – the objects on which behavior is performed (usually information objects and data objects, but may also be physical objects)

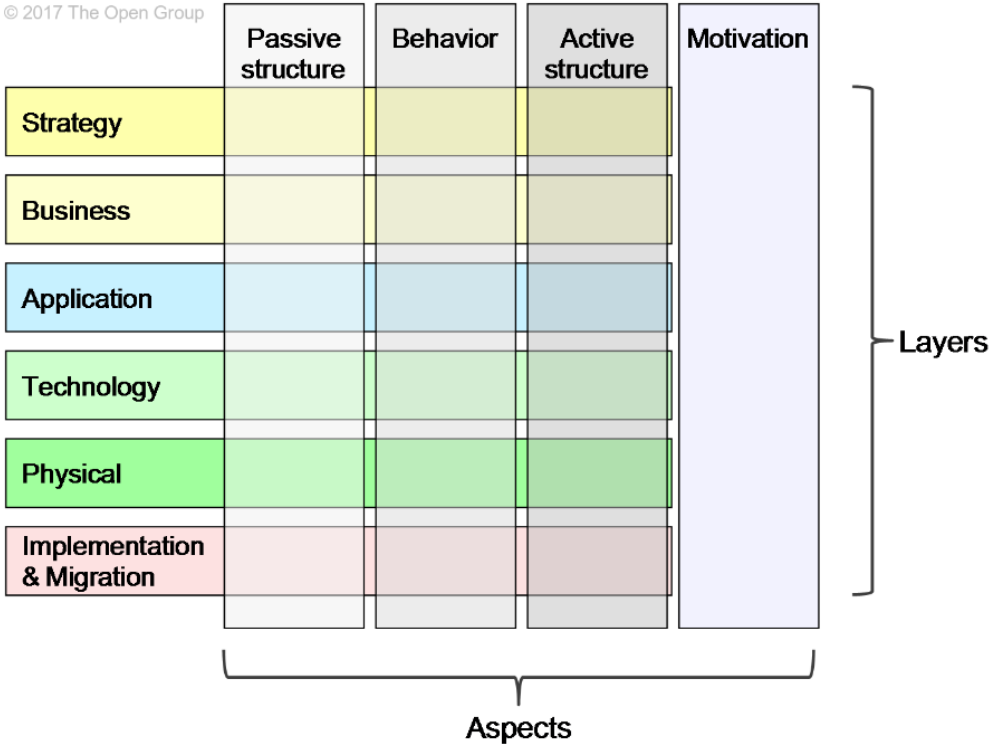




Pictures © Open Group

ArchiMate & TOGAF

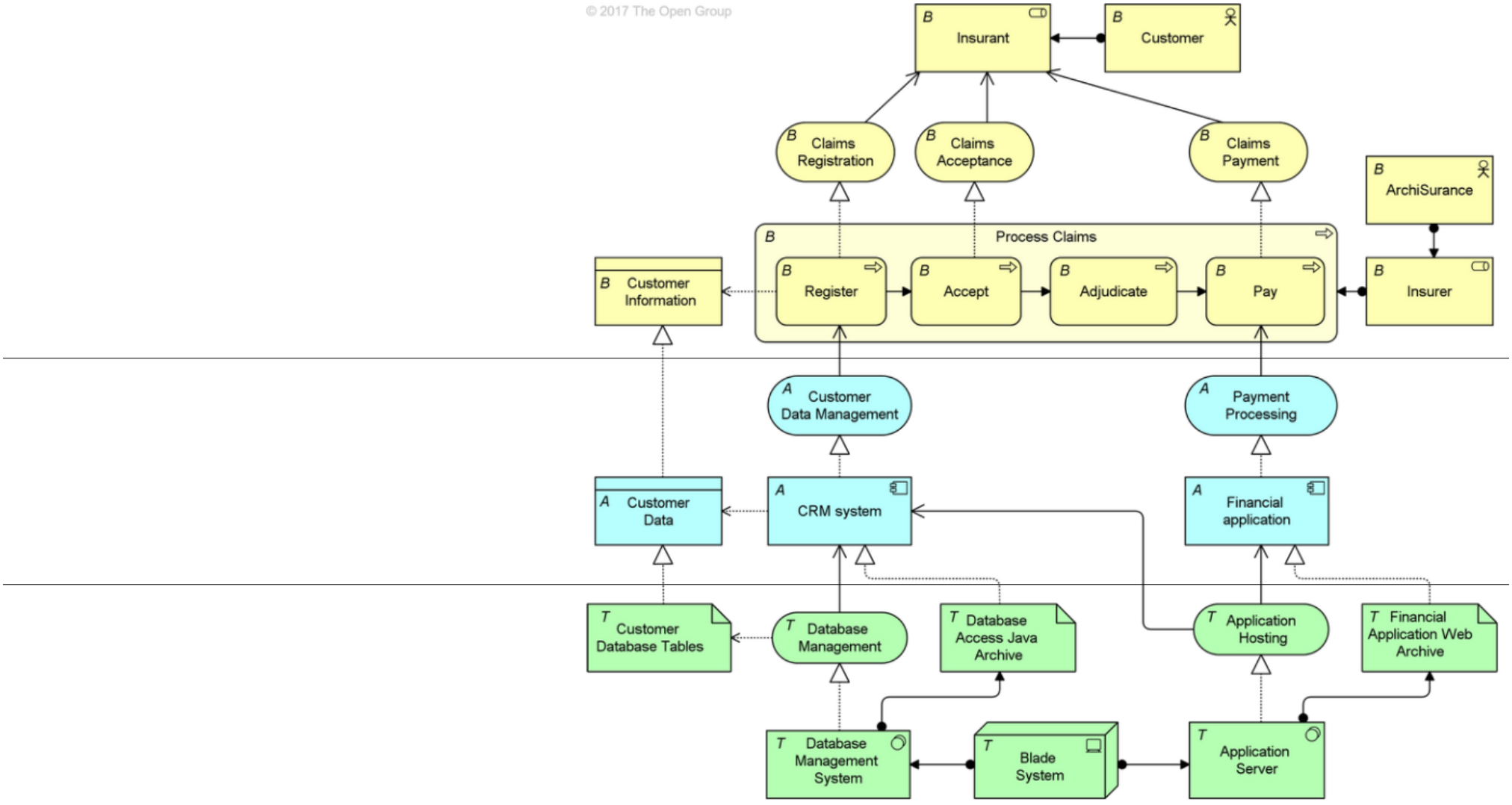




Pictures © Open Group

Example

ArchiMate – Connection between Layers



Pictures © Open Group

Content

Strategy without tactics is the slowest route to victory
Tactics without strategy is the noise before defeat

Sun Tzu

- Larger context
  - Hierarchy of Systems
  - System of Systems
- Enterprise Architecture
  - Need for Larger Context and Structured Approach
  - Some Approaches to Enterprise Architecture
  - Standard for Enterprise Architecture – TOGAF
  - (Industry) Reference Architectures
  - Language for Enterprise Architecture – ArchiMate
- Conclusions

Conclusions

Strategy without tactics is the slowest route to victory Tactics without strategy is the noise before defeat

Sun Tzu

- It is important to **understand the larger context**
  - what is the super-system and how it changes
  - what are the other (peer) systems in the super-system
  - what are the constraints that super-system imposes
- System of Systems
  - a collection of systems, that pool their resources and capabilities
  - often with separate management and authority
  - **offer more** functionality and performance **than simply the sum of the constituent systems**
- Enterprise is a complex system – a System of Systems
- **Isolate your system** from the environment/context – build for
  - *interoperability* – reconcile differences between interfaces
  - *change* – detect changes and adapt to it
  - *failure* and *unexpected* – detect failures and recover from these
  - *security* – detect threats and neutralize these

Conclusions

Strategy without tactics is the slowest route to victory Tactics without strategy is the noise before defeat

Sun Tzu

- When architecting System of Systems,
  - concentrate on *interfaces* (build platforms)
  - provide *collaboration incentives*
  - design so that value can be delivered even by incomplete System of Systems
- When analyzing/designing a complex system
  - ask WHY, WHAT, HOW, WHO, WHEN, WHERE
  - describe the answers for different stakeholders (interested parties)
- Enterprise Architecture is a
  - **holistic view** on whole enterprise – a description of the enterprise to provides a common understanding
  - **strategic planning tool** – a bridge between Strategy and Execution
- Reference architectures provide for a particular domain, a common vocabulary, reusable designs and industry best practices

12. The architect observes the world
but trusts his inner vision.
He allows things to come and go.
His heart is open as the sky.

Lao Tsu (by Philippe Kruchten)

Thank You!

Questions

- Why is important to understand and know the context of the system?
- What kind of Systems of Systems there are?
- What are the properties of Systems of Systems?
- What principles should be followed when architecting system of systems?
- What is the usual context for the IT systems in an enterprise?
- What is the difference of enterprise architecture from the software architecture?
- What Zachman Framework provides?
- How Zachman Framework organizes architecture descriptions(s)?
- What are the common viewpoints in Enterprise Architecture frameworks?
- What are the main parts of TOGAF?
- What are the stages of TOGAF ADM?
- What is enterprise continuum?
- What is reference architecture?
- How to use (industry) reference architectures?
- What are the main aspects of ArchiMate?

Literature

- https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=506213
-
- http://iansommerville.com/software-engineering-book/files/2015/08/Ch-20-Systems-of-systems.pdf
-
- https://strategyzer.com/canvas/business-model-canvas
-
- https://www.zachman.com/about-the-zachman-framework
- https://www.zachman.com/ea-articles-reference/54-the-zachman-framework-evolution
  https://agilearchitect.azurewebsites.net/useful_material/iaf/
- http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html
- http://pubs.opengroup.org/architecture/archimate3-doc/
-
- https://bian.org/
- https://acord.org/
- https://www.tmforum.org/tm-forum-frameworx-2/
- https://news.microsoft.com/download/presskits/msfinancial/docs/MIRAB.pdf
-
- … Google “enterprise architecture” ...

John Boyd's OODA Loop