

IT

# Architecture



June 04, 2024  
Lionel Pilorget

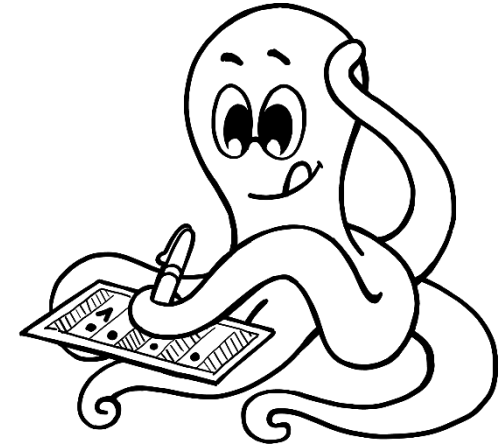


**KNOW***Digital*



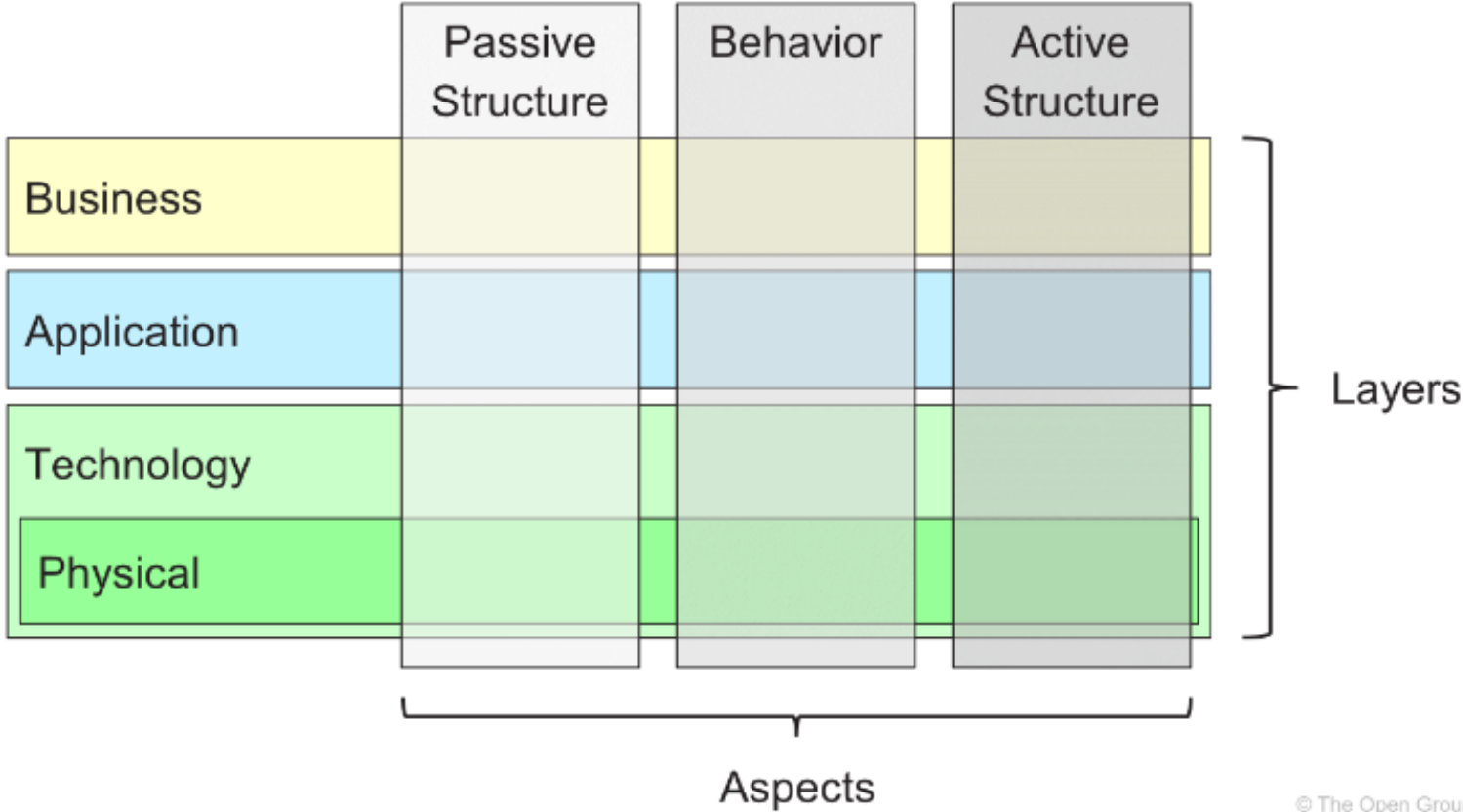


- ArchiMate in a Nutshell
- Modeling an IT Architecture
- IT Architecture Management
- Enterprise Architecture Frameworks
- Futuristic IT Architectures





**ArchiMate Enterprise Architecture modeling language**  
=> visual and graphical language with a set of default  
iconography for the representation of Architecture  
Descriptions

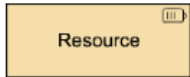





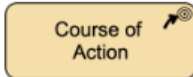



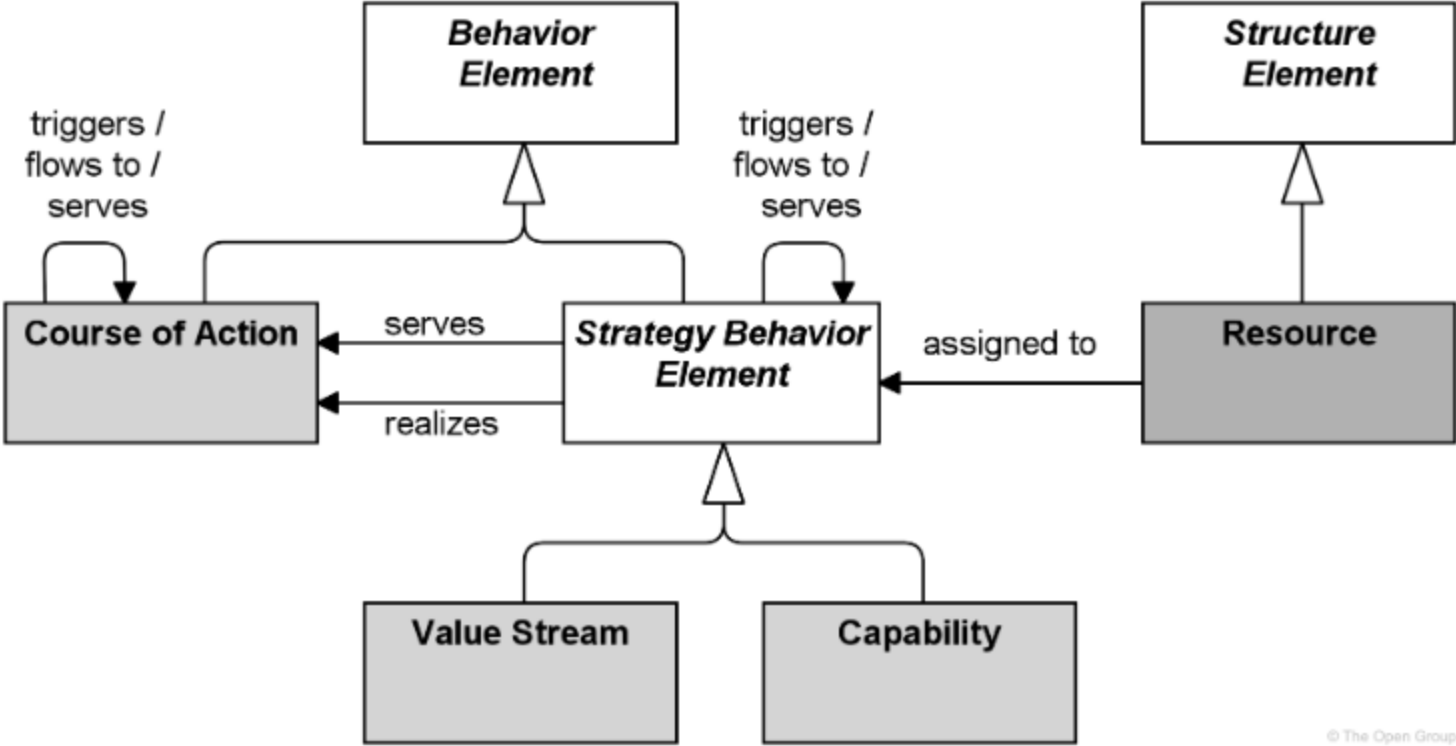
© The Open Group



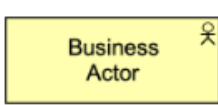



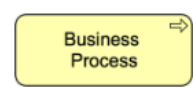
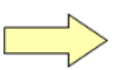




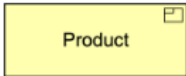
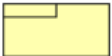
Element	Definition
Business Layer	Business services offered to customers, which are realized in the organization by business processes performed by business actors
Application Layer	Application services that support the business, and the applications that realize them
Technology Layer	Processing, storage, and communication technology in support of the application world and Business Layers, and operational or physical technology with facilities, physical equipment, materials, and distribution networks
Active Structure Aspect	Structural elements like business actors, application components, and devices that display actual behavior; i.e., the “subjects” of activity
Behavior Aspect	Behavior (processes, functions, events, and services) performed by the actors
Passive Structure Aspect	The objects on which behavior is performed; these are usually information objects in the Business Layer and data objects in the Application Layer, but they may also be used to represent physical objects



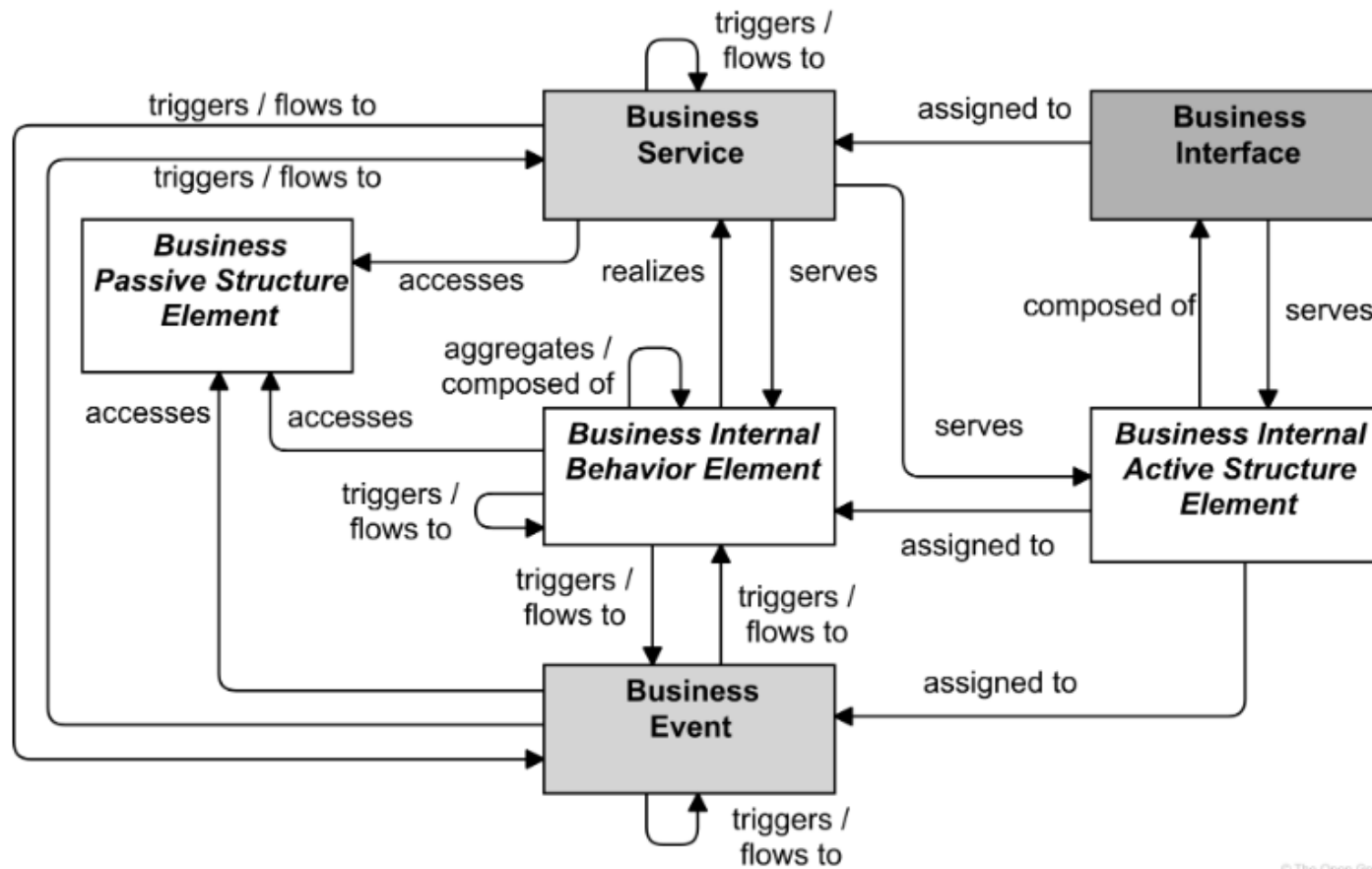
Element	Description	Notation
Resource	Represents an asset owned or controlled by an individual or organization	 
Capability	Represents an ability that an active structure element, such as an organization, person, or system, possesses	 
Value Stream	Represents a sequence of activities that create an overall result for a customer, stakeholder, or end user	 
Course of Action	Represents an approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal	 




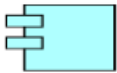

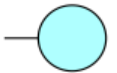

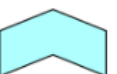
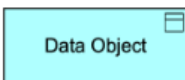
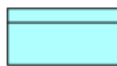


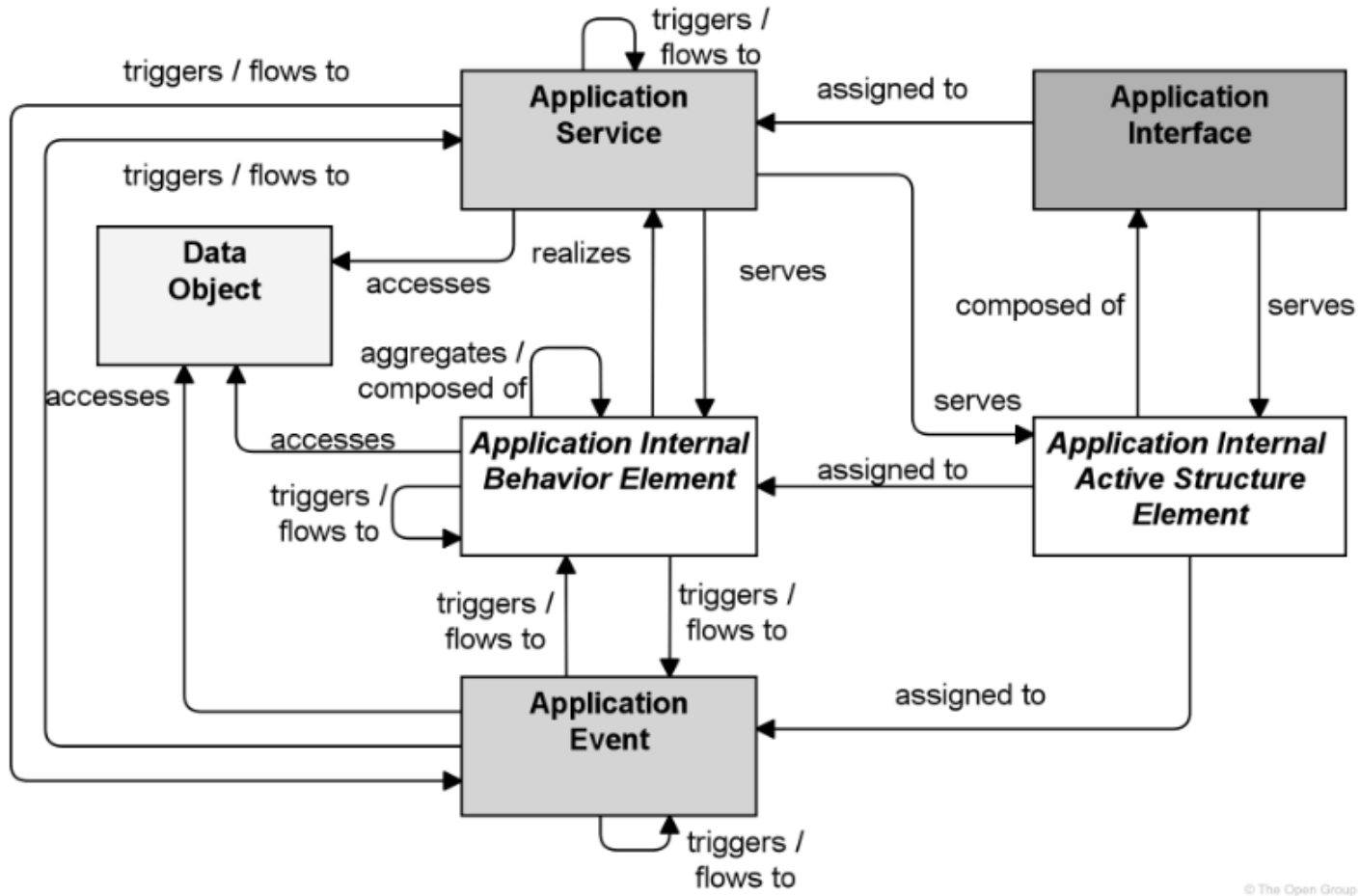
Element	Description	Notation	
Business Actor	Represents a business entity that is capable of performing behavior		
Business Role	Represents the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event		
Business Process	Represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services		
Business Event	Represents a business-related state change		
Business Service	Represents explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment		
Product	Represents a coherent collection of services and/or passive structure elements, accompanied by a contract, which is offered as a whole to (internal or external) customers		





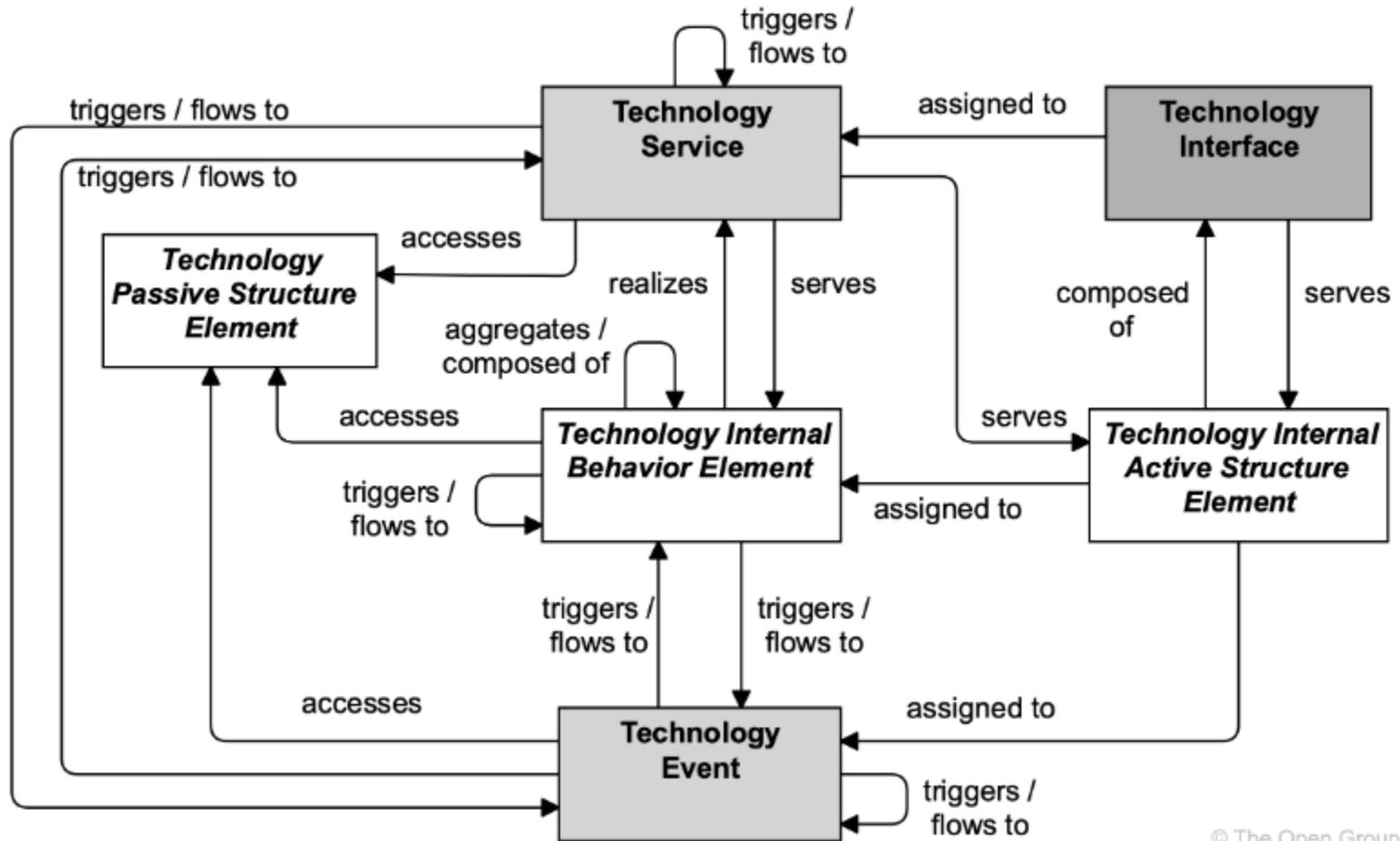


Element	Description	Notation
Application Component	Represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable	 
Application Interface	Represents a point of access where application services are made available to a user, another application component, or a node	 
Application Function	Represents automated behavior that can be performed by an application component	 
Data Object	Represents data structured for automated processing	 





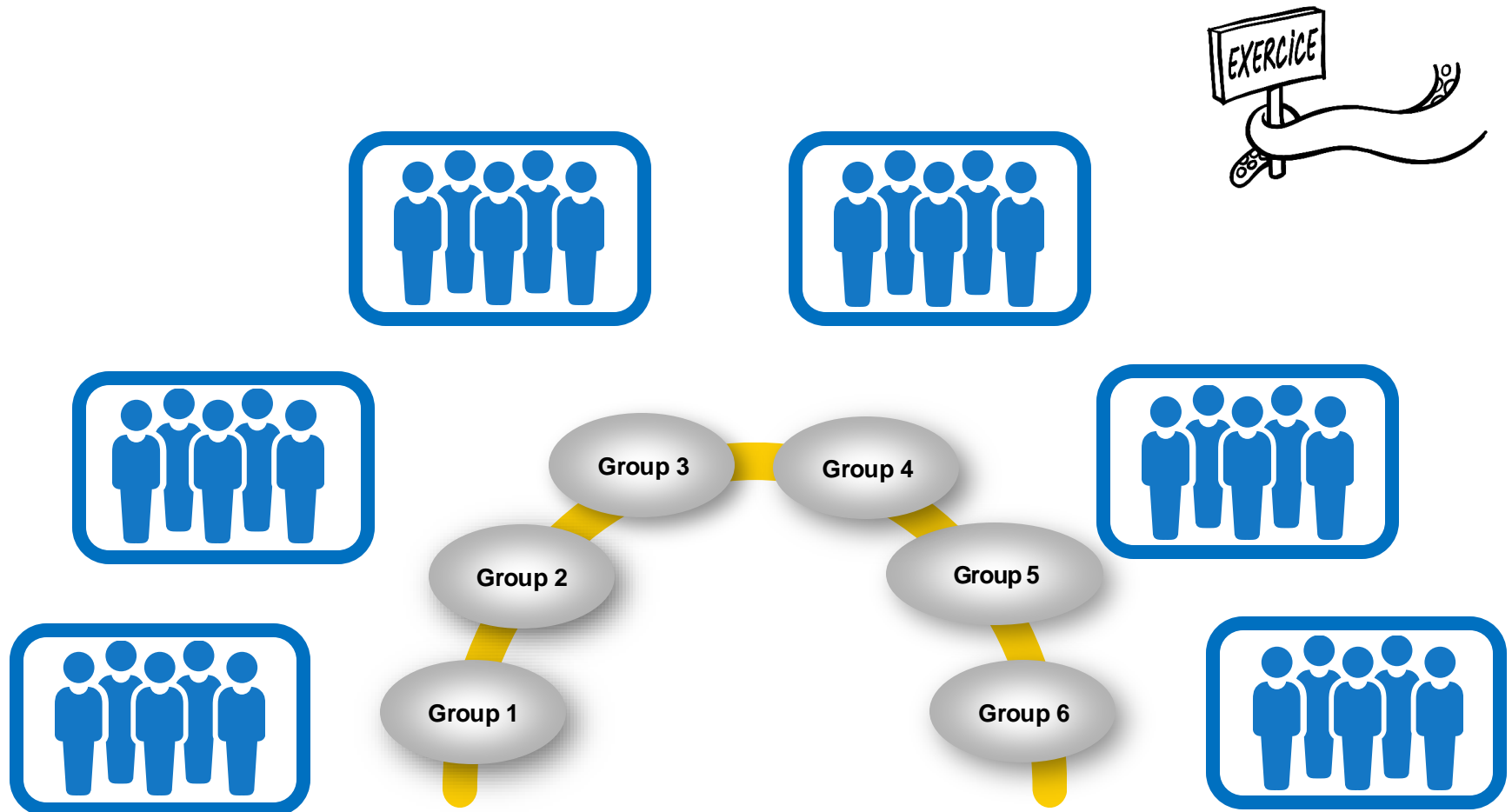
Element	Description	Notation	
Node	Represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources		
Device	Represents a physical IT resource upon which system software and artifacts may be stored or deployed for execution		
System Software	Represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it		
Path	Represents a link between two or more technology internal active structure elements, through which these elements can exchange data, energy, or material		
Artifact	Represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system		
Material	Represents tangible physical matter or energy		



# Describe the Architecture of the following systems



1. Library Management System
2. Online Shopping Website
3. Student Enrollment System
4. Travel Booking Application
5. Employee Performance Management System
6. Online Food Delivery Service





System	Questions
<b>Library Management System</b>	<ul style="list-style-type: none"><li>• Identify the business processes such as borrowing, returning, and cataloging books</li><li>• Model the business actors (e.g., librarian, borrower) and their interactions</li><li>• Specify the application components (e.g., library database, user interface) supporting these processes</li><li>• Represent the underlying infrastructure elements (e.g., server, database server) required to host the application components</li></ul>
<b>Online Shopping Website</b>	<ul style="list-style-type: none"><li>• Define the business functions such as browsing products, adding to cart, and checkout</li><li>• Model the business roles (e.g., customer, administrator) and their responsibilities</li><li>• Specify the application services (e.g., product catalog service, payment service) facilitating these functions</li><li>• Represent the technology components (e.g., web server, database) supporting the application services</li></ul>
<b>Student Enrollment System</b>	<ul style="list-style-type: none"><li>• Identify the business processes such as student registration, course enrollment, and grade submission</li><li>• Model the business objects (e.g., student, course, grade) and their relationships</li><li>• Specify the application components (e.g., student portal, course management system) managing these objects</li><li>• Represent the infrastructure nodes (e.g., application server, database server) hosting the application components</li></ul>

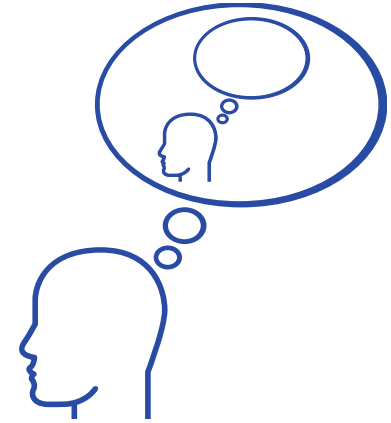


System	Questions
<b>Travel Booking Application</b>	<ul style="list-style-type: none"><li>• Define the business interactions such as search for flights, hotel reservations, and car rentals</li><li>• Model the business services (e.g., flight booking service, hotel booking service) facilitating these interactions</li><li>• Specify the application components (e.g., booking engine, payment gateway) supporting the business services</li><li>• Represent the technology nodes (e.g., web server, booking server) hosting the application components</li></ul>
<b>Employee Performance Management System</b>	<ul style="list-style-type: none"><li>• Identify the business processes such as goal setting, performance appraisal, and feedback</li><li>• Model the business actors (e.g., manager, employee) and their roles in these processes</li><li>• Specify the application functions (e.g., goal setting module, appraisal module) supporting these processes</li><li>• Represent the infrastructure elements (e.g., server, database) required to deploy the application functions</li></ul>
<b>Online Food Delivery Service</b>	<ul style="list-style-type: none"><li>• Define the business processes such as order placement, food preparation, and delivery tracking</li><li>• Model the business roles (e.g., customer, delivery driver) and their interactions</li><li>• Specify the application components (e.g., ordering system, delivery management system) supporting these processes</li><li>• Represent the infrastructure nodes (e.g., cloud server, GPS device) required to operate the application components</li></ul>





- Was the exercise easy?
- Is it easy to use ArchiMate?
- What are the main challenges?
- Is it worth doing such an exercise in a company?
- Is it possible to represent all the IT systems in a company?

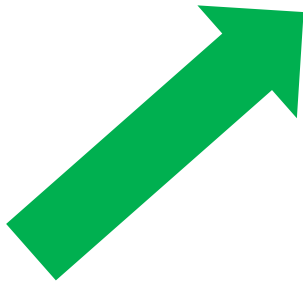




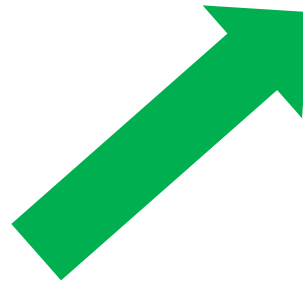


- Avoid redundant functionalities
- Ensure compatibilities between systems
- Guarantee the sustainability of the IT systems
- Ensure the strategy implementation by analyzing how the business can achieve its target outcomes and also by identifying the means to do so
- Identify risks and propose mitigation measures (security, personal data...)
- Optimize the value of the company's assets from a business or technical perspective

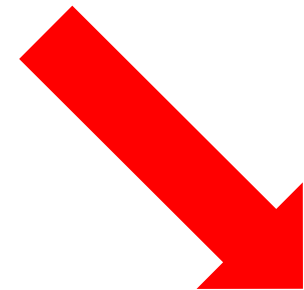
Growth

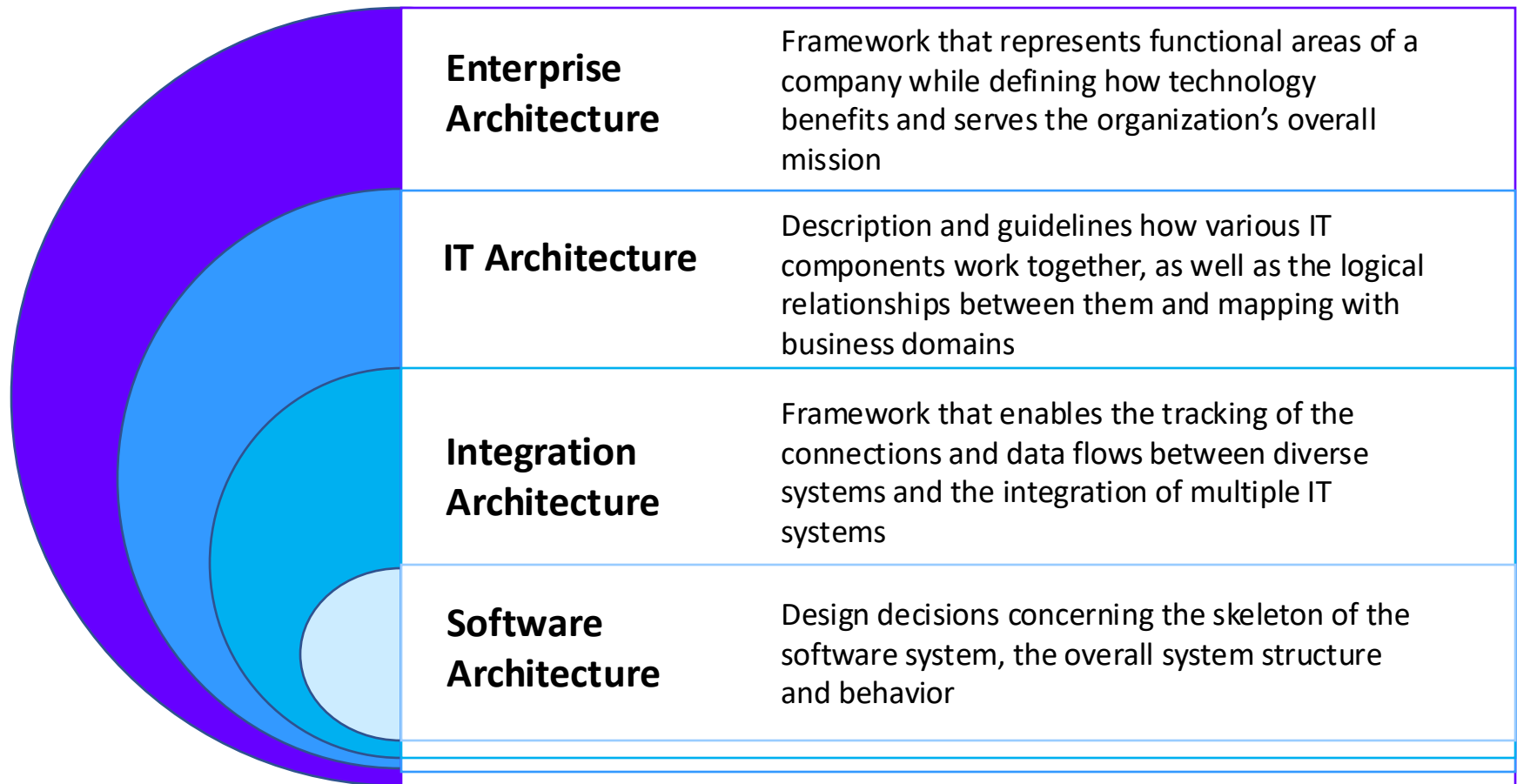


Compliance



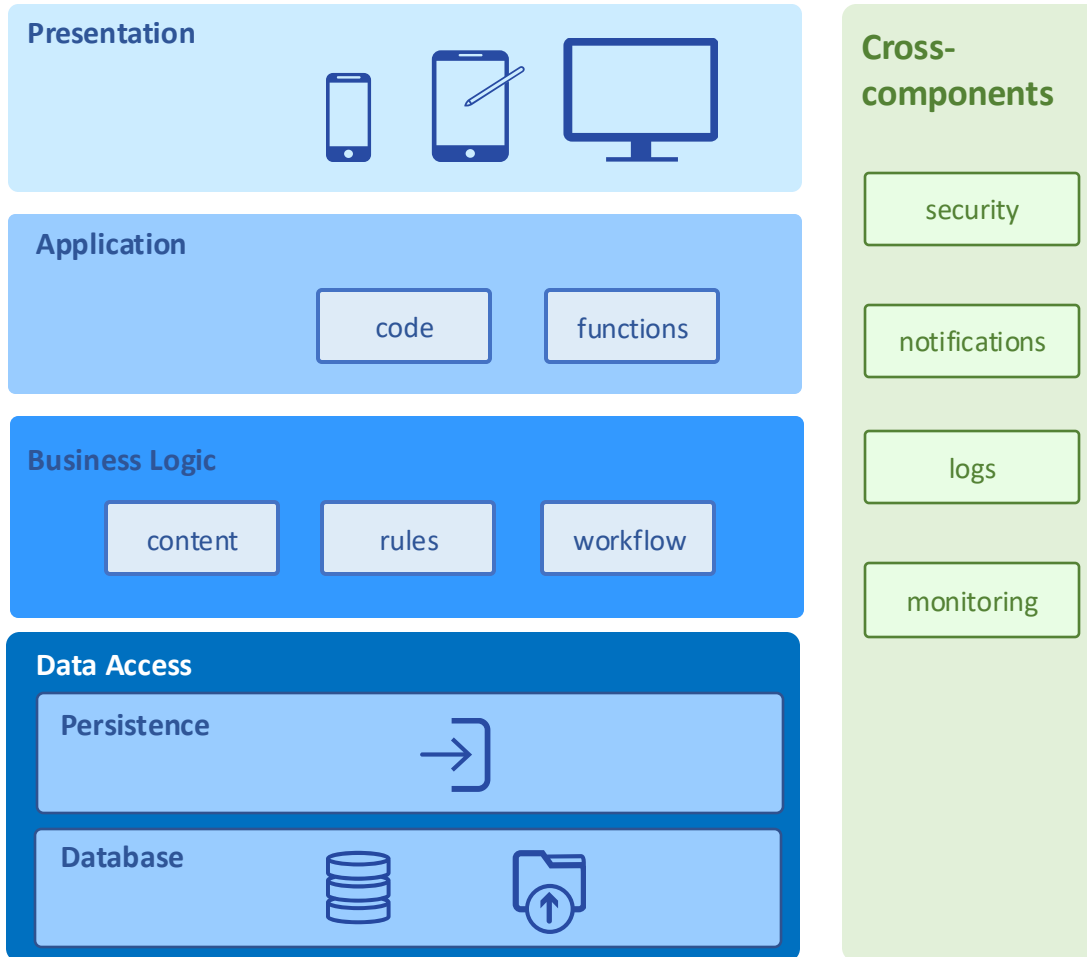
Complexity







## Layered architecture



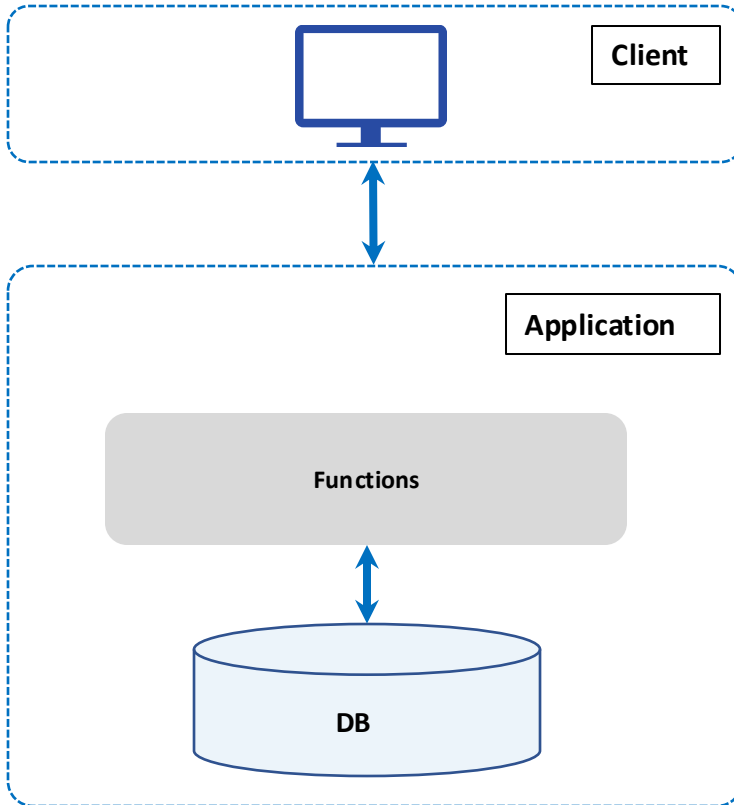
## 4-tier architecture

- Presentation (User Interface)
- Application (Service Layer)
- Business Logic (Domain layer)
- Data Access (Persistence)

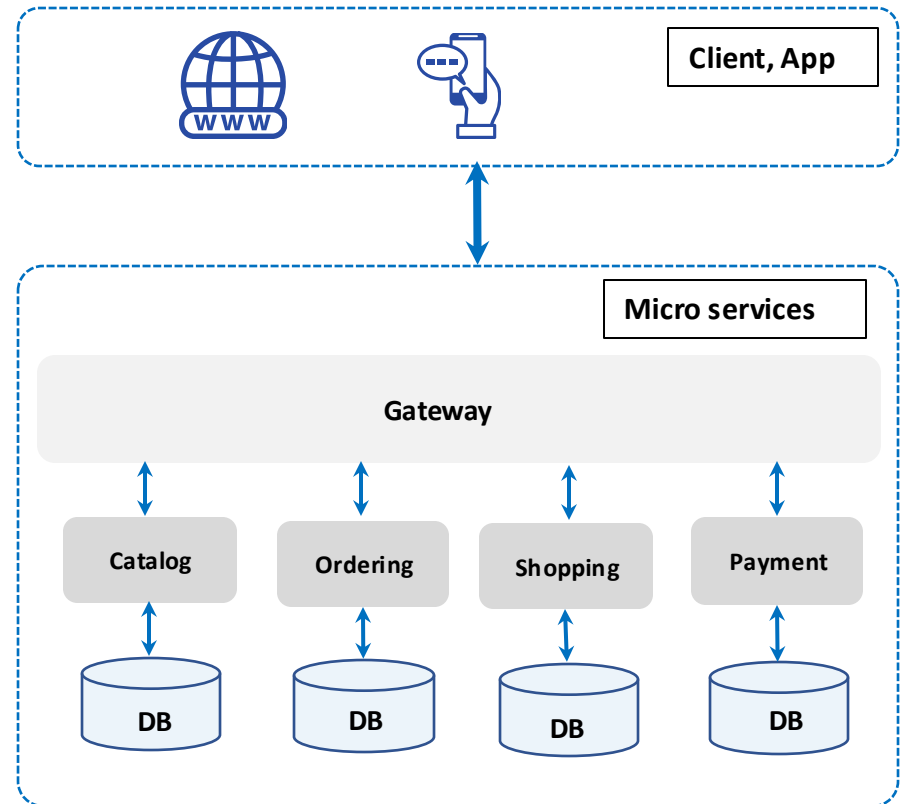
# Monolithic versus Micro service Architecture



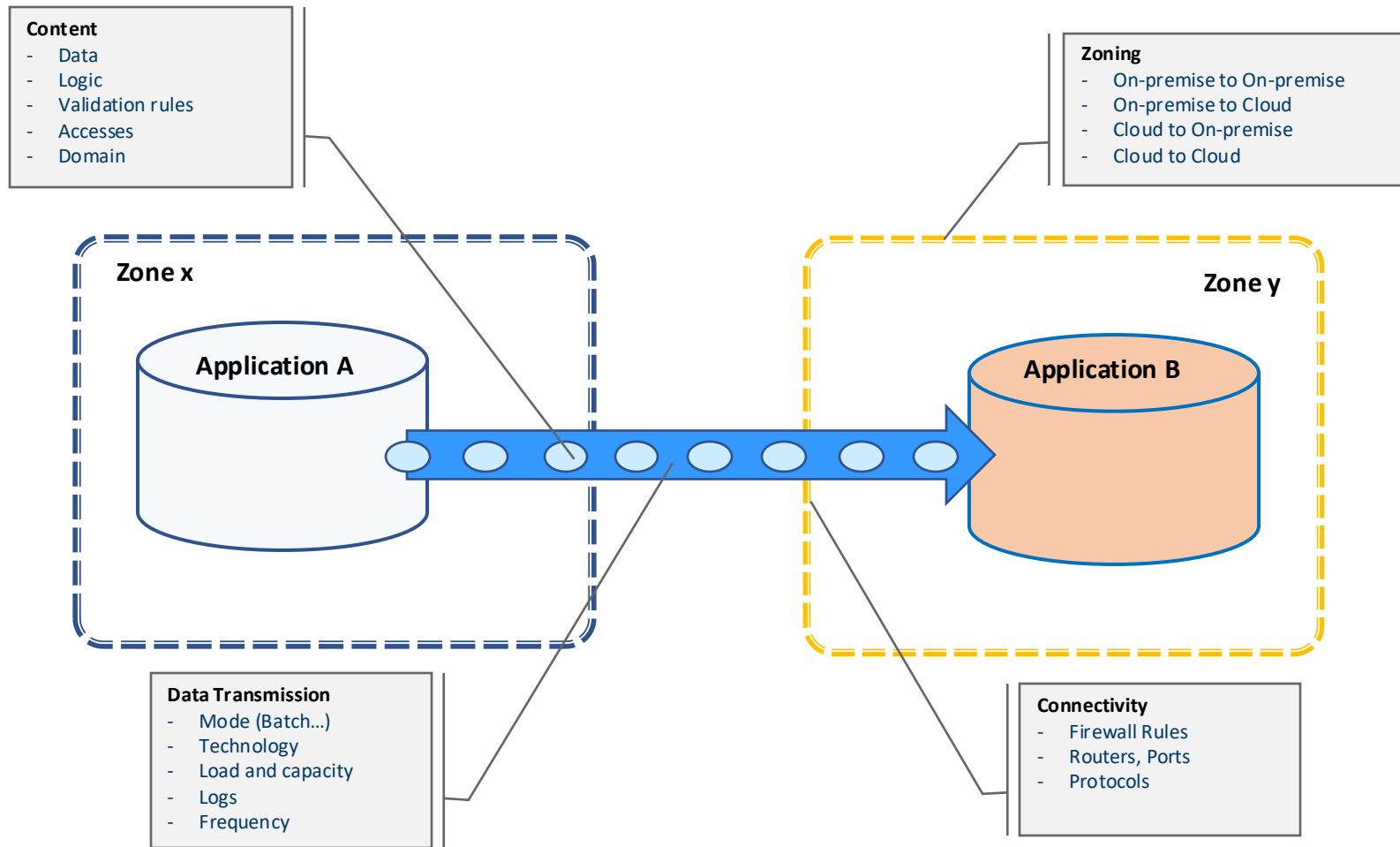
## Monolithic



## Micro services

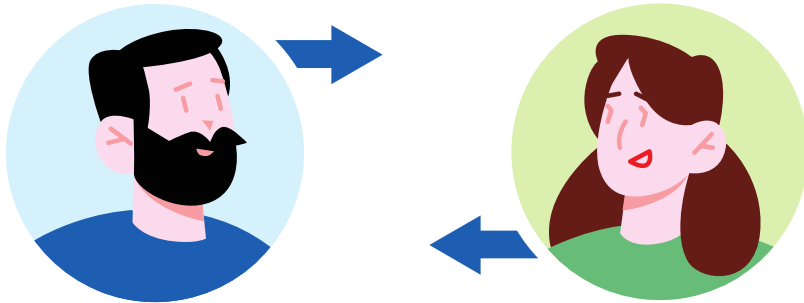


# Interfacing two IT Systems



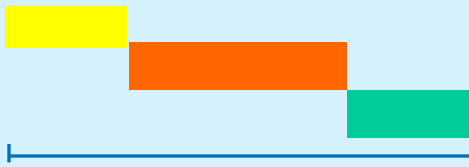


## synchronous

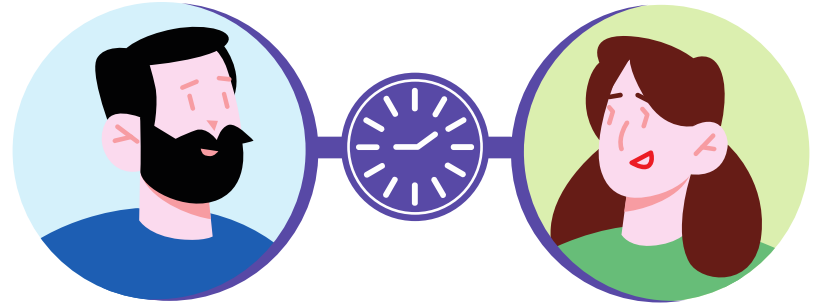


- 
- Video-conferences
  - Chat
  - Audio Calls

one operation or program running at a time



## asynchronous



- 
- Emails
  - Message boards
  - Dashboards
  - Collaboration platform

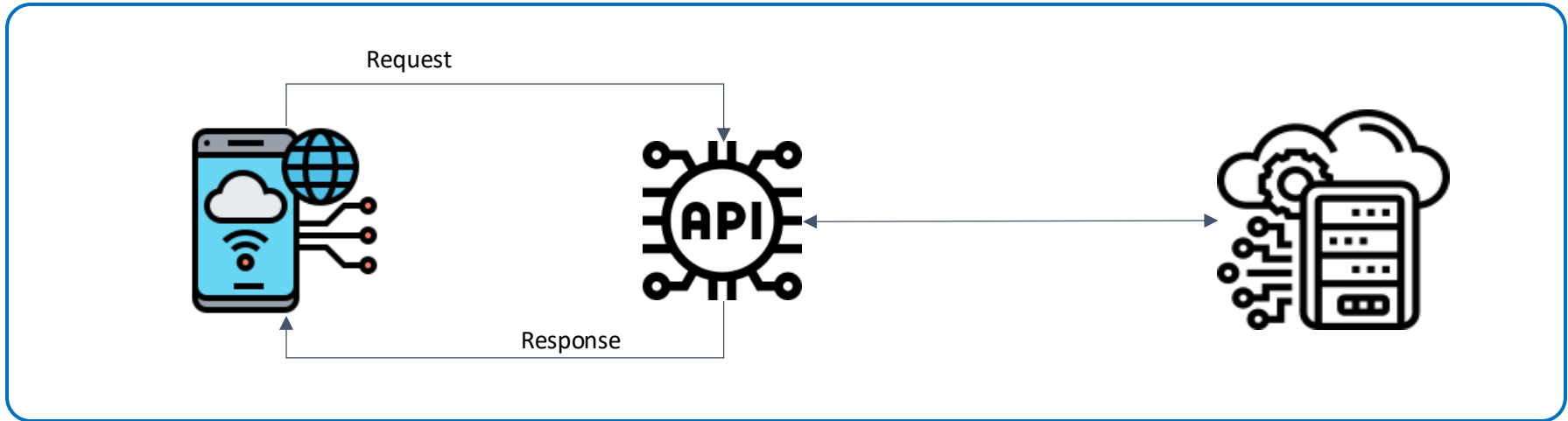
Non-blocking, no dependency















Technology	Description
<b>File Transfer Protocol (FTP)</b>	Standard network protocol that enables the transfer of files between a client and a server. Good for transferring large volumes of data that do not require high security or integrity.
<b>Application Programming Interface (API)</b>	Set of rules and specifications that define how different software components can communicate and exchange data. It allows real-time, bi-directional, and customized data exchange but requires development and maintenance effort.
<b>Extract, Transform, Load (ETL)</b>	Process that involves extracting data from one or more sources, transforming it into a common format, and loading it into a destination system. Powerful and efficient method of data transfer, as it allows for data integration, consolidation, and analysis but costly.
<b>Message Queue (MQ)</b>	Asynchronous and reliable transfer of messages between different systems.
<b>Data Streaming (DS)</b>	Continuous and real-time transfer of data between different systems. high bandwidth and resources, and may encounter data security, privacy, or governance issues.



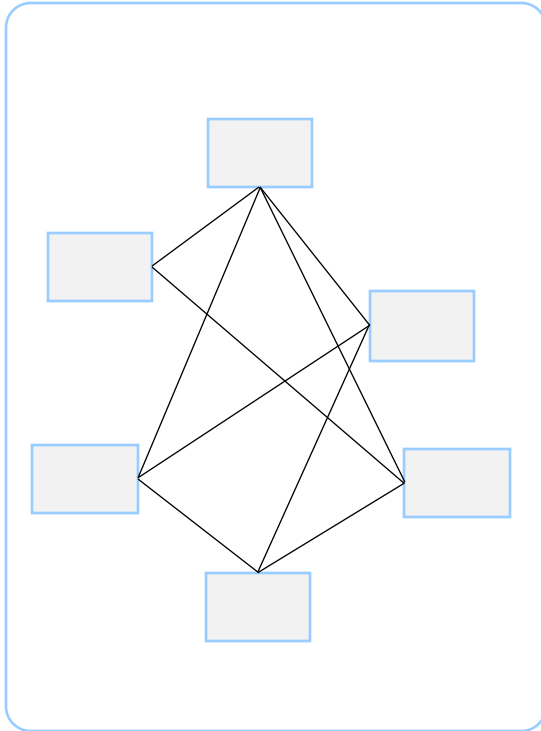
- **API:** a set of rules, protocols and routines that define how two machines talk to each other  
It delivers a user request to a system and sends the system's response back to a user
- **Rest-API:** common method for connecting components and applications in a micro service architecture. A REST API is based on the HTTP protocol and uses HTTP requests to POST (create), PUT (update), GET (read), and DELETE data
- **SOAP API:** highly structured using XML data format, more secure (payment, customer management...)
- Two major underlying technologies used in building real-time APIs are **WebSocket** and **SSE** (Server-Sent Events)



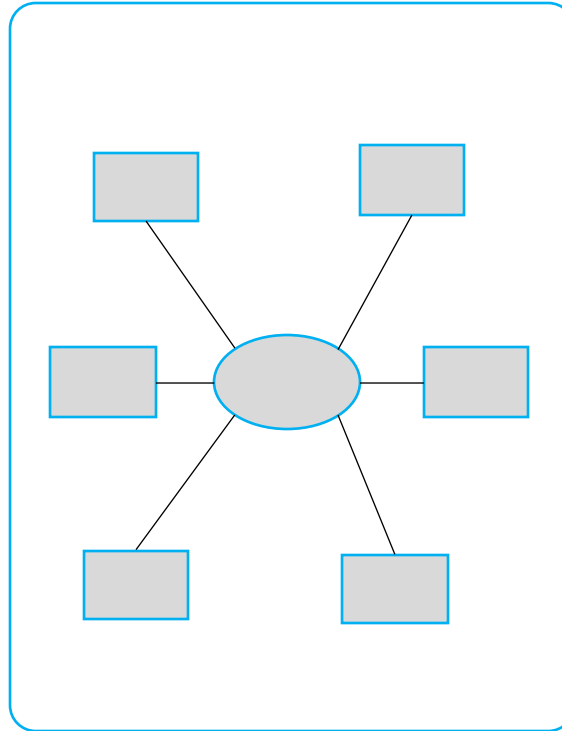
<p>{ REST }</p> <p></p> <p>SOAP API</p>	<p><b>REST API, SOAP API</b> (synchronous)</p> <p><u>REST API</u>: application programming interface commonly used in web and mobile applications to retrieve or modify resources and data on remote systems</p> <p><u>SOAP API</u>: standard messaging protocol that operating systems use to communicate via Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML)</p>
<p> </p>	<p><b>Microservice, RabbitMQ, Kafka</b> (asynchronous)</p> <p><u>Microservice</u>: code required for a particular application function</p> <p><u>RabbitMQ</u>: “Open source” message broker software that implements the Advanced Message Queuing Protocol</p> <p><u>Kafka</u>: free software from the Apache Software Foundation used in particular for processing data streams</p>
<p> </p>	<p><b>Java, Spring Boot</b></p> <p><u>Java</u>: object-oriented programming language</p> <p><u>Spring Boot</u>: “Open source” Java framework used to create a micro service</p>
<p> </p> <p></p>	<p><b>Docker, Container, Kubernetes, Openshift</b></p> <p><u>Docker</u>: a platform for developers and system administrators to build, share and run the applications with containers</p> <p><u>Container</u>: packages of software that contain all of the necessary elements to run in any environment and that virtualize the operating system</p> <p><u>Kubernetes</u>: an “open source” system developed by Google for managing container applications</p> <p><u>Openshift</u>: a product line developed by Red Hat based on Kubernetes</p>



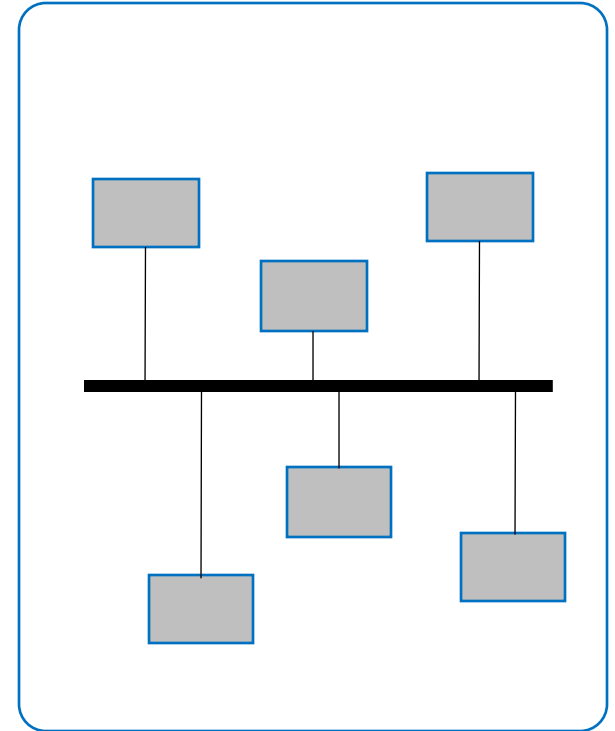
**Point-to-point**



**Hub**

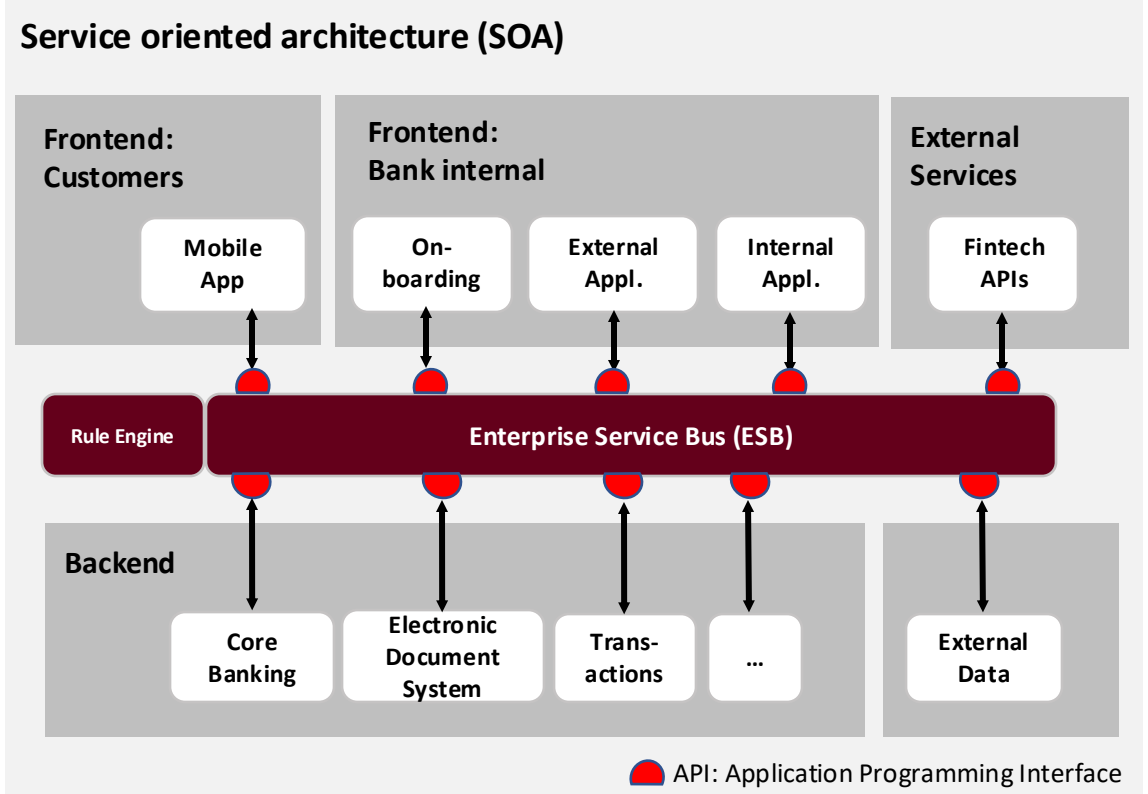
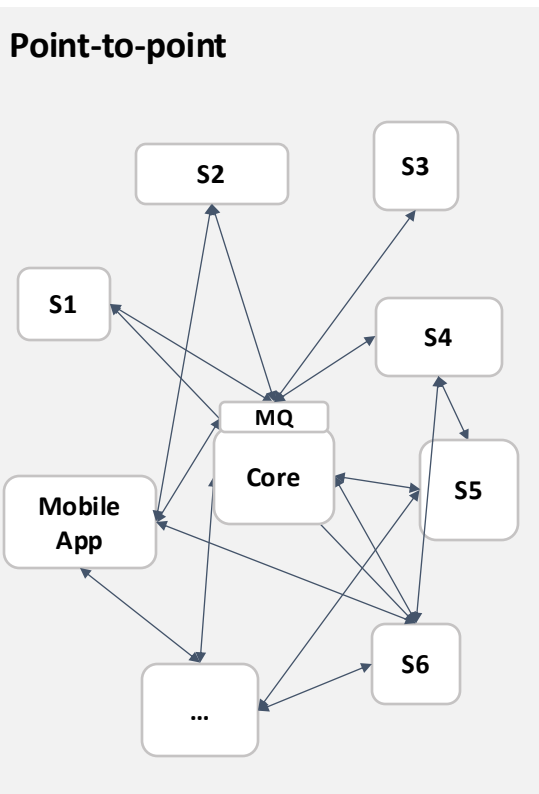


**Bus**

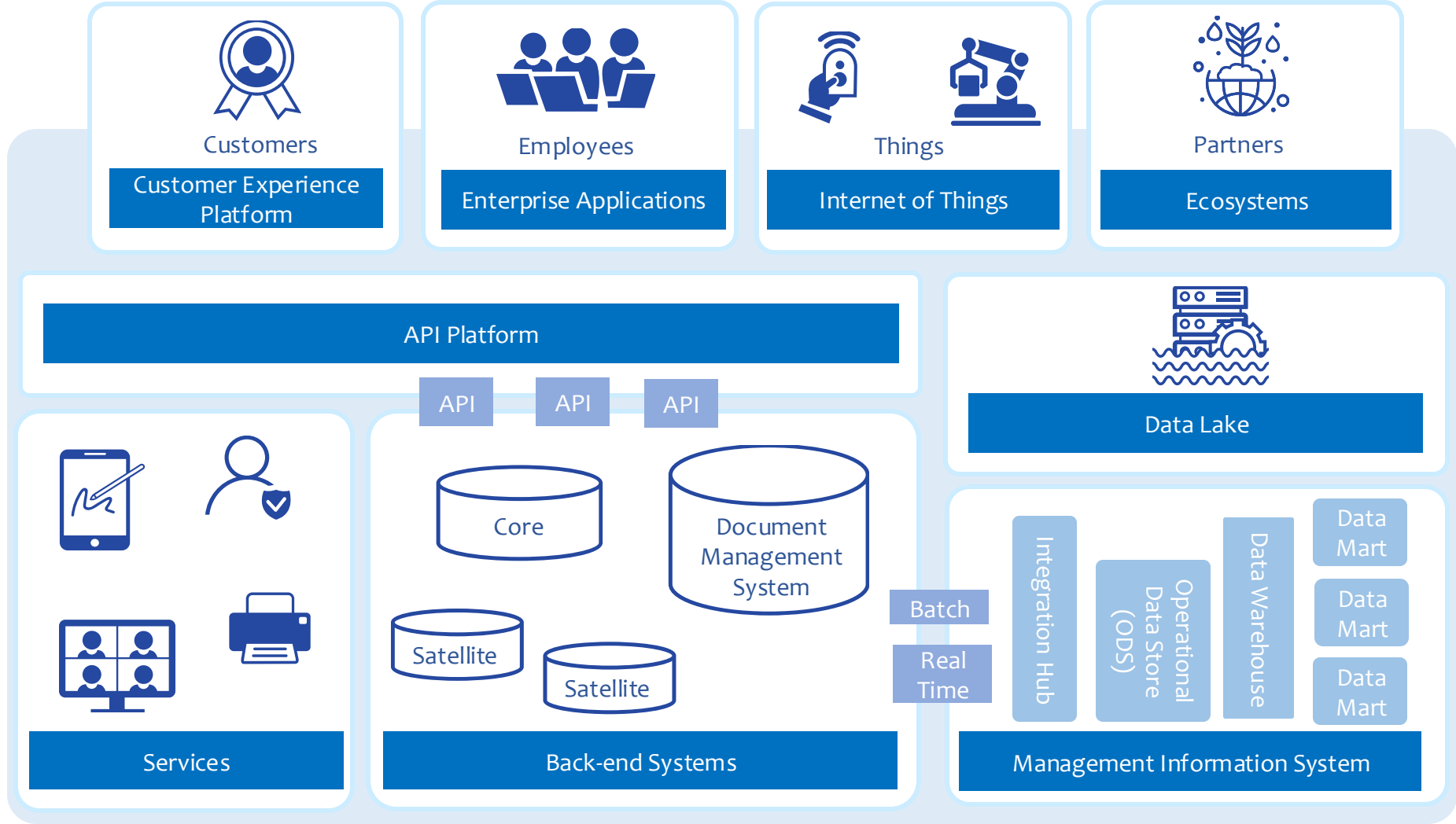




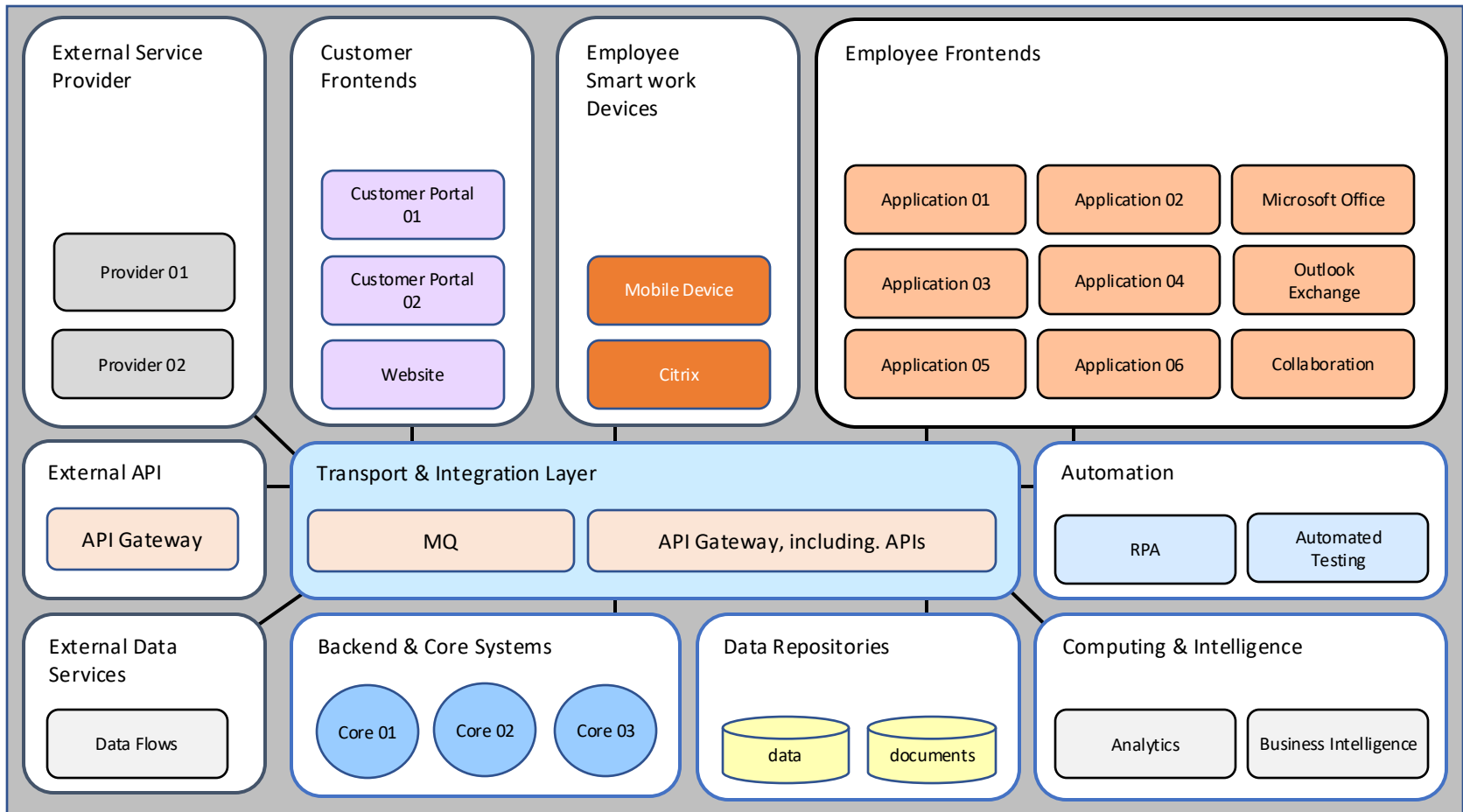
## From Point-to-point to SOA



# A Reference for a Target IT Architecture



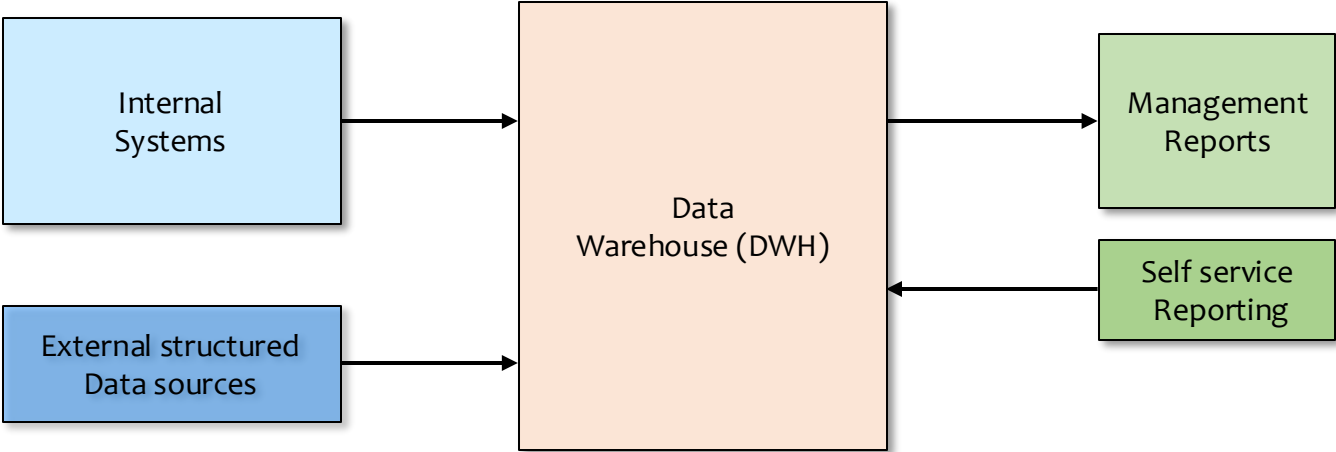
# Example of an Application Landscape



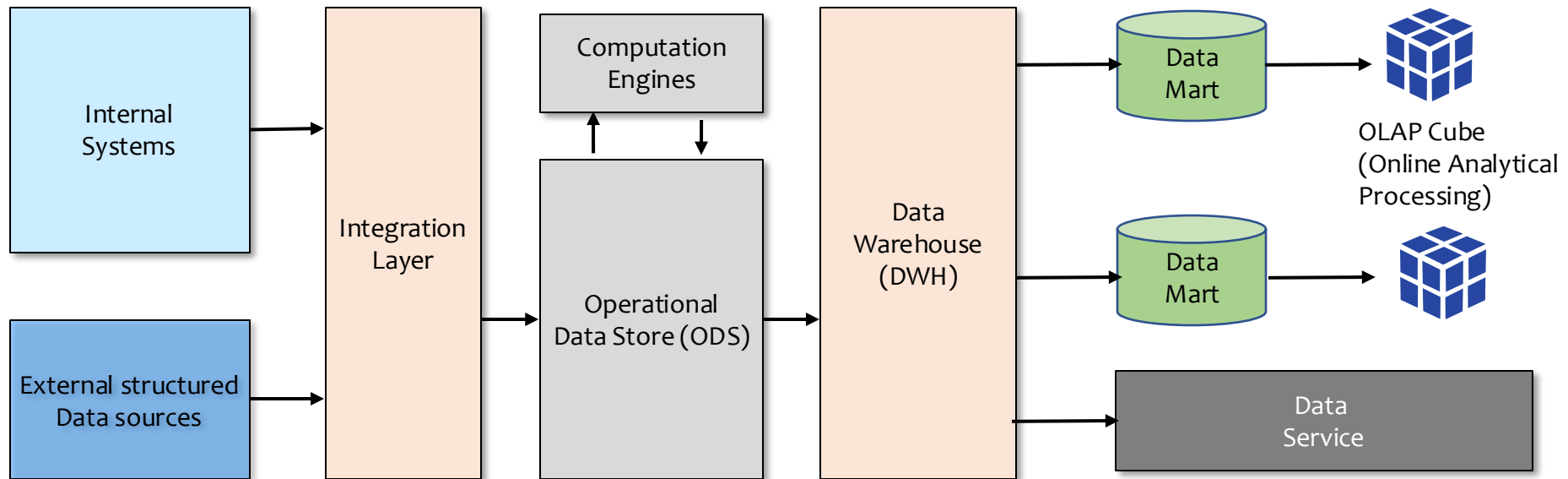
MQ: Middleware-Software from IBM for Message Queueing

API: Application Programming Interface

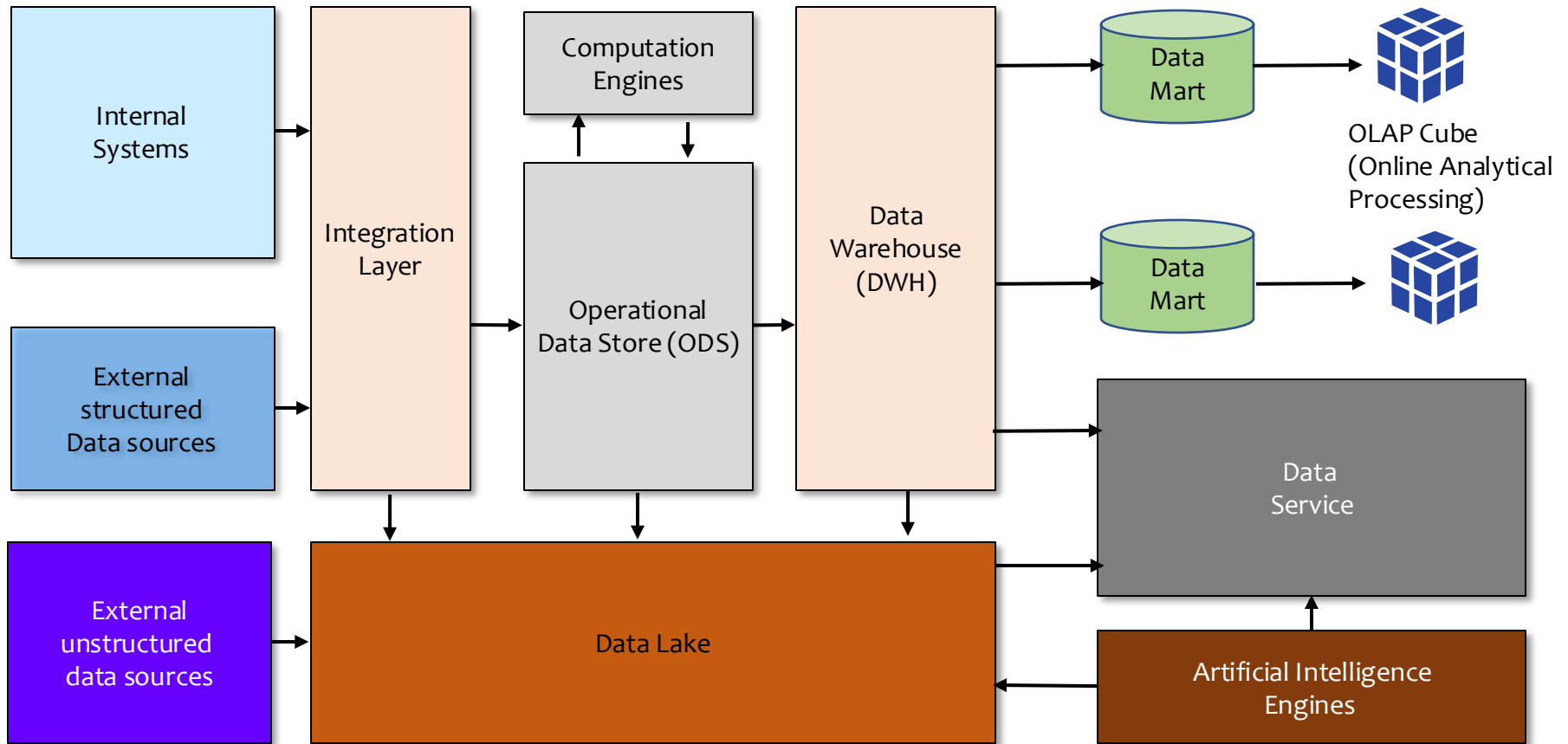
RPA: Robotic Process Automation



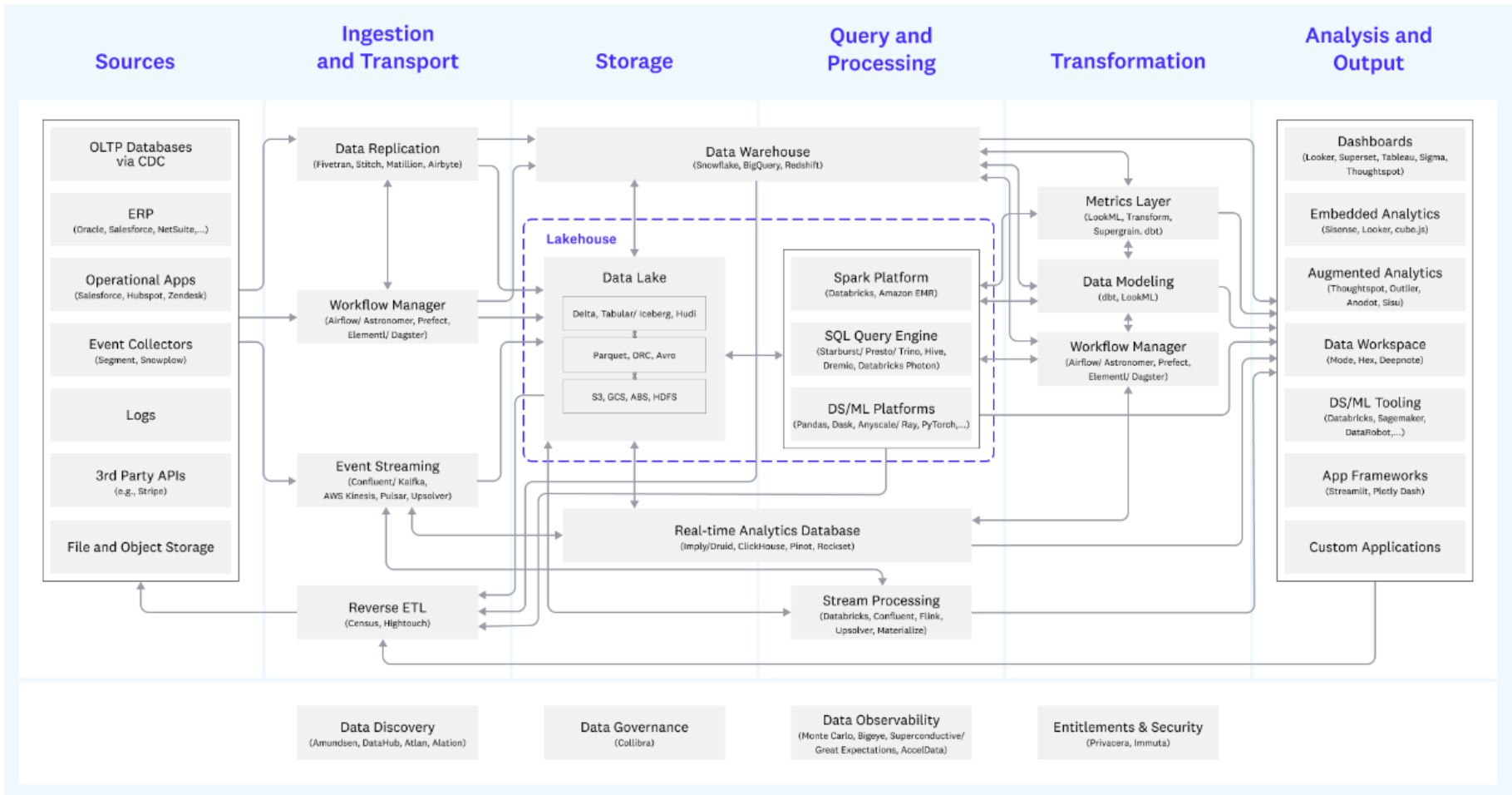




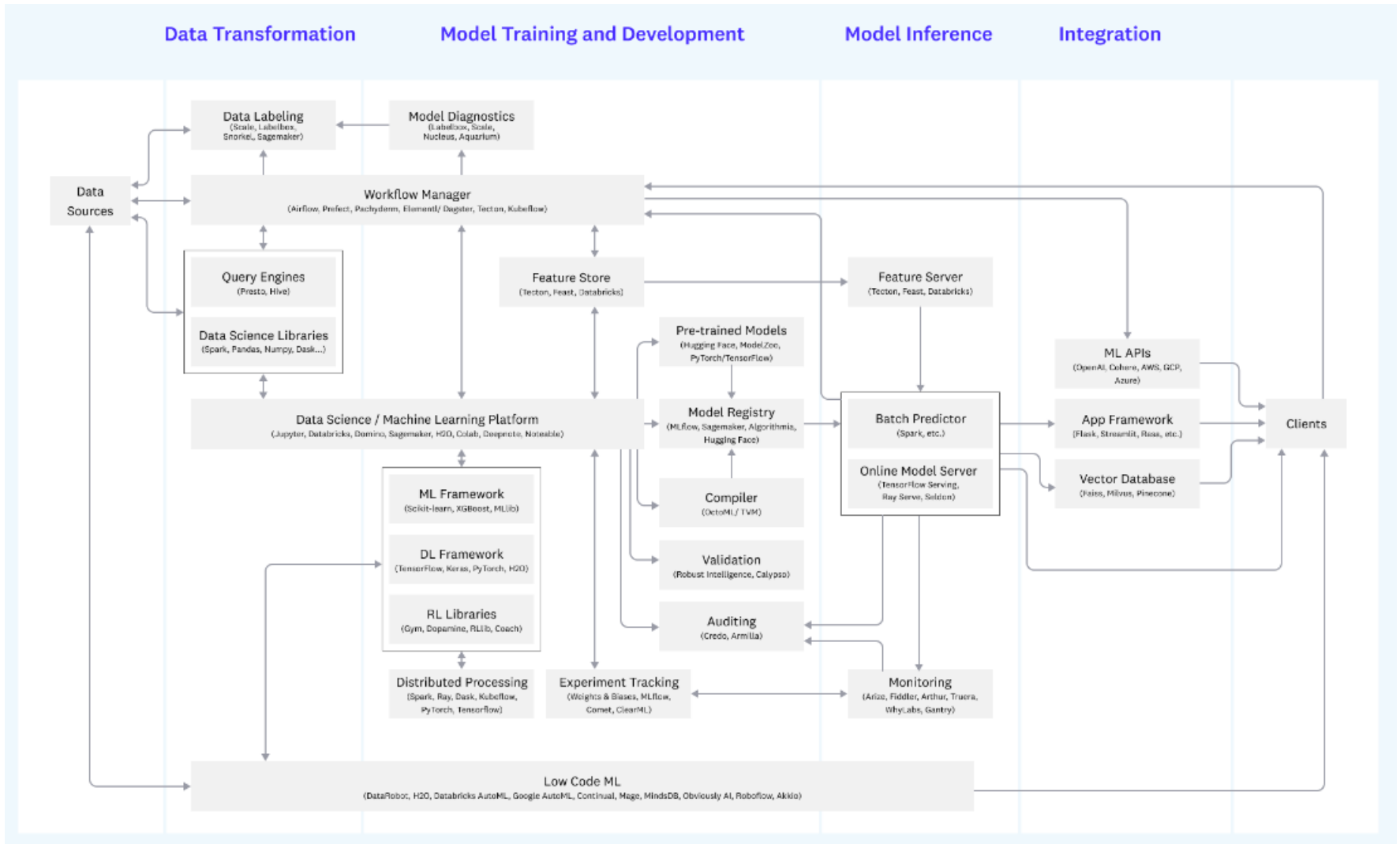
# Data Architecture for Big Data



# Unified Data Infrastructure (2.0)

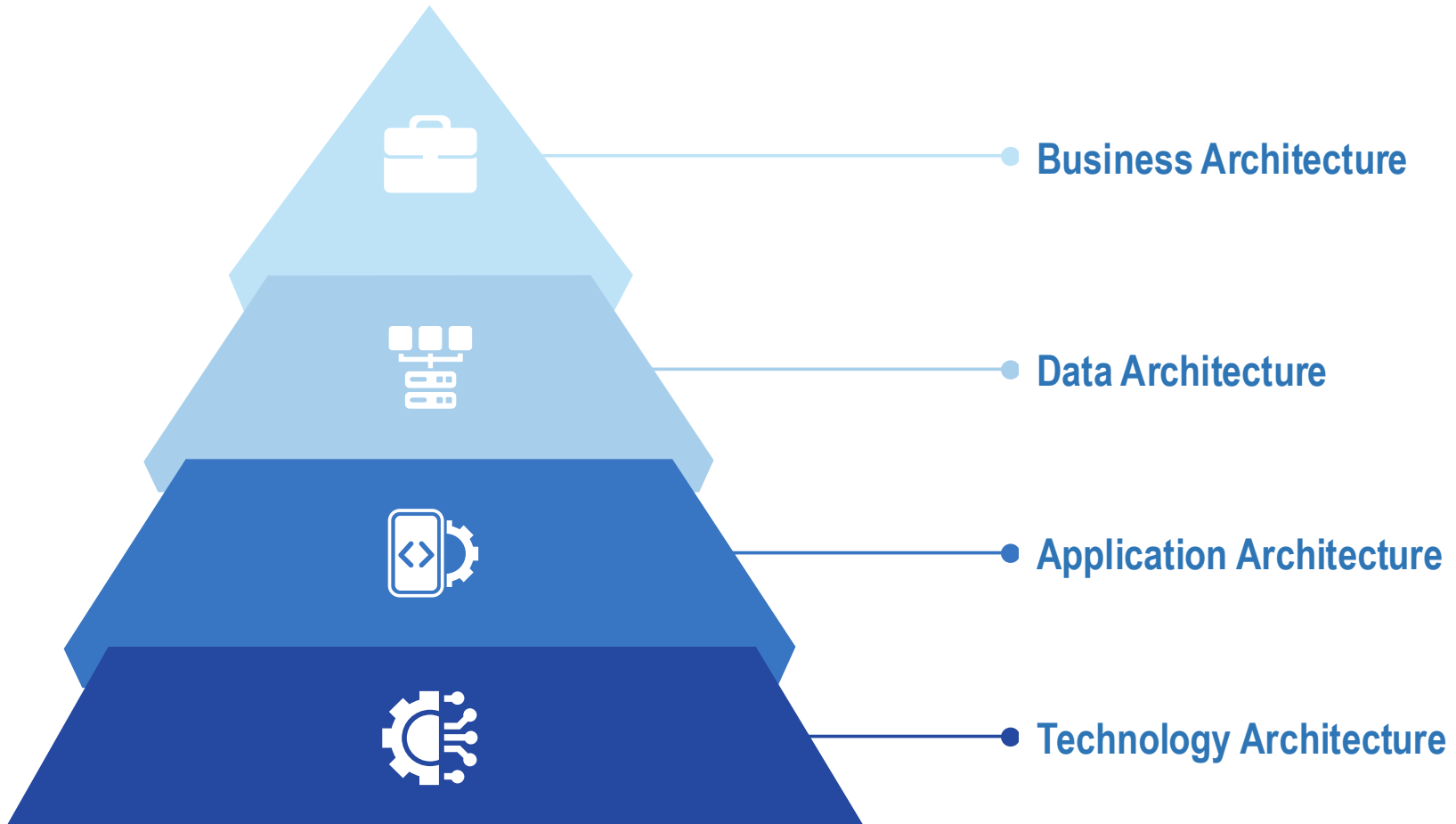


# Machine Learning Infrastructure (2.0)



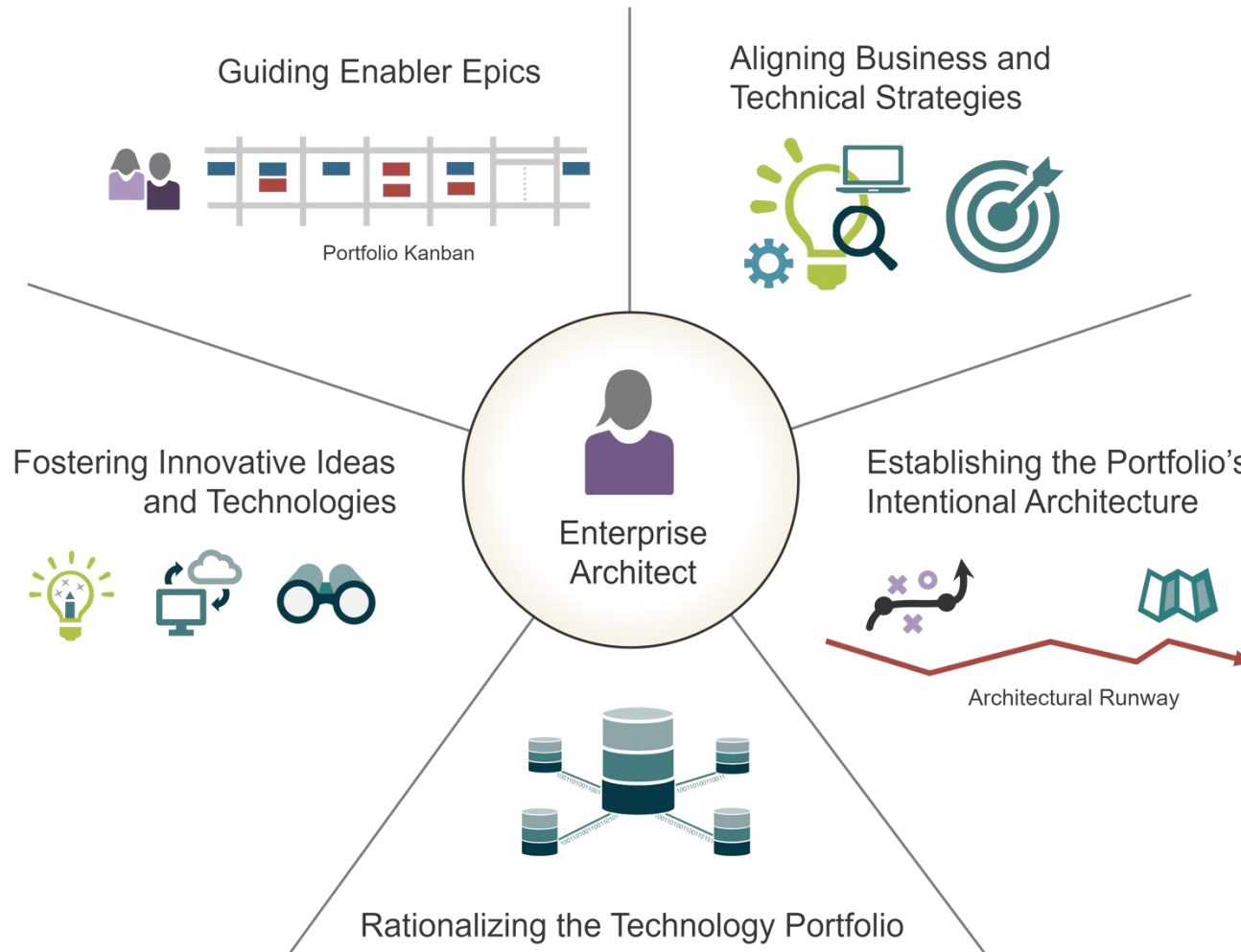


## Different Information Layers concerning the Enterprise





## From the SAFe Framework

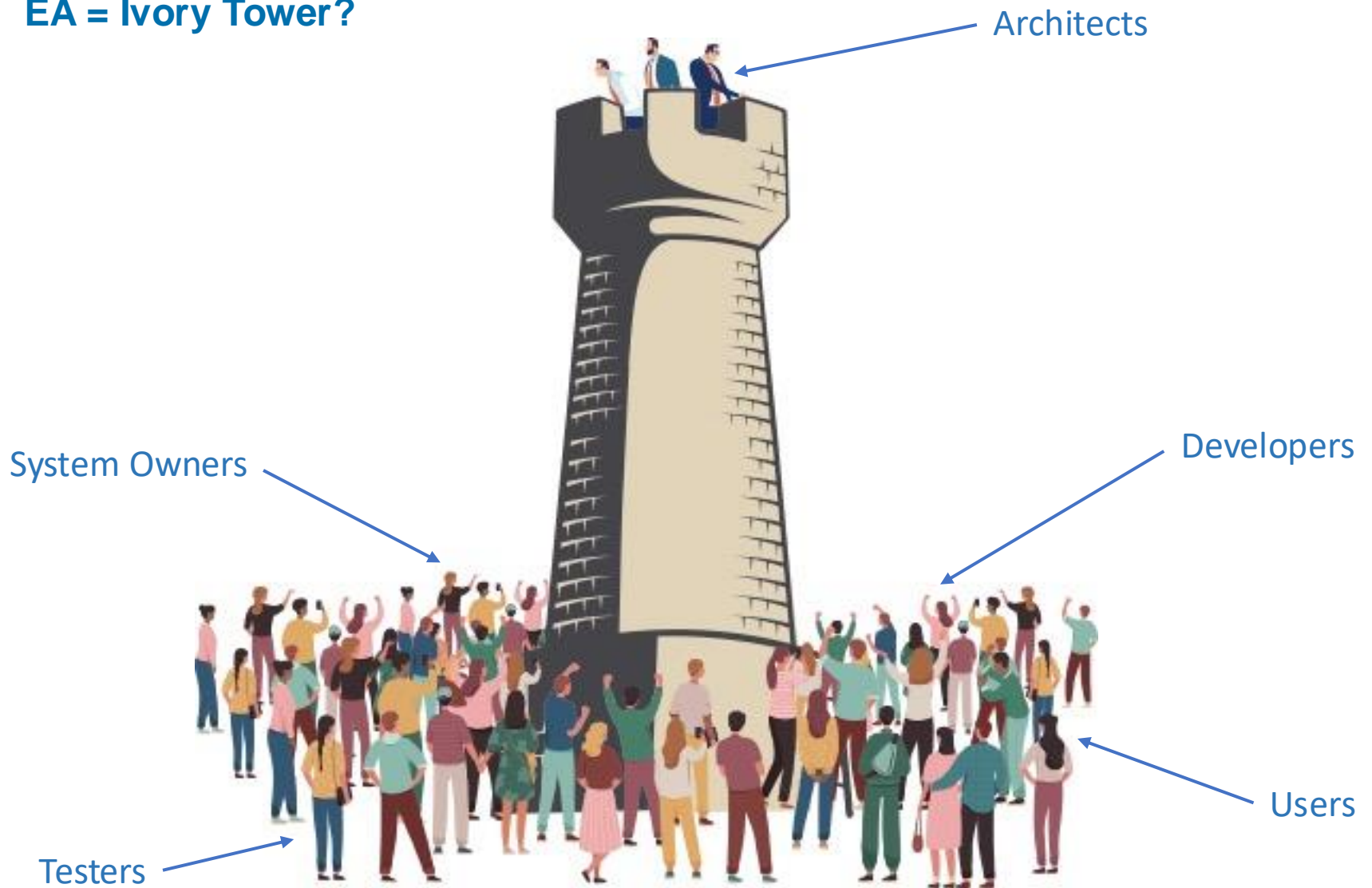


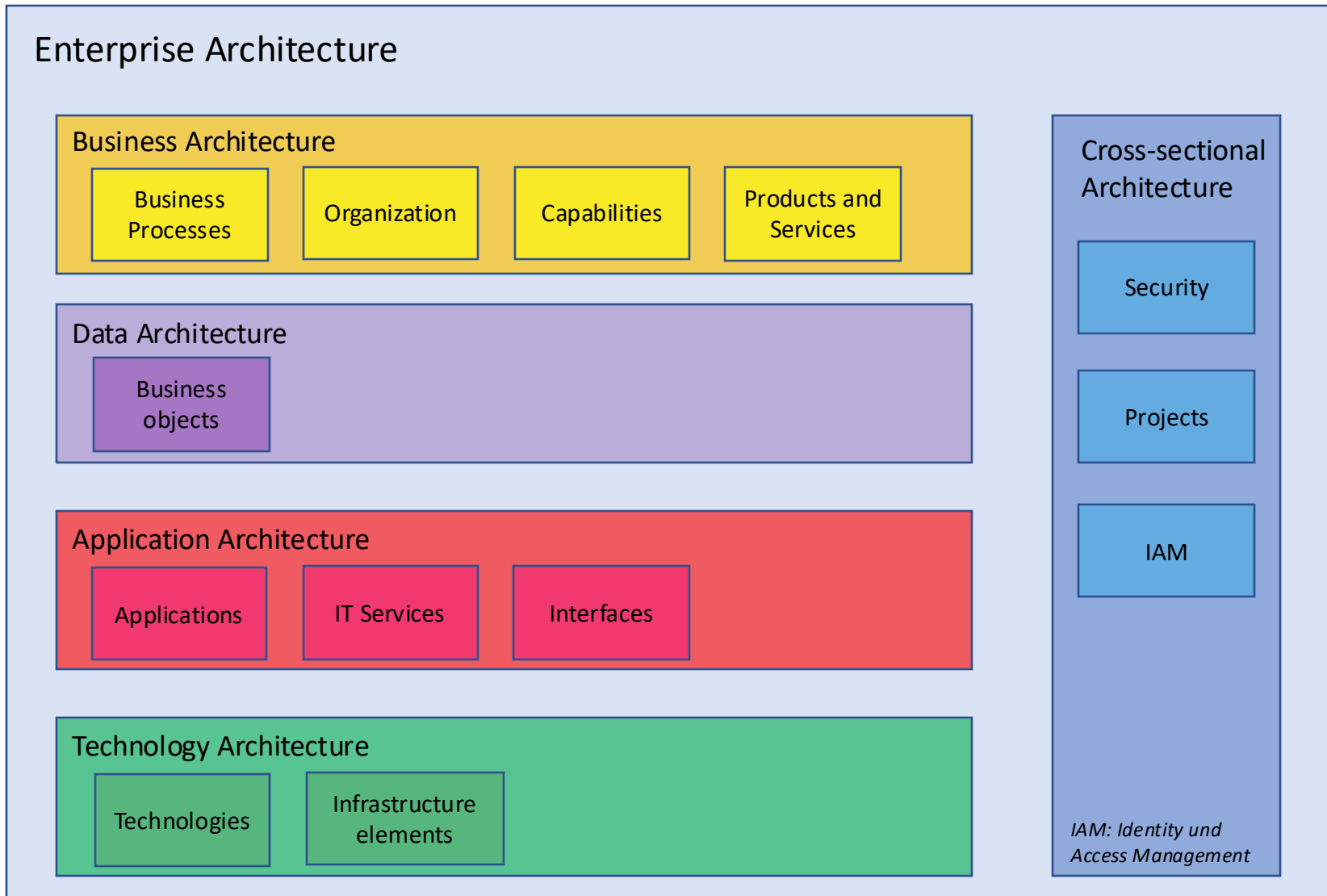
© Scaled Agile, Inc.

# What is not Enterprise Architecture?



## EA = Ivory Tower?









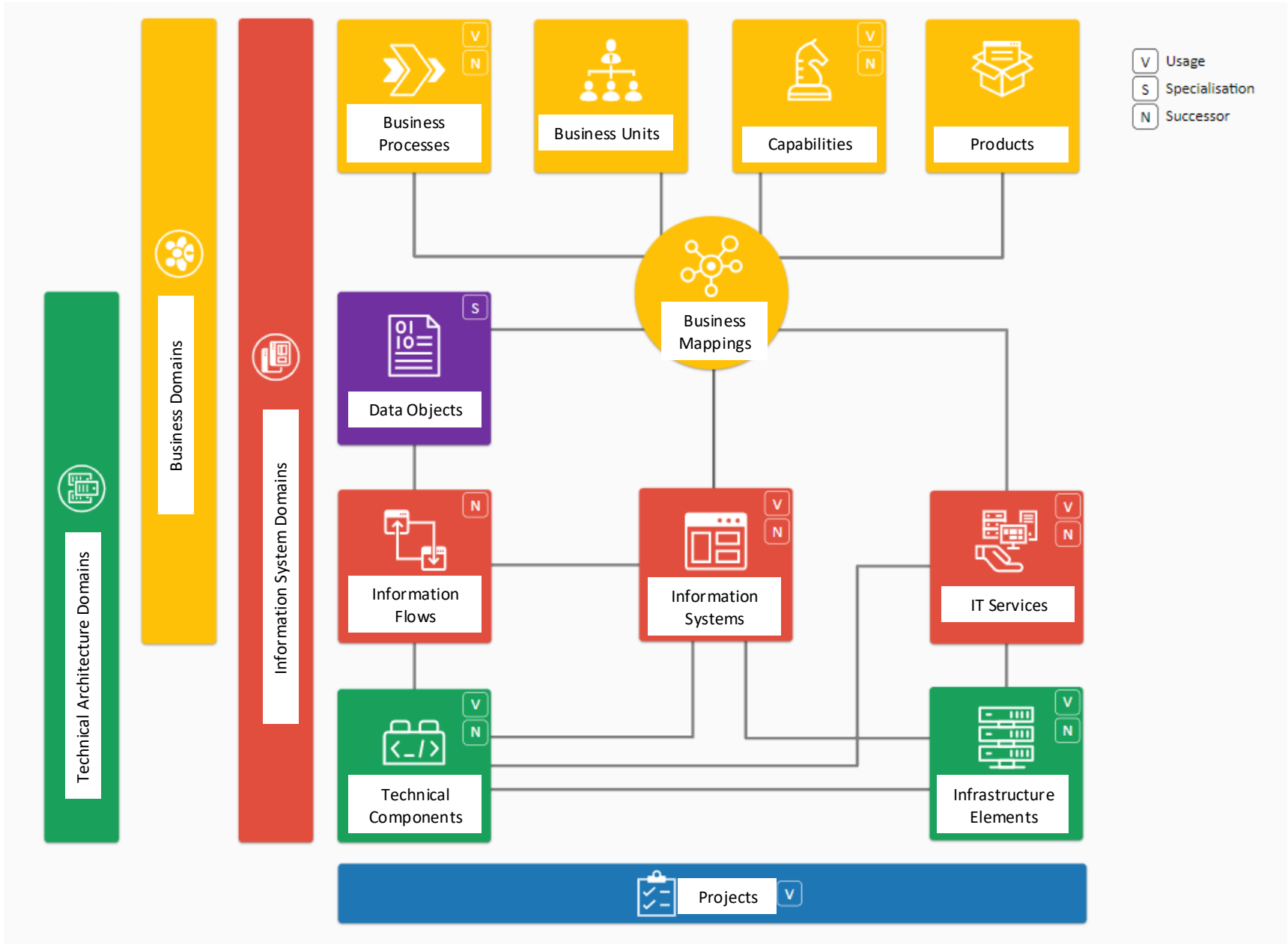
## Requirements (proposal)

- Modern technology (e.g., OpenShift, web client)
- "On Premise" solution or Cloud solution
- Flexible configuration and integration due to interfaces
- Strong visualization options
- Simple onboarding and uncomplicated knowledge exchange

## Tool Evaluation

Tool	Functional Degree of Completion	User-friendliness	Licensing	Overall
LUY	High degree of fulfilment, flexible configuration, modern technology	user-friendly, easy navigation, quick results	based on company size	Good!
ADOIT	Medium level of fulfilment, inflexible configuration, outdated technology	based on Visio 2.0	writers and readers	Average
LeanIX	High degree of fulfilment, potential for Lifecycle Management and AI	modern, social-media inspired	based on the number of users	Good, cloud solution only

# Entry mask





Landscape



Nested Cluster



Nested Layer



Information Flow



Masterplan



Cluster



Pie Chart



Bar Chart



Tree Map



Hotspot

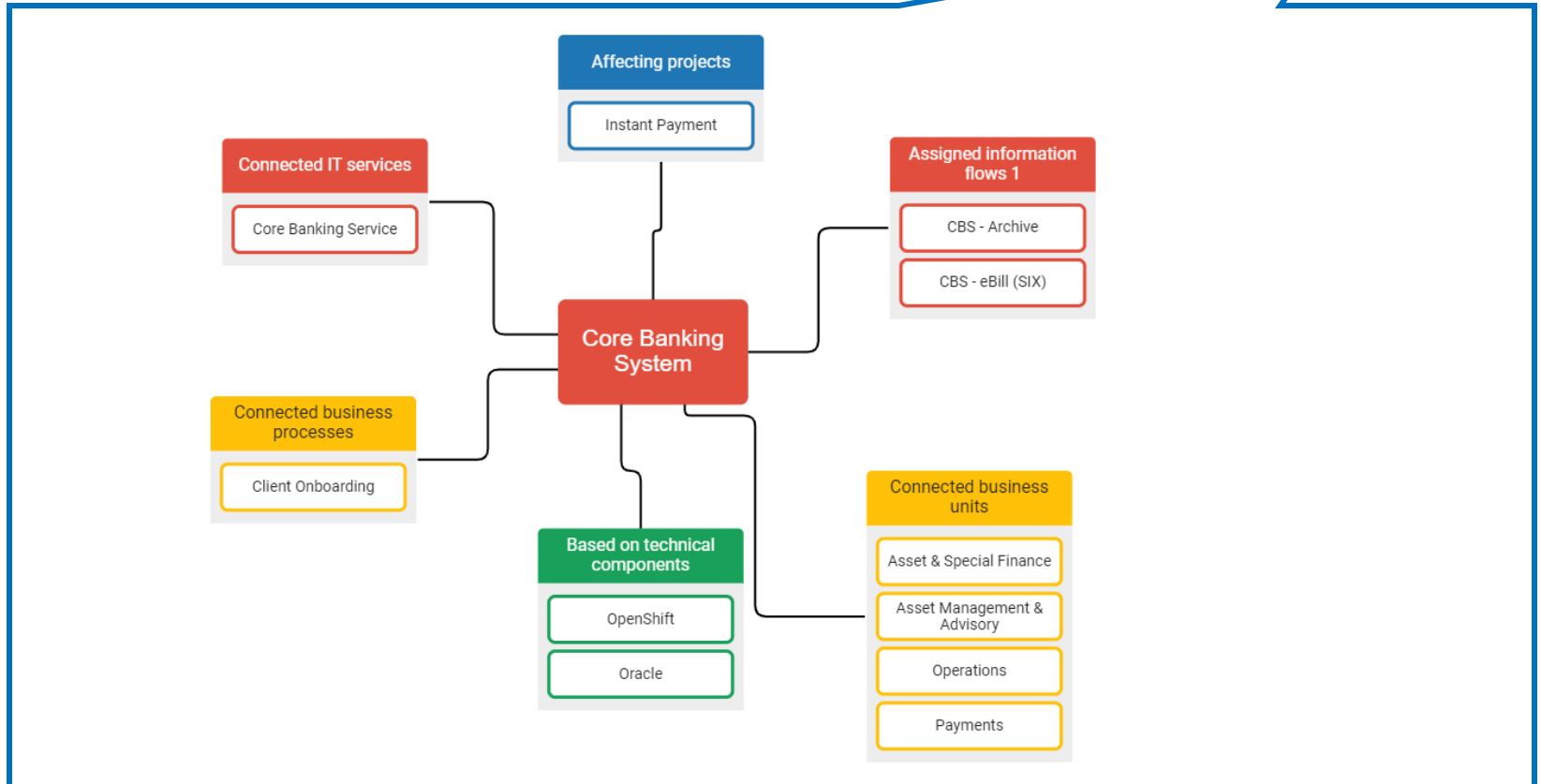
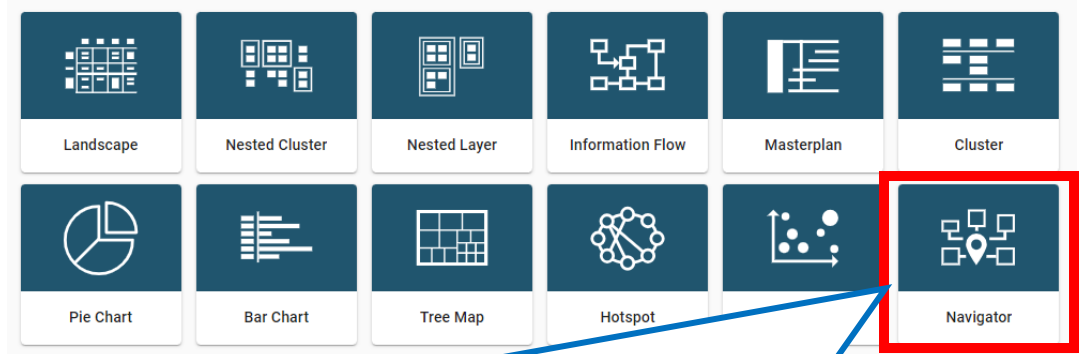


Portfolio

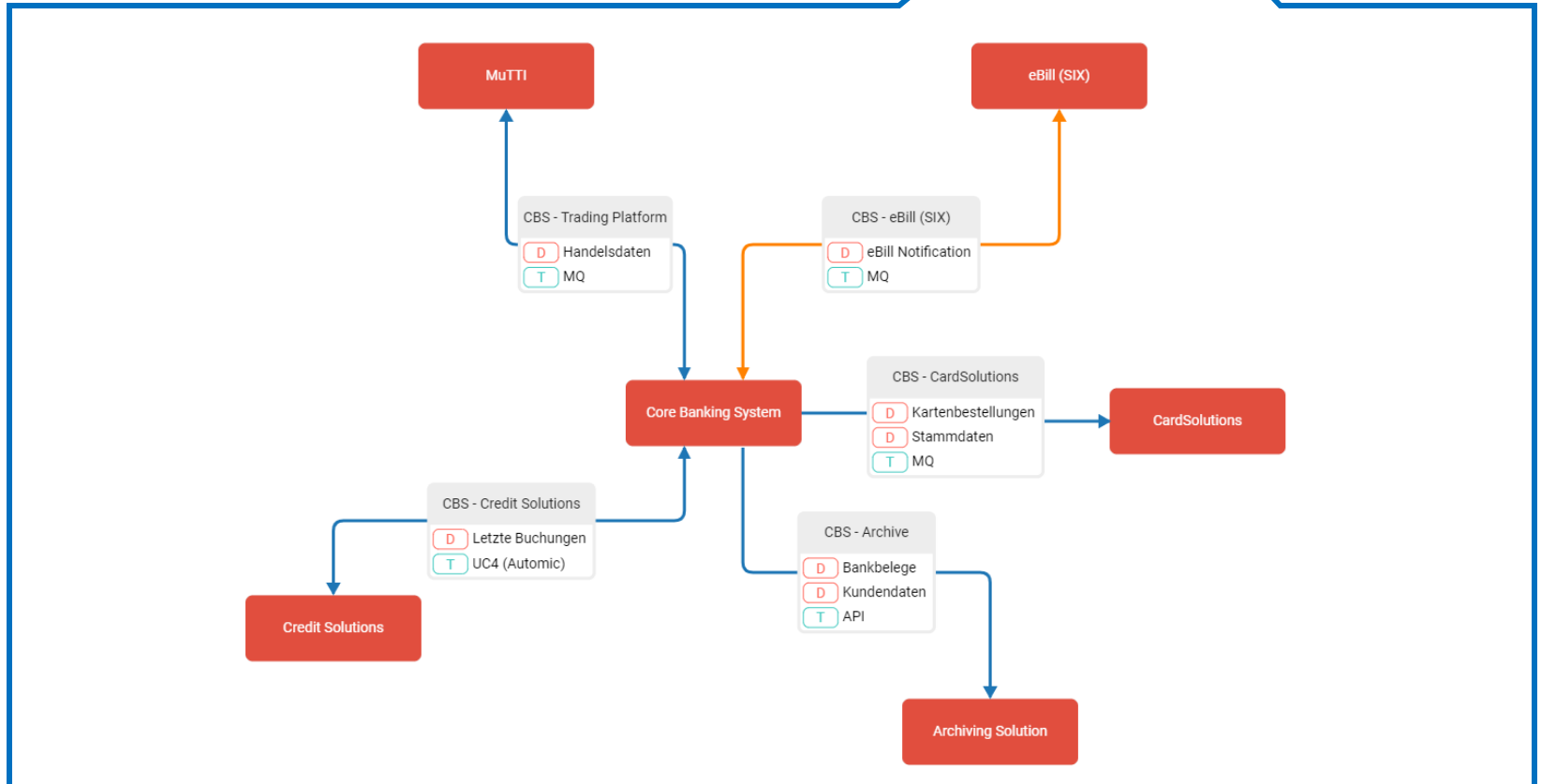
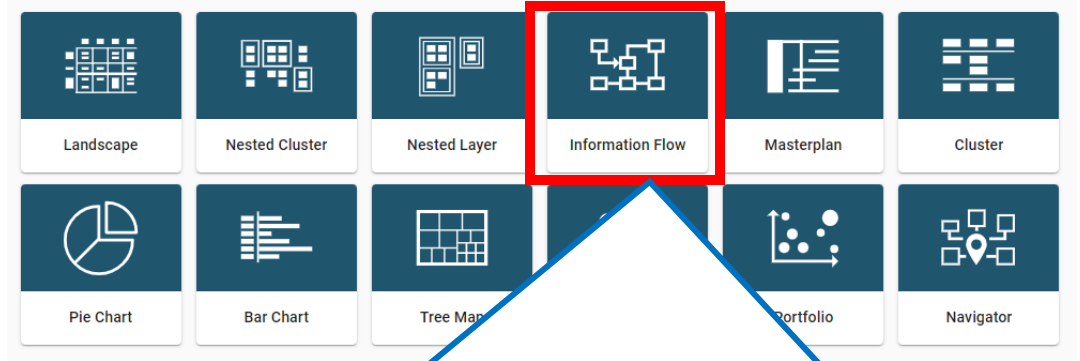


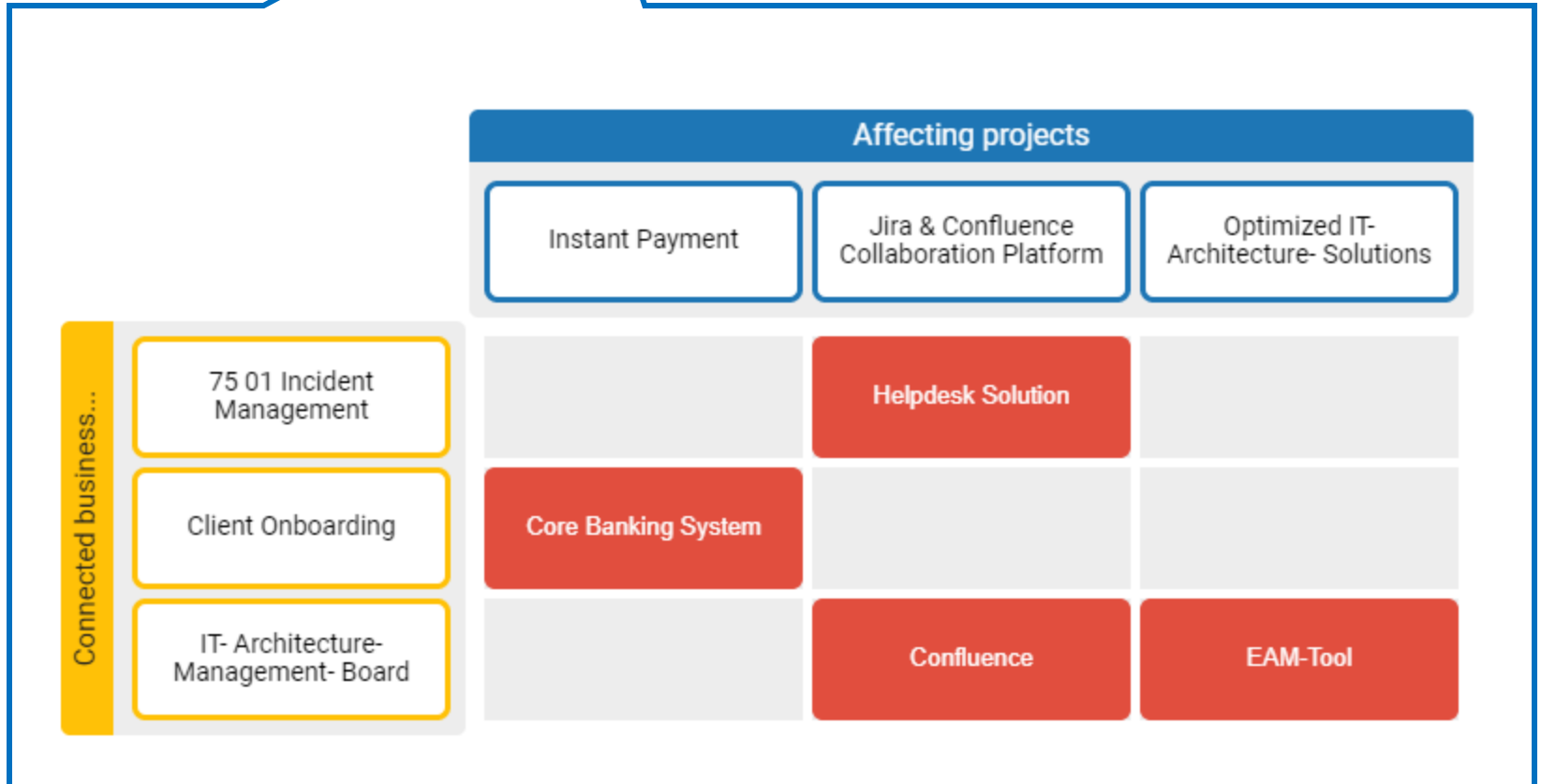
Navigator

# Navigator: Example



# Information Flow: Example

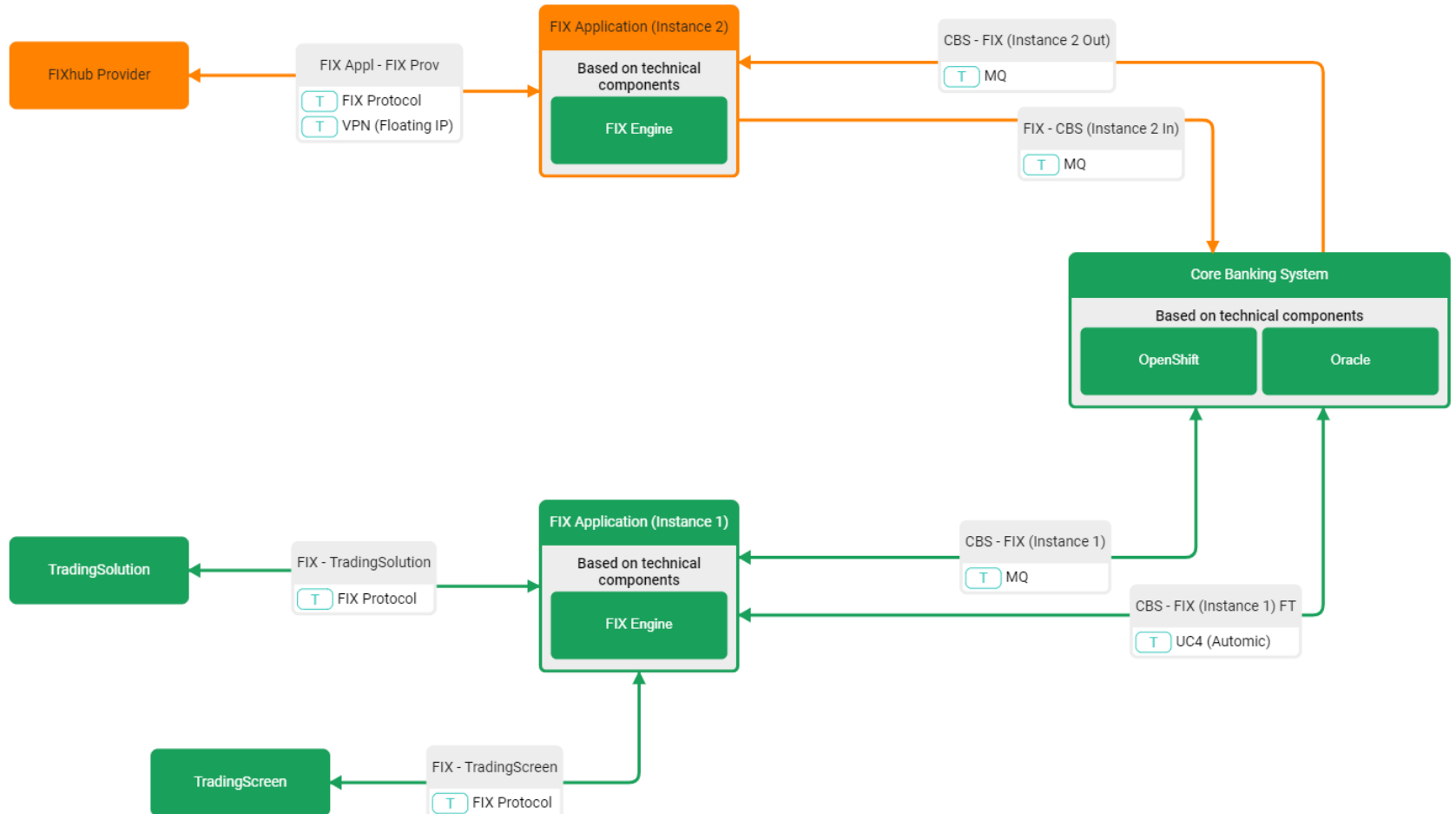


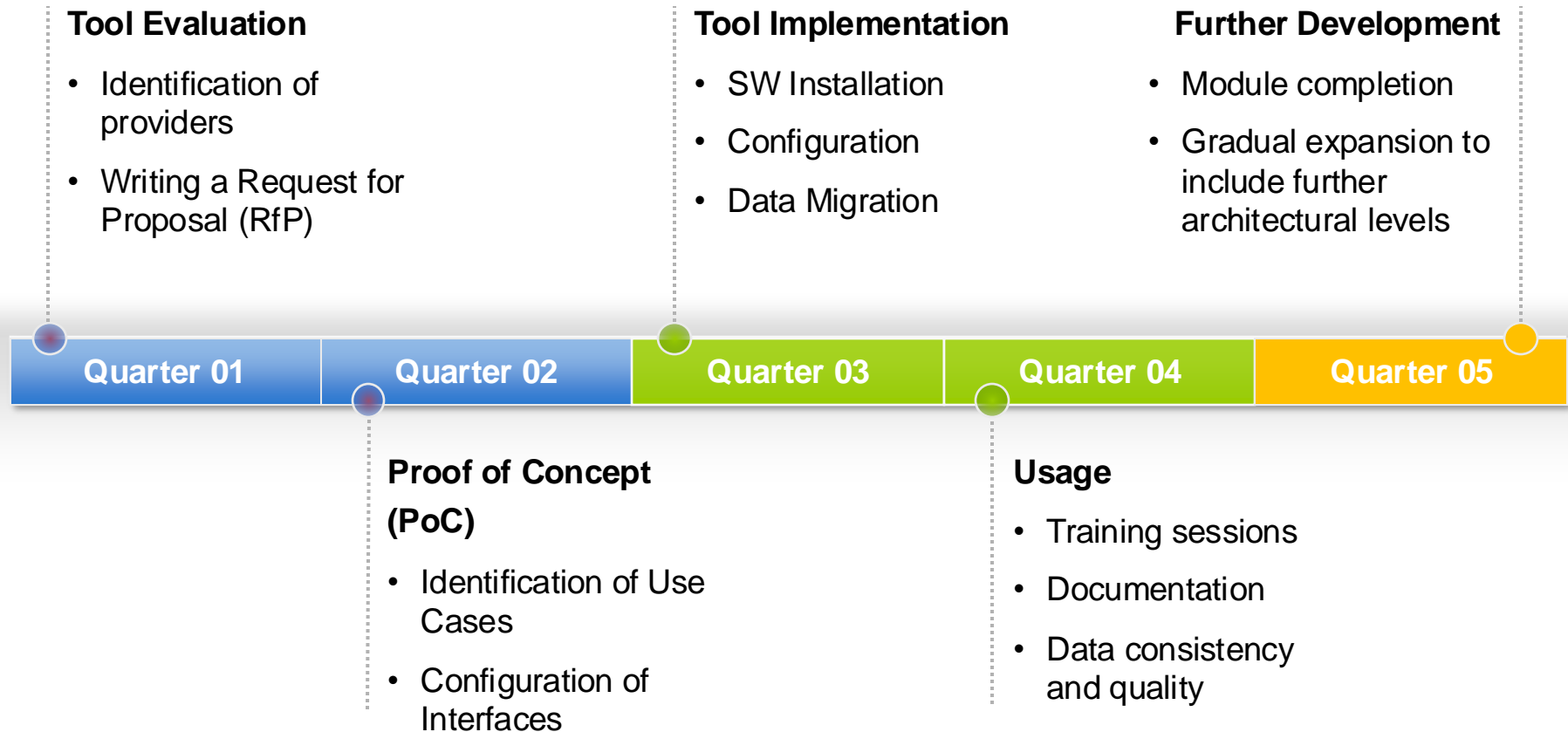


# Target Architecture: Example



**Green = CURRENT**  
**Orange = TARGET**









**No current solution to capture data automatically**, but possibly

- Interfaces to other systems like a CMDB
- AI can do certain basic tasks (i.e. write descriptions)

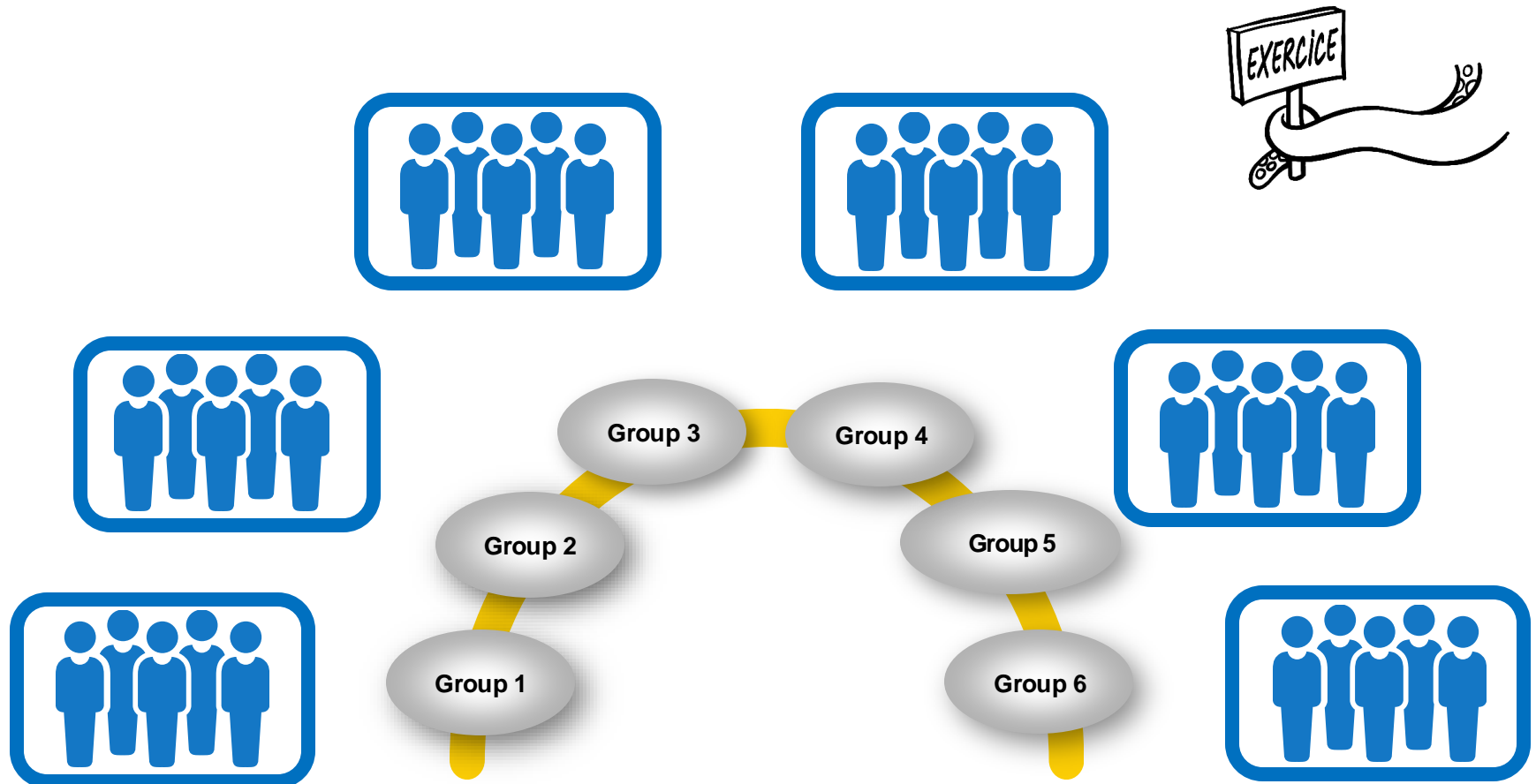
**A data maintenance Governance needs to be defined:**

- Application Owners (operating/current view)
- Change-Management (project/future view)
- IT-Architect (Solution Designs, Sanity checks)



## Explore how EA can support the development and implementation of...

1. Business Continuity Management
2. Lifecycle
3. Projects
4. Skill Management
5. Risk and Compliance
6. Communication





## A development from structured, methodology-driven approaches to more adaptive, business-focused strategies

### **Zachman Framework** (late 1980s)

Developed by John Zachman in the late 1980s, this framework is considered one of the earliest and most influential EA frameworks. It provides a structured way of viewing and defining an enterprise, categorizing architecture artifacts into a 6x6 matrix based on different perspectives (e.g., executive, architect, engineer) and aspects (e.g., data, function, network).

### **TOGAF** (1995)

Developed by The Open Group, TOGAF is based on the US Department of Defense's Technical Architecture Framework for Information Management (TAFIM). It provides a comprehensive approach for designing, planning, implementing, and governing enterprise information architecture.

### **DoDAF** (2003)

Detailed framework for the development of enterprise architecture within the US Department of Defense, focusing on mission-critical systems and interoperability among defense systems

### **BIZBOK** (Late 2010s to 2020s)

Developed by the Business Architecture Guild, BIZBOK (Business Architecture Body of Knowledge) focuses on the practice of business architecture, complementing traditional IT-focused EA frameworks by emphasizing business strategy, value streams, and capabilities



	Why	How	What	Who	Were	When
Contextual	Goal List	Process List	Material List	Organisational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organisational Unit & Role Relationship Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details



## 1. Business Architecture

- Business Strategy: Overall direction and goals of the organization
- Business Processes: Workflows and procedures that achieve business objectives
- Organizational Structure: Roles, responsibilities, and relationships
- Governance: Policies, procedures, and standards for business operations and decision-making
- Business Capabilities: Essential functions and competencies the organization needs to achieve its objectives

## 2. Applications Architecture

- Application Portfolio: Catalogue of all the software applications
- Application Interactions: How different applications communicate and integrate with each other
- Functional Requirements: Specific business functions and services that applications need to support.
- Application Lifecycle: Development, maintenance, and retirement of applications

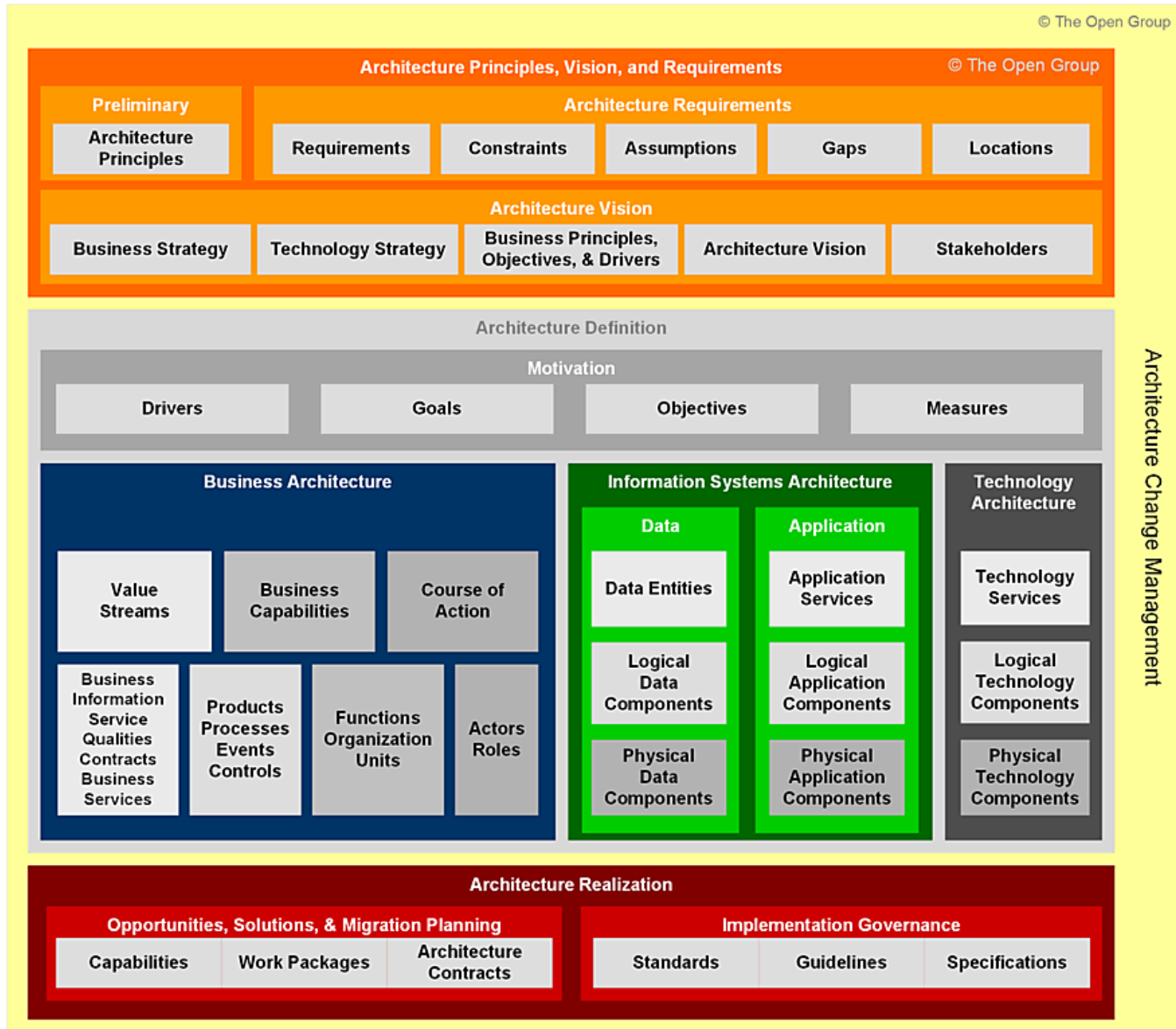
## 3. Data Architecture

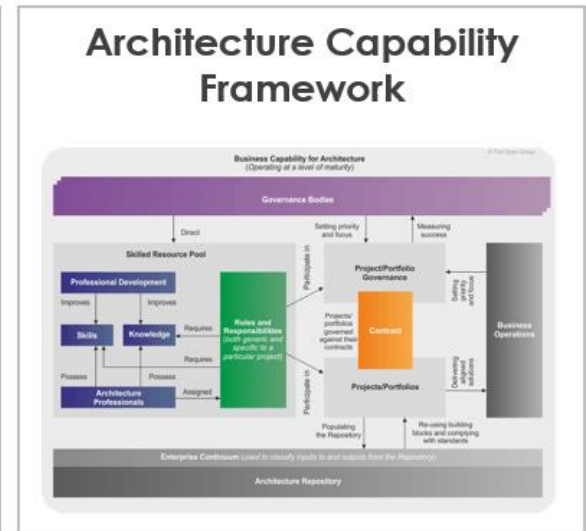
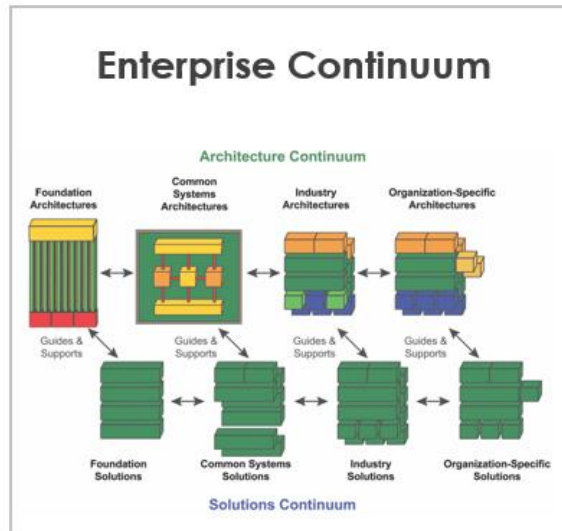
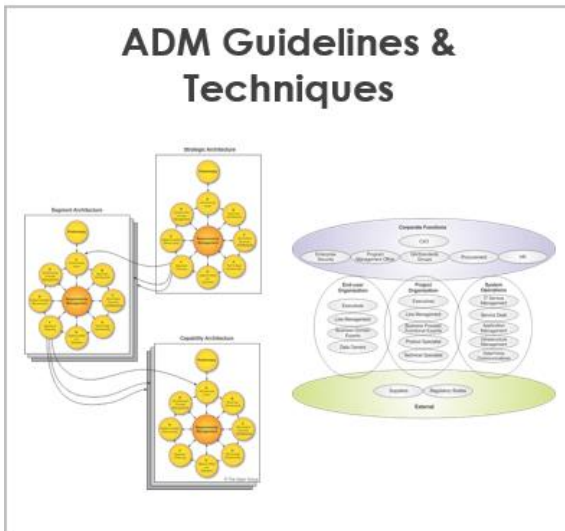
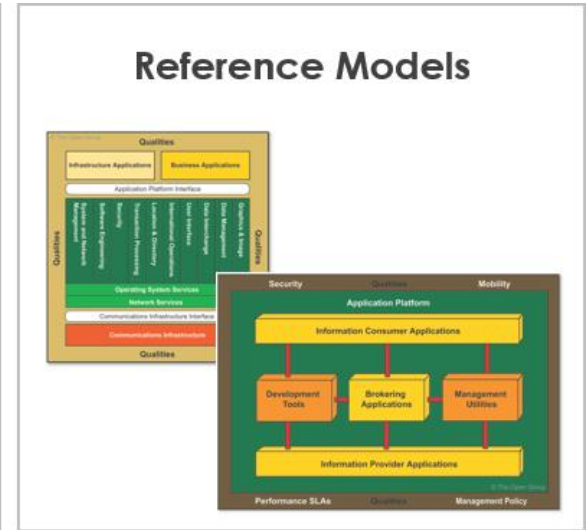
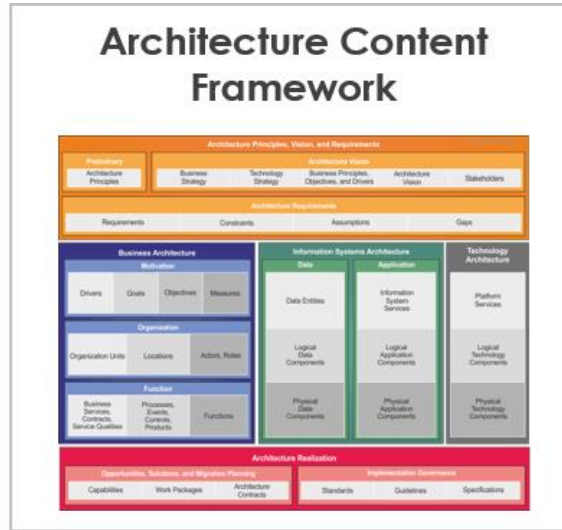
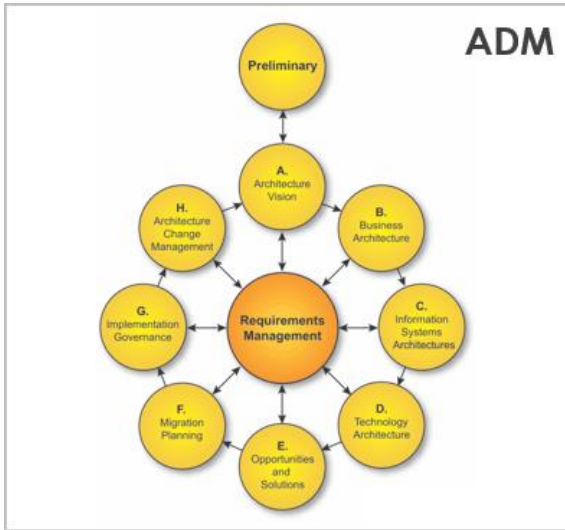
- Data Models: Logical and physical structures that represent data entities and their relationships.
- Data Management: Practices and policies for data governance, quality, and lifecycle management.
- Data Flows: Movement and transformation of data between different parts of the organization.
- Data Storage: Physical and virtual storage solutions for housing data

## 4. Technology Architecture

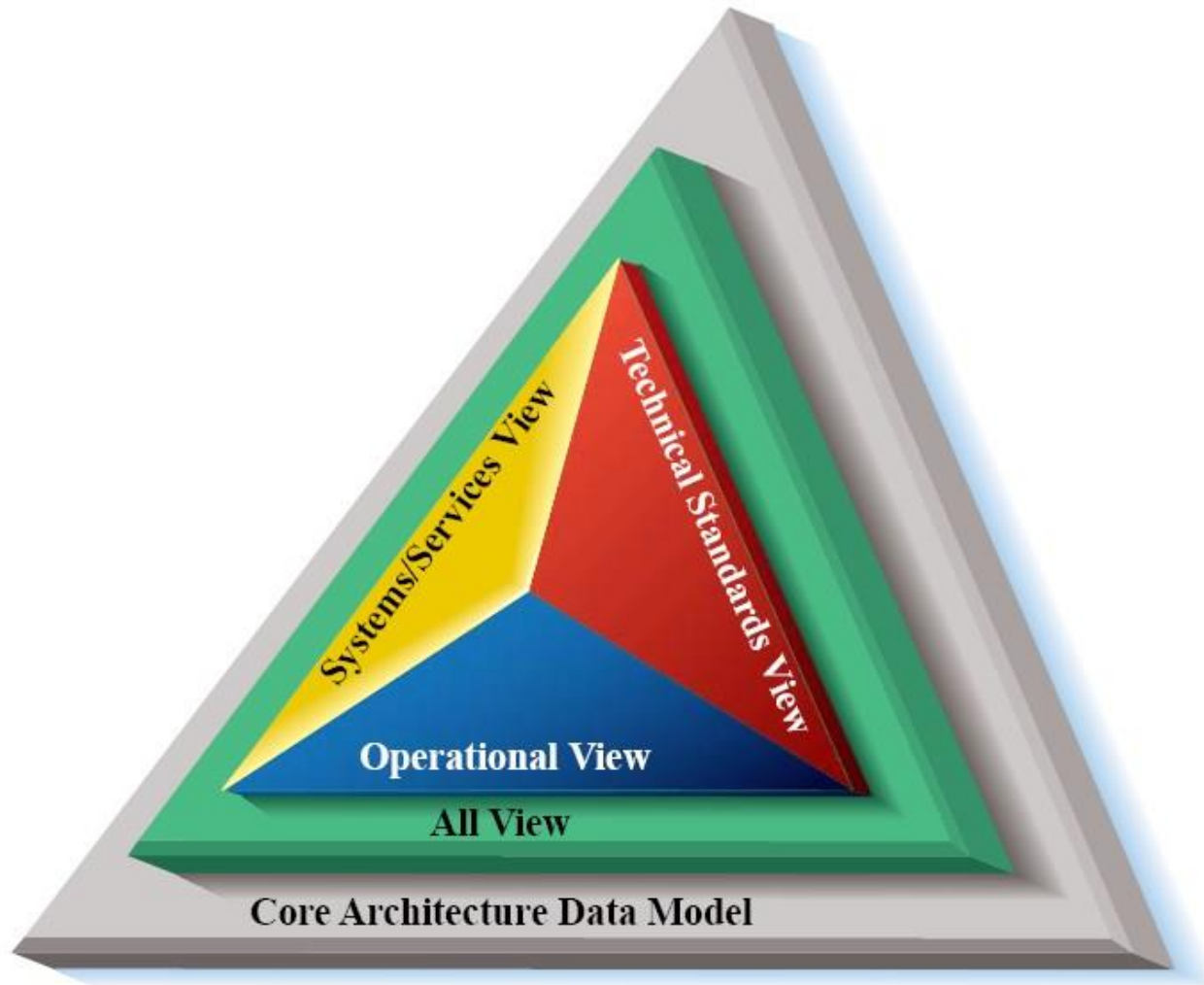
- Technology Components: Hardware and software that make up the IT environment.
- Technical Standards: Guidelines and protocols for ensuring compatibility and interoperability of technology components.
- Infrastructure Services: Underlying IT services that support business operations, such as networking, security, and cloud services.
- Technology Roadmap: Plan for future technology upgrades and innovations

# The TOGAF Content Framework

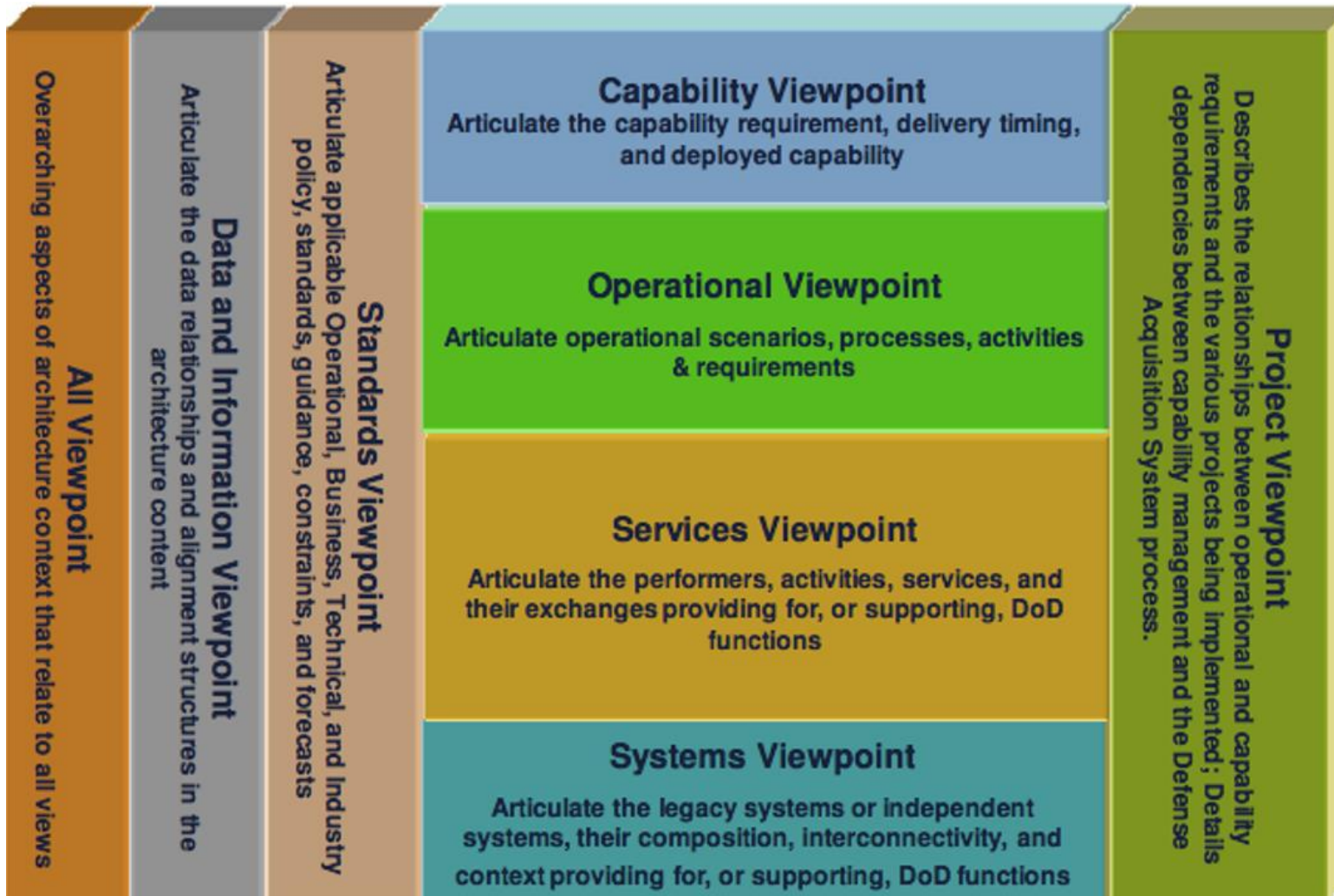




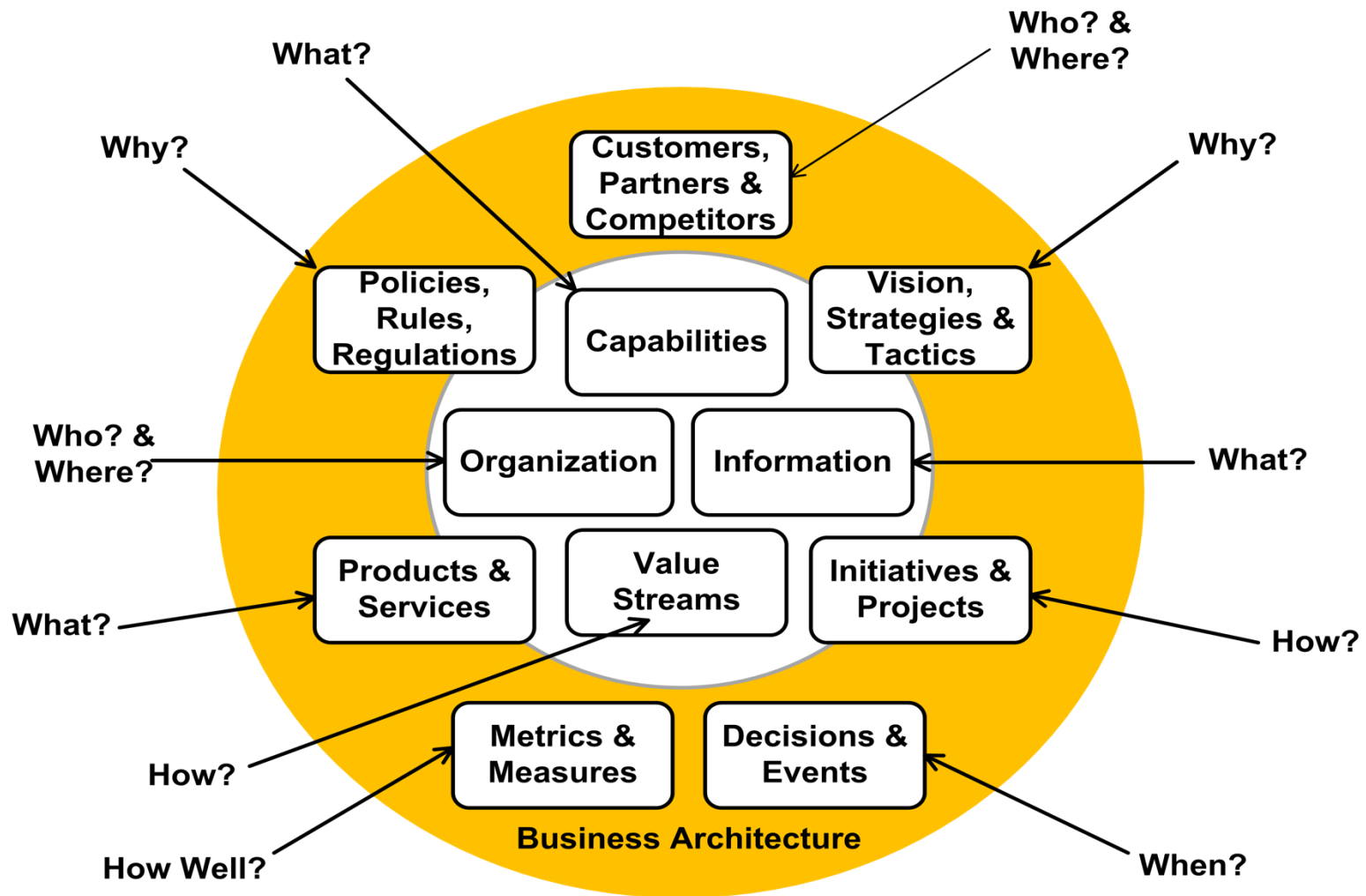
ADM: Architecture Development Method

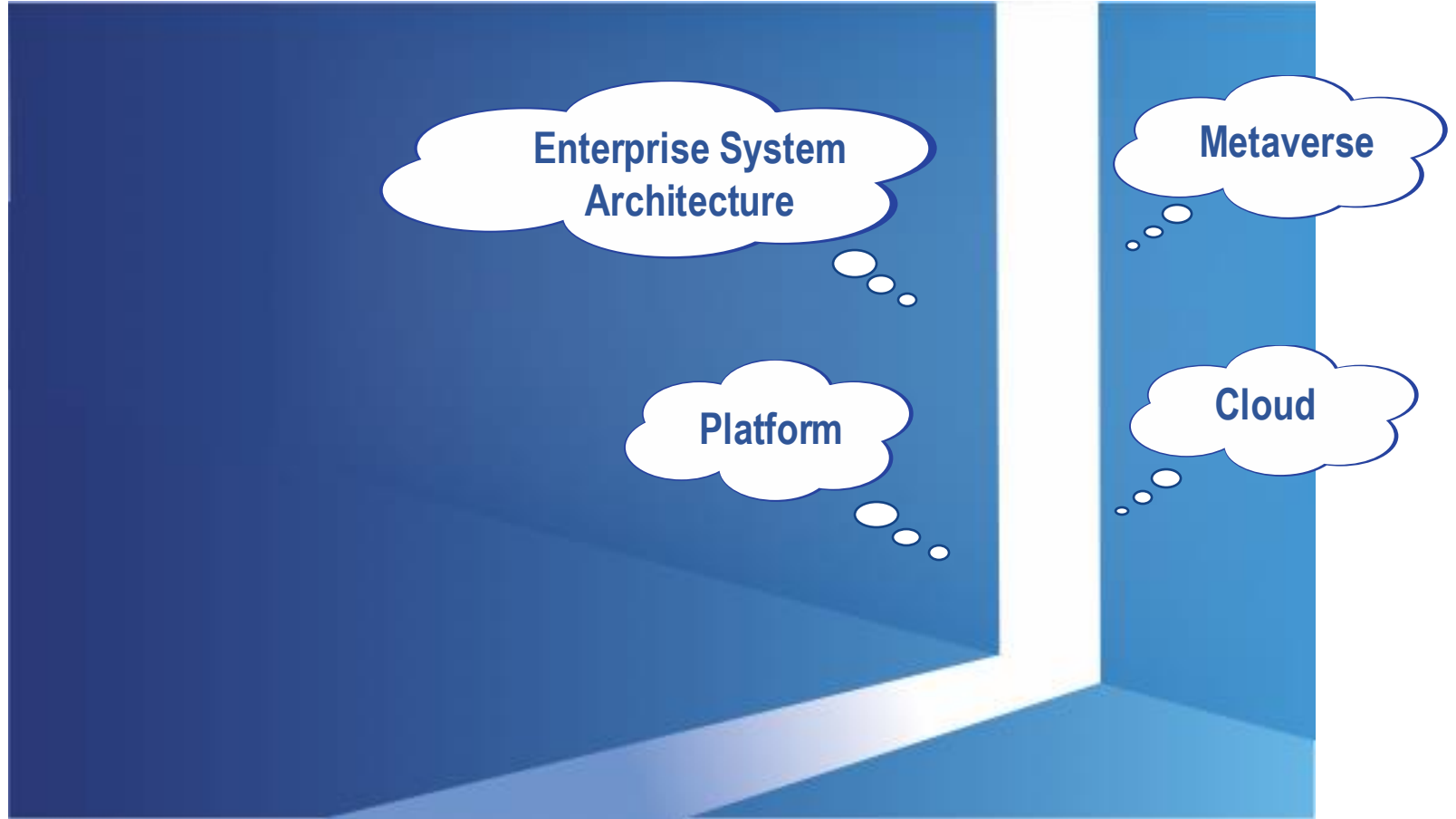




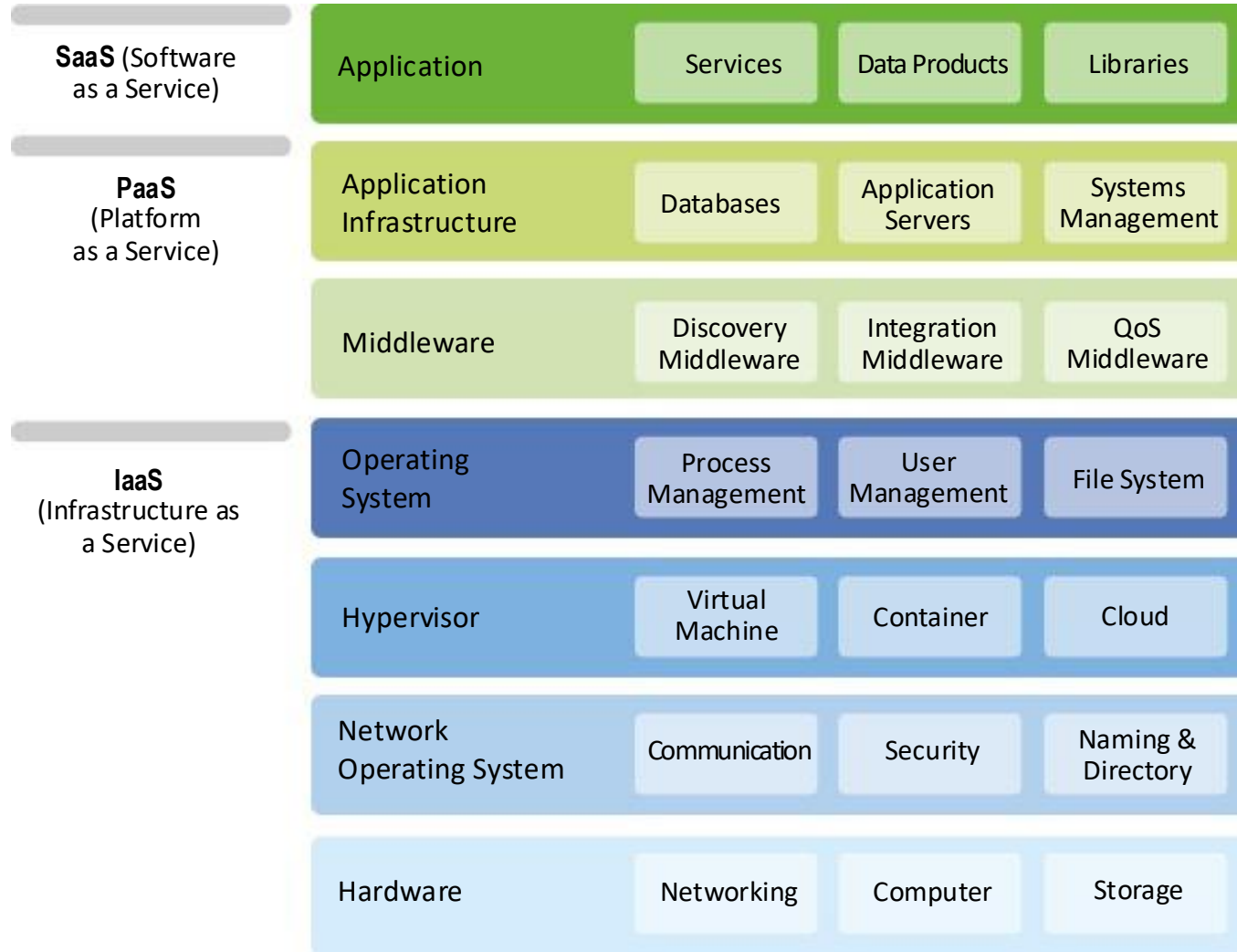


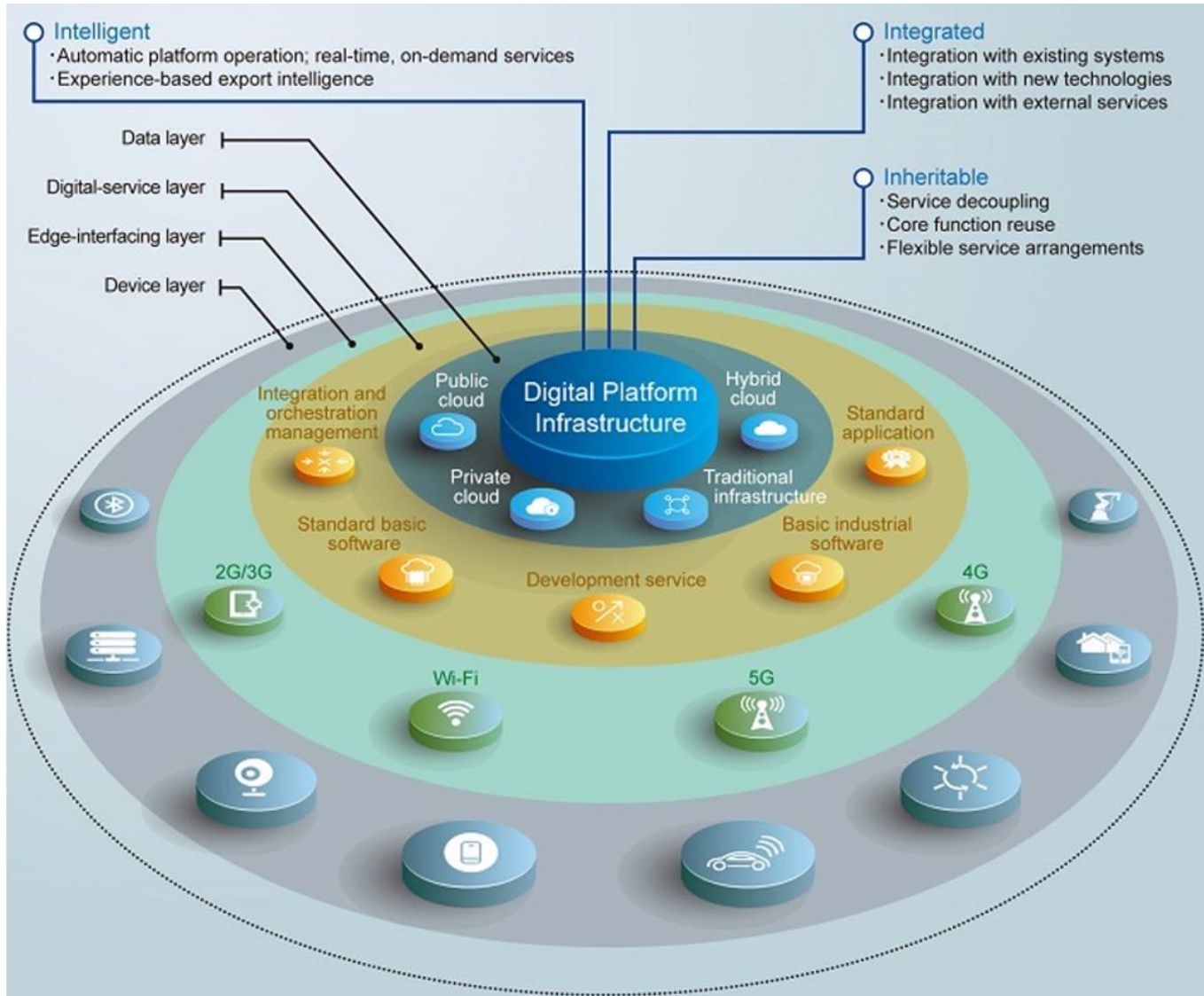
# Business Architecture Diagram (BIZBOK)





# Cloud Architecture: IaaS, PaaS, SaaS





# Example: Open Banking with bLink from SIX



**Service Provider**  
(data provider)



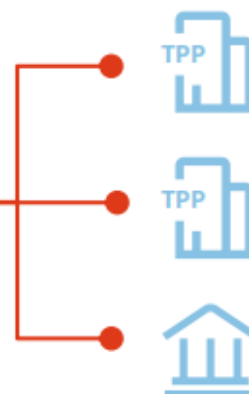
**Platform**



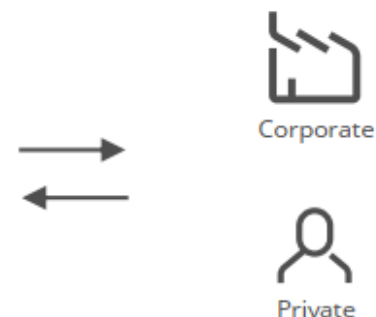
API-Standards:



**Service User**  
(data consumer)

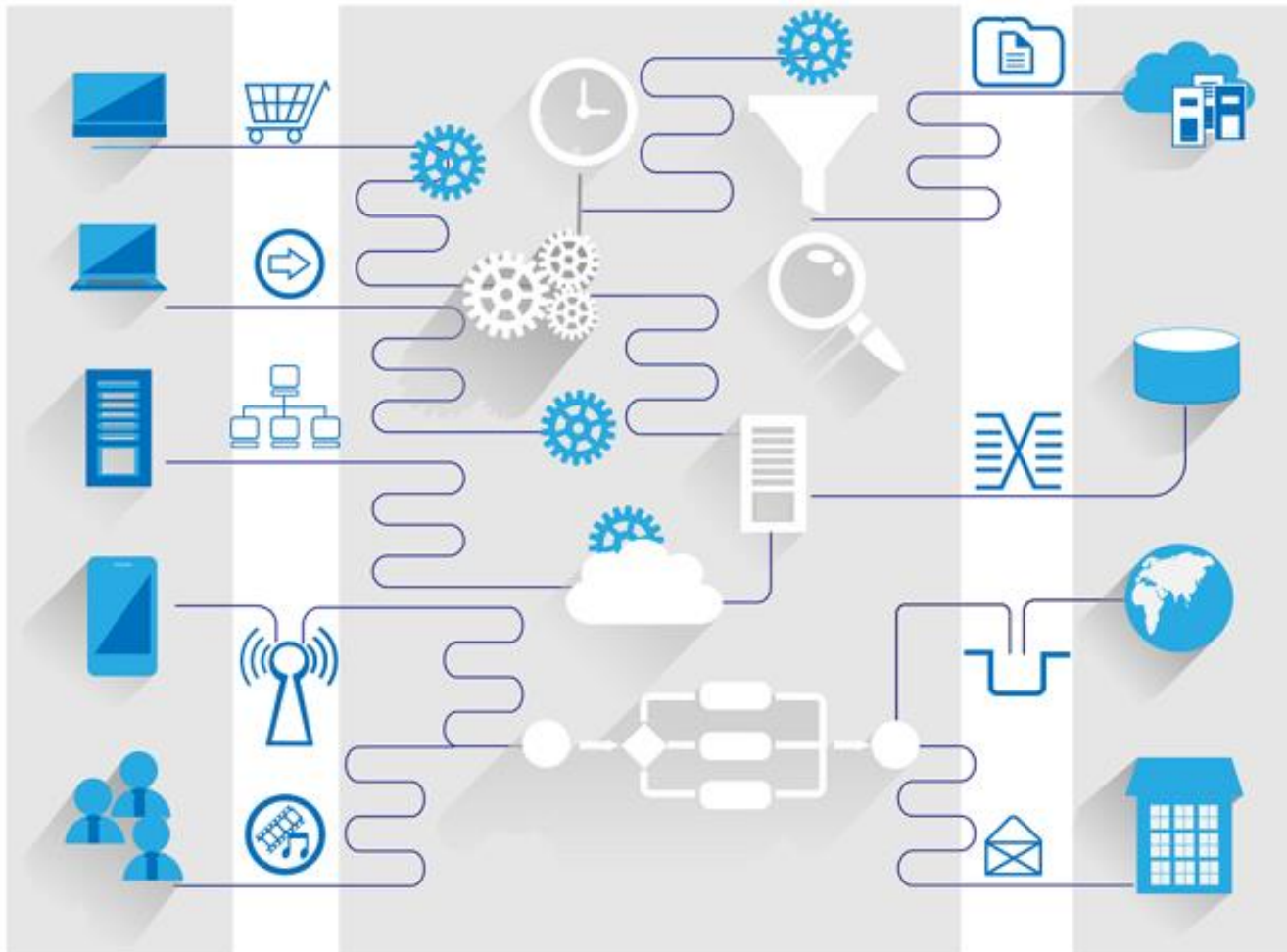


**End Customer**





**ESA as a sub-discipline within EA, specifically addressing the architecture of IT systems to support the overall enterprise strategy**







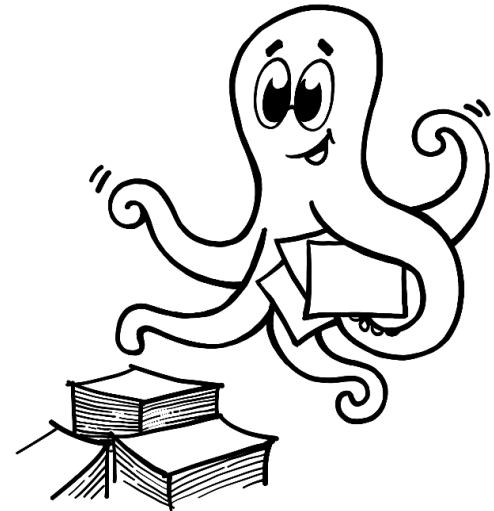
**Set of parameters and rules, and then use computational tools to automatically generate complex and dynamic forms based on these inputs**







- To be able to model the architecture of an IT system
- To understand the different layers and views of an architecture model
- To understand the importance of the relationships between the different elements of the architecture
- To understand the relevance of architectural visualization
- To be familiar with EA frameworks and understand the relevance of best practices
- To anticipate future developments in architecture





- Arnold, I. et al. (2009) Software-Architektur, Spektrum
- Evans E. (2003) Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison-Wesley Professional
- Fowler, M. and Rice, David, (2003) Patterns of Enterprise Application Architecture, Addison-Wesley Professional
- Hanschke I. (2024) Enterprise Architecture Management - einfach und effektiv, Hanser
- Kotusev S. (2021) The Practice of Enterprise Architecture, SK-Publishing
- Martin Robert C. (2017) Clean Architecture: A Craftsman's Guide to Software Structure and Design, Pearson



**KNOWLEDGE**