1. The Concept of Enterprise Architecture

What is Architecture? - "Architecture refers to the conceptual framework and structure that defines the elements and components of a system and their relationships."

What is an Enterprise? - "An Enterprise is a large organization or complex system, typically consisting of multiple departments, processes, and resources working together to achieve common objectives."

Definition of Enterprise Architecture (EA) - "Enterprise Architecture (EA) is a comprehensive framework used to manage and align an organization's IT assets, business processes, information systems, and technology infrastructure with its business goals and strategies."

EA Domains

EA encompasses various domains, including:

- **Business Architecture**: "Defines the business strategy, governance, organization, and key business processes."
- **Data Architecture**: "Describes the structure and organization of an enterprise's data assets."
- **Application Architecture**: "Defines the applications needed to manage data and support business processes."
- **Technology Architecture**: "Covers the hardware, software, networks, and other technology that support applications and data."

Where Did Enterprise Architecture Start?

- **Origin**: "The concept of Enterprise Architecture originated in the late 1980s and early 1990s as organizations began to recognize the need for a more integrated approach to managing IT resources and aligning them with business objectives."
- **Pioneer**: "One of the first formal approaches to EA was developed by John Zachman in 1987, known as the Zachman Framework, which is still influential today."
- **Evolution**: "The field has since evolved with other frameworks like TOGAF (The Open Group Architecture Framework), FEAF (Federal Enterprise Architecture Framework), and others."

Goals of Enterprise Architecture

The primary goals of Enterprise Architecture include:

- **Alignment of IT and Business Strategy**- "Ensuring that technology investments and initiatives support the organization's business goals."
- **Improved Decision Making**- "Providing a clear framework and guidelines for decision-making processes related to IT and business operations."
- **Optimized Resource Use**- "Enhancing the efficiency and effectiveness of IT resources and reducing redundancies."
- **Agility and Flexibility**- "Enabling the organization to respond quickly to changes in the business environment."
- **Risk Management**- "Identifying and mitigating risks related to IT and business processes."
- **Compliance**- "Ensuring that the organization's IT systems and processes comply with regulatory requirements and standards."
- **Innovation Support**- "Fostering innovation by providing a structured approach to integrating new technologies and processes."

Evolution of Enterprise Architecture

Enterprise Architecture (EA) has evolved significantly since its inception:

- **1980s: Emergence of EA Concepts**
    - "The need for aligning IT with business strategies led to the creation of formal EA frameworks."
    - "John Zachman introduced the Zachman Framework in 1987, which provided a structured approach to organizing enterprise architecture."
- **1990s: Framework Proliferation**
    - "The 1990s saw the development of several EA frameworks, such as TOGAF (The Open Group Architecture Framework) and FEAF (Federal Enterprise Architecture Framework)."
    - "Organizations began adopting these frameworks to better manage their IT resources and align them with business goals."
- **2000s: Integration and Maturity**
    - "EA began focusing more on integrating business and IT strategies."
    - "The concept of Service-Oriented Architecture (SOA) emerged, promoting modular and reusable IT services."

    - "EA practices matured, with an emphasis on governance, standardization, and interoperability."

- **2010s: Digital Transformation and Agility**
    - "EA adapted to the rise of digital transformation, cloud computing, and agile methodologies."
    - "EA frameworks evolved to support rapid innovation and scalability."
    - "Concepts like Business Capability Modeling and Agile EA gained traction."
- **2020s: AI, Data-Driven EA, and Continuous Evolution**
    - "The adoption of AI, machine learning, and data-driven decision-making influenced EA practices."
    - "EA became more dynamic, with a focus on continuous improvement, real-time data integration, and strategic foresight."
    - "Organizations began leveraging EA for sustainable growth and resilience in a rapidly changing environment."

Importance of Implementing Enterprise Architecture

Implementing Enterprise Architecture is crucial for several reasons:

1. **Strategic Alignment**: "Ensures that IT investments and initiatives are aligned with the organization's business goals and objectives."
2. **Improved Efficiency**: "Reduces redundancies and optimizes resource use, leading to cost savings and better performance."
3. **Enhanced Agility**: "Provides the flexibility to quickly adapt to changes in the business environment or market conditions."
4. **Risk Management**: "Identifies and mitigates risks related to IT and business processes, ensuring operational continuity."
5. **Compliance and Governance**: "Ensures that the organization's IT systems and processes comply with regulatory requirements and standards."
6. **Innovation Support**: "Facilitates the adoption of new technologies and processes, enabling continuous improvement and innovation."

Roles in an Enterprise Architecture

- **Enterprise Architect**: "An Enterprise Architect is a senior IT professional responsible for developing and implementing an organization's Enterprise Architecture." "They ensure that the organization's IT infrastructure aligns with business goals, integrates smoothly across various domains, and supports the overall strategy."

  - **Key responsibilities include**:
    - "Developing and maintaining the EA framework."
    - "Aligning IT initiatives with business strategy."
    - "Ensuring scalability, security, and interoperability of IT systems."
    - "Collaborating with stakeholders to understand business needs and translate them into architectural solutions."

- **Other Important Roles**:
  - **Business Architect**: "Focuses on defining and aligning business processes and strategies with IT solutions."
  - **Data Architect**: "Responsible for data management, modeling, and governance to ensure data quality and accessibility."
  - **Application Architect**: "Designs and oversees the implementation of software applications that support business processes."
  - **Technology Architect**: "Manages the technology infrastructure, ensuring that it meets the needs of the business and supports the overall architecture."
  - **Solution Architect**: "Provides technical leadership and ensures that specific solutions align with the overall EA framework."
  - **EA Governance Board**: "A group of senior leaders responsible for overseeing the EA strategy, ensuring compliance with standards, and making key decisions."

Enterprise Architecture Planning (EAP)

- **Definition**: "Enterprise Architecture Planning (EAP) is the process of defining the architecture of an enterprise in a structured and strategic way." "It involves creating a roadmap that guides the organization’s IT investments and development efforts to align with business goals."
- **Activities Involved in EAP**:
  - **Assessing Current Architecture**: "Analyzing the existing IT infrastructure, processes, and systems."
  - **Defining Future Architecture**: "Developing a target architecture that aligns with the organization's long-term goals."
  - **Gap Analysis**: "Identifying gaps between the current and future architecture and developing strategies to bridge them."

    - **Developing a Roadmap**: "Creating a detailed plan for transitioning from the current state to the target architecture."
    - **Governance and Compliance**: "Establishing governance structures to ensure that the architecture is implemented according to plan and adheres to standards."
    - **Communication and Change Management**: "Engaging stakeholders, communicating the EAP, and managing the changes required for its implementation."

Steps in the Enterprise Architecture Life Cycle (EALC)

- **Architecture Vision**: "Define the scope, objectives, and stakeholders for the EA initiative."
- **Business Architecture**: "Develop a detailed understanding of the business strategy, processes, and requirements."
- **Information Systems Architecture**: "Design the data and application architecture to support business needs."
- **Technology Architecture**: "Develop the technology infrastructure needed to support the information systems architecture."
- **Opportunities and Solutions**: "Identify potential solutions and opportunities for improvement."
- **Migration Planning**: "Develop a plan to move from the current state to the target architecture."
- **Implementation Governance**: "Monitor and guide the implementation to ensure alignment with the architecture."
- **Architecture Change Management**: "Manage changes to the architecture as the organization evolves."

Benefits of EAP (According to CompTIA)

- **Strategic Alignment**: "Ensures IT initiatives are aligned with business strategies."
- **Cost Reduction**: "Identifies redundant processes and systems, leading to cost savings."
- **Improved Agility**: "Enhances the organization’s ability to respond quickly to changes in the business environment."
- **Better Decision Making**: "Provides a structured framework for making informed decisions about IT investments and initiatives."
- **Increased Efficiency**: "Optimizes resources and processes, improving overall efficiency."

- **Enhanced Security**: "Strengthens the security of IT systems through a well-defined architecture."

Architecture Myths

- **EA is Only About IT**
  - **Myth**: "EA is solely focused on IT."
  - **Reality**: "EA encompasses business strategy, processes, data, and technology."
- **EA Stifles Innovation**
  - **Myth**: "EA creates rigid structures that hinder innovation."
  - **Reality**: "EA provides a structured approach that supports innovation and adaptability."
- **EA is a One-Time Activity**
  - **Myth**: "EA is a project with a start and end date."
  - **Reality**: "EA is an ongoing process that evolves with the organization."
- **EA is Only for Large Organizations**
  - **Myth**: "Only large enterprises need EA."
  - **Reality**: "EA is valuable for organizations of all sizes to align IT and business strategies."
- **EA is Too Expensive**
  - **Myth**: "EA is costly and not worth the investment."
  - **Reality**: "EA can lead to significant cost savings and improved efficiency over time."

Types of Enterprise Architecture (Sparx Systems)

According to Sparx Systems, the different types of EA frameworks include:

- **The Zachman Framework**: "A matrix-based approach that organizes architecture into different perspectives and aspects."
- **The Open Group Architecture Framework (TOGAF)**: "A widely adopted framework that provides a detailed methodology for developing EA."
- **Federal Enterprise Architecture Framework (FEAF)**: "A framework used by the U.S. federal government to ensure interoperability and efficiency across agencies."

- **Gartner Methodology**: "A flexible approach that emphasizes continuous improvement and adaptability in EA."
- **DNV GL's EA3 Cube Framework**: "A framework that views the enterprise as a cube, with different layers representing various aspects of the architecture."

Enterprise Structure

- **Definition**: "Enterprise Structure refers to the organizational framework that defines how an enterprise is organized and how its various components interact."
- **This structure can include**:
    - "Organizational Hierarchy"
    - "Governance Model"
    - "Business Processes"
    - "Information Flow"
    - "Technology Infrastructure"
- **Note**: "For effective functioning of an organization, all units must be interlinked to perform a common goal." "In an enterprise management system like SAP (Systems Applications and Products in Data Processing), the structure is the basis for the configuration of other parameters."
- **Relationship to EA**: "The enterprise structure is a critical component of Enterprise Architecture (EA), as it provides the context within which the architecture is designed and implemented." "EA must consider the existing structure while also guiding changes to optimize performance, efficiency, and alignment with strategic goals."

IT Value Chain

- **Definition**: "The IT Value Chain is a conceptual framework that describes how IT functions and processes create value within an organization." "It mirrors the traditional business value chain, focusing on how IT contributes to the overall performance and competitiveness of the organization by enabling business processes, driving innovation, and supporting strategic goals."
- **Components of the IT Value Chain**:
    - "IT Strategy and Governance"
    - "IT Infrastructure Management"
    - "Application Development and Management"

  - "Data Management and Analytics"
  - "IT Service Management"
  - "IT Security and Risk Management"
  - "Innovation and Emerging Technologies"

Summary of Key Concepts

- **Enterprise Architecture (EA)**
  - "EA is a framework that aligns an organization's IT infrastructure, business processes, and strategy to achieve business goals."
  - "It encompasses various domains like business, data, application, and technology architecture."
- **Enterprise Structure**
  - "The enterprise structure is the organizational framework that includes the hierarchy, governance, business processes, information flow, and technology infrastructure."
  - "It provides the context within which EA is designed and implemented."
- **Relationship Between Enterprise Structure and EA**
  - "The enterprise structure forms the foundation for EA design, guiding how IT systems and processes are aligned with business goals."
  - "EA, in turn, optimizes the enterprise structure to enhance efficiency, adaptability, and strategic alignment."
- **IT Value Chain**
  - "The IT Value Chain describes how IT functions... create value for the organization."
  - "It highlights the importance of IT in driving efficiency, innovation, and competitive advantage."
- **Alignment and Integration**
  - "EA ensures that all components of the IT Value Chain are aligned with the enterprise structure and strategic objectives, optimizing resources, reducing redundancies, and enhancing overall organizational performance."
- **Importance**
  - **Strategic Alignment**: "EA ensures that IT initiatives support business goals."
  - **Efficiency**: "The IT Value Chain helps identify areas for process improvement."

  - **Innovation**: "EA and the IT Value Chain foster innovation by integrating new technologies and approaches."
  - **Risk Management**: "EA provides governance and risk management to protect IT assets and ensure compliance."

---

2. Enterprise Architecture Frameworks

Enterprise Architecture (EA)

- **Definition**: "An enterprise architecture (EA) is a conceptual blueprint for conducting enterprise analysis, design, planning, and implementation, using a comprehensive

---

Enterprise Architecture Frameworks

- **Definition**: "An enterprise architecture framework (EA framework) defines how to create and use an enterprise architecture, which provides principles and practices for creating and using the architecture description of a system."
- **Function**: "It provides tools and approaches that help enterprise architects to optimize different architecture domains across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy."

The Major EA Frameworks

- **Architecture Framework Definition**: "Architecture Framework is a prefabricated structure that organizes the system architecture of an enterprise into complementary projections called Views."
- **Three different types of frameworks**:
  1. "Template Framework – Zachman Framework"
  2. "Content Framework, i.e. The Open Group Architecture Framework (TOGAF), the DoD Architecture Framework (DoDAF), the variants (MODAF and NAF),"
  3. "Unified Profile for EA frameworks from OMG (UDPM, UAF):"

The Zachman Framework

- **Definition**: "Zachman Framework is an enterprise ontology and is a fundamental structure for Enterprise Architecture which provides a way of viewing an enterprise and its information systems from different perspectives and showing how the components of the enterprise are related."

- **Scope**: "The Zachman Framework goes beyond IT. It offers structural connections into any aspect of an enterprise."
- **Rows**: "The rows of Zachman Framework focus on describing the enterprise from six viewpoint perspectives of the stakeholders." "These six perspectives are based on English language interrogatives 'what', 'where', 'who', 'when', 'why', and 'how' (known as W5H)."
- **Columns**: "The columns of the framework consist of a set of artifacts that are a description of the enterprise from a specific viewpoint of a group of stakeholders." "The stakeholders are generally grouped as planners, owners, designers (architects), implementers, sub-constructors, users, or sometimes represented as viewpoints: scope context, business concepts, system logic, technology, physics, component assemblies, and operations classes."

Example: A Healthcare Organization

This table illustrates how the Zachman Framework can be applied:

- **Perspective**: Business
  - **Question**: "What are the business goals and objectives?"
  - **Example**: "Improve patient outcomes, reduce costs, enhance patient satisfaction"
- **Perspective**: Data
  - **Question**: "What data is needed to support the business?"
  - **Example**: "Patient demographics, medical history, treatment plans, billing information"
- **Perspective**: Application
  - **Question**: "What applications are needed to process the data?"
  - **Example**: "Electronic health records (EHR), patient scheduling system, billing system"
- **Perspective**: Technology
  - **Question**: "What technology infrastructure is required to support the applications?"
  - **Example**: "Servers, network, storage, security"
- **Perspective**: People
  - **Question**: "Who is responsible for using the applications and managing the data?"
  - **Example**: "Doctors, nurses, administrators, IT staff"
- **Perspective**: Motivation

  - **Question**: "What are the business drivers and constraints?"
  - **Example**: "Government regulations, industry standards, funding limitations"

Using the W5H Questions with the Zachman Framework

- **Business Perspective**
  - "Who: Who are the key stakeholders and decision-makers?"
  - "What: What are the business goals and objectives?"
  - "When: When do we need to achieve these goals?"
  - "Where: Where will the business operate?"
  - "Why: Why are these goals important to the organization?"
  - "How: How will we measure success and progress towards these goals?"
- **Data Perspective**
  - "Who: Who will be responsible for data management and governance?"
  - "What: What data is needed to support business operations and decision-making?"
  - "When: When will data be collected, stored, and accessed?"
  - "Where: Where will data be stored and processed?"
  - "Why: Why is this data important for the business?"
  - "How: How will data be collected, stored, and protected?"
- **Application Perspective**
  - "Who: Who will develop, maintain, and use the applications?"
  - "What: What applications are needed to support business processes and data requirements?"
  - "When: When will applications be developed, deployed, and updated?"
  - "Where: Where will applications be hosted and accessed?"
  - "Why: Why are these applications necessary for the business?"
  - "How: How will applications be integrated with other systems and processes?"
- **Technology Perspective**
  - "Who: Who will manage the technology infrastructure and support applications?"
  - "What: What technology components are needed to support the applications and data?"

    - "When: When will technology infrastructure be acquired, deployed, and maintained?"
    - "Where: Where will technology infrastructure be located and managed?"
    - "Why: Why is this technology infrastructure necessary for the business?"
    - "How: How will technology infrastructure be secured, monitored, and updated?"
- **People Perspective**
    - "Who: Who will be involved in using, supporting, and managing the enterprise architecture?"
    - "What: What skills and training will be required for people to effectively use the enterprise architecture?"
    - "When: When will people need to be trained and supported?"
    - "Where: Where will people be located and work?"
    - "Why: Why is it important to have the right people in place to support the enterprise architecture?"
    - "How: How will people be organized, motivated, and rewarded?"
- **Motivation Perspective**
    - "Who: Who is driving the enterprise architecture initiative?"
    - "What: What are the business drivers and constraints that influence the enterprise architecture?"
    - "When: When did the need for the enterprise architecture arise?"
    - "Where: Where are the opportunities and challenges for the enterprise architecture?"
    - "Why: Why is the enterprise architecture important for the organization's success?"
    - "How: How will the enterprise architecture be aligned with the organization's overall strategy?"

Content Frameworks

Includes "The Open Group Architecture Framework (TOGAF), the DoD Architecture Framework (DoDAF), the variants (MODAF and NAF)"

The Open Group Architectural Framework (TOGAF)

- **Origin**: "The Open Group Architectural Framework (TOGAF) was first developed in 1995 created and owned by The Open Group, which was based on the Department of Defense's Technical Architecture Framework for Information Management."
- **Prevalence**: "It is one of the most common framework structures in business today and accounts for over 80 percent of the entire business framework structure."
- **Components**: "TOGAF contains all the needed pieces for a powerful framework." "It has a common vocabulary to use, recommended standards and compliance methods, suggested software and tools, and even a method to define best practices."
- **Process**: "TOGAF is often viewed as more an overarching process." "The details and methods contained within TOGAF help guide businesses through any step of business organization."
- **Key Element**: "A key element of TOGAF is the Architecture Development Method (ADM) which specifies a process for developing enterprise architecture."
- **Six Important Parts**:
    1. "Architecture Development Method (ADM)"
    2. "ADM Guidelines and Techniques"
    3. "Architecture Content Framework"
    4. "Enterprise Continuum & Tools"
    5. "TOGAF Reference Models"
    6. "Architecture Capability Framework"

DOD ARCHITECTURE FRAMEWORK (DODAF)

- **Definition**: "The DoD Architecture Framework (DoDAF) is the industry-standard Enterprise Architecture Framework for defense and aerospace applications." "It defines how to organize the specification of enterprise architectures for the U.S. Department of Defense (DoD) applications."
- **Requirement**: "All major DoD weapons and information technology system procurements are required to document their enterprise architectures using the view products prescribed by the DoDAF."
- **Function**: "DoDAF establishes data element definitions, rules, and relationships and a baseline set of products for consistent development of systems, integrated, or federated architectures."
- **Suitability**: "It is well suited to large systems and systems-of-systems (SoSs) with complex integration and interoperability issues."
- **Objective**: "The objective of DoDAF is to concretely define models and concepts that are usable in the DoD's core processes:"

  - "Joint Capabilities and Integration Development (JCIDS)"
  - "Planning, Programming, Budgeting, and Execution (PPBE)"
  - "Defense Acquisition System (DAS)"
  - "Systems Engineering (SE)"
  - "Operational Planning (OPLAN)"
  - "Capability Portfolio Management (CPM)"

- **Core Process Examples**:

  1. **Joint Capabilities and Integration Development (JCIDS)**
     - **Objective**: "To ensure that new capabilities align with the DoD's strategic objectives and can be effectively integrated into existing systems."
     - **Example**: "DoDAF models can be used to analyze how a proposed new weapon system will interact with existing command and control systems..."
  2. **Planning, Programming, Budgeting, and Execution (PPBE)**
     - **Objective**: "To provide a clear and consistent basis for resource allocation and decision-making."
     - **Example**: "DoDAF models can be used to assess the cost-benefit analysis of different acquisition options..."
  3. **Defense Acquisition System (DAS)**
     - **Objective**: "To ensure that acquisition programs are well-defined, managed, and executed."
     - **Example**: "DoDAF models can be used to develop detailed system specifications, identify dependencies, and track the progress of acquisition programs..."
  4. **Systems Engineering (SE)**
     - **Objective**: "To provide a framework for designing, developing, and maintaining complex systems."
     - **Example**: "DoDAF models can be used to define system boundaries, identify interfaces, and analyze system behavior..."
  5. **Operational Planning (OPLAN)**
     - **Objective**: "To develop and execute operational plans that support the DoD's strategic objectives."

        - **Example**: "DoDAF models can be used to visualize the operational environment, identify potential threats, and assess the effectiveness of different operational options..."

6. **Capability Portfolio Management (CPM)**

    - **Objective**: "To manage the DoD's portfolio of capabilities to ensure that they are aligned with strategic objectives and meet operational needs."

    - **Example**: "DoDAF models can be used to assess the current state of the DoD's capability portfolio, identify gaps, and prioritize investments..."

Federal Enterprise Architectural Framework (FEA)

- **Note**: "The Federal Enterprise Architectural Framework (FEA) is one of the newest attempts to create a solid structure for organizations."

- **Origin**: "The US Federal Government developed it in 2006." "Its predecessor, the FEAF, started in 1996."

- **Structure**: "The FEA combines the best of both the Zachman Framework and TOGAF." "The FEA has five reference models. They cover business, service, components, technical, and data." "These five points combine with a segment model to create a perspective on how best to install enterprise architecture."

- **Legacy**: "FEA was the foundation for a massive restructuring of a high-end government." "As such, the framework is a strong core to follow when building a strong foundation for a future company."

MODAF Framework

- **Definition**: "The Ministry of Defense Architecture Framework (MODAF) is an internationally recognized enterprise architecture framework developed by the Ministry of Defence (MOD) to support defense planning and change management activities."

- **Function**: "It does this by enabling the capture and presentation of information in a rigorous, coherent and comprehensive way that aids the understanding of complex issues."

- **Views**: "MODAF provides a coherent set of rules and templates, known as ‘views’, that, when populated, provide a graphical and textual visualization of the business area being investigated." "Each view offers a different perspective on the business to support different stakeholder interests."

- **7 Categories of Views**:

    - **Strategic views (StVs)**: "define the desired business outcome, and what capabilities are required to achieve it"

    - **Operational views (OVs)**: "define (in an abstract rather than physical terms) the processes, information, and entities needed to fulfill the capability requirements"

  - **Service-oriented views (SOVs)**: "describe the services... required to support the processes described in the operational views"
  - **Systems views (SVs)**: "describe the physical implementation of the operational and service orientated views and, thereby, define the solution"
  - **Acquisition views (AcVs)**: "describe the dependencies and timelines of the projects that will deliver the solution"
  - **Technical views (TVs)**: "define the standards that are to be applied to the solution"
  - **All views (AVs)**: "provide a description and glossary of the contents of the architecture"
- **Summary**: "Each viewpoint contributes to a complete picture: strategy (why), operations (what), services (how), systems (with what), technology (under which standards), and metadata (documentation)."

NAF 4.0

- **Definition**: "The NATO Architecture Framework (NAF) is designed to ensure that architectures developed adhering to it can be understood, compared, justified and related across many organizations, including NATO and other National Defence initiatives."
- **Aim**: "The aim of the NATO Architecture Framework Version 4 (NAFv4) is to provide a standard for developing and describing architectures for both military and business use."
- **Objectives**:
  - "Provide a way to organize and present architectures to stakeholders"
  - "Specify the guidance, rules, and product descriptions for developing and presenting architecture information"
  - "Ensure a common approach for understanding, comparing, and integrating architectures,"
  - "Act as a key enabler for acquiring and fielding cost-effective and interoperable capabilities"
  - "Align with architecture references produced by international standard bodies"
- **Structure**: "NAF 4.0 defines Viewpoints, Views and Architecture Descriptions." "...these viewpoints are organized into a logical and consistent manner and presented as a “grid”."

The Latest Integration UAF from OMG

- **Definition**: "The Unified Architecture Framework® (UAF®) is based on the Unified Profile for DoDAF and MODAF™ (UPDM™)."

- **Function**: "UAF defines ways of representing an enterprise architecture that enables stakeholders to focus on specific areas of interest in the enterprise while retaining sight of the big picture."
- **Scope**: "UAF meets the specific business, operational and systems-of-systems integration needs of commercial and industrial enterprises as well as the U.S. Department of Defense (DoD), the UK Ministry of Defence (MOD), the North Atlantic Treaty Organization (NATO) and other defense organizations."
- **Rules**: "UAF provides a set of rules to enable users to create consistent enterprise architectures (as models) based on generic enterprise and system concepts with rich semantics." "These models then become the repositories from which various views can be extracted."
- **Support**: "UAF supports current DoDAF/MODAF/NAF requirements and can evolve to meet future needs:"
  - "produce standard DoDAF / MODAF / NAF products as well as commercial extensions"
  - "leverage cross-industry, standards-based approaches (e.g., MDA, UML, SysML) to enhance tool and architecture data interoperability"
  - "MDA foundation enables UAF to evolve with DoDAF v2 and beyond (i.e. security, human factors)"
  - "UAF is methodology-agnostic (structured, OO, etc.)"

---

3. Architectural Views and EA Management Plan

Architectural Views

- **View Definition**: "A view is a depiction of an entire system from the standpoint of a group of concerns."
- **Usage**: "It's used to describe a system from the perspective of several stakeholders, including end users, project managers, developers, and testers."
- **Modeling**: "Hence, a software architecture is modeled and documented using architecture views."
- **Definition Methods**: A software architecture can be defined using the following:
  - "UML (Unified Modeling Language)"
  - "Architecture View Model (4+1 view model)"
  - "ADL (Architecture Description Language)"

UML (Unified Modeling Language)

- **Definition**: "The Unified Modeling Language is a standardized general-purpose modeling language and nowadays is managed as a de facto industry standard by the Object Management Group (OMG)."
- **Origin**: "The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design." "It was developed by Grady Booch, Ivar Jacobson, and James Rumbaugh at Rational Software in 1994–1995, with further development led by them through 1996."
- **Primary Goals of UML (by Page-Jones)**:
    - "Provide ready-to-use visual modeling language that allows them to create and share meaningful models"
    - "Extend core concepts by providing extensible and specialized mechanisms"
    - "Be independent of programming languages and development process"
    - "Provide formal and concrete basis for modeling language"
    - "Encourage the growth and use of OO tools"
    - "Support higher-level development concepts that promote collaborations and support frameworks"
    - "Able to integrate best practices"
- **Stakeholders Addressed**:
    - "Analysts"
    - "Designers"
    - "Coders"
    - "Testers"
    - "QA"
    - "Clients"
    - "Technical Authors"

Static vs Dynamic View

- **Static modeling**: "is used to specify the structure of the objects, classes or components that exist in the problem domain." "These are expressed using class, object or component."
- **Dynamic modeling**: "refers to representing the object interactions during runtime." "It is represented by sequence, activity, collaboration, and state."
- **UML diagram views**:

  - **Structural (or Static) view**: "emphasizes the static structure of the system using objects, attributes, operations and relationships." "It includes class diagrams and composite structure diagrams."
  - **Behavioral (or Dynamic) view**: "emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects." "This view includes sequence diagrams, activity diagrams, and state machine diagrams."

Architecture View Model (4+1 view model)

- **Origin**: "The “4 + 1 Architectural Views” were proposed in 1995 and represents the functional and non-functional requirements of software applications and architectures."
- **Goal**: "Separating the architecture into many different perspectives... had the goal of isolating and illustrating distinct components of the design with information tailored to different stakeholders."
- **Views**:
  - "Logical view"
  - "Development view"
  - "Process view"
  - "Physical view"
  - "Scenarios view"
- **View Descriptions**:
  - **Logical view**: "Primarily supports the functional requirements of the system." "It illustrates how the system is decomposed into the different areas and how the it should provide in terms of services to its target users."
  - **Process View**: "Under the process architecture non-functional requirements such as performance, availability, fault tolerance, and the system’s integrity is considered." "The purpose of this view is to illustrate components, actual executables needed by the system, and to capture the flow of information being exchange."
  - **Development View**: "In the software-development environment, the development view focuses on the arrangement of the actual software modules." "Information and descriptions included in this view is primarily intended for developers, and technical team involved in the development."
  - **Physical View**: "Considers the system’s non-functional requirements such as availability, reliability, performance and how the system can be extended." "The

    physical structure such as processing nodes… and networking channels… are considered in this view to allow designers to compute network capacity, latency and performance."

  - **Scenario/Use Case View**: "Scenario or the use case view combines the 4 views, working together seamlessly." "It acts as the driver on determining architectural elements and what is to include in the physical, development, process, and logical view."

ADL (Architecture Description Language)

- **Definition**: "ADL defines the software architecture formally and semantically." "ADL may be any modeling language that provides notation and rules to define system architectures in visualizations such as flow diagrams and organization charts."
- **Usage**: "Visualizations are often supported by EAM tools and are used to address concerns of EAM stakeholders."
- **Common ADL Constructs**:
  - "Components are the building blocks of a system."
  - "Connectors define how components interact."
  - "Systems encapsulate components and connectors."
  - "Ports provide the interface between components and connectors."
  - "Representations provide a way to visualize the system."
  - "Rep-maps define the mapping between different representations."
- **ADL Characteristics**:
  - "It should be able to depict most typical building styles"
  - "It should be able to support analytical capabilities or generate prototype implementations quickly."
  - "It should be suitable for information all concerned parties"
  - "It should be suited for tasks such as creation, refinement, and validation of the architecture"
  - "It should be able to provide basis for further implementation"
- **Mainstream Languages in AD**:
  - "UML"
  - "ArchiMate"

- **Other ADL**:
    - "ACME"
    - "Rapide"
    - "Wright"
    - "Unicord"
    - "ABACUS"

Role of Architecture Views

- **Definition**: "Architecture views are representations of the overall architecture based on the different stakeholders in the system."
- **Note**: "It may include one or more architecture models that are aligned with each other, providing a meaningful description of the system's architecture."
- **TOGAF Note**: "In TOGAF, the term "architecture" is replaced with the term "view", in accordance with ANSI/IEEE Std 1471-2000."

Basic Concepts

- **System**: "a collection of components organized to perform a common or set of functions"
- **Architecture**: "the system's fundamental organization"
- **Architecture Description**: "collection of artifacts that document an architecture"
- **Stakeholders**: "people who have concerns or have key roles in the system"
- **Concerns**: "an area of interest"
- **View**: "representation of the system from a specific perspective"
- **Viewpoint**: "defines the perspective"

Core Taxonomy of Architecture Views

- **TOGAF**: "Included are concepts on architecture views and viewpoint in TOGAF." "TOGAF's core taxonomy of architecture views define the minimum set of views to be considered in the development."
- 1. Stakeholders:
    - "Users"

  - "System and Software Engineers"
  - "Operators, Administrators, and Managers"
  - "Acquirers"
- 2. Views/Viewpoints:
  - "Business Architecture Views"
  - "Data Architecture Views"
  - "Application Architecture Views"
  - "Technology Architecture"
- 3. Description:
  - **Business Architecture Views**: "It addresses concerns of users and includes functional aspects from the perspective of the different users involved in the system such as planners and business managers."
    - "i.People View – focuses on human actors in the system"
    - "ii.Business Process View – deals with user processes"
    - "iii.Business Function View – deals with required functions"
    - "iv.Business Information View -deals with required information"
    - "v.Usability View – considers usability aspects"
    - "vi.Business Performance View – focused on the usability aspect"
  - **Data Architecture Views and Applications Architecture Views**: "Addresses concerns regarding databases and focused on how it will be implemented from the perspective of engineers responsible for security, software, data and others."
    - "Data Flow View – deals with storage, retrieval, processing and security of data"
    - "Software Engineering View – concentrates on the software development considerations such as constraints and possible opportunities"
    - "System Engineering View – provides different ways on how software and hardware components can be assembled into a system"
  - **Technology Architecture Views**:
    - "Communications Engineering View – deals on structuring effective communications facilities"
    - "Acquirer’s Views – provides appropriate guidance for purchasing (Includes the 1) Cost View and, 2) Standards View"

- **Composite Views**:
    - "Enterprise Manageability View – addresses the concerns in operations, administration, and management of the system"
    - "Enterprise Security View – focuses on the security of the system"

Developing Specific Views

- **Developing an Enterprise Security View**
    - **Concept**: "The Enterprise Security View is concentrated with the security aspects of the system."
    - **Stakeholders**: "This type of view should be developed for security engineers of the system and focuses on the implementation of the system based on the perspective of security"
    - **Modeling**: "Subjects include components that provide security services."
- **Developing a Software Engineering View**
    - **Concept**: "This view is concentrated with the development of software systems."
    - **Major Concerns**:
        - "Development Approach-Refers to the lifecycle models of the different software development approaches such as waterfall, prototyping and others"
        - "Software modularity and re-use-The term modularity refers to the division of a piece of software into number of discrete and logically cohesive sub-units..."
        - "Portability-A concept that concentrates on the ability to take software to a different environment."
        - "Migration and Interoperability-Interoperability refers to the ability of the new system to operate with the existing or legacy systems."
    - **Key Issues**:
        - "Data-intensive and Information-Intensive Software Systems"
        - "Software Tiers"
        - "Data Access Tier"
        - "Distribution"
        - "Interoperability"

- **Developing a System Engineering View**
    - **Concept**: "The system engineering view shows how software and hardware components can be integrated into a working system in a variety of ways." "This architecture's perspective also includes proper computing models for a distributed computing environment to assist legacy system migration."
    - **Stakeholders**: "This perspective should be designed for the system's systems engineering personnel… focuses on how the system is implemented from the hardware/software and networking perspectives."
- **Developing a Communications Engineering View**
    - **Concept**: "This view focuses on structuring communications and networking elements to simplify and create an effective networking plan."
    - **Stakeholders**: "This view should be designed for the system's communications engineers… concentrate on how the system is implemented from the communications engineer's perspective." "It includes concerns on network and communications requirements."
- **Developing a Data Flow View**
    - **Concept**: "This view is concerned with the storing, processing, retrieving, archiving, as well as the security of data and is developed for database engineers."
    - **Stakeholders**: "Stakeholders for this area is concerned with assuring ubiquitous and appropriate access of data for the right people."
    - **Modeling**: "Modeling of the database is usually done using ERD, schema and document type definitions."
- **Developing an Enterprise Manageability View**
    - **Concept**: "This view is concerned with the operations, administration, and management of the system."
    - **Stakeholders**: "Focuses on the understanding on how the system and its components will be and can be managed." "It also includes predicting and identifying necessary preventive maintenance."
    - **Components**: "Security Components" , "Data Assets" , "Software Assets" , "Hardware Assets" , "Networking Assets"
    - **Modeling**: "Business scenarios may be considered to predict preventive maintenance for unplanned events."
- **Developing Acquirer’s View**
    - **Concept**: "This view is for personnel involve or in charged with the acquisition."

  - **Stakeholders**: "Major concerns... involves proper understanding identifying the appropriate components or building blocks of the architecture." "It also includes knowledge on constraints that may arise to purchases of necessary or possible products."
  - **Modeling**: "This view is typically represented as SBBs or Solution Building Blocks that are guided with standards."

EA Management Plan

- **Definition**: "The enterprise's performance gaps, resource requirements, planned solutions, a sequencing plan, and a review of the existing and future architecture are all documented in the EA Management Plan which is also updated at regular intervals that will serve as guides for current and plans."
- **Includes**: "It also includes the governance process, implementation of the methodology, and the documentation of the framework."

---

4. Service Oriented Architecture (SOA)

Service-oriented architecture (SOA)

- **Definition**: "Service-oriented architecture (SOA) is a method of software development that uses software components called services to create business applications."
- **Style**: "it is an architectural style that focuses on discrete services instead of a monolithic design."
- **Application**: "It is also applied in the field of software design where services are provided to the other components by application components, through a communication protocol over a network."
- **Independence**: "SOA is also intended to be independent of vendors, products and technologies."
- **Service Definition**: "A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently, such as retrieving a credit card statement online."

Service orientation

- **Definition**: "Service orientation is a way of thinking in terms of services and service-based development and the outcomes of services"

- **Service Properties**: A service has four properties:
    1. "It logically represents a repeatable business activity with a specified outcome."
    2. "It is self-contained."
    3. "It is a black box for its consumers, meaning the consumer does not have to be aware of the service's inner workings."
    4. "It may be composed of other services."

Six (6) core values (SOA)

- "Business value is given more importance than technical strategy."
- "Strategic goals are given more importance than project-specific benefits."
- "Intrinsic interoperability is given more importance than custom integration."
- "Shared services are given more importance than specific-purpose implementations."
- "Flexibility is given more importance than optimization."
- "Evolutionary refinement is given more importance than pursuit of initial perfection."

Principles of SOA

- **Standardized service contract**: "Services adhere to a standard communications agreement, as defined collectively by one or more service description documents within a given set of services."
- **Service reference autonomy (an aspect of loose coupling)**: "The relationship between services is minimized to the level that they are only aware of their existence."
- **Service location transparency (an aspect of loose coupling)**: "Services can be called from anywhere within the network that it is located no matter where it is present."
- **Service longevity**: "Services should be designed to be long lived."
- **Service abstraction**: "The services act as black boxes, that is their inner logic is hidden from the consumers."
- **Service autonomy**: "Services are independent and control the functionality they encapsulate, from a Design-time and a run-time perspective."
- **Service statelessness**: "Services are stateless, that is either return the requested value or give an exception hence minimizing resource use."
- **Service granularity**: "A principle to ensure services have an adequate size and scope." "The functionality provided by the service to the user must be relevant."

- **Service normalization**: "Services are decomposed or consolidated (normalized) to minimize redundancy."
- **Service composability**: "Services can be used to compose other services."
- **Service discovery**: "Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted."
- **Service reusability**: "Logic is divided into various services, to promote reuse of code."
- **Service encapsulation**: "Many services which were not initially planned under SOA, may get encapsulated or become a part of SOA."

Service-oriented architecture (SOA) patterns

Patterns include:

- "Event-driven SOA"
- "Loose coupling"
- "Enterprise service bus (ESB)"
- "Microservices"
- "Client-server"
- "Abstraction"
- "Autonomy"
- "Discoverability"

---

1. Event-driven SOA

- **Definition**: "Combines the intelligence of event-driven architecture with the organizational capabilities of service offerings"
- **Function**: "In ESOA, services communicate by producing and consuming events. When one service emits an event, other interested services can subscribe to that event and react accordingly."
- **Use**: "This pattern is often used for asynchronous communication and real-time updates."

2. Loose Coupling

- **Definition**: "Loose coupling means that services are designed to be independent of each other and can be modified or replaced without affecting other services."
- **Benefit**: "This promotes flexibility and maintainability."

3. Enterprise Service Bus (ESB)

- **Definition**: "An ESB is a middleware platform that acts as a central hub for communication between services."
- **Features**: "It provides features such as message routing, transformation, and orchestration."
- **Infrastructure**: "An ESB provides a secure, scalable and cost-effective infrastructure that enables real-time data exchange among many systems." "Data from one system, known as a service provider, can be put on the enterprise service bus as a message, which is sent immediately to a service consumer of the data."
- **An ESB can provide**:
    - **Messaging (asynchronous)**: "Exchange data in real-time among systems."
    - **Web Services (e.g., SOAP & REST)**: "Use applications that broker data requests and delivery."
    - **Data Transformation (e.g., XML, XSLT, JSON)**: "Transform the data format to meet your system's needs, or to add value to the data you receive."
    - **Routing Intelligence**: "Get secure access control to services and intelligent routing of data along its intended path."

4. Microservices

- **Definition**: "Microservices are small, independent services that can be developed, deployed, and scaled independently."
- **Structure**: "They are often built around a specific business capability and communicate with each other using APIs."

5. Client-Server

- **Definition**: "In the client-server pattern, one application (the client) requests services from another application (the server)."
- **Use in SOA**: "This is a common pattern in SOA, where services act as servers and clients can be other services or external applications."

6. Abstraction

- **Definition**: "Abstraction involves hiding the internal implementation details of a service and exposing only the necessary information to its clients."
- **Benefit**: "This promotes reusability and reduces complexity."
- **Note**: "Service Abstraction" "Hiding design details and implementation logic from the outside world." "Turning service to a black box by hiding systems information, processing logic, and programmatic (Java, ABAP etc) approaches."

7. Autonomy

- **Definition**: "Autonomous services can operate independently and make decisions without requiring constant intervention."
- **Benefit**: "This can improve scalability and resilience."

8. Discoverability

- **Definition**: "Discoverability refers to the ability of services to be found and accessed by other services or applications."
- **Method**: "This is often achieved through service registries or directories."