**System Architecture Concept Template**

**1. Introduction**

* **Purpose**: Briefly state the purpose of this document – to define the fundamental structure, behavior, and properties of the system and its environment, and to outline the principles guiding its realization and evolution. This artifact guides implementation and supports traceability from high-level requirements to detailed design.
* **Scope**: Define the boundaries of this architectural analysis, specifying which system or project it applies to, and what aspects (e.g., logical, physical, operational) of the system's architecture are covered.

**2. System Overview and Context**

* **Purpose**: To establish a shared understanding of the system's purpose, operational environment, and its interactions with external entities.
* **Content**:
	+ **Problem Statement / Opportunity**: Briefly reiterate the problem or opportunity the system addresses, as identified in earlier concept artifacts.
	+ **Mission Objectives / Strategic Goals**: Summarize the overarching mission or strategic objectives the system must achieve.
	+ **Stakeholder Concerns**: Summarize the key concerns and expectations of major stakeholders that the architecture must address.
	+ **System Boundaries and Context**: Define what is **inside and outside the system-of-interest** (SoI). This should include a visual representation, such as a **System Context Diagram**.
	+ **Operational Scenarios / Use Cases**: Outline key operational scenarios or use cases that demonstrate how the system will be used in its environment, influencing architectural decisions.

**3. Architectural Vision and Guiding Principles**

* **Purpose**: To articulate the fundamental concepts, properties, and principles that will govern the system's design and evolution.
* **Content**:
	+ **Architectural Drivers**: Identify the primary factors shaping the architecture (e.g., performance, security, cost, reliability, scalability, maintainability, interoperability, safety).
	+ **Architectural Style / Pattern (if applicable)**: Describe any high-level architectural styles or patterns chosen (e.g., layered, microservices, event-driven) and the rationale for their selection.
	+ **Design Principles**: Define key principles guiding architectural decisions (e.g., modularity, reusability, open standards, resilience, human-centered design).

**4. Top-Level Architectural Views**

* **Purpose**: To represent the system's structure, behavior, and relationships from various perspectives, addressing different stakeholder concerns.
* **Content**:
	+ **Operational Architecture View (OV)**:
		- Describes how the system will be used to meet stakeholder expectations.
		- May include **OV-1 (High-Level Operational Concept Graphic)** for a pictorial overview of missions/scenarios, and **OV-2 (Operational Resource Flow Description)** for capability requirements and resource flows.
	+ **Logical Architecture View (LV)**:
		- Defines "what" the system must be able to do, independent of implementation.
		- **Functional Architecture**: Describes the inter-related transformative processes and input-output tasks a system performs.
		- **Behavioral Architecture**: Describes sequencing, execution logic, and control flows.
		- **Temporal Architecture (Optional)**: Classifies functions based on execution frequency.
	+ **Physical Architecture View (PA)**:
		- Describes the arrangement of physical elements (hardware, software, humans, etc.) and their interfaces that provide the design solution.
		- Often driven by non-functional requirements.
	+ **All Viewpoint (AV) (Optional)**: Provides an overview of the architectural effort, scope, context, rules, constraints, and vocabulary.
	+ **Service Viewpoint (SvcV) (Optional)**: Depicts services, sub-services, and resource flows, including human interaction with services.

**5. Major Components and Interfaces**

* **Purpose**: To identify the primary structural elements of the system and define their interactions.
* **Content**:
	+ **Structural Decomposition**:
		- Identify major components and subsystems.
		- Use **SysML Block Definition Diagrams (BDD)** to illustrate system components, subsystems, and their hierarchical relationships.
	+ **Interface Concepts**:
		- Identify major interfaces with external systems, users, or internal components.
		- Use **SysML Internal Block Diagrams (IBD)** to define interfaces and data flows between components.
		- Reference or summarize **Interface Control Documents (ICDs)**, communication protocols, and data exchange formats.

**6. Key Capabilities and Functional Allocation**

* **Purpose**: To define the high-level capabilities the system will provide and how these functions are distributed across the architectural elements.
* **Content**:
	+ **High-Level Capabilities**: List and describe the main capabilities the system must provide.
	+ **Functional Decomposition**: Break down major functions and allocate them to subsystems or components, potentially using functional flow diagrams.

**7. Cross-Cutting Concerns**

* **Purpose**: To address aspects that span multiple components or views of the architecture, ensuring they are systematically considered.
* **Content**:
	+ **Performance Metrics**: Define target performance metrics (e.g., latency, throughput, reliability, availability, maintainability, safety, security (RAMSS)).
	+ **Security Considerations**: Outline how security objectives and controls are embedded into the architecture. Reference or summarize the **Security Concept**.
	+ **Risk Mitigation**: Identify technical architectural risks (e.g., component incompatibility, scalability issues) and outline architectural strategies for their mitigation. This feeds into the **Risk Management Concept**.
	+ **Human Systems Integration (HSI)**: Address how human operators and organizational aspects are considered, including usability, training, and safety within the architectural framework.
	+ **Cost and Schedule Impact**: Summarize how the chosen architecture influences cost and schedule, and any associated trade-offs. This links to the **Cost and Schedule Concept**.

**8. Evolution and Refinement**

* **Purpose**: To describe how the architecture will mature and adapt over time.
* **Content**:
	+ **Architectural Evolution Plan**: Outline the progression from conceptual to logical to physical architecture.
	+ **Trade-off Analysis Summary**: Document the analysis and trade-offs that led to the selection of the preferred architectural approach, including feasibility studies and alternative evaluations.
	+ **Iterative Validation**: Describe how the architecture will be continuously verified and validated against requirements through simulations, prototypes, and analyses.

**9. Relationship to Other Artifacts**

* **Purpose**: Explain how this artifact integrates with and influences other systems engineering activities and artifacts.
* **Content**:
	+ **Derived from / Informs**:
		- **Mission Concept** and **Stakeholder Concept**: The System Architecture Concept is informed by the system's mission objectives and stakeholder needs and expectations.
		- **ConOps** and **OpsCon**: It translates the high-level operational vision (**ConOps**) and detailed operational workflows (**OpsCon**) into a technical structure.
		- **Requirements Definition**: Provides the framework for developing detailed functional, performance, and interface requirements.
		- **Design Concept / Design Definition**: Serves as the foundation for subsequent detailed design activities, linking the conceptual "what" to the implementation "how".
		- **Integration Concept**: Defines the structure that guides how system elements will be assembled and tested.
		- **Test and Evaluation Concept**: Informs the verification and validation strategy, as the architecture defines what needs to be tested and how.
		- **Sustainment Concept**: Influences long-term supportability, maintainability, and upgrade strategies.
		- **Security Concept**: Directly informs the placement and interaction of security controls within the system.
		- **Development Concept**: Specifies the detailed design and engineering of the system architecture.

**10. Appendices (Optional)**

* **Detailed SysML Diagrams**: Full set of Block Definition Diagrams, Internal Block Diagrams, Activity Diagrams, State Machine Diagrams, etc..
* **Interface Control Documents (ICDs)**: Full interface specifications.
* **Trade Study Reports**: Detailed reports from architectural trade-off analyses.
* **Glossary of Terms**: Definition of architectural terms specific to the system.