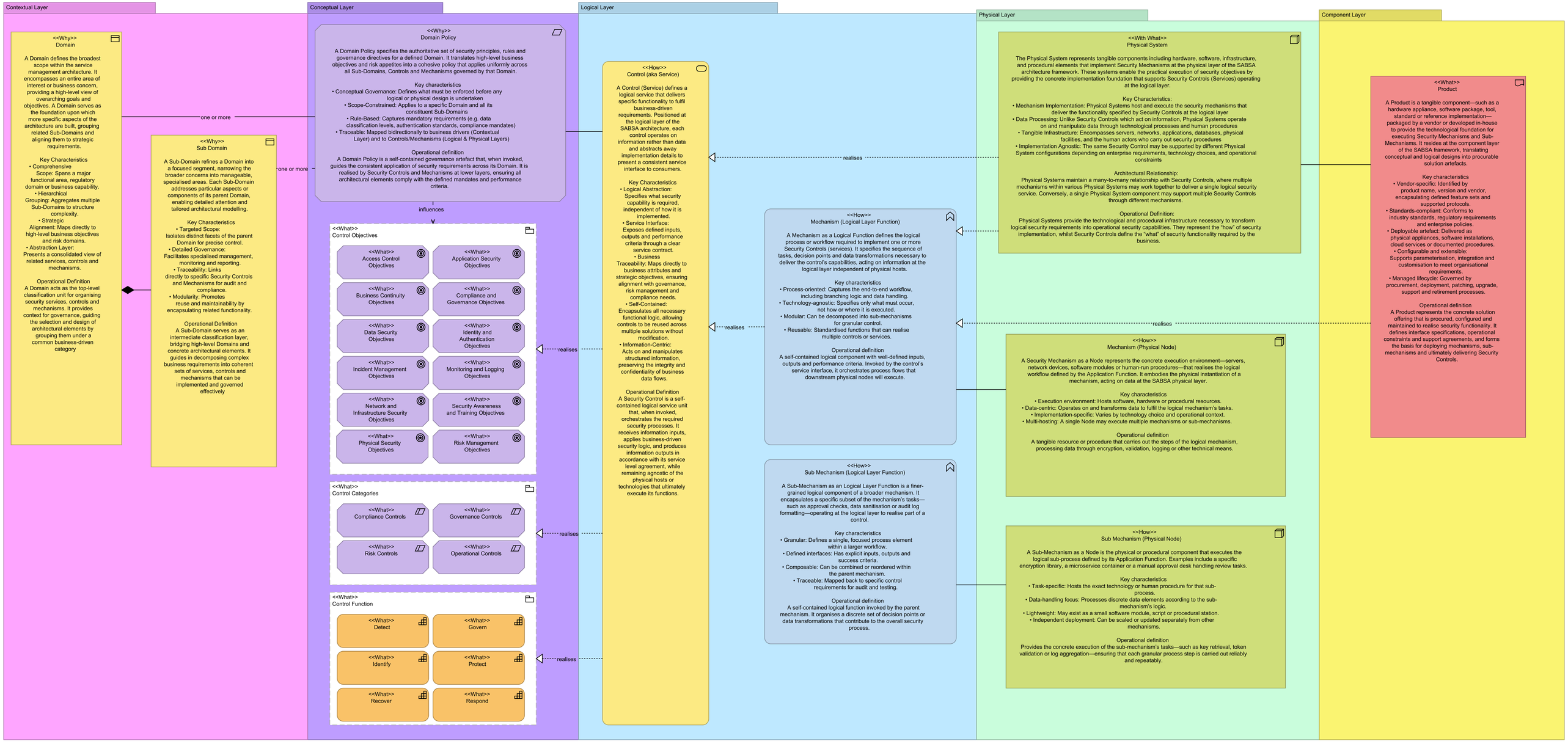
Contextual Layer
Conceptual Layer
Logical Layer
Physical Layer
Component Layer
<<Why>>
Domain
A Domain defines the broadest scope within the service management architecture. It encompasses an entire area of interest or business concern, providing a high-level view of overarching goals and objectives. A Domain serves as the foundation upon which more specific aspects of the architecture are built, grouping related Sub-Domains and aligning them to strategic requirements.
Key Characteristics
• Comprehensive Scope: Spans a major functional area, regulatory domain or business capability.
• Hierarchical Grouping: Aggregates multiple Sub-Domains to structure complexity.
• Strategic Alignment: Maps directly to high-level business objectives and risk domains.
• Abstraction Layer: Presents a consolidated view of related services, controls and mechanisms.
Operational Definition
A Domain acts as the top-level classification unit for organising security services, controls and mechanisms. It provides context for governance, guiding the selection and design of architectural elements by grouping them under a common business-driven category
one or more
<<Why>>
Sub Domain
A Sub-Domain refines a Domain into a focused segment, narrowing the broader concerns into manageable, specialised areas. Each Sub-Domain addresses particular aspects or components of its parent Domain, enabling detailed attention and tailored architectural modelling.
Key Characteristics
• Targeted Scope: Isolates distinct facets of the parent Domain for precise control.
• Detailed Governance: Facilitates specialised management, monitoring and reporting.
• Traceability: Links directly to specific Security Controls and Mechanisms for audit and compliance.
• Modularity: Promotes reuse and maintainability by encapsulating related functionality.
Operational Definition
A Sub-Domain serves as an intermediate classification layer, bridging high-level Domains and concrete architectural elements. It guides in decomposing complex business requirements into coherent sets of services, controls and mechanisms that can be implemented and governed effectively
one or more
<<Why>>
Domain Policy
A Domain Policy specifies the authoritative set of security principles, rules and governance directives for a defined Domain. It translates high-level business objectives and risk appetites into a cohesive policy that applies uniformly across all Sub-Domains, Controls and Mechanisms governed by that Domain.
Key characteristics
• Conceptual Governance: Defines what must be enforced before any logical or physical design is undertaken
• Scope-Constrained: Applies to a specific Domain and all its constituent Sub-Domains
• Rule-Based: Captures mandatory requirements (e.g. data classification levels, authentication standards, compliance mandates)
• Traceable: Mapped bidirectionally to business drivers (Contextual Layer) and to Controls/Mechanisms (Logical & Physical Layers)
Operational definition
A Domain Policy is a self-contained governance artefact that, when invoked, guides the consistent application of security requirements across its Domain. It is realised by Security Controls and Mechanisms at lower layers, ensuring all architectural elements comply with the defined mandates and performance criteria.
influences
<<What>>
Control Objectives
<<What>> Access Control Objectives
<<What>> Application Security Objectives
<<What>> Business Continuity Objectives
<<What>> Compliance and Governance Objectives
<<What>> Data Security Objectives
<<What>> Identity and Authentication Objectives
<<What>> Incident Management Objectives
<<What>> Monitoring and Logging Objectives
<<What>> Network and Infrastructure Security Objectives
<<What>> Security Awareness and Training Objectives
<<What>> Physical Security Objectives
<<What>> Risk Management Objectives
realises
<<What>>
Control Categories
<<What>> Compliance Controls
<<What>> Governance Controls
<<What>> Risk Controls
<<What>> Operational Controls
realises
<<What>>
Control Function
<<What>> Detect
<<What>> Govern
<<What>> Identify
<<What>> Protect
<<What>> Recover
<<What>> Respond
realises
<<How>>
Control (aka Service)
A Control (Service) defines a logical service that delivers specific functionality to fulfil business-driven requirements. Positioned at the logical layer of the SABSA architecture, each control operates on information rather than data and abstracts away implementation details to present a consistent service interface to consumers.
Key Characteristics
• Logical Abstraction: Specifies what security capability is required, independent of how it is implemented.
• Service Interface: Exposes defined inputs, outputs and performance criteria through a clear service contract.
• Business Traceability: Maps directly to business attributes and strategic objectives, ensuring alignment with governance, risk management and compliance needs.
• Self-Contained: Encapsulates all necessary functional logic, allowing controls to be reused across multiple solutions without modification.
• Information-Centric: Acts on and manipulates structured information, preserving the integrity and confidentiality of business data flows.
Operational Definition
A Security Control is a self-contained logical service unit that, when invoked, orchestrates the required security processes. It receives information inputs, applies business-driven security logic, and produces information outputs in accordance with its service level agreement, while remaining agnostic of the physical hosts or technologies that ultimately execute its functions.
realises
<<How>>
Mechanism (Logical Layer Function)
A Mechanism as a Logical Function defines the logical process or workflow required to implement one or more Security Controls (services). It specifies the sequence of tasks, decision points and data transformations necessary to deliver the control's capabilities, acting on information at the logical layer independent of physical hosts.
Key characteristics
• Process-oriented: Captures the end-to-end workflow, including branching logic and data handling.
• Technology-agnostic: Specifies only what must occur, not how or where it is executed.
• Modular: Can be decomposed into sub-mechanisms for granular control.
• Reusable: Standardised functions that can realise multiple controls or services.
Operational definition
A self-contained logical component with well-defined inputs, outputs and performance criteria. Invoked by the control's service interface, it orchestrates process flows that downstream physical nodes will execute.
<<How>>
Sub Mechanism (Logical Layer Function)
A Sub-Mechanism as an Logical Layer Function is a finer-grained logical component of a broader mechanism. It encapsulates a specific subset of the mechanism's tasks—such as approval checks, data sanitisation or audit log formatting—operating at the logical layer to realise part of a control.
Key characteristics
• Granular: Defines a single, focused process element within a larger workflow.
• Defined interfaces: Has explicit inputs, outputs and success criteria.
• Composable: Can be combined or reordered within the parent mechanism.
• Traceable: Mapped back to specific control requirements for audit and testing.
Operational definition
A self-contained logical function invoked by the parent mechanism. It organises a discrete set of decision points or data transformations that contribute to the overall security process.
realises
<<With What>>
Physical System
The Physical System represents tangible components including hardware, software, infrastructure, and procedural elements that implement Security Mechanisms at the physical layer of the SABSA architecture framework. These systems enable the practical execution of security objectives by providing the concrete implementation foundation that supports Security Controls (Services) operating at the logical layer.
Key Characteristics:
• Mechanism Implementation: Physical Systems host and execute the security mechanisms that deliver the functionality specified by Security Controls at the logical layer
• Data Processing: Unlike Security Controls which act on information, Physical Systems operate on and manipulate data through technological processes and human procedures
• Tangible Infrastructure: Encompasses servers, networks, applications, databases, physical facilities, and the human actors who carry out security procedures
• Implementation Agnostic: The same Security Control may be supported by different Physical System configurations depending on enterprise requirements, technology choices, and operational constraints
Architectural Relationship:
Physical Systems maintain a many-to-many relationship with Security Controls, where multiple mechanisms within various Physical Systems may work together to deliver a single logical security service. Conversely, a single Physical System component may support multiple Security Controls through different mechanisms.
Operational Definition:
Physical Systems provide the technological and procedural infrastructure necessary to transform logical security requirements into operational security capabilities. They represent the "how" of security implementation, whilst Security Controls define the "what" of security functionality required by the business.
realises
<<How>>
Mechanism (Physical Node)
A Security Mechanism as a Node represents the concrete execution environment—servers, network devices, software modules or human-run procedures—that realises the logical workflow defined by the Application Function. It embodies the physical instantiation of a mechanism, acting on data at the SABSA physical layer.
Key characteristics
• Execution environment: Hosts software, hardware or procedural resources.
• Data-centric: Operates on and transforms data to fulfil the logical mechanism's tasks.
• Implementation-specific: Varies by technology choice and operational context.
• Multi-hosting: A single Node may execute multiple mechanisms or sub-mechanisms.
Operational definition
A tangible resource or procedure that carries out the steps of the logical mechanism, processing data through encryption, validation, logging or other technical means.
<<How>>
Sub Mechanism (Physical Node)
A Sub-Mechanism as a Node is the physical or procedural component that executes the logical sub-process defined by its Application Function. Examples include a specific encryption library, a microservice container or a manual approval desk handling review tasks.
Key characteristics
• Task-specific: Hosts the exact technology or human procedure for that sub-process.
• Data-handling focus: Processes discrete data elements according to the sub-mechanism's logic.
• Lightweight: May exist as a small software module, script or procedural station.
• Independent deployment: Can be scaled or updated separately from other mechanisms.
Operational definition
Provides the concrete execution of the sub-mechanism's tasks—such as key retrieval, token validation or log aggregation—ensuring that each granular process step is carried out reliably and repeatably.
<<What>>
Product
A Product is a tangible component—such as a hardware appliance, software package, tool, standard or reference implementation—packaged by a vendor or developed in-house to provide the technological foundation for executing Security Mechanisms and Sub-Mechanisms. It resides at the component layer of the SABSA framework, translating conceptual and logical designs into procurable solution artefacts.
Key characteristics
• Vendor-specific: Identified by product name, version and vendor, encapsulating defined feature sets and supported protocols.
• Standards-compliant: Conforms to industry standards, regulatory requirements and enterprise policies.
• Deployable artefact: Delivered as physical appliances, software installations, cloud services or documented procedures.
• Configurable and extensible: Supports parameterisation, integration and customisation to meet organisational requirements.
• Managed lifecycle: Governed by procurement, deployment, patching, upgrade, support and retirement processes.
Operational definition
A Product represents the concrete solution offering that is procured, configured and maintained to realise security functionality. It defines interface specifications, operational constraints and support agreements, and forms the basis for deploying mechanisms, sub-mechanisms and ultimately delivering Security Controls.