SENDĀTRIX

SDX Technical Publications

SDX-TP-003

C Chains™ Security Architecture Guide

Security Architecture, Threat Model, and Independent Evaluation Framework

Structural Protection by Architecture — Designed for Independent Technical Scrutiny

A technical reference for security architects, evaluators, CISOs, auditors, and qualified reviewers.

Version 1.1 | August 2026

Originally issued July 13, 2026 | Greenville, North Carolina, USA | sendatrix.com

Publication Information

Publication	SDX-TP-003
Title	C Chains™ Security Architecture Guide
Subtitle	Security Architecture, Threat Model, and Independent Evaluation Framework
Version	1.1
Original Publication Date	July 13, 2026
Updated Edition	August 2026
Publisher	SENDĀTRIX — Greenville, North Carolina, USA
Patent	U.S. Patent No. 9,405,927

This publication defines the C Chains™ Structural Protection architecture, its operational trust model, security objectives, threat model, and framework for independent technical evaluation.

Table of Contents

Preface

This guide defines the C Chains Structural Protection architecture, its security objectives, its threat model, and a proposed framework for independent technical evaluation.

It is intended for security architects, evaluators, CISOs, prospective customers, auditors, and other qualified reviewers who require a clear statement of what the architecture is designed to accomplish and how those claims should be assessed.

The guide distinguishes between architecture assurance and implementation assurance. Architecture assurance asks whether the documented design logically supports the claimed protection properties. Implementation assurance asks whether the deployed software faithfully realizes that design and maintains those properties under realistic operating and attack conditions.

For consistency, restoration refers to the authorized process by which protected information is made available for a valid transaction. Reconstruction refers to the technical reassembly of protected information, whether authorized or unauthorized. The principal security objective is to prevent unauthorized reconstruction.

This document describes architectural behavior and evaluation objectives. It does not replace conventional cybersecurity controls, operational procedures, or product-specific administration documentation.

Architect's Note

C Chains did not begin as an attempt to improve encryption alone. It began with a different question: why should compromise of a database, application, or administrative account necessarily yield usable sensitive information?

The architecture developed by repeatedly removing concentrations of data, trust, identity, authority, and transactional control. Each design decision led to another question: how can protected information remain useful to authorized applications without remaining directly available to any single operational environment?

The resulting model is intended to make compromise of infrastructure distinct from compromise of protected information. That distinction is the central idea evaluated throughout this guide.

Chapter 1

Executive Summary

Purpose

This document provides an overview of Coordinative Transactional Processing (CTP) and establishes a framework for independent security evaluation of the architecture.

CTP is a patented data protection architecture developed to address a fundamental weakness in conventional information systems: sensitive data typically remains present and accessible within operational environments where authorized users, administrators, applications, and compromised credentials can potentially access it.

Traditional security controls primarily focus on restricting who may access sensitive information. CTP focuses on reducing or eliminating the exposure of sensitive information itself within operational systems.

The Problem

Most modern data breaches do not occur because attackers defeat encryption algorithms or bypass database engines. Instead, attackers commonly obtain legitimate credentials, compromise trusted systems, exploit authorized pathways, abuse administrative privileges, or gain access through application-layer vulnerabilities.

When sensitive information remains directly available within operational databases, successful compromise of those environments can result in exposure of protected information.

Examples include:

- Insider misuse of privileged access.
- Compromised administrator credentials.
- Database theft.
- Storage compromise.
- Application compromise.
- Unauthorized querying of operational records.
- Credential theft leading to legitimate system access.

In these scenarios, traditional access controls may fail because the attacker ultimately gains access to an environment where the sensitive information is already present.

CTP Approach

CTP addresses this problem by separating sensitive information from the operational environment and managing access through a coordinated multi-component architecture.

Rather than storing sensitive information in a directly accessible operational form, CTP transforms and separates protected content into distinct components managed through independent processes and authorization mechanisms.

The architecture is designed so that possession or compromise of an individual architectural dimension or unauthorized subset of the architectural dimensions should not, by itself, result in recovery of protected field values.

This differs from traditional database protection approaches that primarily rely upon:

- Access control.
- Encryption at rest.
- Network segmentation.
- Monitoring and detection.
- Administrative policy enforcement.

CTP is intended to reduce the value of a successful compromise by limiting direct access to protected information even when portions of the environment have been breached.

In CTP, compromise of infrastructure does not necessarily imply unauthorized reconstruction of protected information.

Evaluation Objective

The primary purpose of this evaluation is to assess the architectural and implemented security properties of CTP. Compliance, operational controls, and supporting procedures may be considered where they affect those properties, but they are not substitutes for direct technical evaluation.

The objective is to determine whether protected information can be reconstructed under defined attacker models while CTP protections are operating as designed, and whether the system preserves processing integrity and fails safely during disruption.

The evaluation should focus on realistic attack scenarios, including but not limited to:

- Database compromise.
- Insider access.
- Administrative access.
- Module compromise.
- Restoration-path abuse.
- Metadata analysis.
- Codeset manipulation.
- Application compromise.

Core Security Claim

The central claim under evaluation is:

Protected information should not be recoverable through compromise of operational storage, database access, administrative access, any individual architectural dimension, or any unauthorized subset of the architectural dimensions unless the attacker satisfies the complete reconstruction and authorization requirements defined by the architecture.

Success Criteria

For purposes of evaluation, success is defined as:

The unauthorized reconstruction of original protected information without satisfying the intended restoration controls of the CTP architecture.

Observing metadata, system structure, database schemas, transaction activity, access patterns, or protected fragments does not, by itself, constitute recovery of protected information.

The primary evaluation question is therefore:

Can protected information be reconstructed or exposed under the defined attack scenarios while CTP protections remain in effect?

Chapter 2

*Architecture & **Operational** Data Flow*

Overview

Coordinative Transactional Processing (CTP) is a multi-component Structural Protection architecture designed to reduce the exposure of sensitive information by distributing protection, storage, coordination, and restoration responsibilities across independent architectural dimensions.

The architecture consists of three primary dimensional components:

SPA – Protection Component

SPA operates within the protected application environment and is responsible for initiating protection operations.

SPA receives sensitive field values from the application and coordinates the protection process.

SPA is intentionally designed so that it does not independently possess sufficient information or authority to reconstruct protected information. The same restriction applies to any unauthorized subset of the architectural dimensions.

SPB – Protected Data Component

SPB maintains protected data components and participates in coordinated protection and restoration operations.

SPB is intentionally separated from the operational application environment.

SPB does not independently possess sufficient information or authority to reconstruct protected information. The same restriction applies to any unauthorized subset of the architectural dimensions.

SPC – Coordination Component

SPC maintains coordination information required by the architecture and participates in protection and restoration transactions.

SPC serves as an independent coordination dimension within the architecture. Neither SPC nor any unauthorized subset of the architectural dimensions is intended to possess sufficient information or authority to reconstruct protected information.

Protection Workflow

When a sensitive field is designated for protection:

Step 1

The application submits the sensitive field value to SPA.

Examples may include:

- Patient identifiers
- Social Security numbers
- Financial account information
- Personally identifiable information (PII)
- Protected health information (PHI)

Step 2

SPA initiates a coordinated protection transaction involving SPB and SPC.

A transaction-specific codeset is chosen and assigned to the protection operation.

Step 3

The original sensitive value is processed into protected components and distributed according to CTP processing rules.

Protected elements are maintained across the architecture according to CTP processing rules and the trust boundaries defined in this guide.

Step 4

The operational application retains only the information necessary to continue normal business operations without maintaining direct exposure of the original protected value.

At completion of the protection process, the original sensitive value is no longer present within the operational database in its original form.

Restoration Workflow

Restoration is a separate operation requiring independent authorization.

Protection authority does not automatically imply restoration authority.

Step 1

An authorized restoration request is initiated.

Step 2

The architecture validates the request and associated authorization requirements.

Step 3

Participating components perform coordinated restoration operations.

Step 4

The protected information is reconstructed only for the authorized restoration transaction.

Step 5

The restored information is returned to the requesting process according to established authorization and scope controls.

Authority Separation

The architecture supports separation of operational authority levels.

For example:

Protection Authority

Protection operations may be performed by users or processes authorized to protect sensitive information.

Protection authority alone does not grant restoration authority.

Restoration Authority

Restoration operations require additional authorization and are evaluated independently from protection permissions.

This separation is intended to reduce the likelihood that compromise of a protection operator, application user, or operational administrator automatically results in access to protected field values.

Architectural Objectives

Architectural Reconstruction Requirement
The architecture is designed such that authorized reconstruction of protected information requires the coordinated participation of the complete restoration architecture. Neither an individual architectural dimension nor unauthorized subset of the architectural dimensions is intended to possess sufficient information or authority to reconstruct protected information.

The architecture is intended to ensure that:

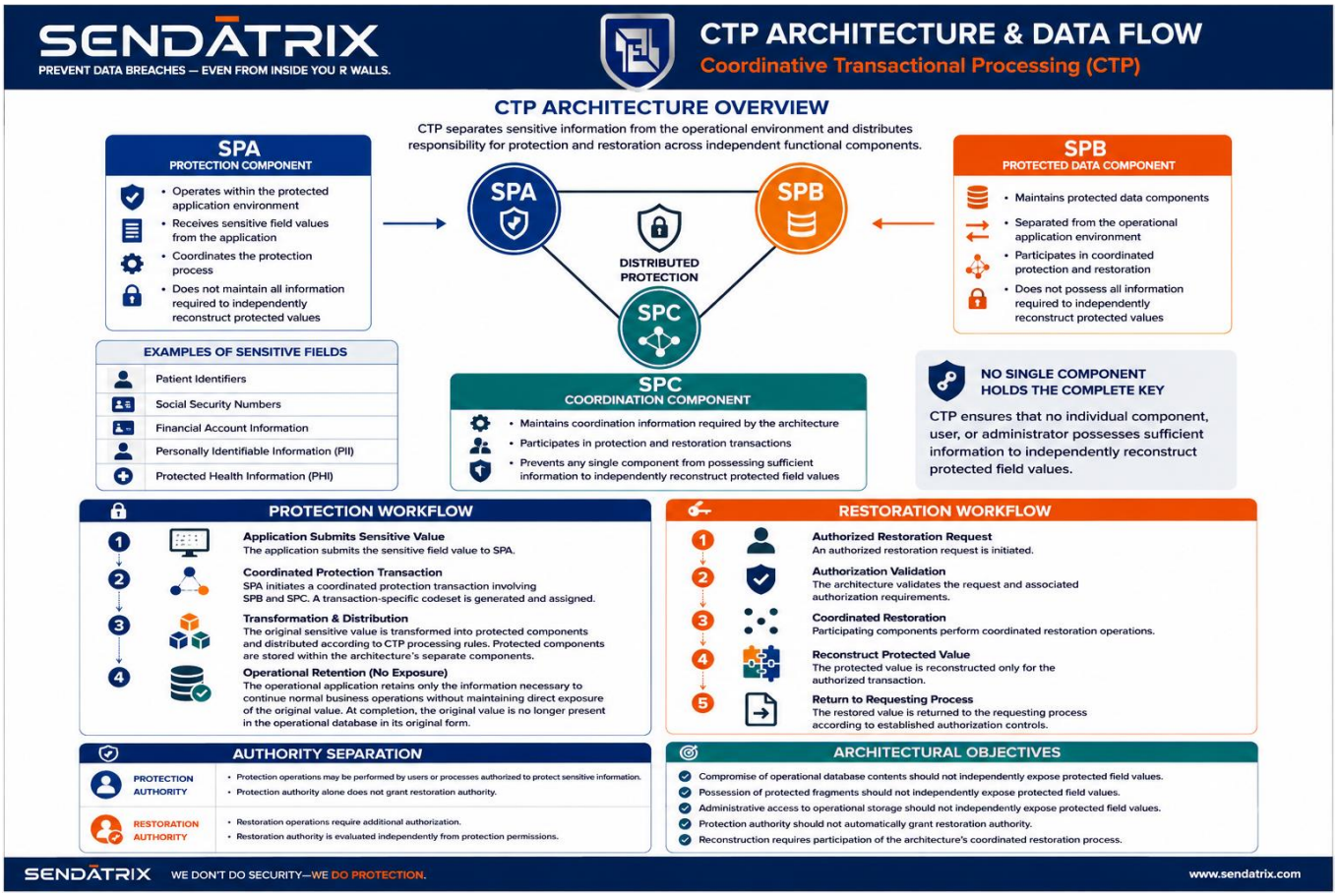
- Protected information cannot be reconstructed solely from operational database contents, any individual architectural dimension, or any unauthorized subset of the architectural dimensions.
- Possession of one or more protected elements, absent coordinated participation of the complete authorized restoration architecture, does not permit unauthorized reconstruction.
- Administrative access to operational storage does not, by itself, provide access to protected information.
- Protection authority remains independent from restoration authority.
- Authorized reconstruction requires coordinated participation of SPA, SPB, and SPC through the complete authorized restoration process.
- Maintain transaction sequence, completeness, validity, and audit continuity throughout protection and restoration operations.
- Fail safely during architectural dimension or communication disruption so that degraded operation does not expose protected information or accept incomplete transactions.
- Support operational recovery without weakening the reconstruction and authorization requirements of the architecture.

Conceptual View

The following diagram illustrates the logical interaction of the three architectural dimensions during protection and restoration operations.

It is intended to communicate the operational relationships among the architectural dimensions rather than implementation-specific deployment details.

Conceptual Architecture Diagram



Operational State After Protection

Following successful protection, the original sensitive field value is removed from operational exposure and replaced by a masked representation suitable for continued application operation.

The original protected value is no longer available to database users, administrators, or application operators through normal database access.

Recovery of the original value requires an authorized restoration transaction through the CTP architecture.

Normal database operations therefore continue without routine exposure of the original protected value.

Scope of Architectural Review

This document intentionally describes architectural behavior and security objectives without disclosing implementation-specific details.

Specific implementation details, algorithms, codeset generation methods, synchronization mechanisms, and source code may be reviewed separately under appropriate confidentiality agreements if required as part of a formal security assessment.

Chapter 3

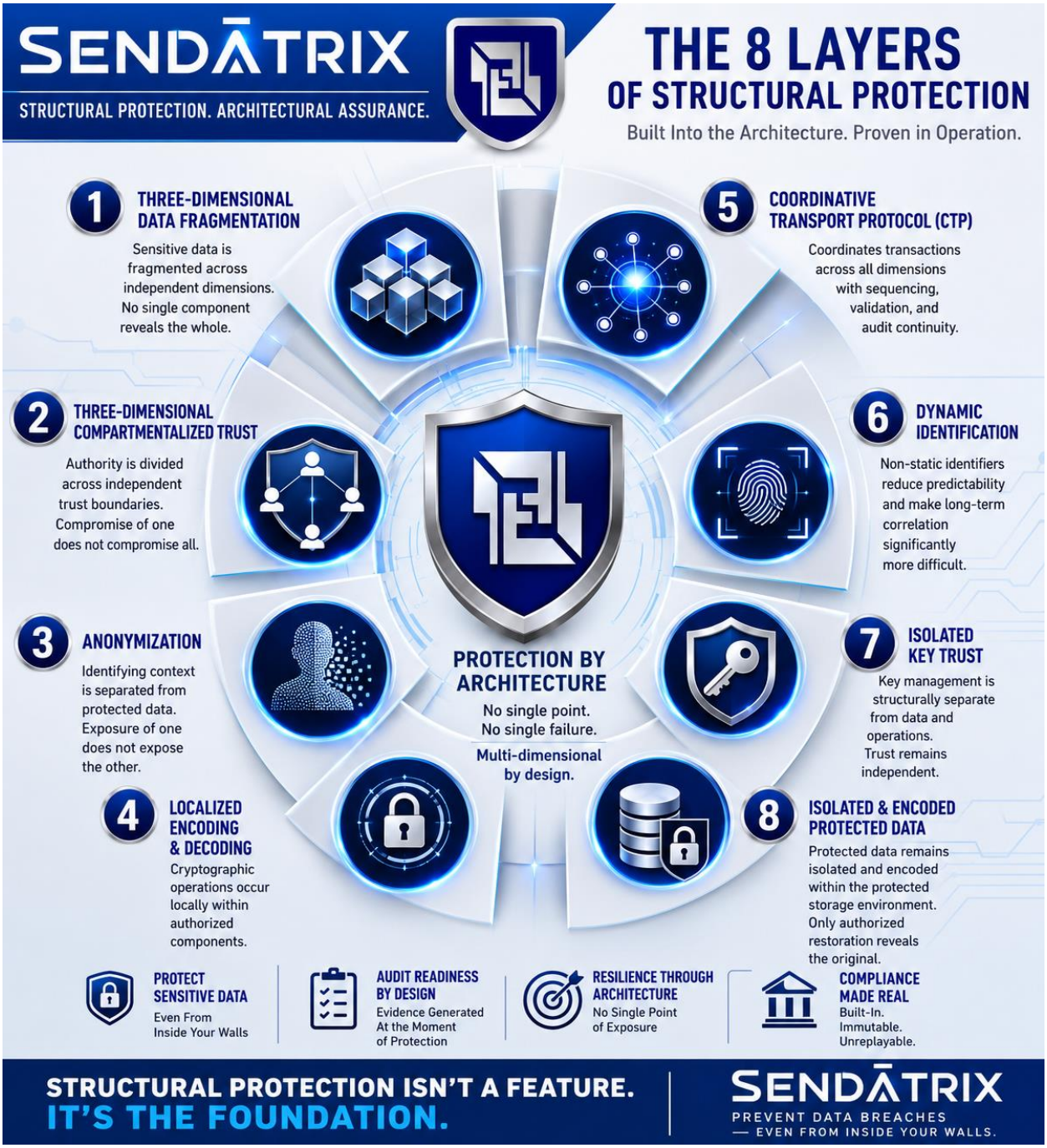
The Eight Architectural Layers of Structural Protection

Purpose

Structural Protection is achieved through the interaction of multiple independent architectural layers rather than reliance upon a single security control.

Each layer contributes a distinct protection property. Together, these layers create a defense model intended to limit exposure of sensitive information even when portions of the operational environment are compromised.

The layers are cumulative. Each builds upon the protection established by the previous layers.



1. Three-Dimensional Data Fragmentation

Sensitive information is structurally fragmented across multiple independent dimensions rather than remaining as complete, recoverable records within a single operational environment.

This fragmentation reduces the value of any individual dimension or incomplete subset while requiring coordinated reconstruction under authorized conditions.

Protection Contribution

Without structural fragmentation, sensitive information remains available as a complete record within a single environment. Fragmentation requires multiple independent architectural dimensions to participate in reconstruction, requiring coordinated reconstruction from multiple independent architectural dimensions before protected information can be recovered.

---

2. Three-Dimensional Compartmentalized Trust

Trust is intentionally distributed across independent architectural dimensions.

Rather than concentrating trust within a single administrator, server, database, or application, authority is divided among multiple trust boundaries.

Compromise of one trust boundary, or an unauthorized combination of fewer than all required dimensions, does not automatically imply compromise of the overall protection model.

Protection Contribution

Without compartmentalized trust, a compromised environment could concentrate sufficient authority to reconstruct protected information. By distributing trust, each dimension retains only the authority necessary for its defined function.

---

3. Anonymization

Protected information is separated from identifying context wherever practical.

Operational systems therefore work with information that minimizes direct exposure of sensitive identities while preserving application functionality.

Protection Contribution

Without anonymization, compromise of protected information could immediately reveal the identity associated with that information. By separating identity from protected content, exposure of one does not automatically expose the other.

---

4. Localized Encoding and Decoding

Encoding and decoding operations occur locally within authorized components rather than through centralized services.

This minimizes unnecessary transmission of sensitive information and limits opportunities for centralized compromise.

Protection Contribution

Without localized encoding and decoding, compromise of a centralized encoding service could increase the attack surface. Localized processing confines encoding and decoding responsibilities to authorized components, reducing dependence upon centralized cryptographic services.

5. Coordinative Transport Protocol (CTP)

The Coordinative Transport Protocol provides the controlled communication framework between architectural dimensions.

CTP coordinates protection and restoration transactions while maintaining synchronization across the participating architectural dimensions.

CTP also coordinates the progression of transaction-specific codesets across the participating dimensions.

It also provides transaction sequencing, validation, coordination, processing integrity, and audit continuity throughout the complete protection and restoration process.

The valid transaction path coordinates the complete A–B–C–B–A sequence. Interruption, duplication, substitution, or out-of-sequence processing is intended to be detected and rejected rather than accepted as a complete transaction.

Protection Contribution

Without CTP, the architecture would lose its coordinated transaction model. Independent components could no longer reliably synchronize protection and restoration operations, verify transaction sequencing, or maintain complete end-to-end audit integrity.

---

6. Dynamic Identification

Protected record identification is based upon dynamically generated operational identifiers rather than fixed, long-lived identifiers.

Dynamic identification reduces predictability and minimizes opportunities for long-term correlation of protected transactions.

Protection Contribution

Without dynamic identification, fixed identifiers could potentially assist an attacker in correlating protected fragments across architectural dimensions. Dynamic identifiers reduce the usefulness of previously observed transactions and make long-term correlation significantly more difficult.

---

7. Isolated Key Trust

Key management remains structurally independent from protected data and operational processing.

Separating trust in this manner limits the impact of compromise within any individual architectural dimension or unauthorized subset of the architectural dimensions.

Protection Contribution

Without isolated key trust, cryptographic key material and protected information could become sufficiently associated to weaken the intended separation between architectural components. Isolating key management preserves independent trust boundaries throughout the architecture.

---

8. Isolated and Encoded Protected Data

Protected information remains both structurally isolated and encoded throughout its protected lifecycle within the protected storage environment.

Only authorized restoration processes operating within the defined architecture are intended to reconstruct the original protected information.

Protection Contribution

Without isolated protected storage, compromise of the storage environment could expose sensitive information directly. Structural isolation and encoding ensure that compromise of the protected storage environment alone does not provide sufficient information to reconstruct protected records.

Architectural Perspective

These layers explain the protection properties of the operational architecture described in Chapter 2 and provide the technical basis for the threat model established in Chapter 6.

Unlike traditional security architectures that frequently rely upon a limited number of centralized controls, Structural Protection employs multiple complementary architectural layers.

Each layer contributes a distinct protection property. No single layer is intended to provide complete protection by itself.

Instead, the security characteristics of the architecture arise from the combined interaction of all eight layers. As successive layers are applied, the conditions required to reconstruct protected information become progressively more restrictive.

Structural Protection therefore derives its effectiveness from architectural composition rather than dependence upon any single security mechanism.

Chapter 4

Functional Role Model

Purpose

C Chains uses authorization levels to control access to specific operational functions within the Structural Protection architecture.

These authorization levels are assigned to applications and operational functions rather than requiring separate personnel for every responsibility.

Organizations may assign multiple functions to a single trusted administrator or distribute responsibilities across several individuals depending upon operational requirements, security policy, compliance obligations, and organizational size.

The architecture supports both approaches.

The objective is not to increase administrative complexity, but to provide organizations with the flexibility to separate responsibilities whenever doing so improves security, accountability, or regulatory compliance.

Design Philosophy

C Chains extends the principle of Structural Protection beyond the protection of sensitive information.

Operational authority itself is compartmentalized.

Rather than requiring administrative capabilities to be concentrated within a single privileged account, C Chains allows operational responsibilities to be independently authorized, audited, and managed.

Organizations remain free to consolidate or separate these responsibilities according to their own operational needs.

Functional Categories

Protection Operations

Primary functions include:

- Protecting sensitive information
- Executing approved protection templates
- Running protection operations

Typical Authorization Level

Level 2

Audit & Compliance

Primary functions include:

- Reviewing protection activity
- Reviewing restoration activity
- Viewing audit records
- Managing audit information

Typical Authorization Levels

Levels 3 and 8

Codeset Administration

Primary functions include:

- Importing approved codesets
- Managing operational codesets
- Generating codesets

Typical Authorization Levels

Levels 4 and 10

Codeset generation normally remains under Sendatrix administration because of the high level of trust associated with this function.

Restoration Operations

Primary functions include:

- Executing the Fetch Engine
- Initiating restoration requests

Typical Authorization Levels

Levels 5 and 6

Organizations may separate these responsibilities or assign both functions to the same trusted individual depending upon organizational policy.

System Administration

Primary functions include:

- Managing operational users
- Maintaining administrative records
- Verifying system operation
- Reviewing system status

Typical Authorization Levels

Levels 7 and 9

Deployment Flexibility

One of the strengths of the C Chains architecture is that organizations may assign responsibilities according to their own operational requirements.

Small Organization

One trusted administrator may perform:

- System administration
- Audit
- Codeset import

Operational users perform:

- Protection
- Restoration

Medium Organization

Responsibilities may be divided among:

- Operations
- Audit
- Administration

Enterprise Organization

Responsibilities may be distributed among:

- Security Administration
- Audit Department
- Compliance
- Operations
- Sendatrix (Codeset Generation)

The architecture accommodates increasing separation of duties without requiring changes to the underlying software.

Audit Accountability

Regardless of how responsibilities are assigned, **every protected operation is associated with:**

- Authenticated user
- Authorization level
- Date and time
- Application used
- Transaction reference
- Audit reference

This provides complete accountability whether a single trusted administrator performs multiple functions or operational responsibilities are distributed across many individuals.

Architectural Significance

The Functional Role Model extends the philosophy of Structural Protection beyond protected data.

Just as sensitive information is compartmentalized within the architecture, operational authority may also be compartmentalized.

This reduces the concentration of privilege while providing organizations the flexibility to assign responsibilities according to their own operational requirements.

Chapter 5

Authorization Level Reference

Purpose

This chapter documents the authorization levels implemented within the C Chains Structural Protection architecture.

Unlike the Functional Role Model presented in the previous chapter, which groups responsibilities by operational function, this chapter provides a detailed reference for each authorization level.

These authorization levels define permissions associated with specific applications and operational functions.

They do **not** imply separate personnel or dedicated job titles.

A single trusted administrator may be assigned multiple authorization levels, while larger organizations may distribute responsibilities among several individuals to support separation of duties.

Design Philosophy

Each operational function within the C Chains environment is associated with one or more authorization levels.

Each authorization level grants only those permissions necessary to perform a specific operational responsibility.

Administrative authority is therefore compartmentalized rather than centralized.

Authorization Level Reference

Authorization Level 2

Protection Operations

Primary responsibility:

Protect sensitive information.

Capabilities include:

- Execute protection operations
- Apply approved protection templates
- Process records for Structural Protection

This is the primary operational level responsible for protecting sensitive information.

Authorization Level 3

Audit Review

Primary responsibility:

Review protection activity.

Using the Protection Audit Management application, this level may review:

- Protected record activity
- User performing protection
- Date and time of protection
- Protection template used
- Transmission/Returned Codes
- Associated log references

This level does **not** expose access to protected information.

Its purpose is operational accountability and auditing.

Authorization Level 4

Codeset Import

Primary responsibility:

Manage operational codesets.

Responsibilities include:

Import approved codesets
Maintain codeset availability

This level performs no protection or restoration operations.

Authorization Level 5

Fetch Engine Authorization

Primary responsibility:

Authorize execution of the Fetch Engine.

This level permits execution of the restoration engine itself but does not necessarily authorize users to initiate restoration requests.

Authorization Level 6

Restoration Operations

Primary responsibility:

Initiate restoration requests.

This level authorizes the application responsible for requesting protected information through the Fetch Engine.

Separating Levels 5 and 6 provides an additional operational boundary between requesting restoration and executing restoration.

Authorization Level 7

System Administration

Primary responsibility:

Maintain core C Chains operational records.

Responsibilities include:

- Verify system integrity
- Review operational records
- Confirm correct system operation

This level normally performs maintenance rather than routine operational work.

Depending upon deployment requirements, this function may remain under Sendatrix administration or be assigned to a trusted third party.

Authorization Level 8

Audit Administration

Primary responsibility:

Manage audit records.

Unlike Level 3, which reviews audit information, this level provides administrative authority over audit management functions.

This level is expected to be used infrequently.

Authorization Level 9

User Administration

Primary responsibility:

Manage operational users.

Responsibilities include:

- Create users
- Modify users
- Assign authorization levels
- Manage user permissions

Because improper assignment could affect system security, this level is typically reserved for trusted administrators.

Authorization Level 10

Codeset Generation

Primary responsibility:

Generate and manage cryptographic codesets.

This is the highest authorization level within the C Chains architecture.

Responsibilities include:

- Codeset generation
- Codeset lifecycle management
- High-trust cryptographic administration

Under normal deployments, this capability remains under Sendatrix administration because of the security significance of codeset generation.

Security Benefits

The authorization hierarchy provides several important architectural security benefits.

- Administrative authority is distributed rather than concentrated.
- Operational functions are independently authorized.
- Auditing responsibilities remain separate from operational responsibilities.
- Codeset management is isolated from protection and restoration operations.
- High-trust administrative functions are reserved for the highest authorization levels.
- Every privileged operation is attributable to an authenticated user and authorization level.

Most importantly, no single authorization level is intended to possess unrestricted authority across the entire Structural Protection architecture.

Architectural Significance

The authorization hierarchy extends the Structural Protection philosophy beyond protected data.

Within C Chains:

- Sensitive information is compartmentalized.
- Operational functions are compartmentalized.
- Administrative authority is compartmentalized.

Together, these complementary design principles reduce the concentration of both data and privilege while supporting comprehensive accountability through the C Chains audit framework.

Chapter 6

Security Objectives, Threat Model & Evaluation Scope

Purpose

The purpose of this chapter is to define the security assumptions, threat model, and intended protection scope of the C Chains Structural Protection architecture.

Structural Protection is designed to reduce the ability of an unauthorized party to reconstruct protected information following compromise of defined architectural components.

Like any security technology, it is designed with specific assumptions and objectives. Understanding those assumptions is essential to evaluating the architecture fairly.

Security Objectives

The purpose of C Chains is not simply to protect databases or encrypt sensitive information. Its primary objective is to reduce the ability of an unauthorized party to reconstruct protected information, even after compromise of individual components within the operational environment.

Primary Architectural Security Objective

To prevent the unauthorized reconstruction of protected information through a structurally compartmentalized architecture.

Authorized reconstruction requires coordinated participation of all three architectural dimensions. Neither one dimension nor an unauthorized combination of two dimensions is intended to be sufficient.

Rather than relying upon a single security mechanism, C Chains distributes the conditions required for reconstruction across multiple independent architectural components. Protected information is therefore intended to remain unavailable unless all required architectural conditions are legitimately satisfied.

Architectural Approach

C Chains achieves this objective through the coordinated application of the Structural Protection architecture described in the preceding chapters.

The architecture compartmentalizes:

- Data
- Trust
- Authority
- Identity
- Coordinated transaction processing

These architectural elements operate together rather than independently. The security properties of the system arise from their combined interaction rather than from any single mechanism.

Supporting Security Objectives

In addition to preventing unauthorized reconstruction, the architecture is designed to:

- Minimize exposure of sensitive information throughout its lifecycle.
- Reduce the impact of compromise of any individual architectural dimension or any unauthorized subset of the architectural dimensions.
- Eliminate unnecessary concentrations of trust.
- Prevent unauthorized restoration outside approved operational workflows.
- Support separation of operational authority through compartmentalized roles.
- Provide comprehensive accountability through immutable audit records.
- Preserve compatibility with existing relational databases and business applications.
- Complement existing cybersecurity controls rather than replace them.
- Preserve processing integrity by detecting incomplete, duplicated, substituted, or out-of-sequence transactions.
- Fail safely during component, service, or communication outages without exposing protected information.
- Support restoration of normal operations without bypassing authorization, reconstruction, or audit controls.

Design Philosophy

Structural Protection is intended to be evaluated on its architectural properties rather than on secrecy of design.

The architecture itself is designed to be openly documented and subject to technical scrutiny. The primary question for independent evaluation is therefore not whether the architecture is publicly understood, but whether the implemented system faithfully delivers the protection properties that the architecture claims.

Accordingly, the security objective of C Chains is not simply to prevent unauthorized access to systems. Its principal objective is to prevent unauthorized reconstruction of protected information, even when portions of the operational environment have been compromised.

Threat Model

The threat model should be interpreted together with the architecture and reconstruction requirements defined in Chapters 2 and 3.

The following threat scenarios define the principal classes of compromise against which the Structural Protection architecture is intended to be evaluated.

Each scenario represents an independent evaluation objective designed to determine whether protected information can be reconstructed under realistic compromise conditions.

This chapter defines the classes of threats the architecture is intended to address.

Designed to Address

1. Database Compromise

Assumption:

An attacker obtains the operational database, database backups, storage snapshots, exported datasets, or equivalent administrative access.

Objective:

Determine whether protected information can be reconstructed.

---

2. Privileged Insider

Examples:

- Database Administrator
- System Administrator
- Backup Administrator

Objective:

Determine whether privileged access alone permits reconstruction of protected information.

---

3. Partial Infrastructure Compromise

Examples:

Compromise of:

- SPA
- SPB
- SPC

individually.

Objective:

Determine whether compromise of any individual architectural dimension, or any unauthorized combination of two dimensions, permits unauthorized reconstruction.

---

4. Credential Abuse

Examples:

- Stolen credentials
- Excessive privilege
- Insider misuse

Objective:

Determine whether compromised credentials permit unauthorized reconstruction beyond assigned authority.

5. Replay / Transaction Abuse

Examples:

- Replay requests
- Transaction manipulation
- Codeset misuse

Objective:

Determine whether restoration can be abused outside authorized operational flow.

6. Processing Integrity Failure

Examples:

- Interrupted transactions
- Duplicated or replayed transaction steps
- Substituted transaction elements
- Out-of-sequence processing
- Incomplete audit chains

Objective:

Determine whether the architecture detects and rejects incomplete, altered, duplicated, substituted, or out-of-sequence transactions without exposing protected information.

7. Availability Disruption

Examples:

- SPA, SPB, or SPC unavailable
- Network interruption between dimensions
- Service outage
- Resource exhaustion or denial of service

Objective:

Determine whether disruption is handled safely, whether incomplete operations are rejected, and whether normal service can be restored without weakening protection controls.

8. Public Information Reconnaissance

Assumption

The attacker has access to publicly available information describing the architecture, including published documentation, presentations, diagrams, and other publicly available materials.

Objective

Determine whether public knowledge of the architecture materially assists unauthorized reconstruction of protected information.

This evaluation reflects the design philosophy that Structural Protection is intended to withstand architectural scrutiny without relying upon secrecy of design.

Not Intended to Address

Examples:

C Chains is **not intended** to replace:

- Firewalls
- Endpoint protection
- Network segmentation
- Identity management
- Multi-factor authentication
- Operating system hardening
- Patch management
- Malware detection
- Physical security

Those remain essential security controls.

Structural Protection complements rather than replaces those technologies by focusing specifically on reducing exposure of protected information.

Security Assumptions

The architecture assumes:

- Authorized software operates as designed.
- Administrative roles are assigned appropriately.
- Cryptographic materials are managed according to documented procedures.
- Supporting operating systems and infrastructure are maintained according to accepted security practices.
- Authorized users operate within their assigned authorization levels.

Evaluation Philosophy

The purpose of evaluation is not merely to determine whether an attacker can gain access to components of the system.

Rather, evaluation seeks to determine whether protected information can be reconstructed under defined compromise scenarios.

Accordingly, evaluation should include:

- Architectural review
- Implementation verification
- Adversarial testing
- Attempted reconstruction
- Evidence collection

Relationship to Penetration Testing

Traditional penetration testing often measures success by demonstrating compromise of systems or infrastructure.

The evaluation of Structural Protection extends beyond initial compromise to examine whether compromise results in unauthorized reconstruction of protected information.

This distinction does not replace conventional penetration testing; rather, it defines the principal architectural security property that independent evaluation is intended to verify.

The security objectives, assumptions, and threat scenarios defined in this chapter establish the basis for the Architectural Security Evaluation Framework presented in Chapter 7. The following chapter translates these objectives into a structured methodology for independent assessment.

Chapter 7

Architectural Security Evaluation Framework

Purpose

The purpose of this chapter is to define a structured methodology for independently evaluating the security properties of the C Chains Structural Protection architecture.

Because C Chains introduces architectural security properties that extend beyond traditional infrastructure protection, evaluation should include:

- *Architectural review*
- *Implementation verification*

The objective is not merely to determine whether components of the system can be compromised, but whether compromise results in unauthorized reconstruction of protected information.

Evaluation Methodology

The phases are derived directly from the security objectives and threat scenarios established in Chapter 6. Conventional penetration testing is one part of the broader evaluation and does not replace architectural review.

The Architectural Security Evaluation Framework is intended to progress from architectural understanding to implementation verification and finally to adversarial testing. Each phase builds upon the findings of the previous phase, allowing evaluators to understand the architectural intent before attempting to validate or challenge its implementation.

Phase 1 — Architectural Review

Objectives

- Review architectural documentation.
- Review threat model.
- Review trust model.
- Review assumptions.
- Challenge architectural claims.
- Identify logical weaknesses.
- Determine whether the claimed architectural security properties logically follow from the documented design.

Phase 2 — Implementation Verification

Implementation verification confirms that the deployed software faithfully implements the documented architecture.

Objectives

  - Verify software behavior.
  - Verify workflows.
  - Verify authorization.
  - Verify audit trail.
  - Verify role separation.
  - Verify restoration controls.
- Verify coordinated participation of SPA, SPB, and SPC during authorized restoration.
- Verify rejection of incomplete, duplicated, substituted, or out-of-sequence transactions.
- Verify safe failure and recovery behavior during component or communication disruption.

Phase 3 — Adversarial Evaluation

Objectives

Representative Evaluation Scenarios

- Database compromise
- SPA compromise
- SPB compromise
- SPC compromise
- Insider misuse
- Replay attacks
- Credential abuse
- Reconstruction attempts
- Unauthorized combinations of two architectural dimensions
- Availability disruption and denial-of-service conditions
- Transaction interruption, substitution, duplication, and sequencing abuse

---

Phase 4 — Performance Evaluation

Evaluate:

- Protection latency
- Restoration latency
- Transaction throughput
- Scalability
- Operational overhead
- Resource utilization
- Operational scalability
- Recovery time and operational continuity following controlled disruption

---

Success Criteria

The evaluation should determine:

- Whether protected information can be reconstructed from any individual architectural dimension or unauthorized subset of the architectural dimensions.
- Whether administrative authority alone permits reconstruction.
- Whether public architectural knowledge materially assists unauthorized reconstruction.
- Whether authorized restoration controls can be bypassed.
- Whether audit evidence remains complete, accurate, and verifiable.
- Whether measured performance is acceptable for the intended deployment environment.
- Whether incomplete or altered transactions are detected and rejected without loss of audit continuity.
- Whether component or communication failures are handled safely without exposing protected information.
- Whether normal service can be restored without bypassing authorization or reconstruction controls.

Expected Deliverables

- Architectural Review Report
- Implementation Verification Report
- Technical Evaluation Report
- Performance Evaluation Summary
- Consolidated Final Assessment

Transition to Chapter 8

The Architectural Security Evaluation Framework defines **what** should be evaluated and the methodology for conducting that evaluation.

Chapter 8 defines the evidence required to document the evaluation, support independent findings, and produce a comprehensive technical assessment.

Chapter 8

Evidence Collection & Reporting Standards

Purpose

The purpose of this chapter is to define the evidence required to support an independent evaluation of the C Chains Structural Protection architecture.

Because the architecture introduces security properties that extend beyond traditional infrastructure protection, conclusions should be supported by documented evidence demonstrating both the testing methodology and the observed results.

The objective is to ensure that evaluation findings are repeatable, technically defensible, and independently verifiable.

Evidence Categories

Evidence collection should distinguish architectural compromise, infrastructure compromise, processing-integrity failure, availability disruption, and actual unauthorized reconstruction.

Evaluation evidence should be collected throughout each phase of the Architectural Security Evaluation Framework.

Evidence may include:

- System configuration
- Test procedures
- Observed behavior
- Audit records
- Performance measurements
- Reconstruction attempts
- Screenshots and diagrams
- Log files
- Supporting technical analysis

Architectural Review Evidence

The Architectural Review should document:

- Architecture reviewed
- Documentation examined
- Security assumptions identified
- Trust boundaries evaluated
- Architectural strengths
- Architectural concerns
- Recommended implementation tests

Deliverable:

Architectural Review Findings

Implementation Verification Evidence

Evidence should demonstrate that the implemented system behaves consistently with the documented architecture.

Examples include:

- Protection workflow execution
- Restoration workflow execution
- Authorization enforcement
- Role separation
- Audit record generation
- Transaction sequencing
- Codeset utilization
- Error handling
- All-three-dimension participation during authorized restoration
- Rejection of unauthorized one- and two-dimension reconstruction attempts
- Detection of incomplete, duplicated, substituted, or out-of-sequence transactions
- Safe failure and recovery behavior during controlled disruption

Deliverable:

Implementation Verification Findings

Adversarial Evaluation Evidence

Each adversarial scenario should document:

- Objective of the test
- Test environment
- Assumptions
- Methodology
- Tools used
- Actions performed
- Results obtained
- Supporting evidence
- Conclusion

Where reconstruction is attempted, the report should clearly state:

- Whether reconstruction succeeded
- Under what conditions
- What architectural assumptions were required
- Whether the result represents expected or unexpected behavior

- Which architectural dimensions were compromised or available to the evaluator
- Whether one, two, or all three dimensions participated in the attempted reconstruction
- Whether transaction sequencing, authorization, or audit controls were bypassed

Deliverable:

Technical Evaluation Findings

Performance Evaluation Evidence

Performance testing should document:

- Test environment
- Hardware configuration
- Database platform
- Number of protected records
- Protection latency
- Restoration latency
- Transaction throughput
- Resource utilization
- Scalability observations
- Availability and recovery behavior under controlled disruption
- Processing-integrity results under interrupted or malformed transaction conditions

Deliverable:

Performance Evaluation Summary

Audit Evidence

Evaluation should verify that audit records accurately document protected operations.

Evidence should include:

- User identity
- Authorization level
- Date and time
- Operation performed
- Transaction identifiers
- Codeset references (where appropriate)
- Associated audit records

The evaluation should confirm that audit information supports accountability without exposing protected information.

Findings Classification

Evaluation findings should be classified according to their significance.

Suggested classifications include:

- Informational

- Low
- Moderate
- High
- Critical

Each finding should include:

- Description
- Impact
- Evidence
- Recommended action
- Architectural relevance

Final Evaluation Report

The completed evaluation should include:

- Executive Summary
- Scope of Evaluation
- Architectural Review Findings
- Implementation Verification Findings
- Adversarial Evaluation Findings
- Performance Evaluation Results
- Audit Verification Results
- Availability and Recovery Results
- Processing Integrity Results
- Findings Summary
- Conclusions
- Recommendations

Where appropriate, supporting evidence should be included as appendices.

Architectural Conclusion

The purpose of the evaluation is not solely to determine whether components of the environment can be compromised.

Its principal objective is to determine whether compromise of those components results in unauthorized reconstruction of protected information.

Accordingly, the final report should distinguish between:

- Successful compromise of infrastructure, and
- Successful reconstruction of protected information.

This distinction defines the principal architectural security property that the C Chains evaluation is intended to verify: compromise of infrastructure does not necessarily imply unauthorized reconstruction of protected information.

Closing Statement

The C Chains Security Architecture Guide has been developed to provide evaluators, security professionals, and prospective users with a structured understanding of the architecture, its security objectives, and the methodology proposed for independent evaluation.

As the architecture undergoes independent review and practical deployment, this guide is expected to evolve based on technical findings, operational experience, and recommendations from qualified security professionals. Its purpose is not only to document the current architecture, but also to support an ongoing process of validation, refinement, and transparency.

Future Revisions

Version 1.1 is intended to serve as the technical review baseline. Future revisions should be controlled, versioned, and based on documented findings from independent architectural review, implementation verification, adversarial testing, deployment experience, and qualified stakeholder feedback.

Revision History

Version	Date	Status	Notes
1.0	July 13, 2026	Technical Review Edition	Original publication.
1.1	August 2026	Updated Technical Review Edition	Publication styling standardized across the SDX Technical Publications series; front matter, typography, headers, footers, and table of contents updated. Technical architecture and evaluation framework retained.

Copyright