Research Report

STEVEN F. COMER, HUNTER PAVELA, VARUN GANDHI, KYLE SILER-EVANS, ERICH DEVENDORF, BEN KELLEY, JAMES GIMBI, JAIR AGUIRRE, GABRIEL KULP, MARK STALCZYNSKI, MATTHEW J. MALONE

Highly Secure Inference Data Centers

A Vertically Integrated Strategy for Security Engineering

The title of this publication was revised slightly in August 2026.

For more information on this publication, visit **www.rand.org/t/RRA4827-1**.

About RAND

RAND is a research organization that develops solutions to public policy challenges to help make communities throughout the world safer and more secure, healthier and more prosperous. RAND is nonprofit, nonpartisan, and committed to the public interest. To learn more about RAND, visit www.rand.org.

Research Integrity

Our research integrity is grounded in RAND's core values of quality and objectivity. Rigorous quality assurance procedures, conflict of interest screening, and transparency in funding ensure that every study is objective and nonpartisan. Learn more at www.rand.org/integrity.

RAND's publications do not necessarily reflect the opinions of its research clients and sponsors.

Published by the RAND Corporation, Santa Monica, Calif.
© 2026 RAND Corporation
RAND® is a registered trademark.

Limited Print and Electronic Distribution Rights

This publication and trademark(s) contained herein are protected by law. This representation of RAND intellectual property is provided for noncommercial use only. Unauthorized posting of this publication online is prohibited; linking directly to its webpage on rand.org is encouraged. Permission is required from RAND to reproduce, or reuse in another form, any of its research products for commercial purposes. For information on reprint and reuse permissions, visit www.rand.org/about/publishing/permissions.

RR-A4827-1

About This Report

This report presents a security strategy to protect artificial intelligence (AI) data centers from the most sophisticated and determined nation-state adversaries. We examine a secure inference data center designed to securely deploy trained models. We present a conceptual security architecture and a methodology for securely implementing features, and we recommend actions to accelerate deployment. The work builds on RAND's prior research on AI security and is intended to inform AI laboratories, cloud providers, and government stakeholders who may need to protect strategically important AI deployments or reduce the risk posed by dangerous AI systems.

Center on AI, Security, and Technology

RAND Global and Emerging Risks is a division of RAND that delivers rigorous and objective public policy research on the most consequential challenges to civilization and global security. This work was undertaken by the division's Center on AI, Security, and Technology, which aims to examine the opportunities and risks of rapid technological change, focusing on artificial intelligence, security, and biotechnology. For more information, contact cast@rand.org.

Funding

This research was independently initiated and conducted within the Center on AI, Security, and Technology using income from operations and gifts and grants from philanthropic supporters. A complete list of donors and funders is available at www.rand.org/CAST. RAND clients, donors, and grantors have no influence over research findings or recommendations.

Acknowledgments

We thank the following experts, as well as those who did not opt to be named, for their insights and contributions to the ideas in this report: William Young, Orr Dermer, Alvin Moon, Andrew Smith, and Sella Nevo.

We are grateful to our peer reviewers, Alex Larson at Dragos and Elizabeth Bodine-Baron at RAND, for their insightful and thoughtful comments and critique of the report.

Not all consulted experts or reviewers support all recommendations and conclusions in the report, and the views expressed in this report do not necessarily represent the views of any institution or organization with which the consulted participants are affiliated, including the U.S. government. The authors are solely responsible for any errors or omissions.

Summary

As frontier artificial intelligence (AI) systems continue to advance, the impact of model theft, manipulation, and misuse increases. Some future AI deployments will require substantially stronger security than ordinary commercial cloud environments can provide. This report presents a security strategy for a secure inference data center (SIDC): a purpose-built, Security Level 5 (SL5) facility for operating trained AI models in high-risk settings in which confidentiality and integrity are paramount.

The SIDC strategy is designed for cases in which protected assets must be defended against highly capable nation-state adversaries. We assess with high confidence that an SIDC built in accordance with our security strategy can preserve the confidentiality and integrity of model weights, algorithms, and inference data over a five-year operational period.

An SIDC can be implemented today using proven, off-the-shelf compute hardware: No fundamental research breakthroughs are required. The security strategy and system design are also agnostic to the model serving inference and to the hardware stack on which it runs. We estimate a cost of $37 million to $50 million for a proof-of-concept facility and $277 million to $345 million for a larger, enterprise-scale version. We predict that construction and deployment activities can be completed rapidly, in as few as 14 months.

This report is focused on conceptual analysis of the SIDC, omitting sensitive research and products on specific operational, architectural, and technical details.

Use Cases

Highly secure inference data centers can be useful for a variety of purposes and relevant in a variety of scenarios, such as the following:

- SIDCs can secure commercial AI systems (that have been trained in nonsecure facilities) that are being deployed in sensitive national security settings, protecting both the confidentiality of classified payloads and the integrity of the critical national security system that the AI system is incorporated into.
- SIDCs can act as emergency (temporary or long-term) hosting facilities in case strategically important or dangerous capabilities emerge (e.g., in biotechnology) or if a nation-state adversary embarks on a campaign to undermine or compromise U.S. AI systems. Such data centers would provide the (potentially urgent) ability to use the system without unacceptable risk to its confidentiality or integrity.
- Prototyping and building inference SIDC-style data centers (which can be far smaller, simpler, and cheaper than the smallest useful training data center) can act as a critical stepping stone and testing ground for developing the capability to construct full highly secure training data centers.

Note that none of the motivations above requires applying this high-security standard to all inference data centers or to commercial data centers providing global-scale services. To ground the discussion and examples in this report, we focus on the first use case described above: deploying commercial AI systems within sensitive national security environments.

Security Approach

The SIDC security strategy follows a concept-to-circuit approach: Security-critical design choices are derived from the strategic outcomes the system must prevent, rather than being added later as controls around general-

purpose infrastructure. We begin with the losses that would be unacceptable (such as theft of model weights or compromise of inference data) and use Systems-Theoretic Process Analysis for Security (STPA-Sec) to identify the constraints needed to prevent them.

Those constraints are then translated into concrete requirements, architectures, components, and implementation mechanisms, with traceability from strategic objective to circuit-level design. This approach favors minimal, purpose-built features because excess capacity introduces unnecessary risk. For the most security-critical elements, the SIDC strategy requires high trustworthiness, including formal methods and other rigorous assurance techniques, to demonstrate that components satisfy their required security properties.

Next Steps

Key early steps to deploying an SIDC include the following:

- **Identify an implementing entity and begin detailed engineering.** An AI laboratory, cloud provider, U.S. government agency, or public-interest organization should step forward, engage a system integrator, and initiate detailed design.
- **Select a site early.** Because major construction phases are largely sequential, early site selection is critical to minimizing the total project duration. For rapid deployment, housing the SIDC within an existing government facility, e.g., in an aircraft hangar, can bypass or significantly shorten key development steps.
- **Prototype key security features and integration now.** Critical security components, including formally verified protocols and unidirectional dataflows, can be tested on short timescales, independent of facility construction. Doing so is a low-cost opportunity to reduce technical risk and prevent downstream delays.

Contents

About This Report iii
Summary v
Figures and Tables ix

CHAPTER 1
Introduction 1
Security Objective 1
Scope 2
Limitations 3
Organization of This Report 3

CHAPTER 2
Approach 5
System-Theoretic Process Analysis for Security 5
Time and Cost Estimates 6

CHAPTER 3
Security Strategy 9
Threat Model and Security Objective 9
System Design 10
Security Guarantees 10
Operational Limitations 14

CHAPTER 4
Concept to Circuit 17
Example: Cross-Realm Solution 17
Assurance Path for the CRS 18
Trustworthiness and Formal Methods 19
Concept-to-Circuit Methodology 20
Next Steps 22

CHAPTER 5
Conclusions and Recommendations 23
Effective Security Requires Purpose-Built, Security-First Facilities 23
Such Security Is Appropriate for Only Targeted Use Cases 23
Recommended Next Steps 24
Conclusion 24

APPENDIXES
A. STPA-Sec Outputs 25
B. Time and Cost to Implement an SIDC 27
C. Concept-to-Circuit Tools and Methods 33

Abbreviations 37
References 39
About the Authors 41

Figures and Tables

Figures

3.1. Conceptual Diagram of the SIDC 13
4.1. Concept-to-Circuit Path 21
B.1. Timeline to Implement the SIDC, Normal Pace and Accelerated Pace 29

Tables

2.1. Example STPA-Sec Outputs 6
4.1. Assurance Chain Summary 19
A.1. STPA-Sec Outputs for the SIDC 25
B.1. Cost Estimates for Proof-of-Concept and Deployment-Scale SIDCs 28

CHAPTER 1

Introduction

As artificial intelligence (AI) models become strategically important to protect, highly secure data centers will be critical for ensuring that U.S.-developed frontier models remain protected from theft, sabotage, and misuse by sophisticated and well-resourced attackers. The need for such facilities could arise quickly, and preparation lead times may be short, making it critical to plan and build capacity now.

This report aims to advance the state of the art for secure inference data centers (SIDCs). Specifically, the report provides

- a high-level system architecture for an SIDC
- a vertically integrated security strategy, from national security objectives to technical controls
- a concept-to-circuit methodology for adding critical features without undermining the security effectiveness
- a worked example of a cross-realm communication mechanism
- cost, schedule, and implementation recommendations.

This report extends the foundational research from the prior RAND reports *Securing AI Model Weights* and *Securing AI Algorithmic Insights*.[1] While those previous works identified attack vectors and defined security levels, a significant gap remains in translating those security levels into implementable security strategies. This work addresses that gap by providing a conceptual security architecture and procedures to effectively translate from concept to circuit-level implementation.[2]

Security Objective

A key aim of this work is to define security strategies capable of defending AI infrastructure against Operational Capacity 5 (OC5) attacks, the most sophisticated threat category identified in previous research.[3] OC5 operations are defined as

> [o]perations roughly less capable than or comparable to 1,000 individuals who have experience and expertise years ahead of the (public) state of the art in a variety of relevant professions (cybersecurity, human

[1] Sella Nevo, Dan Lahav, Ajay Karpur, Yogev Bar-On, Henry Alexander Bradley, and Jeff Alstott, *Securing AI Model Weights: Preventing Theft and Misuse of Frontier Models*, RAND Corporation, RR-A2849-1, 2024; Asher Brass-Gershovich, Rachel Steratore, Wesley Hurd, Henry Alexander Bradley, Anjay Friedman, and Sella Nevo, *Securing AI Algorithmic Insights*, RAND Corporation, RR-A4685-1, 2026.

[2] Nevo et al., 2024.

[3] This work targets the same threat level as Security Level 5 (SL5) defined in the report by Nevo et al. (2024)—protection against OC5 adversaries—and shares substantial overlap in security approach. However, because the designs presented in this report were not constrained by the specific measures in that framework, we do not use the SL5 designation.

> intelligence gathering, physical operations, etc.) spending years with a total budget of up to $1 billion on the specific operation, with state-level infrastructure and access developed over decades and access to state resources such as legal cover, interception of communication infrastructure, and more.[4]

The security objective prioritizes the confidentiality and integrity of model weights, algorithms, the input and output tokens of the model, and other components critical to AI data center operations. Defending against an OC5 actor is an extremely high bar, exceeding established security practices for commercial and sensitive computing environments. Meeting this standard requires a combination of stringent security measures and significant operational constraints.

To state our security claim directly, we believe that an SIDC implemented and operated according to the concepts, specifications, and designs described in this report can withstand OC5 attacks for a five-year operational period without experiencing the unacceptable losses we aim to prevent.[5] This claim is specific to the OC5 threat model and to the system's stated security objective: preserving confidentiality and integrity, even when doing so requires sacrificing availability.

Scope

SIDCs are compact facilities designed to securely deploy already trained models. The focus of this report is a technical analysis to develop security engineering strategies that can credibly achieve the stated security objectives. We do not specify the thresholds at which such security is warranted, how to balance security with other objectives, or policy mechanisms to achieve data center security (e.g., regulation, incentives, public-private partnerships). Further details about the size, time, and cost of our representative data center are provided in Appendix B.

The SIDC considered in this report is intended for scenarios in which the risks of compromise are exceptionally grave and warrant extreme security. Our primary focus is a representative national security application. As AI systems increasingly support the analysis of sensitive intelligence and inform critical national security decisions, high-assurance AI infrastructure becomes essential. These systems are likely to be exceptionally high-value targets for nation-state adversaries: A single facility could contain highly capable AI models (representing billions of dollars in research and development investment), multiple streams of classified information, and the potential to manipulate strategic government decisionmaking.

The SIDC strategy presented here provides high-assurance security to prevent OC5 adversaries from manipulating model inputs or outputs to covertly influence decisionmaking or exfiltrating model weights and classified data.

While we employ the national security use case as a concrete example, we emphasize that the approach and resulting insights presented in this report should be viewed more broadly. A similar approach could be used to implement SIDC facilities to provide emergency response capacity if high-security requirements arise rapidly—for example, as a result of dangerous emerging capabilities (e.g., biological weapon design) or coordinated attacks on AI infrastructure. In addition, a small-scale SIDC facility could serve as a pilot project to derisk investment in larger secure data centers.

Readers should treat the security strategy presented here as an illustrative example of what it takes to accomplish OC5-level protection generally. Relevant stakeholders are encouraged to use the approach and

[4] Nevo et al., 2024, p. 10.

[5] Although we expect the SIDC to hold security value beyond five years, the ability to withstand long-duration OC5 campaigns becomes less robust the longer the facility remains in operation.

principles illustrated in this report to implement purpose-built security for their unique use cases, operational constraints, and security requirements.

Limitations

We note several important limitations that should inform how readers interpret and apply these findings. First, the analysis is primarily conceptual rather than operational. This report does not include architectural blueprints, detailed technical implementations, or step-by-step deployment procedures. These are omitted not because we think that these details should be immune from scrutiny. On the contrary, they require robust verification. Those evaluations must occur privately with relevant stakeholders, not in a public report that hands attackers a precise map of the technical controls employed to secure the SIDC. Interested parties should engage with RAND to obtain additional specifications.

The cost and schedule estimates contained in Appendix B are derived from conceptual designs and are inherently imprecise; they should be regarded as order-of-magnitude estimates suitable for strategic planning, not as procurement-ready budgets.

Finally, no absolute method or benchmark exists for establishing sufficient security against the most-sophisticated adversaries. Although we use structured methods and rely on expertise within and external to the project team, no collection of security controls can provide an absolute guarantee of successful defense against an OC5 attack, and no security claims can hold for an indefinite period of time.

Organization of This Report

This report is organized as follows: Chapter 2 briefly describes the analytical methods, including the security design approach and the basis for the time and cost estimates. Chapter 3 details the security strategy. Chapter 4 shows the strategy continuity from concept down to the circuit level. Chapter 5 concludes with key findings and recommended next steps for AI laboratories, cloud providers, and government stakeholders who may need to secure strategically important AI capabilities.

CHAPTER 2

Approach

This chapter describes our approach to (1) developing security strategies capable of defending against the most-sophisticated and most-determined nation-state adversaries and (2) estimating the time and cost to implement the envisioned facilities. The methods for each are summarized below.

System-Theoretic Process Analysis for Security

To develop a security strategy capable of resisting OC5-level threats, we employed System-Theoretic Process Analysis for Security (STPA-Sec), a top-down methodology rooted in systems engineering that treats security as a fundamental control problem.[1] Rather than layering a collection of security features onto existing architectures, STPA-Sec seeks to develop a *concept of secure function* by modeling a holistic system. In our case, this means a data center serving sensitive AI operations. Security is treated as an emergent property of the whole system, including humans and the ways they control, operate, and interact with it, rather than a feature of any individual component.

The process begins by asking: What are the goals of the system, and which losses are unacceptable? The methodology then works systematically through *hazards* (system states that can lead to losses), *constraints* (conditions that prevent those hazards), and *controls* (mechanisms that enforce those constraints). Examples of these outputs are shown in Table 2.1, and the full set of goals, losses, and hazards is provided in Appendix A. The STPA-Sec structure creates explicit traceability from each security control back to the unacceptable losses that it prevents.[2] Stakeholders can evaluate any proposed control by weighing its cost or operational impact against the specific losses that it is designed to prevent.

Our engineering effort adopted a security-first posture: When design choices created tension between security and other considerations, such as system availability, operational convenience, or cost, we consistently prioritized security. The resulting security strategy is intended to withstand OC5-level attacks with high confidence. As noted in Chapter 1, readers should view the resulting security strategy as an illustrative example, and relevant stakeholders are encouraged to use the approach and principles illustrated here to implement purpose-built security for their unique use cases, operational constraints, and security requirements. The same structure supports systematic verification, red-team evaluation organized around the hazard-constraint framework, and audits that test the efficacy of deployed controls.

[1] SAE International, "Standard J3307_202503: System Theoretic Process Analysis (STPA) Standard for All Industries," March 24, 2025; William Young and Nancy G. Leveson, "An Integrated Approach to Safety and Security Based on Systems Theory," *Communications of the ACM*, Vol. 57, No. 2, February 2014.

[2] In many cases, controls will have a one-to-many mapping to constraints, hazards, and losses.

TABLE 2.1
Example STPA-Sec Outputs

Element	Example
System goals	Maintain international lead in AI.
Unacceptable losses	Confidentiality of model weights is lost.
Hazards	SIDC allows unauthorized information transaction across realm boundary.
System constraints	Channel controller prevents unauthorized information transactions across realm boundary.
Security controls	Formally verified field-programmable gate array (FPGA) implements specified state machine and guarantees only specified behavior.

Complementing STPA-Sec, we draw on important concepts from National Institute of Standards and Technology's (NIST's) *Engineering Trustworthy Secure Systems*.[3] We use the Security Design Order of Precedence (SecDOP), which establishes engineering design priorities for eliminating hazards. The SecDOP guided our emphasis on minimizing and eliminating unnecessary features. As Chapters 3 and 4 will show, we approach secure data centers as purpose-built facilities. Functions not essential to the system goals are excluded rather than merely controlled.

We use the principle that rigorous secure systems engineering must address three distinct contexts: problem, solution, and trustworthiness. This led us to invest substantially more effort in the problem and trustworthiness contexts than is typical. STPA-Sec enables us to address the problem context, traditional systems engineering addresses the solution context, and we generally specify "high" levels of trustworthiness for the solution at critical points. In this context, high trustworthiness generally translates to heavy use of formal methods to prove that a system or component possesses certain desirable properties.

The security strategy for the SIDC (Chapter 3) is built around five system goals, five unacceptable losses, and 11 primary hazards (detailed in Appendix A).

Time and Cost Estimates

To estimate the cost required to construct secure data centers, we primarily used estimation by analogy, comparing data center costs on a dollar-per-megawatt or dollar-per-square-foot basis using publicly available data from recent projects and industry benchmarks. This top-down approach is well suited to estimating projects that are in the conceptual phase, for which detailed design information is unavailable.[4] In select cases, we used bottom-up estimating to cross-check results and provide additional precision on specific security controls. Compute hardware was estimated separately from facility construction because hardware can represent more than half of the total project costs.[5]

The cost estimates in this report should be interpreted as order-of-magnitude estimates suitable for strategic planning, not detailed procurement budgets. Actual costs will vary substantially based on site selection, final design, market conditions, and other project-specific factors.

[3] Ron Ross, Mark Winstead, and Michael McEvilley, *Engineering Trustworthy Secure Systems*, National Institute of Standards and Technology, NIST SP 800-160v1r1, November 2022.

[4] Warfighting Acquisition University, "Cost Estimation Methods," webpage, undated; Christine Green, "Estimating as an Art—What It Takes to Make Good Art," paper presented at PMI Global Congress 2006, Madrid, Spain, October 2006.

[5] Didier Vila, "Deconstructing the Data Center: A Look at the Cost Structure Igniting the AI Boom!" Alpha Matica, undated.

For time estimates, we drew on historical data from recent projects and industry planning estimates to establish typical development timelines across all major phases: land acquisition, design and engineering, permitting, construction, and commissioning. We then assessed whether specialized security requirements would extend critical path activities or could be done in parallel with standard development steps. Finally, we considered how timelines might be compressed under emergency conditions, in which building the secure data center becomes a top national priority.

CHAPTER 3

Security Strategy

This chapter presents our minimum viable concept for an SIDC, which serves as a foundation for protected AI inference deployments in security-sensitive environments. The chapter aims to build familiarity with the secure base architecture and critical security guarantees.

We define the security objectives and threat model, explain the core design principles and high-level security architecture, and enumerate operational limitations. In the subsequent chapter, we share a concept-to-circuit example and provide a methodology by which users can add features to the SIDC without compromising security.

Threat Model and Security Objective

The SIDC facility described in this chapter is designed for a representative national security deployment in which AI models are used to inform strategic planning and decisionmaking. We assume a persistent adversary with OC5-level capabilities: that is to say, a well-funded, state-backed adversary with deep technical expertise, access to sovereign infrastructure and authorities, and the capacity to sustain a dedicated, multiyear campaign against a single target. This encompasses sophisticated cyber, supply chain, and physical attacks against data center infrastructure and personnel.[1]

Under this threat model and with our selected objectives, any compromise of the protected information and system properties is considered unacceptable. Accordingly, the SIDC prioritizes *confidentiality* and *integrity* over *availability*. Because the system is highly isolated and minimally featured, we expect such outages to be rare, but the system includes a fail-safe state where features are temporarily disabled in circumstances in which protected information would otherwise be disclosed or corrupted. The system operates in this state while investigation and recovery procedures occur. Regarding those elements of protected information, the SIDC must ensure confidentiality and integrity for

- copies of model weights, inference algorithms, and reference data that exist inside the SIDC
- inference prompts and responses as and while they exist inside the SIDC.

Furthermore, the SIDC makes the following additional security claims:

- Each inference response is verifiable, given valid model weights, model architecture, and the retrieved context from reference data storage.
- Compromise of any single subsystem or person does not enable other activity that violates these protection objectives.

[1] See Nevo et al., 2024.

The system design, security mechanisms, and verification methods presented in the remainder of this chapter are derived from these objectives.

System Design

The SIDC is a cyber-physical system that executes AI model inference workloads under explicit constraints on information flow, physical access, matter and energy transfer, and operational authority. It enforces the principle of least privilege under a default-deny, fail-secure policy. Our design philosophy is described by three architectural invariants (detailed further in the "Security Guarantees" section): realm partitioning, fixed diode-gated realm topology, and formally verified cross-boundary protocols. Each is summarized in the next section.

The system is partitioned into a finite set of mutually isolated realms, each with a formally defined set of privileges. A *realm* is a physically isolated domain with exclusive control over its computational resources, storage, devices, and personnel access. Each realm defines inbound and outbound interfaces and a set of capabilities governing cross-realm interactions. All computation, storage, devices, sensors, actuators, and human interactions are assigned to specific realms such that no component operates outside the realm model and no realm shares a writable state with any other.

Cross-realm interactions—whether data, personnel, or matter—occur only through predefined interfaces with authenticated end points; all other interactions are prohibited. Energy flows, such as power delivery, thermal dissipation, and electromagnetic emissions, are continuously measured and bounded, and any deviations from predefined profiles are treated as unauthorized interactions. The SIDC enforces fail-secure semantics: Any deviation from expected control, data, matter, energy, or personnel flows triggers a hazard, followed by immediate containment actions. All realms are equipped with a secure audit system that guarantees synchronous log availability for all intra-realm and inter-realm interactions for hazard analysis and resolution. Together, these mechanisms ensure that compromise of any single realm cannot yield unauthorized privileges or lateral access to other realms.

Security Guarantees

The security strategy is derived from the STPA-Sec product, in which the defined security measures enforce constraints to prevent hazardous states and thereby avoid unacceptable losses. Appendix A provides the enumerated goals, losses, and hazards for the SIDC. These measures are not intended to be applied selectively; they function as an integrated system whose effectiveness depends on the integrity of the whole. Because comprehensive prevention of hazard states is not possible, the security strategy is underpinned by tailored detection and response procedures to recover from fail-secure events.

Confidentiality and integrity of inference computation together form the central security objective of the SIDC. Under the threat model defined previously, the system's architecture ensures that conditions leading to a violation of confidentiality or integrity are unreachable. As mentioned above, this assurance rests on three architectural invariants:

1. **Realm partitioning.** The system is decomposed into mutually distrustful security domains, each treated as a potentially adversarial principal. No realm assumes the correctness or benign nature of any other.

2. **Fixed diode-gated realm topology.** All inter-realm communication traverses data diodes arranged in a fixed, immutable topology. These diodes enforce unidirectional information flow at the physical layer, eliminating reverse-channel and covert-channel creation by construction.
3. **Formally verified cross-boundary protocols.** Every dataflow crossing a realm boundary is mediated by a protocol implementation that has been verified against its security specification. This reduces the trusted software surface to the cross-realm protocol layer, which can be compromised only in the event that an attacker is successful in bypassing the trusted setup procedures or as a result of an insider threat that is beyond the SIDC's tolerable threshold. Intra-realm software, except cryptographic key management, is untrusted by design and cannot alone cause an unacceptable loss of confidentiality or integrity.

These three architectural invariants do not merely harden the system but also enable enforcement of the following five security properties:

- **P1: Cross-boundary flow containment.** Every flow of data, energy, matter, or personnel between realms is an element of a fixed set of authorized interactions established during trusted setup. No flow outside this set can be instantiated at runtime without triggering a hazard.
- **P2: Realm-level computational-state confidentiality.** Plaintext computational state is confined to the realm in which it originates and cannot cross a realm boundary except through explicitly authorized output channels. In the primary case, all Trusted Inference Compute Sanctum (TICS) runtime state, including model weights, inference algorithm, intermediate tokens, and reference data, exists in plaintext only within the TICS boundary, with no computational state exiting the TICS except finalized inference responses delivered to the Interface Room (IR).
- **P3: Integrity of inference.** Dataflow integrity of the inference response is enforced throughout its life cycle. This is guaranteed through cryptographic integrity verification of the inference computation, the context retrieved from the reference data storage, and periodic verification of model weights and reference data storage against the golden state maintained in the Trustworthy Recovery Vault (TRV) realm.
- **P4: Assured recovery.** For all failure modes except physical compromise of the TRV, the system can recover to a known-good, authenticated operational state.
- **P5: Cross-realm privilege separation.** Compromise of any single realm yields no unauthorized capability in any other realm.

Each security property is maintained by a pipeline of hazard prevention, detection, and response. Architectural constraints prevent defined classes of violations by construction. Where prevention is infeasible, continuous monitoring detects hazard conditions, and the system executes an immediate fail-secure state transition. Together, these security properties encapsulate our defense-in-depth approach: P1 contains all cross-boundary flows to a fixed, enumerated set. P2 layers a confidentiality guarantee over the computational state of a realm. P3 ensures that the computation isolated by P2 is authentic. P4 makes detection of any violation of P1, P2, and P3 meaningful by providing a credible recovery path. P5 bounds the impact of any breach to a single realm.

Conceptual Architecture and Realm Descriptions

The facility implements defense-in-depth through nested physical and logical boundaries as shown in Figure 3.1. A guarded perimeter fence defines the outer site boundary. Within this perimeter, the facility is partitioned into eight functionally and physically isolated realms: the IR, the Payload Exchange Room (PER),

the TICS, the TRV, Information Technology Management (ITM), the Protection Operations Center (POC), Utility, and Restroom. We describe each realm below:

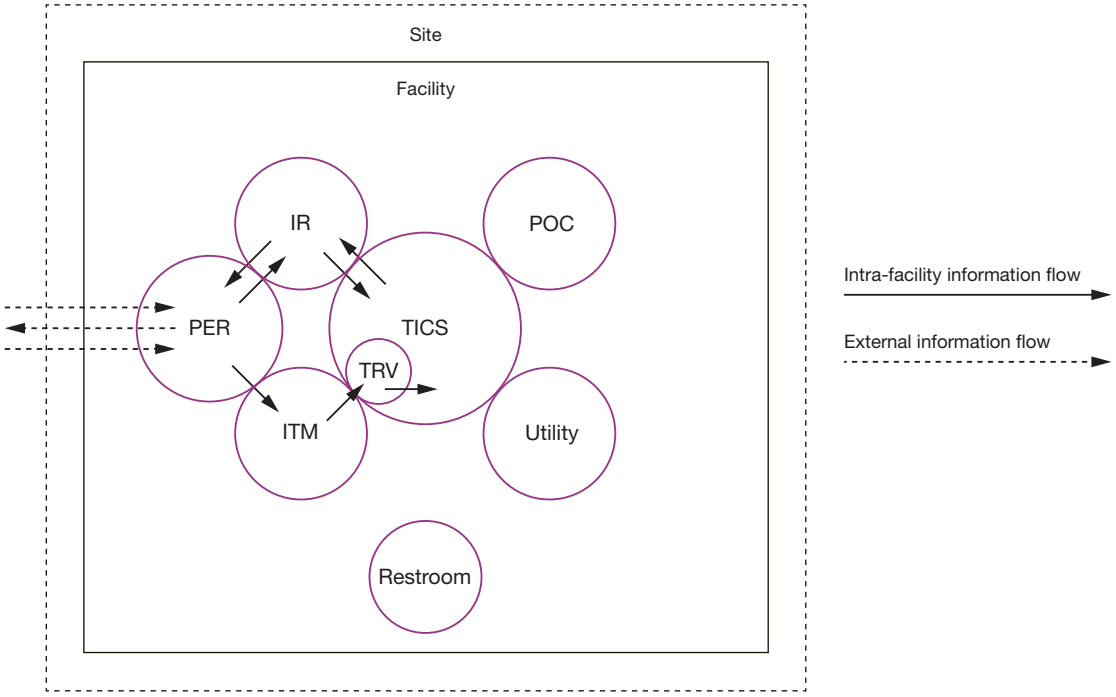
- **IR**—the sole interface between external inference users and the TICS. The IR receives authenticated inference requests and delivers responses, screening plaintext inputs and outputs but never handling model weights or inference runtime state.
- **PER**—the proxy for legitimate payloads inbound to and outbound from the SIDC facility. Inbound, only two legitimate payloads are accepted. These are the inference prompt (proxied to the IR) and reference data updates (proxied to the ITM). Outbound, the only accepted payload is a legitimate inference response.
- **ITM**—trusted to ingest and validate vetted reference data under two-person integrity controls and is required for the trusted setup phase of SIDC, including compute hardware, encrypted model weights, and initial state for the reference data storage.
- **TICS**—the most critical security perimeter, housing model weights, the inference algorithm, reference data storage, and all compute hardware. All cross-boundary flows are eliminated or reduced to single-bit control signals, with the sole exceptions of inbound inference requests from the IR, outbound inference responses to the IR, and inbound pre-staged reference data from the ITM.
- **TRV**—the integrity anchor for the TICS. The TRV stores protected, read-only copies of critical system state and reference measurements used for integrity verification and recovery. It is connected to only the TICS and the ITM, providing reference baselines against which both realms verify the authenticity of their operational state. The TRV remains quiescent during normal operation and is activated only during system initialization or in response to a hazard.
- **POC**—the facility's centralized security oversight and incident response authority. The POC receives hazard signals and telemetry from all realms through dedicated unidirectional inbound channels and maintains outbound control and signaling channels to all realms, including the sole authority to trigger a temporary or destructive shutdown. The POC is fully isolated from model data and computation (i.e., its channels carry only control signals and hazard telemetry, never inference data).
- **Utility**—supplies power to other realms in the facility, performs conditioning on energy flows crossing the facility boundary to mitigate the risk of unauthorized information channels being constructed in the energy flow, and is responsible for cooling activities.
- **Restroom**—routine use for facility personnel provides a source of drinking water, sink water, and toilet waste disposal water. Supplied by an on-site well for inbound water and an on-site septic system for outbound waste. Both directions of matter flow are designed with special attention to the risk of covert information channels.

Inter-realm information channels are actuated using unidirectional control signals transmitted through diodes and after actuation are authenticated using post-quantum cryptographic primitives. Personnel movement and matter exchange across realm boundaries are governed by strict access control and two-person integrity controls, respectively.

Trusted Setup Procedures to Establish Integrity Basis

Prior to operational deployment, the SIDC undergoes a trusted setup phase that establishes the root of trust for all subsequent security guarantees, encompassing secure construction of the facility, hardware ingestion, cryptographic key initialization, and ingestion of model weights and the initial state of the reference data storage.

FIGURE 3.1
Conceptual Diagram of the SIDC



Facility construction is conducted under continuous physical security and chain-of-custody controls. All hardware undergoes destructive sampling and verification prior to installation. After verification, components that pass are sealed, logged, and installed under two-person integrity controls. Cryptographic key material, including realm identity, channel authentication, and attestation keys, is generated within the facility boundary, bound to verified hardware, and never derived from externally held secrets. Once the cryptographic infrastructure is established, model weights and the reference data storage are ingested into the TICS and verified against provider-supplied reference measurements.

All reference measurements established during trusted setup, including hardware attestation baselines, cryptographic key bindings, model weight digests, and reference data fingerprints, are stored in the TRV as the authoritative basis for integrity verification throughout the facility's operational lifetime.

An important note on the effectiveness requirement for both the hardware and software screening activities: Unless otherwise noted (e.g., for hardware-based cryptographic root of trust), these screening and risk-reduction activities for the hardware and software acquired and ingested for use in the facility are assumed to fail in comprehensively achieving their aims. Furthermore, we assume that successful supply chain compromise of commodity compute hardware and of software components in the inference stack is not only possible but likely. We plan accordingly in the security strategy. The purpose of undertaking these imperfect risk-reduction activities is to increase the cost and complexity of a successful attack.

Inference Life-Cycle Activities

Following the trusted setup, the SIDC enters its operational state, and the TICS holds verified model weights and reference data storage. The life cycle of a single inference request then proceeds as follows:

- Assumptions
  - The SIDC site is within walking distance of a facility with classified network connectivity.

  - The SIDC and the adjacent classified facility are connected by three one-way, tailored point-to-point links. Each of these links is a separate instance of the feature that we label the Secure Transfer Inter-facility Conduit (STIC). Each STIC is responsible for a single message type: inference prompts inbound, inference responses outbound, and reference data inbound.
  - Prompts and responses are encrypted and digitally signed before being transported between facilities.
  - Required SIDC constraints are enforced during each transaction.

- Outside the SIDC
  - A national security official in a classified facility (hereafter referred to as the *end user*) specifies an inference prompt. This prompt is transferred via a classified network from the end user's location to the classified facility adjacent to the SIDC site.
  - The prompt is transferred from the classified network to the inbound prompt STIC for transport into the SIDC.

- Inside the SIDC
  - The PER ingests the prompt from the STIC and transfers it to the IR.
  - IR personnel screen the input to assess compliance and to protect against adversarial inputs.
  - The screened request is transmitted to the TICS over an authenticated channel.
  - The TICS executes the inference workload using the resident model and reference data storage.
  - The TICS verifies that the resident model weights and reference data storage match the reference measurements stored in the TRV, ensuring that the inference response was generated using authenticated artifacts.
  - The TICS returns the finalized response to the IR.
  - IR personnel screen the output, as on the inbound path.
  - IR transfers the response back to the PER.
  - The PER transfers the response to the outbound response STIC.

- Outside the SIDC
  - The adjacent classified facility receives the response via its response STIC.
  - The response is transferred via a classified network back to the end user.

We have estimated the total round-trip time for a user to submit a prompt to the SIDC and receive an inference response back, including those activities that occur outside the SIDC. The total figure is an estimate of the overhead, not including the time required for the model to perform inference computation. A conservative range for the overhead of a full inference-response life cycle is three to five minutes. Human-mediated review steps in the IR are the rate-limiting factor. It is plausible that the lower bound of the time estimate could be reduced further by investing in systems that remove human-mediated processes and replace them with machine automation—as long as such investments maintain the security constraints defined in the Intent Specification. Such investments would likely bring the security overhead to less than one minute in total.

Operational Limitations

The SIDC operates under extreme security conditions that impose operational limitations, which we broadly describe here. Users desiring additional features can add them by implementing the procedures described

in Chapter 4. The system starts with and requires a trusted setup phase in which reference measurements for model weights and the baseline contents of reference data storage are established; it cannot detect pre-ingestion compromise if the trusted setup itself is compromised. The facility secures the infrastructure, not the model's reasoning. It does not defend against undetected biases or latent model behaviors (i.e., model outputs being fallible).

Additionally, the SIDC's security architecture imposes significant operational constraints, which are a significant departure from typical commercial operations. These trade-offs stem from prioritizing confidentiality and integrity against OC5-level adversaries over system availability and operational convenience. Key trade-offs include the following:

- **Availability and latency.** The SIDC prioritizes confidentiality and integrity over system availability. The facility tolerates significant latency, necessary for screening inbound and outbound data, and may enter a permanently inoperable state under extreme security conditions, such as detected physical compromise of hardware in the inner sanctum. Specifically, we estimate the total round-trip overhead time (on top of model computation time) for an inference request to be in the range of three to five minutes, and we believe that straightforward automation of a few non-security-critical processes could reduce it to less than one minute. As a single physical facility, the SIDC could be rendered effectively inoperable through physical destruction or sustained denial of access. Although the TRV preserves state for recovery from hazards, operational continuity requires either a redundant site or acceptance of downtime. While these trade-offs are likely unacceptable in globally served and commercially focused, low-latency environments, the architecture remains suitable for some national security and other use cases in which security requirements justify reduced availability and throughput.
- **Limited connectivity and fixed software.** The facility eliminates traditional network connectivity across the perimeter; inference prompts, responses, and reference data updates cross the facility boundary via the highly constrained STIC link rather than a traditional network. Software is fixed at deployment, with no updates during the five-year operational life except for scrutinized configuration changes by on-site engineers. This ensures that attackers cannot introduce malicious code through software updates or network channels but means that this specific facility's software and algorithms will lag external developments by up to five years at the end of the facility's life. A mitigation here, given the relatively modest costs described in Appendix B, which can achieve both security and progress velocity, could be to build new small data centers on a rolling basis, if one desires.
- **Special-purpose design.** Industry-standard data centers make heavy use of general-purpose computing and network equipment. At the decisive points where we choose to throttle potential attacks on the SIDC, we have deliberately avoided general-purpose components and have instead opted for special-purpose design. Minimal and tailored component engineering greatly reduces the attack surface that would otherwise be present in general computing equipment. This decision comes with additional time and capital requirements.
- **Restricted physical inputs.** Matter transfers are tightly constrained and largely limited to humans and facility-authorized clothing. All hardware, including spare components for anticipated failures and destructive testing, is staged during construction. This mitigates the risk of postdeployment supply chain attacks during operations but locks in compute capabilities at construction time, which degrade relative to leading-edge chips over the facility's lifetime. A possible mitigation for this limitation mirrors the one proposed above for software: rolling construction of small data centers.
- **Minimal human presence.** Personnel access is restricted to essential functions only. Maintenance, emergency response, and routine operations are performed by permanent facility employees rather than by those with an ephemeral facility presence, minimizing the counterintelligence and insider-threat

burden for presence inside the facility. This reduces flexibility and may require maintaining specialized expertise on site.

These constraints trade operational flexibility for high-confidence protection. While severe by general-purpose commercial standards, they address scenarios in which security requirements justify substantial inconvenience (i.e., national security applications involving classified information or emergency containment of dangerous AI capabilities). We acknowledge that the security strategy presented up to this point may sound familiar to national security professionals. We believe that the SIDC differs most heavily from the status quo in its tailored path to engineering closure, described in the next chapter.

CHAPTER 4

Concept to Circuit

This chapter describes a concept-to-circuit template that extends the SIDC security strategy. The central idea is that the design and engineering of security-critical features should not begin with general-purpose components that are constrained after the fact. Instead, these features must derive from (and trace back to) the strategic losses that the system must prevent, the hazards that could lead to those losses, and the system constraints that prevent those hazards. The resulting components should realize only the minimum function needed to satisfy feature requirements and security constraints. Excess capacity in any system, beyond the minimum required to support specified requirements, represents unnecessary risk and a possible attack surface that an adversary can leverage.

As stated above, many efforts to design and deploy secure information systems begin by attempting to constrain the behavior of general-purpose components and protocols. We fundamentally diverge from this approach. When we choose to enforce a security boundary, we start with specification and strict enforcement of transaction policies across all forms of matter, energy, and information. The starting constraint is that all transactions of all types are unauthorized. The system enforces this constraint using SecDOP: Prevent hazards when possible, and monitor, detect, respond, and recover when prevention is infeasible.

After achieving sufficient confidence that unauthorized transactions are impossible or detectable, the design process builds upward from the physical layer and hardware to support the specific features required, and nothing else. For example, at the physical layer, an information channel should not have any information-theoretic capacity that is not strictly required. At selected decisive points in environments in which information-related losses have catastrophic consequences, we believe that this security approach is necessary.

The purpose of this chapter is not to provide a complete implementation manual for an SIDC. Rather, it demonstrates how the security strategy can be made concrete for one security-critical subsystem, the cross-realm solution (CRS). The CRS example illustrates how a feature moves from system-level security intent to a protocol, from a protocol to finite-state behavior, and from finite-state behavior to hardware-realizable circuit logic with verification evidence.

Example: Cross-Realm Solution

The most mature example of the concept-to-circuit approach is the CRS, the mechanism that mediates authorized transactions across SIDC security boundaries. The CRS is not a generic data transfer device; each instance is engineered to its authorized transaction and constraints. Each CRS facilitates a single cross-realm flow. For all other cases, the CRS prevents or detects and fails securely.

The CRS exists to prevent losses involving disclosure or corruption of protected information. At the hazard level, it addresses unauthorized information transfer across realm boundaries. At the constraint level, it enforces the rule that only authorized transactions of specified types may cross a boundary and only under specified conditions. At the requirement level, this becomes a set of rules about message type, directional-

ity, sequencing, end-point authentication, malformed input handling, state transitions, and hazard response. The CRS is therefore a pathfinder for the SIDC's larger engineering claim that security-critical controls should be derived from system constraints, not from the latent capabilities of general-purpose communication equipment.

The SIDC architecture depends on the ability to permit a very small number of controlled transactions between otherwise isolated realms. The most sensitive of these interactions involve the TICS boundary, where model weights, inference runtime state, and protected computation reside. The CRS is intended to provide a repeatable pattern for these high-consequence boundary crossings.

The CRS supports the SIDC's architectural invariants in the following ways:

- **Realm partitioning.** The CRS assumes that the source and destination realms are separate security domains. It does not require either realm to trust the internal correctness of the other.
- **Fixed diode-gated topology.** The CRS is designed around a physically constrained direction of high-bandwidth dataflow. It does not provide a general bidirectional network connection.
- **Formally specified cross-boundary behavior.** CRS behavior is expressed as a finite-state protocol with defined transitions, timing bounds, authentication requirements, and fail-secure states.

This combination matters because a cross-realm mechanism is one of the places where small design errors can invalidate the security architecture. A conventional network connection between realms would introduce a large, general-purpose attack surface. The CRS instead narrows the boundary to a specific transaction pattern: Authenticate the end point, enable a one-way data path for an authorized transfer, and close or block the path when the transaction completes or when any hazard is detected.

The CRS pathfinder decomposes the subsystem into circuit-relevant elements. At a high level, these include

- transmit-side logic (TX), responsible for preparing or presenting authorized outbound transactions
- receive-side logic (RX), responsible for accepting only authorized inbound transactions
- channel-control logic (CC), responsible for enforcing the transaction policy and state-machine behavior that governs the channel.

Assurance Path for the CRS

The CRS is a useful pathfinder because it exercises the full concept-to-circuit assurance chain. The assurance argument does not rest on any single tool. Instead, each layer supports a different type of claim.

At the system level, STPA-Sec identifies unauthorized information transfer across a realm boundary as a hazard and derives constraints on authorized transactions. System modeling then records the CRS boundary, interfaces, component responsibilities, and requirement allocation.

At the first level of protocol specification, the CRS is specified as a finite-state interaction between the source transmitter and the destination channel controller, with explicit states, transitions, timing constraints, authentication requirements, and hazard behavior. This specification is written in a form suitable for formalization.

At the formal protocol level, cryptographic analysis evaluates whether the authentication and session-establishment design supports such claims as secrecy, authentication, replay resistance, and key freshness under the stated assumptions. Symbolic and computational tools are useful for different parts of this level.

At the state-machine level, model checking can evaluate whether the abstract CRS control logic permits unsafe reachable states, such as data transfer before authentication, simultaneous incompatible modes, or failure to enter a blocked state after a hazard.

At the hardware-design level, the CRS behavior can be translated into synthesizable hardware logic. Circuit-level assertions can then check that selected temporal properties hold in the register-transfer level implementation, such as "data-transfer active implies authenticated state" or "hazard assertion forces transition to block." The resulting assurance chain is summarized in Table 4.1.

This evidence does not amount to absolute proof that the CRS, or the SIDC, is secure. Rather, it provides structured, reviewable evidence for specific claims at specific layers of abstraction. That is the appropriate role of formal methods in a security-critical system.

Trustworthiness and Formal Methods

Formal methods are foundational to the SIDC pathfinder, but they are not a panacea. They do not eliminate the need for systems security engineering, expert judgment, testing, red-team evaluation, operational discipline, or accreditation. Formal verification can only prove properties that have been specified within the assumptions of the model and the scope of the tool. If the wrong property is specified, if a key assumption is omitted, or if the implementation departs from the verified model, the formal result may provide false confidence.

The value of formal methods in the model presented in this report is more precise. Their use delivers reproducible evidence of trustworthiness at security-critical points in the architecture. It forces ambiguity out of specifications, exposes hidden assumptions, and makes certain classes of errors machine checkable. When connected to STPA-Sec and systems engineering artifacts, it also supports traceability and auditing. A circuit-level assertion can be traced to a component requirement all the way up to an unacceptable loss at the system level.

TABLE 4.1
Assurance Chain Summary (Highest Abstraction to Lowest)

Assurance Layer	Example Artifact	Primary Claim Supported
STPA-Sec and system constraints	Hazard and constraint traceability	CRS is necessary and addresses a specific class of unacceptable losses.
System model	Realm and interface allocation	CRS components and interfaces are correctly placed within the architecture.
Protocol specification	CRS finite-state protocol	Authorized behavior, sequencing, timing, and fail-secure transitions are defined.
Cryptographic protocol analysis	Authentication and session-establishment model	End-point authentication, secrecy, replay resistance, and key freshness hold under the stated assumptions.
State-machine model checking	Abstract CRS control model	Unsafe control states are unreachable within the modeled behavior.
Hardware implementation	Synthesizable logic	The protocol can be realized without importing general-purpose communication behavior.
Circuit-level assertions	Temporal assertions over hardware logic	Selected safety properties hold at the signal and cycle level.
Test and red-team evaluation	Integrated system evidence	The implemented CRS behaves correctly under realistic operational and adversarial conditions.

Producing trustworthy formal-methods evidence requires several ingredients:

- **A clear security objective at the appropriate level of abstraction.** The property being verified must be tied to a system-level constraint or security guarantee.
- **A sufficiently precise specification.** Informal protocol or component descriptions must be translated into formal models without losing essential behavior.
- **Explicit assumptions.** Environmental conditions, adversary capabilities, timing assumptions, trust boundaries, and input constraints must be stated.
- **Tool-appropriate scope.** Different tools support different claims.
- **Traceability.** Verification artifacts should connect back to requirements, constraints, hazards, losses, and goals.
- **Independent review.** Formal models and proofs require expert review because the most-important errors often appear in the modeling assumptions rather than in the tool output.
- **Integration with ordinary systems engineering life-cycle activities.** Testing, verification, validation, red-team evaluation, configuration management, and operational monitoring remain necessary.

The SIDC security strategy treats formal methods as part of a broader trustworthiness argument. They are strongest when used to support narrow, security-critical claims at well-defined boundaries. They are weakest when treated as a general substitute for systems security engineering.

Concept-to-Circuit Methodology

The top-down path from concept to circuit requires multiple tools and methods, each operating at a different level of abstraction and retaining the intent of its parent levels. Not every tool applies to every feature, and not every feature requires circuit-level realization. The point of the stack is not to impose a rigid sequencing or selection but to maintain security continuity throughout the engineering life cycle.

Within a realized component, every circuit-level operation, protocol rule, and subsystem behavior verifiably enforces security constraints derived from the STPA-Sec results.

In practice, the concept-to-circuit process asks five recurring questions:

1. Why is this control necessary?
   - The control must be justified by a system goal or user need. If the control is not necessary, it should not exist.

2. Which losses and hazards does the control affect?
   - The control must be traceable to the unacceptable losses that it helps prevent and to the hazards that it could help create or mitigate.

3. Which constraints govern the control?
   - The control must have explicitly defined authorized and unauthorized behaviors, including how it handles malformed input, timing deviations, abnormal state transitions, and hazardous conditions.

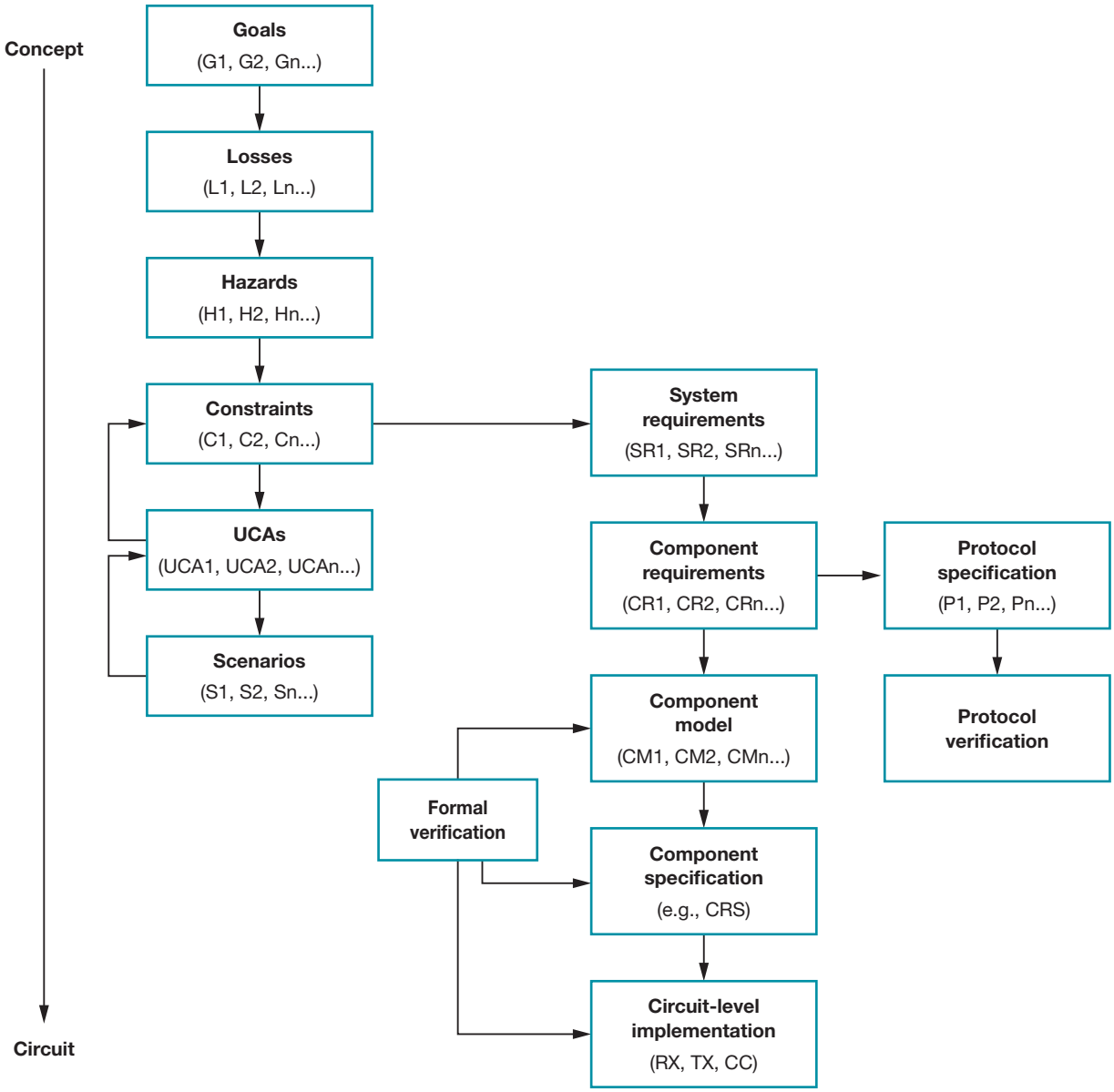
4. What is the minimum mechanism that satisfies those constraints?
   - The control should be implemented with the least functionality, least privilege, and least information-carrying capacity needed to perform its intended role.

5. What evidence supports trust in the control?
   – The control must have verification, validation, testing, review, and operational evidence appropriate to its security significance.

This structure provides a disciplined method to determine whether a proposed control is justified, what it is allowed to do, how it must fail, and how its correctness should be evaluated. Appendix C documents and summarizes the specific tools and methods used to develop the CRS.

At a high level, our concept-to-circuit path follows the structure shown in Figure 4.1.

FIGURE 4.1
Concept-to-Circuit Path



NOTE: CC = channel controller; RX = receiver; TX = transmitter; UCA = unsafe control action.

Next Steps

The CRS implementation work is relatively mature compared with other backlog items. We have developed detailed designs for a protocol-specific CRS and have begun mapping the design to formal specification and verification artifacts. However, this report intentionally provides only limited CRS implementation detail. The important point for the public version of this report is not the specific circuit design but the method leading to its creation. Security-critical cross-realm behavior is derived from constraints and security guarantees, not from the capabilities of general-purpose communication equipment.

The concept-to-circuit process must also be applied to all other security-critical features before the SIDC begins operating. Some of these features may require CRS variants; others may require new system models, protocols, or verification artifacts. At this stage, we treat these as implementation backlog elements rather than as fully specified solutions.

Features not yet engineered to the circuit level of precision include the following:

- **Hazard management.** This includes a facility controller in the POC realm, a realm controller in every realm, and a realm child template for both sensors and actuators.
- **Recovery and reconstitution.** The system must define how the trusted state is preserved, restored, or deliberately destroyed under different hazard conditions. This also includes development of the components in the TRV.
- **Reference data ingestion.** Controlled updates to reference data storage require thorough treatment because they introduce new information into protected environments.
- **Evaluation.** If users require model evaluation or monitoring capabilities, those features must be added without creating unauthorized channels into or out of the TICS.
- **Agentic interaction.** More-interactive or more-agentic use cases require new protocols, hazard analyses, and constraints because they can involve repeated tool use, stateful interaction, or more-complex transaction patterns.

Again, these backlog items must not be completed by importing general-purpose capabilities into the SIDC. Each should be justified by a user need, traced to system goals, analyzed for unique hazards, constrained, specified, and verified in proportion to its security significance.

The concept-to-circuit process does not replace the rest of traditional systems engineering. In parallel with the steps above, the other elements of engineering and assurance must still be performed, including testing, verification, validation, red-team evaluation, configuration management, accreditation, operational procedure development, and maintenance planning. These activities are necessary to ensure that the designed feature performs as expected in the integrated system and under realistic operational conditions.

CHAPTER 5

Conclusions and Recommendations

This chapter summarizes the overarching conclusions drawn from the analysis and recommended next steps for implementation.

Effective Security Requires Purpose-Built, Security-First Facilities

General-purpose data centers present too much attack surface to confidently protect against the most-sophisticated and most-motivated adversaries. We conclude that OC5-level security cannot be achieved by layering controls onto general-purpose systems. Although this approach may be appropriate for lower levels of security, it is inadequate to defend against the most-sophisticated adversaries.

It is our assessment that meeting the security objective with current technology requires (1) purpose-built facilities where functions not essential to the system goals are excluded rather than merely controlled, (2) a security-first design approach in which security is consistently prioritized over other factors, and (3) significant operational restrictions, many of which will incur productivity penalties relative to standard commercial operations.

Such Security Is Appropriate for Only Targeted Use Cases

The SIDC is estimated to cost $37 million to $345 million depending on scale (see Appendix B). Although cost is not insignificant, it is unlikely to be the decisive barrier for stakeholders considering such a facility. Rather, operational restrictions (e.g., fixed software, limited throughput, strict access controls) will constrain potential use cases and present barriers to broader adoption. Organizations seeking to prioritize model availability over security will likely find the SIDC incompatible with their operational requirements. We therefore see the SIDC as most appropriate for three potential use cases:

- **National security applications.** Inference deployments supporting the most sensitive intelligence analysis and strategic decisionmaking will be exceptionally high-value targets for nation-state adversaries seeking to manipulate model outputs, exfiltrate classified data, or influence decisions at the highest levels. The SIDC presented in this report is intended to provide high-assurance security for such cases.
- **Emergency response capability.** If security requirements arise rapidly because of emerging capabilities (e.g., biological weapon design), the ability to rapidly secure frontier models becomes critical. An SIDC with purpose-built security for that use case can meet this need.
- **Pilot and testbed deployment.** A small-scale SIDC can be used to validate security architectures, operational procedures, and integration approaches to derisk larger-scale build-outs and investments.

Recommended Next Steps

Key early steps to deploying an SIDC include the following:

- **Begin procurement.** For the national security use case, the U.S. government should initiate procurement, engage a system integrator, and begin detailed design and engineering. Industry or public-interest organizations could pursue similar efforts in parallel for other use cases.
- **Select a site early.** Major construction phases are largely sequential, making early site selection critical to minimizing the total project duration. For rapid deployment, housing the SIDC within an existing facility, such as a building on an existing cleared campus or even an aircraft hangar, can bypass or significantly shorten key development steps.
- **Prototype key security features and integration now.** FPGA-based channel control with formally verified protocols and optical diode–enforced unidirectional dataflows can be tested immediately by any interested party, independently of steps above. Integrating this relatively novel application into a small-scale, end-to-end inference system with other targeted security mechanisms presents a lower-cost opportunity to reduce technical risk and prevent downstream delays.

Conclusion

The SIDC can be constructed today using a novel combination of proven components and established security techniques: No fundamental research breakthroughs are required. We have high confidence that a well-implemented SIDC, following the specifications and designs presented in this report, can successfully defend against OC5-level threats. We recommend that relevant stakeholders move rapidly to construct an SIDC, either for critical national security applications or to provide emergency response capabilities.

APPENDIX A

STPA-Sec Outputs

As described in Chapter 2, system goals, unacceptable losses, hazards, and system constraints are key outputs from the STPA-Sec analysis. Table A.1 presents the goals, losses, and primary hazards for the SIDC. The complete analysis includes sub-hazards and system constraints, which are excluded from this public report.

TABLE A.1
STPA-Sec Outputs for the SIDC

Identifier	Output
System goals	
G1	Maintain national technical and military advantage as outlined in the National Security Strategy.
G1.1	Maintain a six-month lead on adversaries and competitors.
G1.1.1	Maintain or increase blue velocity.
G1.1.2	Deny red acceleration from U.S. AI development.
G2	Prevent global catastrophic harm from U.S. frontier models.
G3	Support efficient, effective, frontier AI model operations.
G4	Develop a safe, usable system for national security application.
G5	Enforce containment of the AI model within the SIDC.
Unacceptable losses	
L1	Model weights disclosed outside the SIDC
L2	Tokens disclosed outside the SIDC
L3	Algorithmic insights disclosed outside the SIDC
L4	An incident related to an SIDC mission directly causes another mission to fail
L5	Loss of or damage to SIDC
Hazards	
H1	Unauthorized or “malformed” data enter or exist inside the facility.
H2	Unauthorized or “malformed” data leave the facility.
H3	Unauthorized person enters or exists inside the facility.
H4	Unauthorized person leaves the facility.
H5	Malicious, unwitting, or opportunistic insider compromises the facility.
H6	Unauthorized entity reads SIDC data.
H7	Non-POC element of the SIDC remains in operation while a hazard is unresolved.
H8	Unauthorized modification of SIDC occurs.

Table A.1—Continued

Identifier	Output
H9	Unauthorized matter transits a realm boundary.
H10	Unauthorized energy transits a realm boundary.
H11	Unauthorized information transits a realm boundary.

APPENDIX B

Time and Cost to Implement an SIDC

This appendix provides estimates for the cost and time to implement an SIDC.

System Size

We consider two representative cases: (1) the minimum scale needed for a functioning proof of concept and (2) a deployment-scale facility capable of hosting multiple models and providing inference at scale.

Proof of Concept (250 kW)

The minimal configuration is a 3,000 ft^2 facility with 250 kW of compute (about 150 inference accelerators). This scale represents a minimal configuration for a fully functional facility serving low-throughput inference with a frontier model. This example is most relevant for those seeking to implement a proof of concept, either to generate learning for security best practices that can be applied elsewhere or to derisk larger-scale build-outs in the future.

Enterprise Scale (3 MW)

For deployment scale, we assume 3 MW of compute (about 400 inference accelerators) within a 30,000 ft^2 facility. This configuration could host multiple frontier-scale models, provide operational redundancy against hardware failures, and support an inference query throughput to support many simultaneous users (not commercial scale but ample for the use cases envisioned here).

Cost

For new construction, including land acquisition, the proof-of-concept SIDC is projected to cost $37 million to $50 million, while the enterprise scale facility is projected to cost $277 million to $345 million. The cost breakdown is shown in Table B.1.

The estimates above represent up-front capital expenditures required to deploy the system. Operational costs for staffing, utilities, maintenance, and other ongoing costs are not included in these estimates.

An important cost element is the development of the FPGA-based channel control system. Formally verified FPGA logic is a critical element of the overall security strategy. While FPGAs are not expensive, developing custom, formally verified protocols will require a specialized and limited talent pool. A functional prototype is estimated to cost less than $10 million; if successful, it should scale to full SIDC implementation at relatively modest additional cost. We estimate the cost for an implemented solution (including prototyping) to be $10 million to $15 million for the proof-of-concept SIDC and $15 million to $22.5 million for the deployment-scale SIDC.

TABLE B.1
Cost Estimates for Proof-of-Concept and Deployment-Scale SIDCs

Component	Proof of Concept (million $)	Deployment Scale (million $)	Notes
Land	0.4	4.0	Assuming $266,000 per acre (2025 data);[a] 1.5 and 15 acres for a proof-of-concept and deployment-scale SIDC, respectively
Facility construction	5.6	48.0	Assuming $1,600 per ft^2; 3,500 ft^2 and 30,000 ft^2[b]
Compute hardware	12.0–16.0	150–200	AI-optimized server racks plus extra for destructive hardware testing and on-site stockpile to account for failure rates; 150 graphics processing units (GPUs) and 375 GPUs[c]
Networking and balance of system	1.7	21.6	Switches, fiber-optic diodes, and rack management security controls, plus replacement hardware stockpile (one-third of components)
Software inspection	1.4–3.2	2.1–4.8	Review of existing code for vulnerabilities; selective memory-safe rewrites not included[d]
Channel controller development	10.0–15.0	15.0–22.5	Formally verified FPGA channel controllers to establish strict controls on the multi-bit channels and optical diodes to enforce unidirectional dataflows
Initial security vetting of personnel	2.3	6.9	Background investigations and adjudications for security clearances; ~100 cleared staff for a proof-of-concept SIDC and 300 for a deployment-scale SIDC[e]
General contractor overhead and profit	4.0–5.3	29.7–36.9	12% of above costs
Total estimated cost	≈$37–$50	≈$277–$345	Order-of-magnitude estimate

[a] Brian Ungles, John McWilliams, Ethan Tribble, Adrian Conforti, and Jason D'Orlando, *Data Center Development Cost Guide 2025*, Cushman & Wakefield, 2025.

[b] For reference, the high end of construction cost for a Sensitive Compartmented Information Facility (SCIF) is $1,000 to 1,200 per ft^2. See Alex Leutwyler, "Decoding SCIF Costs: What Drives the Price of Sensitive Compartmented Information Facilities," Emblem Builders blog, last updated October 18, 2025; Charles Simmins, "What Is a SCIF," ClearanceJobs blog, April 1, 2024.

[c] Costs are based on vendor quotes for GPUs as of summer 2025. Public analysis finds a comparable cost per rack. See Dylan Patel and Daniel Nishball, "H100 vs GB200 NVL72 Training Benchmarks—Power, TCO, and Reliability Analysis, Software Improvement over Time," SemiAnalysis, August 20, 2025. Although we used GPUs for the cost estimate, we are agnostic to the underlying compute hardware (GPUs, tensor processing units, application-specific integrated circuits).

[d] Costs are estimated based on the average from several Forrester case studies. See Forrester Consulting, *The Total Economic Impact of Revenera: Cost Savings and Business Benefits Enabled by Revenera*, May 2025; Forrester Consulting, *The Total Economic Impact of the Veracode Application Risk Management Platform: Cost Savings and Business Benefits Enabled by Veracode*, August 2024b; and Forrester Consulting, *The Total Economic Impact of Checkmarx One: Cost Savings and Business Benefits Enabled by the Checkmarx One Platform*, March 2024a.

[e] Top Secret background investigations cost $6,739 per person (plus $3,866 for periodic reinvestigations) according to fiscal year 2026 billing rates. See Federal Investigations Notice No. 24-01, "Products and Services Reimbursable Billing Rates for FY 2025 and FY 2026," Defense Counterintelligence and Security Agency, August 30, 2024.

Software inspections are another notable cost element. We recommend scrutinizing general-purpose and inference-specific software before deploying in the SIDC, which is expected to cost a few million dollars.[1] Vulnerabilities discovered in that process may warrant memory-safe rewrites or other remediations, the cost of which are not included here. We emphasize, however, that these measures are best-effort and their completion is not mandatory for the security strategy. In contrast, the custom FPGA implementations governing information flow across realm boundaries require formal verification and absolute correctness.

[1] This estimate was derived from several 2024 to 2025 Forrester consulting case studies that included software developer and security efficiency improvements costs per year over a three-year period. Our cited cost is the midpoint average of similar software inspection efforts. See Forrester Consulting, 2025; Forrester Consulting, 2024a; and Forrester Consulting, 2024b.

Implementation Timeline

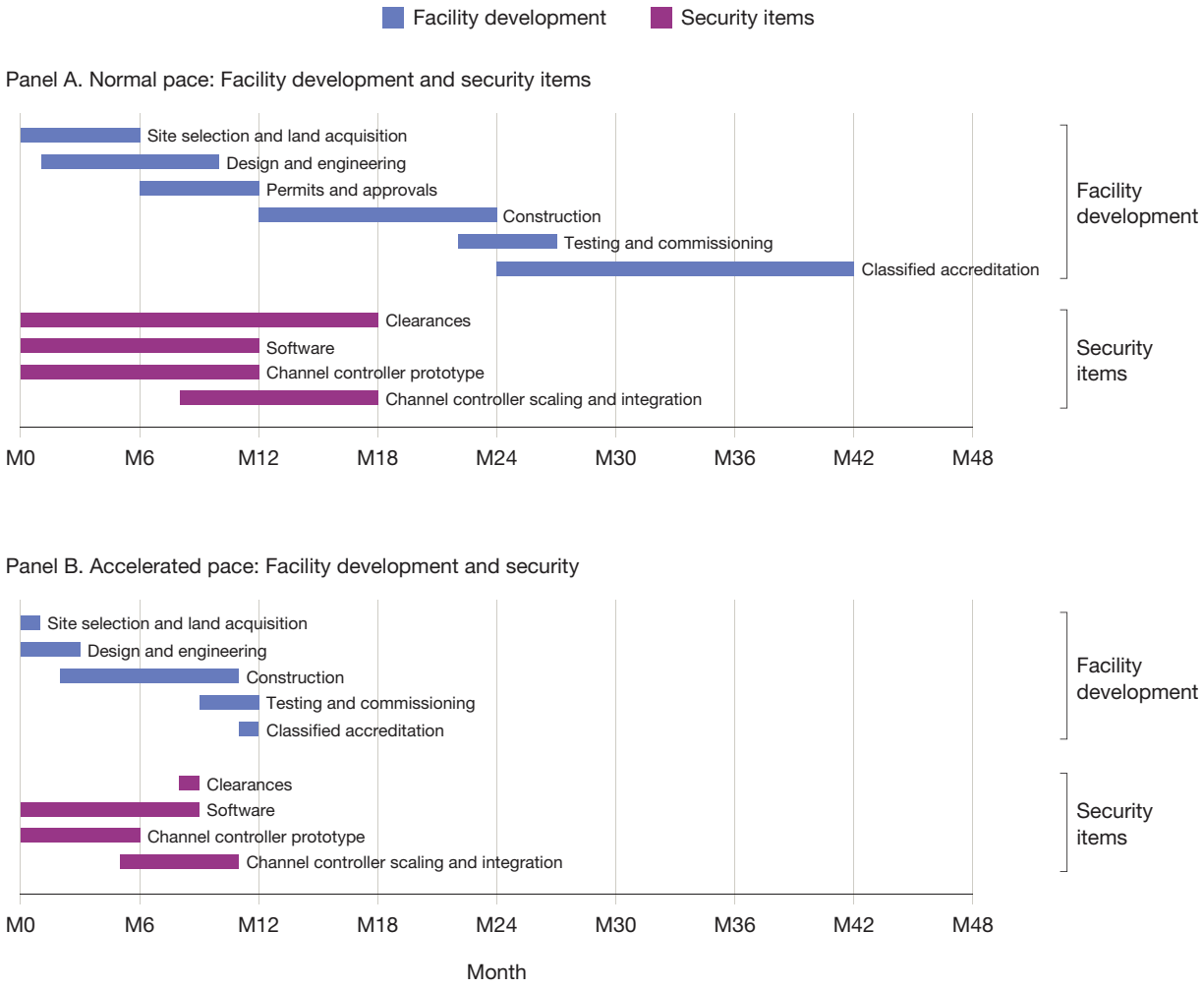
Deployment Timeline Under Normal Conditions

Under normal conditions, we expect two to two and a half years to deploy the SIDC, with an additional year to complete accreditation for handling classified materials. This timeline assumes new construction, including land acquisition, and is based on typical procurement, permitting, construction, and accreditation timelines. Note that this timeline is a reasonable baseline estimate for either the proof-of-concept or the deployment-scale SIDC. Despite there being a roughly ten-times size difference, we conclude that the time to complete key steps does not shift dramatically (e.g., permitting, constructing the facility itself, classified accreditation).

Figure B.1, Panel A, shows the sequence of key development steps to implement the facility (blue bars): site selection and land acquisition, design and engineering, permits and approvals, construction, testing and commissioning, and classified accreditation.

FIGURE B.1

Timeline to Implement the SIDC, Normal Pace and Accelerated Pace



SOURCES: Facility construction steps are based on data from Arlene Hidalgo, "Data Center Construction: Costs, Timeline, and Delivery Steps," Mastt, last updated February 20, 2026; and Bray Dohrwardt, "How Long Does It Take to Develop a Data Center? A Step-by-Step Timeline," January 27, 2025. Timelines for software review and channel controller development are the authors' estimates.

Permitting is a significant variable, with typical timelines ranging from six to 18 months depending on the project and jurisdiction. We assume 18 months for accreditation, longer than the construction phase itself.[2] While some accreditation activities can proceed in parallel with earlier steps, it remains a key time driver under baseline conditions.

On the security side (violet bars), we highlight four time drivers. First, we assume that all staff working in the SIDC must hold a Top Secret security clearance, with typical background investigations and adjudications taking 8 to 18 months.[3] If staff are identified early, vetting can proceed in parallel with other development phases. Second, software screening is assumed to take one year. This timeline can be accelerated by dedicating additional resources; because it is a best-effort exercise rather than a critical security control, it should never be allowed to become a bottleneck.

Finally, we divide channel controller development into two phases: prototype development (one year), followed by scaling and integration into the SIDC (ten months, partially overlapping with prototype development).

In summary, under normal conditions, standard facility development steps and accreditation for classified data are the key timeline drivers. Elements of the security strategy are not barriers, provided they are pursued in parallel with facility construction.

Compressed Timelines If Pursued as a National Priority

Under emergency or national priority conditions, we estimate that the SIDC could be constructed in approximately 14 months (Figure B.1, Panel B). While limited data are available to inform these estimates, we assume a government-owned facility constructed within an existing aircraft hangar or warehouse on an existing government campus. In this scenario, site selection, land acquisition, permitting, and approval processes could be virtually eliminated.

We assume that construction times are 20 to 30 percent faster than lower-bound benchmark data, reflecting accelerated efforts under national priority conditions and time savings from using an existing foundation and utilities. Similarly, integration and testing are compressed and, where possible, conducted in parallel with final construction stages. Because the U.S. government controls the accreditation process, we assume that it is expedited to one month rather than the normal 12 to 18 months.

On the security side, we assume that clearances are expedited or staff are drawn from existing cleared personnel, eliminating this as a potential bottleneck. As noted above, software screening can be accelerated as needed and should never be allowed to constrain the schedule.

Developing the channel controllers is a potential schedule constraint. Creating and formally verifying the FPGA-based channel control system requires highly specialized, low-density expertise and cannot be substantially accelerated through additional funding alone. We believe that the accelerated timelines shown (six months each for prototype and scaling and integration, with partial overlap) are plausible, although these estimates reflect our best judgment rather than data-driven figures. Developing the channel controllers

[2] Typical accreditation timelines for facilities handling classified information are based on Intelligence Community Directive 705, *Sensitive Compartmented Information Facilities*, Office of the Director of National Intelligence, May 26, 2010; and Department of Defense Manual 5205.07, *Special Access Program Security Manual*, U.S. Department of Defense, January 17, 2025. New construction generally requires 12 to 18 months for full accreditation. See Brian Metras and Tom Taylor, "Navigating the New Era of SCIF Construction," Area Development, Q1 2025; and Briana Becker, Gordon Griffin, and Susannah L. Gilmore, "Leasing SCIF Space: How the New ICD-705 Affects Leases," Holland & Knight, October 30, 2025.

[3] Lindy Kyzer, "How Long Does It Take to Get a Security Clearance? Times Go Up in 2024," ClearanceJobs, November 13, 2024; GCheck Editorial Team, "How Long Does a DoD Contractor Background Check Take? 2025 Timeline," GCheck blog, undated.

should begin early and can proceed in parallel with the construction phases shown in Figure B.1. The organization developing the prototype need not be the same entity building the facility.

Note that deployment under compressed timelines adds risk. The same pressures that aim to deliver the facility more quickly can encourage those involved to cut corners in ways that could harm security objectives. For this reason and especially in this scenario, effective verification and validation activities are essential to ensure that the fielded system delivers the required security outcomes.

APPENDIX C

Concept-to-Circuit Tools and Methods

This appendix documents and summarizes the specific tools and methods used to formally specify the CRS.

NIST Special Publication 800-160

NIST Special Publication (SP) 800-160, Vol. 1, Rev. 1, *Engineering Trustworthy Secure Systems*, provides the systems security engineering foundation for this work.[1] It emphasizes that trustworthy secure systems require attention to three distinct contexts: the problem context, the solution context, and the trustworthiness context. This framing is central to the SIDC effort. The problem context asks which losses must be prevented and what hazards could lead to them. The solution context asks which system architecture, interfaces, functions, and operational procedures can prevent or control those hazards. The trustworthiness context asks why stakeholders should believe that the solution will perform as required in a hostile environment.

NIST SP 800-160 also provides the SecDOP, which informs the SIDC's least-function strategy. Rather than relying primarily on detection or administrative controls, the design seeks first to eliminate unnecessary function and prevent hazardous system states by construction. Where prevention is infeasible, the design then specifies monitoring, detection, fail-secure behavior, and recovery mechanisms. In the concept-to-circuit template, NIST SP 800-160 provides the overarching engineering philosophy and helps connect strategic security objectives to concrete design decisions.

STPA-Sec

STPA-Sec is the structured method we use to connect high-level security strategy to concrete engineering decisions, primarily delivering content for the problem context mentioned above.[2] It begins with the system's goals and unacceptable losses, then identifies the hazardous system states that could lead to those losses, the constraints needed to prevent them, the hazardous control actions that may violate those constraints, and the scenarios in which those hazardous actions arise.

In this report, STPA-Sec provides the upper portion of the concept-to-circuit traceability chain. It translates strategic objectives, such as protecting model weights and preventing adversarial manipulation, into system-level hazards and security constraints. Those outputs then become the basis for the system model, concept of secure function, protocol specifications, subsystem designs, and ultimately circuit-level mechanisms. In this way, STPA-Sec helps ensure that low-level design choices remain explicitly traceable to the strategic losses that the SIDC is intended to prevent.

[1] Ross, Winstead, and McEvilley, 2022.

[2] See Nancy G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press, 2012.

System-Theoretic Concept Development

System-theoretic concept development extends the STPA-Sec logic into the formulation of the system concept itself.[3] Rather than treating the architecture as a given, this step asks what kind of system could satisfy the constraints derived from the losses and hazards. It supports the development of the *concept of secure function*: a concise statement of how the system is intended to operate securely, what it must prevent, what it must detect, and how it should respond when prevention fails.[4]

For the SIDC, this concept development step is where abstract constraints become architectural commitments. For example, if unauthorized information transfer across a realm boundary is a hazard, the system concept must specify how realm boundaries are defined, what transactions are authorized, what transactions are impossible by construction, and what hazard response occurs when anomalous flows are detected. The concept of secure function therefore provides the bridge between STPA-Sec outputs and the system architecture.

System Modeling: Systems Modeling Language

For modelable elements of the system, model-based systems engineering (MBSE) can be used to represent the system of interest and its boundaries, interfaces, functions, requirements, and allocation relationships. These tools are most valuable where they help preserve traceability across abstraction layers: from goals and losses to hazards, constraints, requirements, components, interfaces, and verification activities. For the SIDC effort, we chose the Systems Modeling Language (SysML) to model relevant systems.

In the SIDC context, MBSE is useful for accurately describing realm structure, authorized interactions, interface definitions, requirements allocation, and dependencies among features. It can also support design review by making explicit which components or operational processes satisfy which constraints. However, system modeling should not be confused with evidence of security. A model can organize requirements and relationships, but it does not prove that the system is secure. Its value depends on the quality of the underlying analysis and the rigor of the verification activities connected to the model.

Protocol Verification: Tamarin and CryptoVerif

For security-critical components realized in information exchange protocols, Tamarin can be used to analyze cryptographic protocol properties. Tamarin is designed to reason about symbolic protocol behavior and can support claims about authentication, secrecy, key exchange, and adversarial interaction under formalized assumptions. It is particularly useful when a security boundary depends on cryptographic guarantees rather than only on physical separation or simple state-machine behavior.

In the concept-to-circuit path, Tamarin sits below the English-language protocol specification and above the implementation-specific circuit or software artifacts. Its role is to help evaluate whether the protocol, as specified, satisfies the cryptographic security properties required by the system constraints. As with all formal tools, Tamarin's results are conditional on the model. It can increase confidence that a modeled protocol satisfies specified properties, but it cannot prove that the implementation is correct, that all relevant assumptions were captured, or that the system architecture around the protocol is sound.

---

[3] Alexander P. Hillman, *A Systems-Theoretic Approach to Design of Early Concepts for Novel, Complex Systems in Aerospace*, dissertation, Massachusetts Institute of Technology, February 2025.

[4] Ross, Winstead, and McEvilley, 2022.

CryptoVerif is an open-source formal prover written in OCaml, which, unlike Tamarin, proves security properties for protocols in the computational model. Inputs to CryptoVerif require a description of the security of the underlying primitives to a protocol, as well as user-provided heuristics for convergence of proofs, which may require academic expertise. Assuming a probabilistic polynomial-time adversary, CryptoVerif takes, as input, a protocol specification in terms of cryptographic primitives, channels, and oracles and produces a sequence of mathematical models called *games*, which formally prove bounds on the probability of the adversary breaking given security properties. The granular output of CryptoVerif fills an analytic gap for Tamarin, which assumes ideal behavior of cryptographic primitives; use of CryptoVerif methods guarantees that both models of protocol security are addressed by formally proved security results.

Formal Specification: Temporal Logic of Actions+

Temporal Logic of Actions+ (TLA+) is the formal method we used to model protocol control logic as a distributed system and check high-level safety assumptions across the full abstract state space rather than only in hand-selected scenarios. Applying it required translating the protocol into an explicit finite-state specification and deciding which behaviors could be treated as atomic. That translation was itself a major source of value: Forcing each transition and signal into precise rules exposed places where the written protocol needed sharper operational assumptions.

The Temporal Logic Checker (TLC) then lets us reason over all reachable modeled behaviors, producing a TLA+ model and machine-checked safety results about the protocol's control behavior. These claims are about abstract protocol safety and state-machine behavior, not cryptographic soundness, implementation correctness, or hardware-faithful timing. Those claims must be addressed using separate tools. In the concept-to-circuit path, TLA+ helps bridge the gap between a prose protocol and a precise behavioral model whose safety properties can be exhaustively checked within the bounds of the model.

Hardware Translation

For the CRS, we elected to use FPGAs as the implementation substrate. Others might choose to close using application-specific integrated circuits or custom silicon. Hardware description language is implementable on an array of possible hardware, each with its own security considerations and supply chain risks. Engineers are free to make trades of this kind as long as the choice fulfills SIDC security constraints. Note that the preceding tooling is applicable to human-operated systems and protocols (as in the hazard management system) and to hardware design and implementation. Subsequent tools are more specific to the exemplar circuit-level subsystem.

Synthesizable Hardware Specification: Bluespec SystemVerilog

Bluespec SystemVerilog (BSV) bridges the gap between a protocol specification and synthesizable hardware by allowing designers to express circuit behavior as guarded atomic rules rather than as clock-by-clock register transfers. This separates protocol correctness from implementation scheduling: Each rule defines a single atomic state transition with an explicit guard condition, while the compiler handles scheduling and synthesis to standard Verilog.

That rule-based structure has a direct correspondence to the abstract protocol models produced upstream, preserving traceability from specification to circuit. The register-transfer–level (RTL) code that the compiler

produces then serves as the input to circuit-level formal verification. The claims BSV can support are structural rather than fully formal. Certain violations are prevented by construction because a rule cannot fire unless its guard is satisfied, but BSV does not prove that guard conditions are correct or complete. That obligation falls to the verification layer.

Inline Temporal Verification: SystemVerilog Assertions

SystemVerilog Assertions (SVA) operate at the layer where protocol requirements become verifiable claims against the actual circuit, taking over where BSV's structural guarantees end. SVA works directly on RTL code and requires engineers to translate protocol requirements into temporal sequences. That translation is itself a specification audit: Expressing requirements as falsifiable temporal predicates routinely surfaces assumptions that the original specification left unstated.

SVA produces a layered artifact library: *Assert property statements* enforce safety invariants, *cover property statements* confirm reachability, and *assume property statements* constrain the formal tool's input space. Its scope is the signal and cycle level. SVA cannot verify what was never asserted, so coverage is not completeness, and system-level or probabilistic claims remain outside its reach. In the concept-to-circuit path, SVA is the final formal link between requirements and circuit behavior for selected hardware mechanisms.

Abbreviations

AI	artificial intelligence
BSV	Bluespec SystemVerilog
CRS	cross-realm solution
FPGA	field-programmable gate array
GPU	graphics processing unit
IR	Interface Room
ITM	Information Technology Management
MBSE	model-based systems engineering
NIST	National Institute of Standards and Technology
OC5	Operational Capacity 5
PER	Payload Exchange Room
POC	Protection Operations Center
RTL	register-transfer level
SecDOP	Security Design Order of Precedence
SIDC	secure inference data center
SL5	Security Level 5
SP	Special Publication
STIC	Secure Transfer Inter-facility Conduit
STPA-Sec	System-Theoretic Process Analysis for Security
SVA	SystemVerilog Assertions
TICS	Trusted Inference Compute Sanctum
TLA+	Temporal Logic of Actions+
TRV	Trustworthy Recovery Vault

References

Becker, Briana, Gordon Griffin, and Susannah L. Gilmore, "Leasing SCIF Space: How the New ICD-705 Affects Leases," Holland & Knight, October 30, 2025.

Brass-Gershovich, Asher, Rachel Steratore, Wesley Hurd, Henry Alexander Bradley, Anjay Friedman, and Sella Nevo, *Securing AI Algorithmic Insights*, RAND Corporation, RR-A4685-1, 2026. As of July 20, 2026: https://www.rand.org/pubs/research_reports/RRA4685-1.html

Department of Defense Manual 5205.07, *Special Access Program Security Manual*, U.S. Department of Defense, January 17, 2025.

Dohrwardt, Bray, "How Long Does It Take to Develop a Data Center? A Step-by-Step Timeline," January 27, 2025.

Federal Investigations Notice No. 24-01, "Products and Services Reimbursable Billing Rates for FY 2025 and FY 2026," Defense Counterintelligence and Security Agency, August 30, 2024.

Forrester Consulting, *The Total Economic Impact of Checkmarx One: Cost Savings and Business Benefits Enabled by the Checkmarx One Platform*, March 2024a.

Forrester Consulting, *The Total Economic Impact of the Veracode Application Risk Management Platform: Cost Savings and Business Benefits Enabled by Veracode*, August 2024b.

Forrester Consulting, *The Total Economic Impact of Revenera: Cost Savings and Business Benefits Enabled by Revenera*, May 2025.

GCheck Editorial Team, "How Long Does a DoD Contractor Background Check Take? 2025 Timeline," GCheck blog, undated.

Green, Christine, "Estimating as an Art—What It Takes to Make Good Art," paper presented at PMI Global Congress 2006, Madrid, Spain, October 2006.

Hidalgo, Arlene, "Data Center Construction: Costs, Timeline, and Delivery Steps," Mastt, last updated February 20, 2026.

Hillman, Alexander P., *A Systems-Theoretic Approach to Design of Early Concepts for Novel, Complex Systems in Aerospace*, dissertation, Massachusetts Institute of Technology, February 2025.

Intelligence Community Directive 705, *Sensitive Compartmented Information Facilities*, Office of the Director of National Intelligence, May 26, 2010.

Kyzer, Lindy, "How Long Does It Take to Get a Security Clearance? Times Go Up in 2024," ClearanceJobs, November 13, 2024.

Leutwyler, Alex, "Decoding SCIF Costs: What Drives the Price of Sensitive Compartmented Information Facilities," Emblem Builders blog, last updated October 18, 2025.

Leveson, Nancy G., *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press, 2012.

Metras, Brian, and Tom Taylor, "Navigating the New Era of SCIF Construction," Area Development, Q1 2025.

Nevo, Sella, Dan Lahav, Ajay Karpur, Yogev Bar-On, Henry Alexander Bradley, and Jeff Alstott, *Securing AI Model Weights: Preventing Theft and Misuse of Frontier Models*, RAND Corporation, RR-A2849-1, 2024. As of July 20, 2026:
https://www.rand.org/pubs/research_reports/RRA2849-1.html

Patel, Dylan, and Daniel Nishball, "H100 vs GB200 NVL72 Training Benchmarks—Power, TCO, and Reliability Analysis, Software Improvement over Time," SemiAnalysis, August 20, 2025.

Ross, Ron, Mark Winstead, and Michael McEvilley, *Engineering Trustworthy Secure Systems*, National Institute of Standards and Technology, NIST SP 800-160v1r1, November 2022.

SAE International, "Standard J3307_202503: System Theoretic Process Analysis (STPA) Standard for All Industries," March 24, 2025.

Simmins, Charles, "What Is a SCIF," ClearanceJobs blog, April 1, 2024.

Ungles, Brian, John McWilliams, Ethan Tribble, Adrian Conforti, and Jason D'Orlando, *Data Center Development Cost Guide 2025*, Cushman & Wakefield, 2025.

Vila, Didier, "Deconstructing the Data Center: A Look at the Cost Structure Igniting the AI Boom!" Alpha Matica, undated.

Warfighting Acquisition University, "Cost Estimation Methods," webpage, undated. As of February 19, 2026: https://www.waru.edu/acquipedia-article/cost-estimation-methods

Young, William, and Nancy G. Leveson, "An Integrated Approach to Safety and Security Based on Systems Theory," *Communications of the ACM*, Vol. 57, No. 2, February 2014.

About the Authors

Steven F. Comer is a senior AI security resident at RAND. His research improves security outcomes for high-consequence software systems with tailored systems security engineering. Comer holds an M.S. in applied computer science.

Hunter Pavela is a senior AI security resident at RAND, where he focuses on AI and cybersecurity. Pavela holds an M.S. in cybersecurity.

Varun Gandhi is an AI security researcher at RAND. His work lies at the intersection of systems security, cryptography, formal methods, and AI control, focusing on hardware-enforced isolation and cryptographic verification mechanisms for secure and verifiable computation in advanced AI systems. Gandhi holds a Ph.D. in computer science.

Kyle Siler-Evans is a senior engineer at RAND and a policy lead in the Center on AI, Security, and Technology. His prior work focused on disaster management, infrastructure planning and resilience, cost analysis, and decisionmaking under uncertainty. Siler-Evans holds a Ph.D. in engineering and public policy.

Erich Devendorf is a senior AI security resident at RAND. He conducts technical and policy research on AI security, control, alignment, and evaluation. Devendorf holds a Ph.D. in mechanical engineering.

Ben Kelley was an AI security resident at RAND. He conducted technical and policy research on AI and cybersecurity. Kelley holds an M.S. in cybersecurity and public policy.

James Gimbi is a senior AI security resident at RAND. He has a background in cyber crisis response, and his research focuses on secure AI practice and policy. Gimbi holds a B.S. in information security.

Jair Aguirre is a senior information scientist at RAND. He conducts technical and policy research on offensive cyber and spectrum operations, AI, and other emerging technology topics. Aguirre holds an M.S. in applied economics.

Gabriel Kulp is adjunct staff at RAND, continuing work he pursued through the Technology and Security Policy Fellowship. His research improves the security of frontier AI systems as they become more important to national security and geopolitical stability. Kulp holds an M.S. in computer science.

Mark Stalczynski is a senior quantitative analyst at RAND. He undertakes research in public finance, budget and cost analysis, and energy and the environment. Stalczynski has an M.S. in public policy and management.

Matthew J. Malone is a senior policy analyst at RAND, where he focuses on AI security and cybersecurity. Malone holds an M.A. in security policy studies.