One Mission View

Connecting Security Requirements, Design Principles, and Assurance Evidence

Dr. Ron Ross
CEO, RONROSSECURE, LLC
ISTS FELLOW, DARTMOUTH COLLEGE

Abstract

Security requirements and design principles provide complementary foundations for engineering trustworthy secure systems. This article proposes a unified mission view that connects system security requirements, principle-informed design decisions, relevant security controls, and assurance evidence. Two engineering sets—controls associated with security requirements (R) and controls associated with the application of security design principles (P)—are combined and compared with the tailored security control baseline (B). The proposed model distinguishes a recorded association from implementation, verified effectiveness, and a supported assurance conclusion. A water treatment facility example illustrates how traceability, dependency analysis, and explicit assumptions support the assurance case. Maintaining these relationships throughout the system life cycle can expose gaps, identify evidence affected by change, and give authorizing officials a clearer basis for evaluating mission consequences and residual risk. The model is an author-proposed organization of systems engineering information, not a set structure prescribed by NIST.

1. The Mission View

A security control identifier alone tells an authorizing official very little about whether a system can sustain its mission under adversity. An informed authorization decision requires a traceable connection from the protection need to the system requirement, the architecture that realizes it, and the evidence supporting confidence in the result.

Security requirements and security design principles together help establish that connection. Organizing them into a single *mission view* gives engineers, assessors, and authorizing officials a shared understanding of how the system protects what matters to the mission.

I propose organizing that view around two engineering perspectives on controls: their association with system security requirements (set R) and their association with the application of security design principles (set P). Comparing the combined view (**P** ∪ **R)** with the tailored control baseline (**B**) while preserving the rationale and evidence behind every association bridges the gap between compliance artifacts and systems engineering.

This is a proposed way to organize engineering and authorization information. NIST does not prescribe the **P**, **R**, and **B** sets described below. The approach draws on the systems security engineering foundation in SP 800-160, Volume 1, Revision 1 and the control selection, assessment, and authorization practices of the Risk Management Framework (RMF). [1] [2]

2. Requirements Establish the Obligations

System security requirements are the system requirements that have security relevance. They translate validated stakeholder requirements into technical obligations for the system. Those obligations can concern protection capabilities, behavior, performance, architecture, assurance activities, and evidence. Security also places constraints and conditions on other system requirements. [1]

© 2026 RONROSSECURE, LLC | CC BY 4.0.

Requirements therefore extend beyond a list of security functions. A functional requirement might constrain which operations an entity may perform. A structural requirement might specify separation between system elements so that a failure or compromise cannot produce a prohibited effect across a defined boundary.

Requirements must be precise enough to guide realization and support verification. “Limit access to authorized users” leaves important questions unanswered. Which resources and operations? Under which conditions? Who establishes the authorization? What happens when that authority is unavailable or compromised?

Those questions matter because a correctly authenticated and authorized entity can still request an operation with unacceptable consequences.

3. Principles Shape the Engineering Decisions

Security design principles guide decisions about how the system will satisfy its requirements and achieve trustworthy behavior. *Domain Separation*, *Mediated Access*, *Least Privilege*, and *Trustworthy System Control* provide different, complementary engineering perspectives on the same system.

Architecture and design are central places to apply these principles. Their influence also extends to requirements analysis, implementation, integration, verification, operation, maintenance, and recovery. A boundary that is effective during normal operation can disappear during maintenance or restoration. Applying a principle requires examining those transitions as well as the intended operating state.

A principle is general engineering guidance. It becomes concrete through decisions about a particular system. Applying *Domain Separation*, for example, can reveal a need for additional structural requirements or constraints on administrative access. Applying *Least Privilege* can expose authority that is broader than the assigned function requires.

The relationship is iterative. Requirements guide the architecture. Architecture analysis reveals derived requirements. Verification and validation findings can require changes to both.

4. Use One Connected Assurance Argument

Verification results and design assurance evidence belong in the same body of evidence. A test showing that an implementation satisfies a requirement can also support an assurance claim about the design. Analysis may explain why the test coverage is adequate, why alternate paths have been considered, and which assumptions limit the result.

Requirement satisfaction alone does not establish that the requirements adequately capture stakeholder protection needs. An assurance argument must also address their adequacy and coverage, the correctness of the design and implementation, and the conditions under which the conclusions hold. SP 800-160 explicitly cautions against expressing assurance claims solely as restatements of requirements. [1]

Principle-specific claims and subclaims can help structure that argument. They are propositions to substantiate for a defined system, with stated operating conditions, adversities, dependencies, and scope. A claim associated with a principle is not an inherent guarantee supplied by the principle. Listing candidate evidence does not establish that the evidence exists or supports the claim.

For an authorizing official, the useful question is how the available evidence supports the mission-relevant conclusion and what remains *uncertain*.

© 2026 RONROSSECURE, LLC | CC BY 4.0.

5. Two Engineering Perspectives on Controls

NIST SP 800-53 controls provide a common vocabulary for expressing and assessing security and privacy measures. [4] Engineers can associate relevant controls with requirements and with the design decisions made when applying principles.

In the proposed model, ***P*** is the set of controls associated with the documented application of security design principles. Each association records the design decision, its rationale, the relevant assurance proposition, and the scope of the relationship.

R is the set of controls associated with documented system security requirements. Each association identifies the requirement, the relevant control provisions, and how the implemented solution addresses the obligation.

Neither set membership establishes implementation or effectiveness. The model distinguishes four states that must not be collapsed:

- **Proposed association:** a candidate linkage recorded for review
- **Accepted association:** a reviewed and retained linkage with documented rationale and scope
- **Implemented measure:** the actual technical or procedural realization in the system
- **Evidence-supported result:** results from verification or other evaluation, with the applicable scope, conditions, assumptions, and limitations recorded

The record must preserve those differences. Verification results support an assurance conclusion alongside the argument, assumptions, and other evidence.

The combined engineering view is **P** ∪ **R**. A control appearing in both sets **P** ∩ **R** has two recorded sources of engineering rationale. That is richer traceability, but confidence still depends on the quality and applicability of the supporting evidence. Ten weak mappings do not outweigh one well-supported argument.

6. Read the Differences as Engineering Questions

A control in both P and R connects a requirement with an application of one or more principles. Review whether the records concern the same system elements, operating modes, control provisions, and implementation. A shared control identifier can conceal very different scopes.

A control in P alone prompts a review of the requirements and traceability. The design decision may expose a missing requirement, implement an existing requirement whose association was not recorded, or represent a justified design choice requiring further documentation. Missing verification evidence is a separate question.

A control in R alone prompts a review of the recorded design rationale. Some requirements concern organizational or operational measures for which a direct association with a particular design principle may add little value. Other cases may expose an unrecorded or inadequate application of a principle. The purpose is to identify meaningful gaps, not manufacture a principle mapping for every control.

Control relationships are often partial. A requirement may address only a portion of a control, and a design decision may contribute to several controls without fully implementing any of them. Record the relevant provisions and organization-defined parameters so that a broad label does not imply complete coverage. For example, a requirement may address access enforcement for one application's requests without

© 2026 RONROSSECURE, LLC | CC BY 4.0.

addressing access through other system interfaces. The control association should state that limited scope explicitly.

7. Compare the Engineering View with the Control Baseline

B is the tailored baseline applicable to the system—a key output of the NIST Risk Management Framework (RMF) Select step. It reflects the applicable selection and tailoring decisions and provides the comparison point for **P** ∪ **R**. The engineering analysis can inform further tailoring and supplementation. [2] [3]

Controls appearing in both views ((**P** ∪ **R**) ∩ **B**) have a recorded baseline basis and an engineering association. That agreement helps explain their inclusion. It does not prove adequate implementation or justify closing an assessment finding.

Controls in **B** alone (**B** \ (**P** ∪ **R**)) need review across engineering, system ownership, and assessment responsibilities. They may be inherited common controls, organizational measures, operational obligations, or controls whose engineering rationale has not been captured. They may also reveal a gap in requirements or design. Assigning every such item to the assessor would obscure those distinctions.

Controls in **P** ∪ **R** but not in **B** ((**P** ∪ **R**) \ **B**) identify engineering considerations beyond the current baseline record. They may warrant supplementation, a correction to the mapping, or another documented treatment. Their absence from the baseline does not invalidate the underlying mission need.

Table 1 summarizes how engineers, assessors, and authorizing officials should interpret set overlaps and differences.

Set Relationship	What It Represents	Primary Engineering and Authorization Action
P ∩ **R**	Control tied to both a requirement and a design principle	Verify that both records share the same system scope, operating modes, and evidence.
(**P** ∪ **R**) ∩ **B**	Baseline control backed by explicit engineering rationale	Confirm implementation and test evidence; do not treat mapping agreement as verification.
B \ (**P** ∪ **R**)	Baseline control with no recorded engineering mapping	Classify as inherited, organizational, operational, or a genuine requirements/design gap.
(**P** ∪ **R**) \ **B**	Engineered safeguard outside the current tailored control baseline	Evaluate for baseline supplementation or document as a system-specific assurance measure.

The comparison is diagnostic. It should improve both engineering completeness and the control baseline. It should never become an automatic rule for removing or adding controls.

8. Follow a Water Treatment Example

Consider a water treatment facility where a supervisory application can request chemical dosing changes. The facility needs to prevent unacceptable dosing behavior while preserving specified local control when the supervisory service fails or becomes compromised.

A requirement might state that supervisory dosing requests must be evaluated against facility-defined limits and relevant process conditions before they can affect local dosing control. A related requirement might specify continued local control or transfer to an approved protective mode when requests are rejected or the supervisory service is lost.

© 2026 RONROSSECURE, LLC | CC BY 4.0.

The design principle of *Mediated Access* guides the treatment of consequential requests and possible bypass paths (ensuring no supervisory command reaches an actuator without passing through local validation logic). The principle of *Least Privilege* limits the supervisory application's authority (restricting it to proposing set-point requests rather than writing directly to local controller registers). The principle of *Trustworthy System Control* guides the dependable enforcement of the constraints (including deterministic fallback to local control on watchdog timeout). The principle of *Domain Separation* helps constrain the application's influence over the control mechanism and its dependencies (isolating the local safety-enforcement controller from the supervisory network segment).

An engineering review might associate relevant provisions of NIST SP 800-53 controls AC-3 (Access Enforcement), AC-6 (Least Privilege), and SC-7 (Boundary Protection) with parts of that solution. Those associations would require examination of the actual control text, parameters, implementation, and scope; they would not establish that the control identifiers collectively satisfy the dosing requirements. [4]

Evidence could include architecture and dependency analysis, implementation inspection, tests of disallowed parameters and alternate paths, and demonstrations of specified behavior during supervisory loss. The assurance argument must address whether a compromised supervisory application can alter the enforcement logic, configuration, or process information on which the decision depends.

If the mechanism depends on a trusted process-data source, that dependency belongs in the assurance claim. Tolerating compromise of the supervisory application does not establish tolerance of compromise of every supporting dependency. Facility-specific engineering must establish the numerical limits, timing constraints, and acceptable protective behavior.

A hypothetical comparison shows how the three sets might be read. Suppose the engineering records associate AC-3 and AC-6 with both the requirements and the design decisions, SC-7 with the *Domain Separation* decision only, SI-10 with the request-evaluation requirement only, and CP-12 with the protective-mode requirement only. Suppose also that the tailored baseline contains AC-3, AC-6, SC-7, and SI-10 but not CP-12. These assignments are illustrative; real ones depend on the actual control text and the actual system.

Under those assumptions, AC-3 and AC-6 appear in P, R, and B, so the review checks that the records concern the same elements and operating modes. SC-7 appears in P and B but not R, which raises the question of whether the separation decision exposes a missing structural requirement. SI-10 appears in R and B but not P, which prompts a check on whether the request-evaluation design has recorded rationale tied to *Mediated Access*. CP-12 appears in R but not B, which makes it a candidate for supplementation or another documented treatment. None of these outcomes shows that a control is implemented or effective; each directs engineering attention to a specific question.

9. Keep the Information Where the Work Occurs

Requirements should be managed where they are developed and maintained, with links to the design decisions, implementations, verification results, and assurance claims that depend on them. Architecture decisions and evidence should retain their own authoritative records. An authorization view can assemble those relationships without creating a second, disconnected engineering record.

Each relationship should identify its rationale, scope, review status, responsible party, and applicable configuration. Evidence needs a result and limitations, not merely a hyperlink. Inherited and shared controls need explicit responsibility and dependency information. The depth of the record should scale with mission consequence: capture full detail first for the functions whose failure would have the most serious mission effects and keep records with fewer details elsewhere.

© 2026 RONROSSECURE, LLC | CC BY 4.0.

When a requirement, design, configuration, or assumption changes, the relationships help identify which claims, evidence, and assessment conclusions need review. For example, if a facility upgrade adds a remote maintenance interface to the local dosing controller, traceability immediately flags the *Domain Separation* and *Mediated Access* claims—and their associated SC-7 and AC-3 evidence—for re-verification. A contradiction should prompt investigation. It should not disappear because a dashboard still displays the same control identifier.

10. Give AI a Bounded Engineering Role

AI can help decompose requirements, suggest candidate control associations, locate supporting artifacts, and identify inconsistent terminology or unsupported claims. Engineers should review each proposed association against the authoritative source material and the actual system.

The tool should distinguish an AI suggestion from an accepted mapping and a supported conclusion. It should preserve why the reviewer accepted or rejected the suggestion. AI-generated rationale cannot substitute for missing engineering evidence, and an automated consistency check cannot establish system trustworthiness.

The value is a shorter path to the questions that require engineering judgment, with the supporting information available for review.

11. Strengthen the Authorization Decision

The RMF already calls for assessing whether controls are implemented correctly, operate as intended, and achieve the relevant outcomes. Engineering traceability can strengthen that assessment by connecting control conclusions to the system architecture, protection needs, and mission consequences. [2]

The authorizing official should be able to see why a security measure is needed, how it is realized, what evidence supports its effectiveness, and which assumptions or unresolved findings limit confidence. The same view should reveal what needs reassessment when the system changes.

That is a practical direction for next-generation Authorizations to Operate (ATOs): decisions grounded in a current understanding of the engineered system and its residual risk.

References

[1] R. Ross, M. Winstead, and M. McEvilley, *Engineering Trustworthy Secure Systems*, NIST Special Publication 800-160, Volume 1, Revision 1, National Institute of Standards and Technology, November 2022. https://doi.org/10.6028/NIST.SP.800-160v1r1

[2] Joint Task Force, *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*, NIST Special Publication 800-37, Revision 2, National Institute of Standards and Technology, December 2018. https://doi.org/10.6028/NIST.SP.800-37r2

[3] Joint Task Force, *Control Baselines for Information Systems and Organizations*, NIST Special Publication 800-53B, National Institute of Standards and Technology, October 2020 (updated December 2020). https://doi.org/10.6028/NIST.SP.800-53B

[4] Joint Task Force, *Security and Privacy Controls for Information Systems and Organizations*, NIST Special Publication 800-53, Revision 5, National Institute of Standards and Technology, September 2020 (updated December 2020). https://doi.org/10.6028/NIST.SP.800-53r5

© 2026 RONROSSECURE, LLC | CC BY 4.0.