

Visual Guide to Computer Systems

Zero Trust A Visual Guide

Sample Book — Chapters 1 to 2

Understand Identity, Access, Policy, and Security Architecture from the Ground Up

Reiner Braun

Introduction

This book is designed for readers who are taking their first steps with Zero Trust, including systems engineers who want to build the right foundation from the start.

Rather than asking you to memorize more than you need, the book focuses on the essential concepts that will support your future study and day-to-day work. More advanced topics are included only when they are likely to become useful to many readers, and even then they are introduced only far enough to explain their purpose, context, and overall role.

Processes, relationships, and structures that can be difficult to picture from text alone are supported by diagrams. By moving between explanations, visuals, and examples, you can build a clearer and more practical understanding of the subject.

Product names, service names, logos, icons, and other identifying marks mentioned in this book are trademarks or registered trademarks of their respective owners. This book is not an official publication, certified training resource, or endorsed learning material of any company or organization.

How to Use This Book

- **Read the chapters in order.** The sequence builds the required background step by step, from the Zero Trust mental model to access decisions, enforcement, operations, and troubleshooting.
- **Review “What You’ll Be Able to Do” before each chapter.** These objectives identify the practical knowledge and skills to focus on while reading.
- **Use the diagrams to understand relationships, processes, and structures.** They support the main text by making access paths, responsibilities, dependencies, state changes, and security boundaries easier to see.
- **Try the code, commands, and hands-on examples whenever practical.** Reproducing a small example is often the fastest way to confirm how the concept behaves in a real environment; use a safe test environment rather than a production system.
- **Use each Chapter Summary and Review Questions block to check retention.** Explain the answers in your own words and revisit the relevant section when an answer is unclear.
- **Treat advanced-topic overviews as signposts.** You do not need to memorize their implementation details at this stage; remember why the topic exists, where it fits, and when you may need to study it more deeply.

Table of Contents

Introduction

How to Use This Book

Main Text

Part I — The Zero Trust Mental Model

Chapter 1 — Zero Trust: The Big Picture

- The central idea
- Why the network perimeter is no longer enough
- Three principles that organize Zero Trust
- The practical meaning of the seven Zero Trust tenets
- Protect resources, not just zones
- Strategy, architecture, and products are different
- Security outcomes and limits
- An airport analogy
- Misconceptions that cause design errors
- What a systems engineer should retain

Chapter 2 — The Anatomy of an Access Request

- Every decision answers the same core questions
- The access lifecycle, step by step
- Running example: opening the monitoring application
- Login is not the same as resource access
- Static and dynamic evidence
- More outcomes than allow or deny
- Continuous verification without constant interruption
- A practical tracing method
- Two distinctions that prevent major mistakes

Main Text

Zero Trust is a security approach that protects each resource—such as an application, service, data store, or administrative capability—through explicit, least-privilege access decisions rather than assuming that a user, device, or workload is safe because it is inside a particular network or was trusted earlier.

A **workload** is running software, such as a service, scheduled job, or container, that can request access under a non-human identity.

The practical question is not simply, “Is this request coming from inside the company?” It is:

Who or what is requesting which action on which resource, using what device or workload, under what current conditions, and according to which policy?

A sound Zero Trust design answers that question before access is granted, enforces the answer on the actual path to the resource, and can change or revoke access when the evidence changes.

This text develops that foundation in four parts. Part I establishes the Zero Trust mental model and the flow of an access decision. Part II explains the information required to make trustworthy decisions: inventories, ownership, identities, device posture, and authorization policy. Part III applies those decisions through enforcement, workload identity, telemetry, and resilient failure handling. Part IV turns the architecture into a phased implementation and an operating practice.

Part I — The Zero Trust Mental Model

Chapter 1 — Zero Trust: The Big Picture

What You'll Be Able to Do

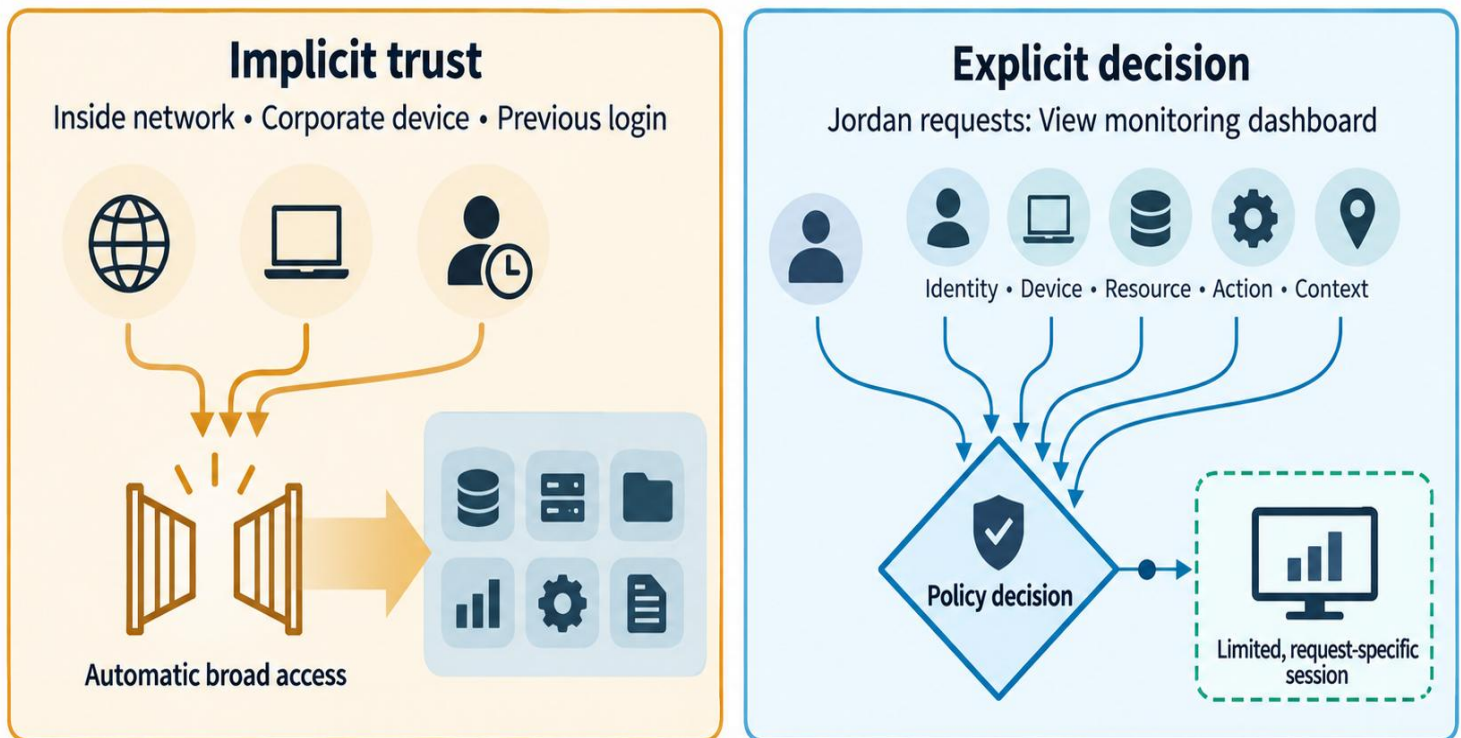
- Explain Zero Trust as a resource-specific, evidence-based security approach rather than a product or a synonym for remote access.
- Describe how **verify explicitly**, **use least privilege**, and **assume breach** work together.
- Distinguish Zero Trust strategy, architecture, implementation controls, and the security practices that remain necessary alongside them.

The central idea

Zero Trust removes implicit trust from access decisions. **Implicit trust** is access granted mainly because of a broad condition such as network location, device ownership, or a previous successful login, without checking whether the current request should be allowed.

Chapter 1: Zero Trust Big Picture

1-1 Implicit Trust vs Explicit Decisions



Remove automatic technical trust; do not distrust people.

001

Figure 1-1: Implicit trust versus an explicit, request-specific decision

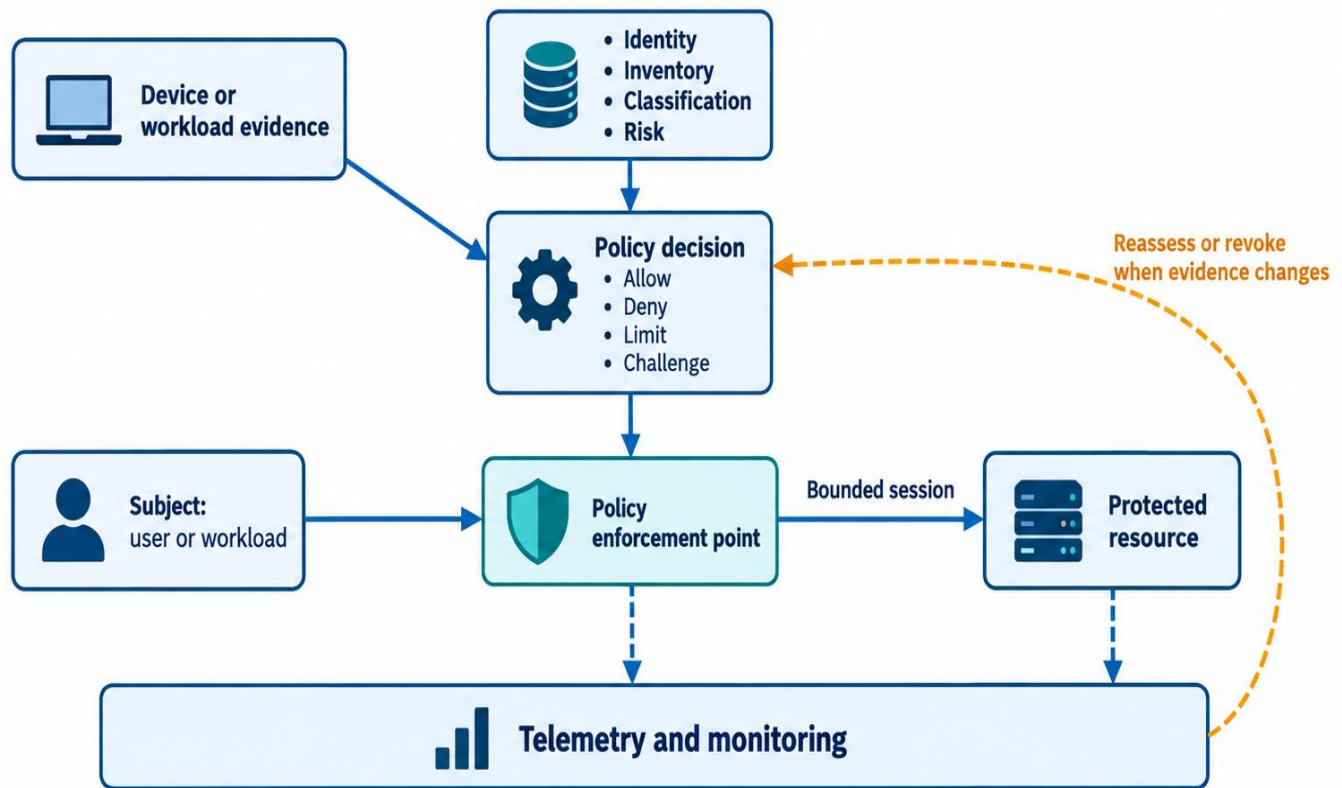
Zero Trust does not mean that every person is suspected of malicious intent. It means that technical systems should not treat any identity, device, workload, or network location as permanently safe. A legitimate employee account can be stolen. A company laptop can become infected. A trusted service can be misconfigured. A cloud credential can leak. Security therefore depends on evidence that is relevant to the specific request.

A **Zero Trust architecture** is a system design that applies this approach through identity services, device and workload evidence, access policies, enforcement controls, and monitoring. **Authentication** establishes the identity making a request, while **authorization** decides whether that identity may perform the requested action. A **session** is the bounded

period during which approved access remains active. Zero Trust architecture is not one product and it is not a synonym for remote-access technology.

Chapter 1: Zero Trust Big Picture

1-1 The Complete Zero Trust Big Picture



002

Figure 1-2: The complete Zero Trust big picture

Why the network perimeter is no longer enough

A **network perimeter** is the boundary between a network treated as internal and networks treated as external, often enforced by a **firewall**, which allows or blocks network traffic according to rules. Traditional designs did not rely on the perimeter alone, but many gave users and devices broader reach after they connected to the corporate network or a **virtual private network**, which creates an encrypted connection into a private network.

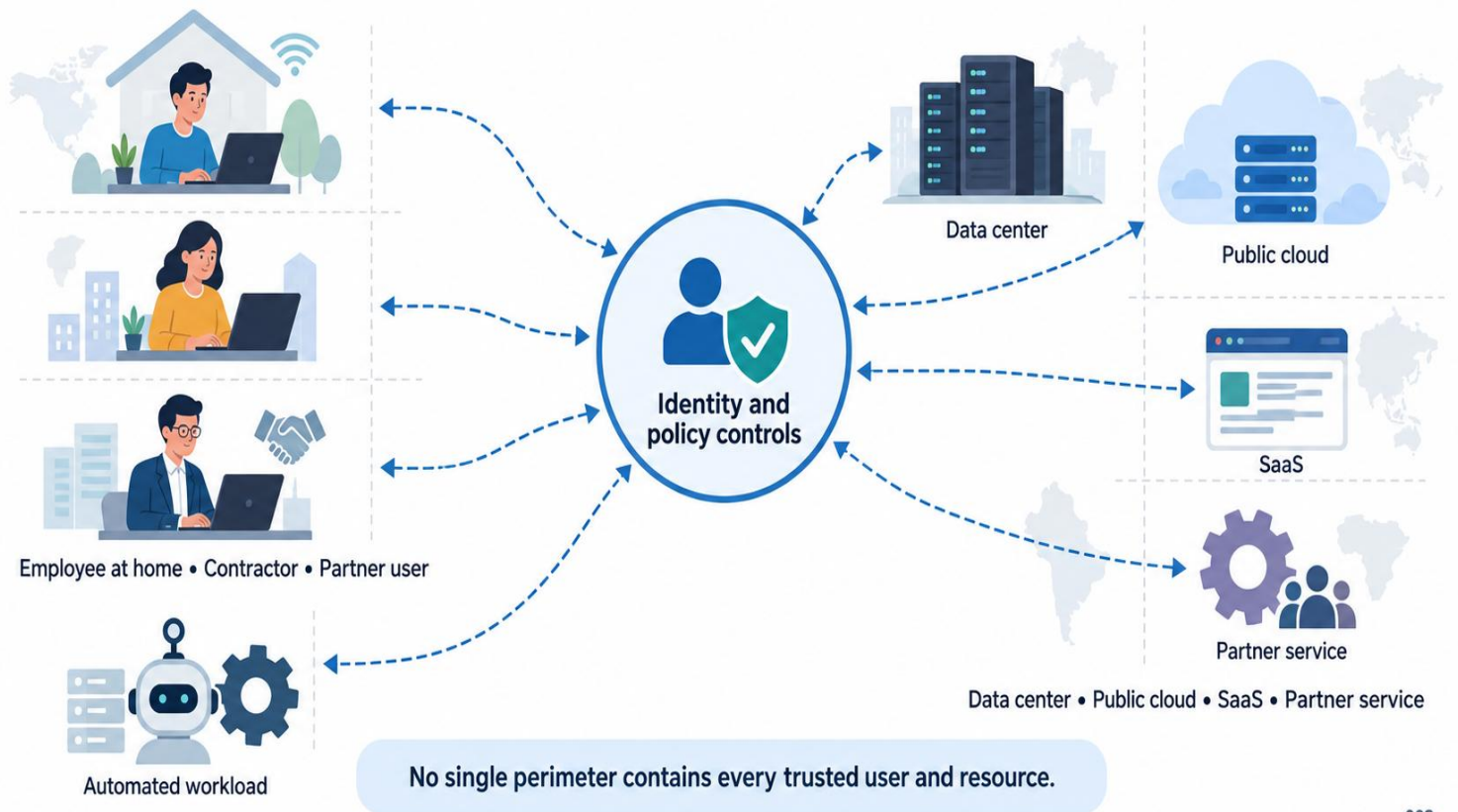
That assumption is increasingly weak because important resources and users no longer sit inside one stable boundary:

- Employees work from homes, customer sites, and temporary locations.
- Applications run in data centers, public clouds, and **software-as-a-service platforms**, where a provider operates the application and users access it over a network.
- Contractors and partners need limited access without becoming ordinary internal users.
- Personally owned and partner-managed devices may connect to selected services.
- Internal accounts, **endpoints** such as user computers and mobile devices, and servers can be compromised.

- Automated services communicate across cloud accounts, clusters, and organizations.

Chapter 1: Zero Trust Big Picture

1-2 Users and Resources Are Distributed



003

Figure 1-3: Users and resources distributed across home, data center, cloud, SaaS, and partner environments

Consider a systems engineer who connects from home to review an internal monitoring dashboard. A perimeter-focused design might treat the engineer as trusted after a virtual private network connection succeeds. That connection can expose routes to many systems even though the engineer needs only one application and one set of read-only functions.

A Zero Trust design asks for narrower proof and grants narrower access. It may verify the engineer's identity, confirm that the laptop is managed and compliant, check that the requested action is permitted, and establish a session only to the monitoring application. The user does not gain general network trust merely by connecting.

Chapter 1: Zero Trust Big Picture

1-2 Broad VPN Reach vs a Narrow Application Session

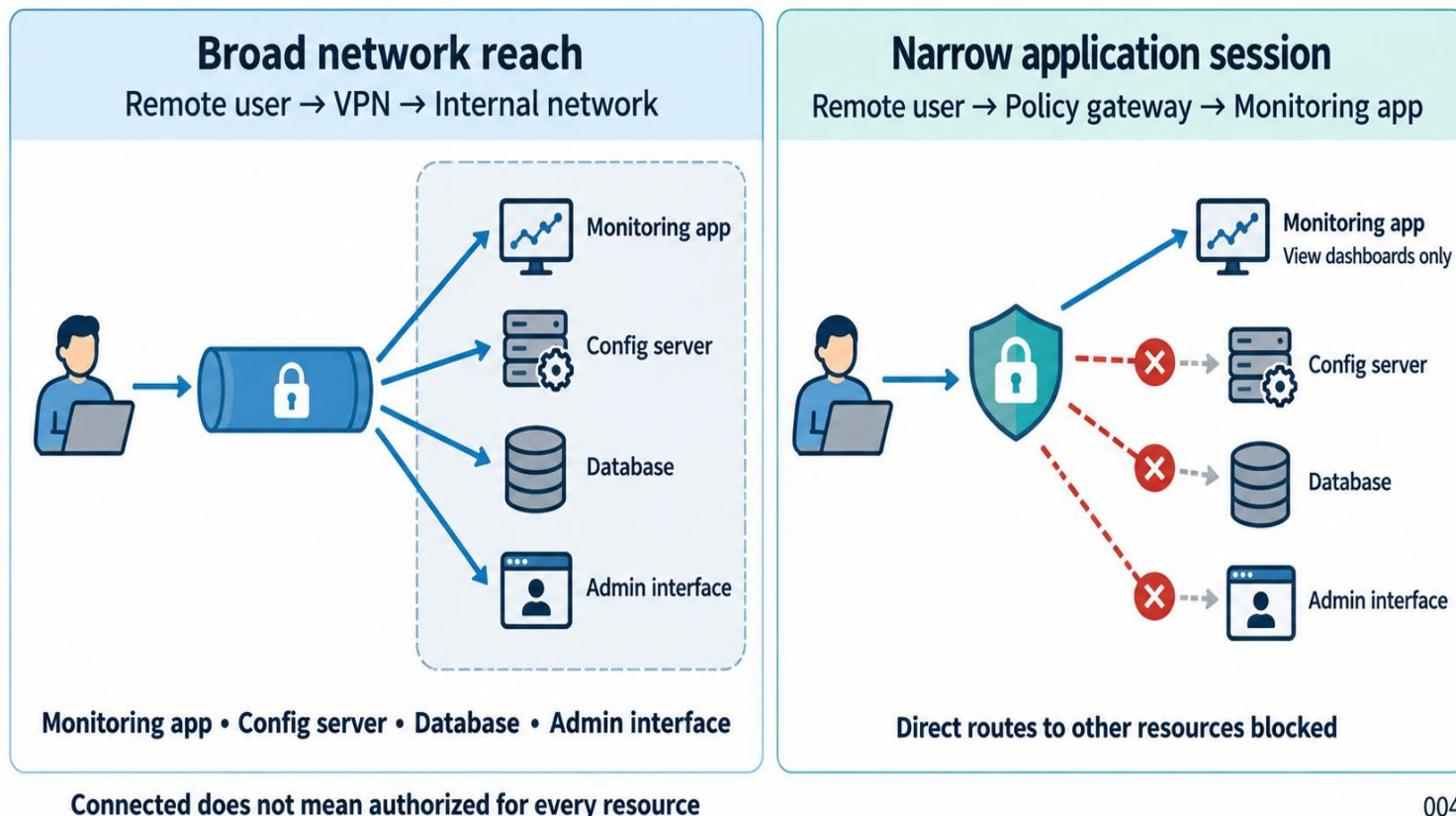
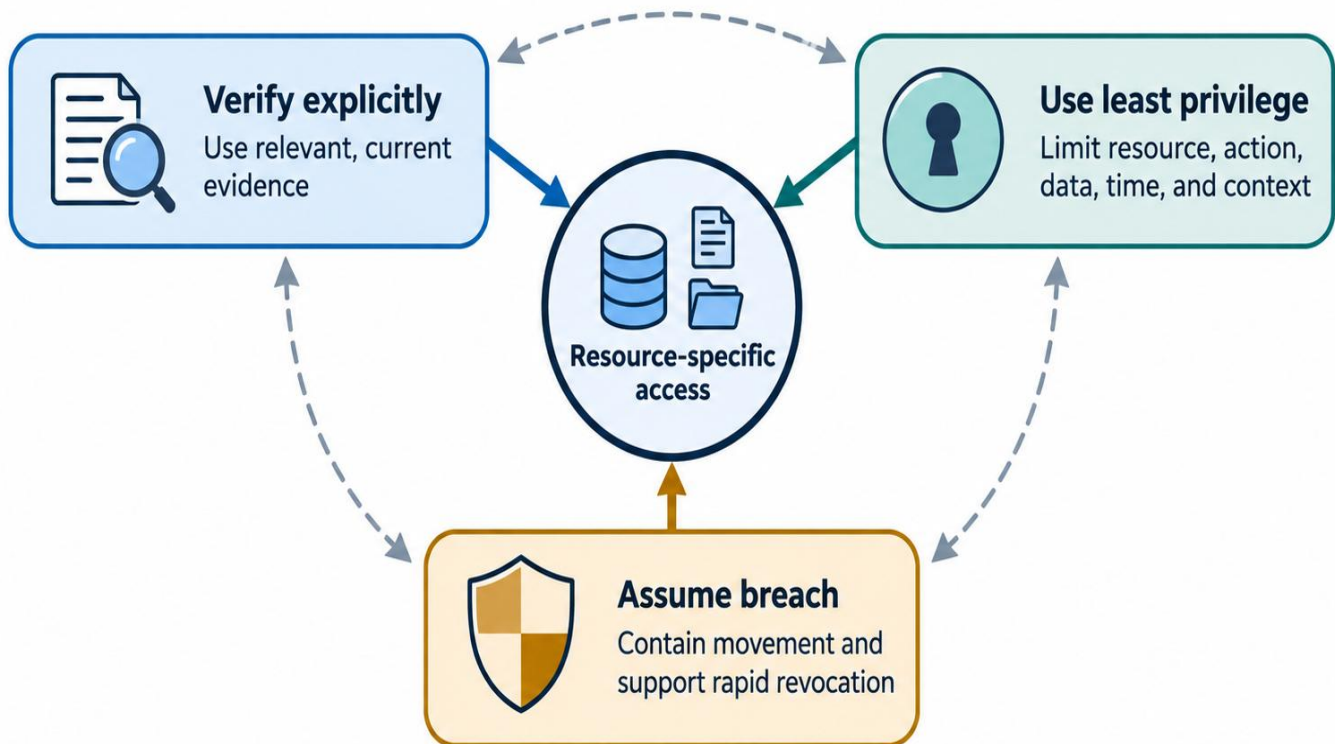


Figure 1-4: Broad VPN reach versus a narrow application session

Chapter 1: Zero Trust Big Picture 1-3 Three Principles Working Together



Evidence decides the narrowest safe path; containment limits failure.

005

Figure 1-5: How verify explicitly, least privilege, and assume breach work together

Verify explicitly

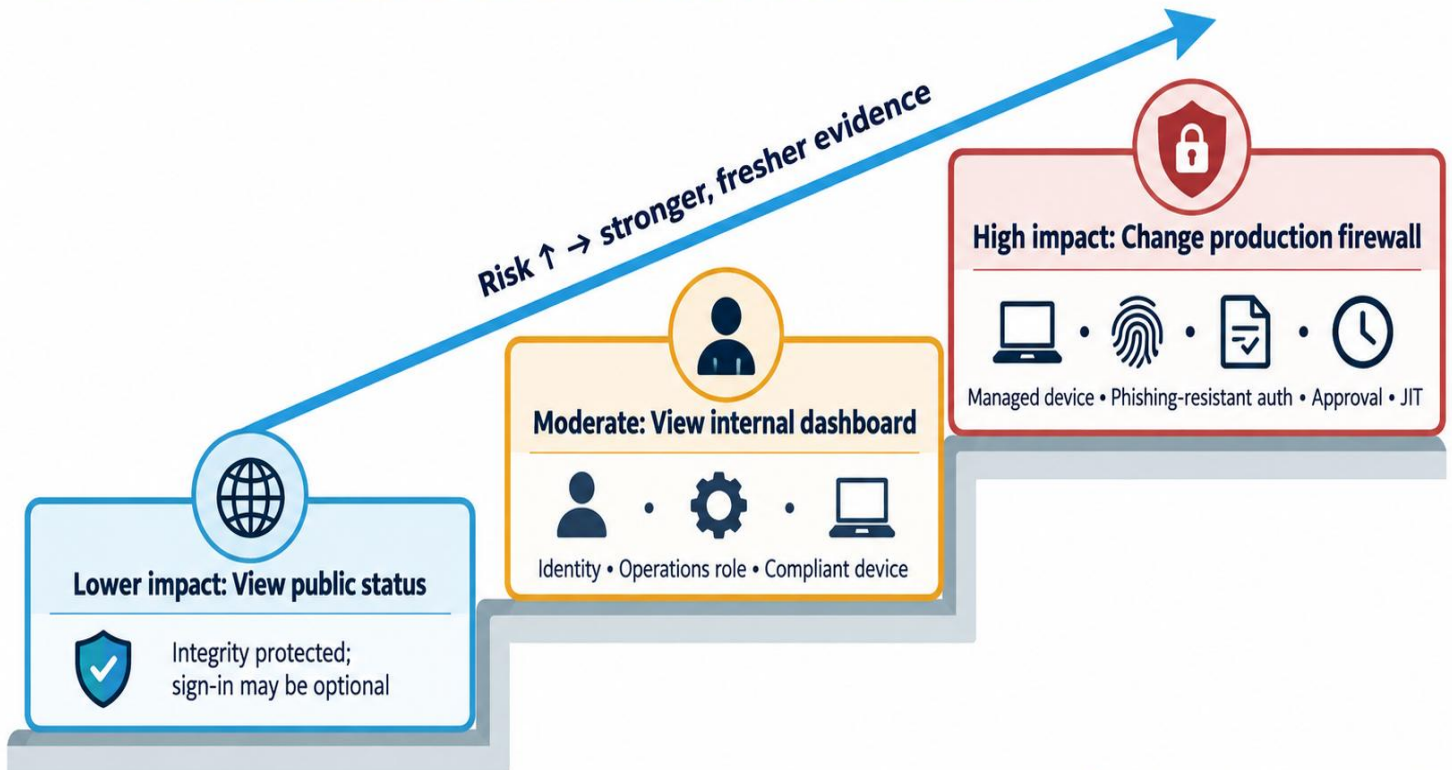
Verify explicitly means making an access decision from relevant, current evidence instead of relying on a broad trust shortcut.

The evidence may include the user or workload identity, authentication strength, device security posture, resource sensitivity, requested action, time, location, and current risk information. Not every request needs every signal. The policy should use evidence that materially improves the decision.

For example, reading a low-sensitivity status page may require a valid workforce identity. Changing a production firewall rule may additionally require a managed device, **phishing-resistant authentication**, which uses cryptographic methods that cannot be successfully reused at a fake sign-in site, explicit approval, and a time-limited privileged role.

Chapter 1: Zero Trust Big Picture

1-3-1 Evidence Scales with Action Risk



Example; requirements vary. 006

Figure 1-6: Evidence requirements increase with action risk

Use least privilege

Least privilege means granting only the permissions required for the task, only to the required resources, and only for as long as necessary.

Least privilege has several dimensions:

- **Resource scope:** access one application rather than an entire network segment.
- **Action scope:** view alerts rather than change alert rules.
- **Data scope:** read one operational dataset rather than all production data.
- **Time scope:** hold administrative rights for thirty minutes rather than permanently.

- **Context scope:** permit the action only from a compliant device using strong authentication.

Chapter 1: Zero Trust Big Picture

1-3-2 Five Dimensions of Least Privilege

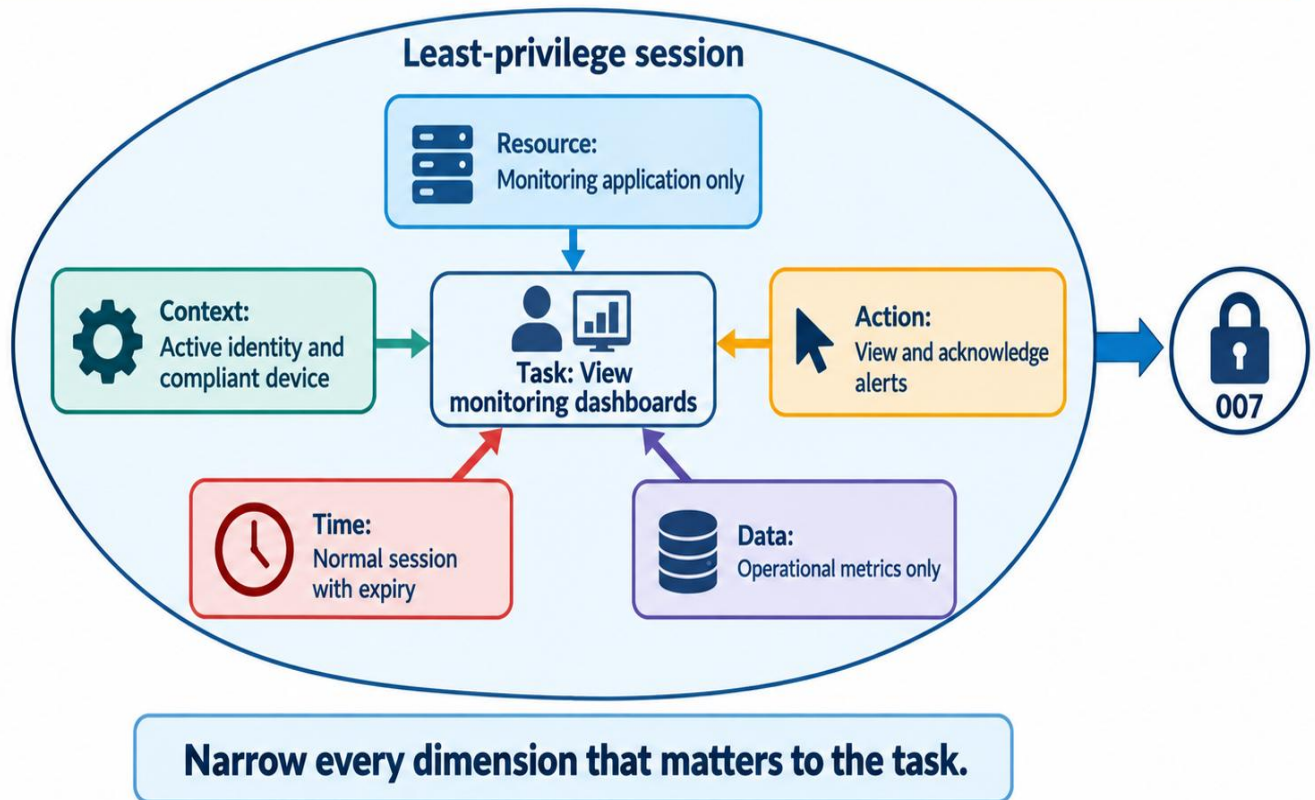


Figure 1-7: Five dimensions of least privilege

Least privilege is not merely a small set of user groups. It is the continuing practice of matching access to a real task.

Assume breach

Assume breach means designing as though an identity, endpoint, service, or network segment may already be compromised, even when no compromise is currently known.

This principle changes architecture. Systems should limit lateral movement, protect administrative paths, use separate identities for different trust levels, record decisions, and support rapid revocation. **Lateral movement** is the process by which an attacker who compromises one identity or system reaches additional systems.

Chapter 1: Zero Trust Big Picture

1-3-3 Assume Breach Limits Blast Radius

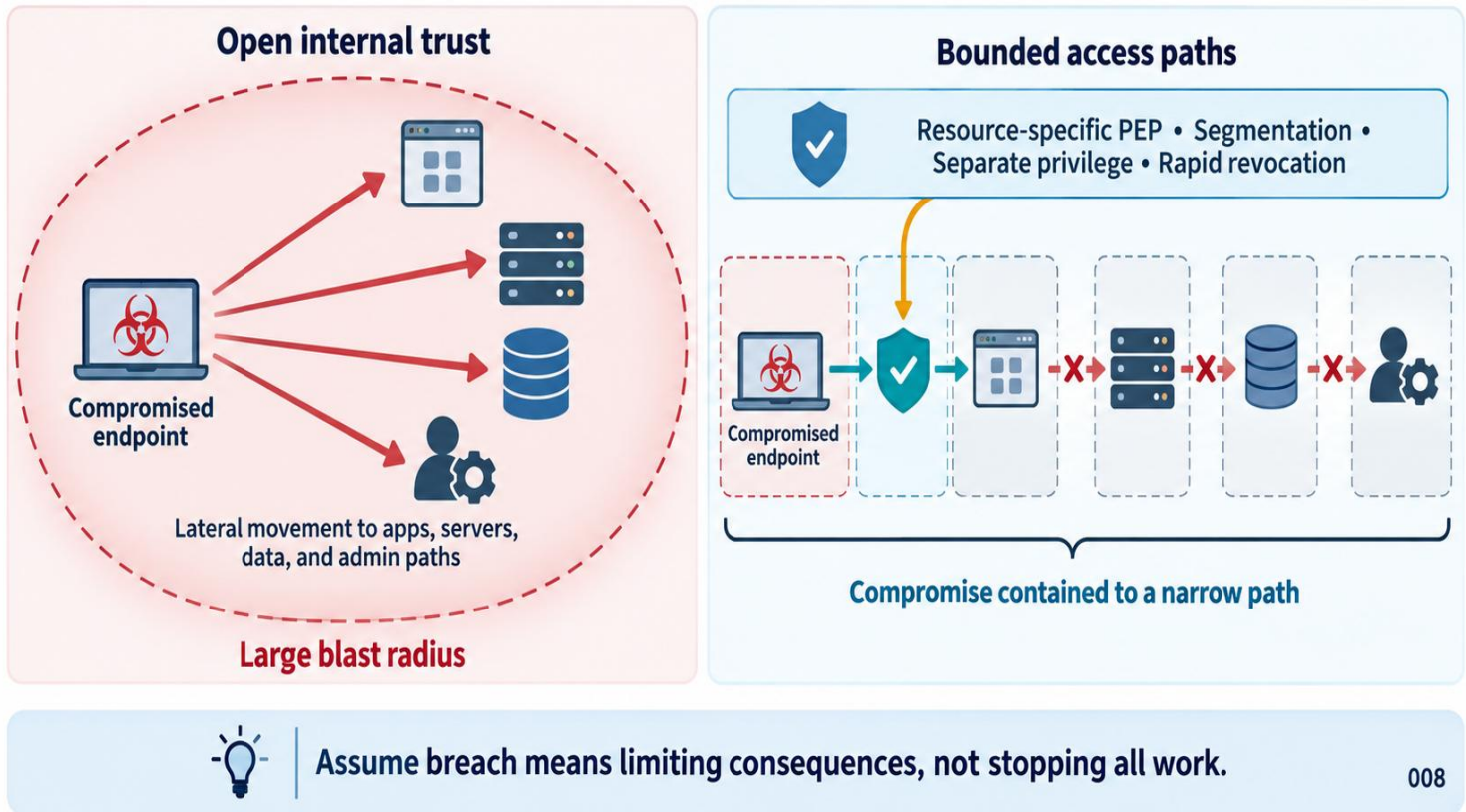


Figure 1-8: Limiting lateral movement and blast radius after one compromise

Assuming breach does not mean declaring the environment unsafe and stopping all work. It means avoiding designs in which one successful compromise provides broad, durable control.

The practical meaning of the seven Zero Trust tenets

An **application programming interface (API)** is a defined way for software components to request data or operations from another system.

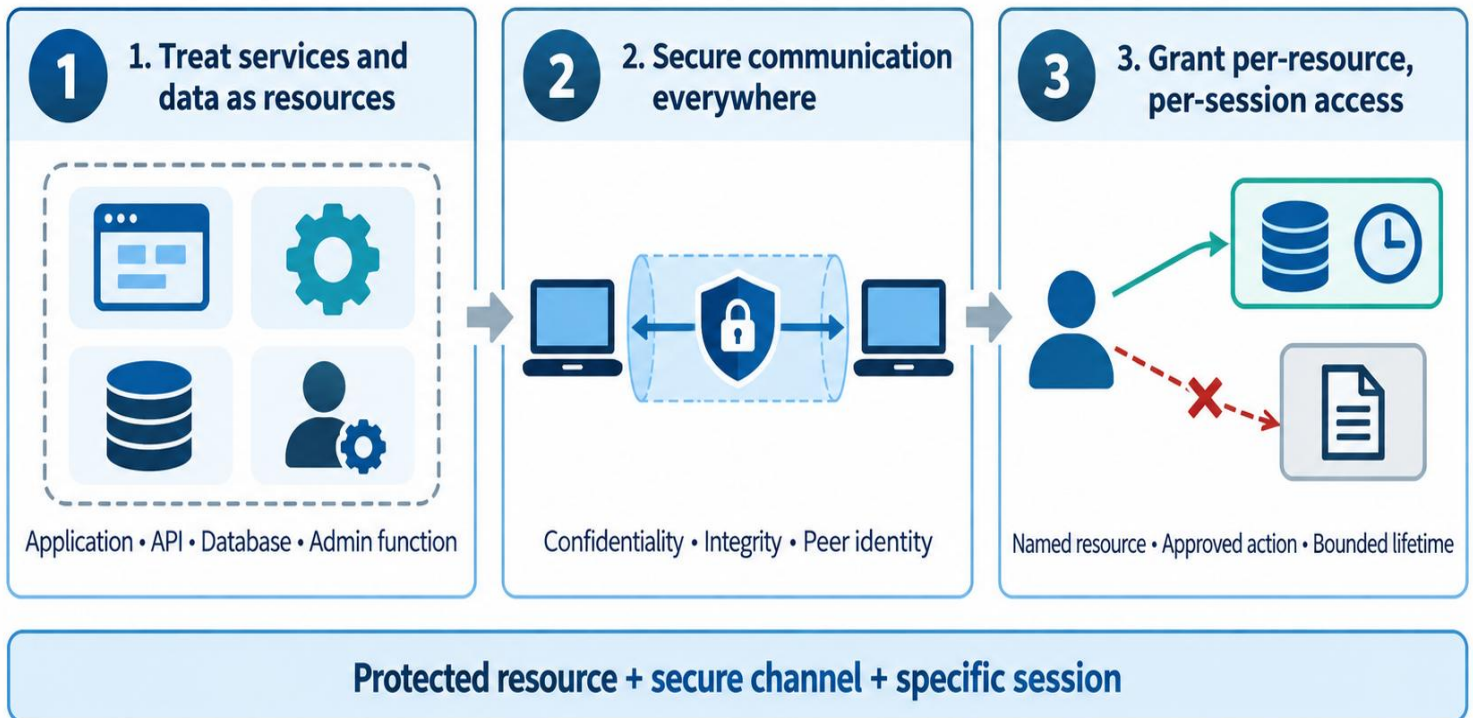
The model in [NIST SP 800-207, Zero Trust Architecture](#) is commonly summarized through seven tenets. In practical systems-engineering terms, they mean the following:

1. **Treat data sources and computing services as protected resources.** Applications, servers, APIs, databases, administrative functions, and important data all require deliberate protection.
2. **Secure communications regardless of network location.** Internal traffic is not automatically safe; confidentiality, integrity, and peer identity still matter.

3. **Grant access to individual resources per session.** Approval for one resource or session should not silently authorize unrelated resources or future sessions.

Chapter 1: Zero Trust Big Picture

1-4 Zero Trust Tenets 1-3



009

Figure 1-9: Tenets 1–3: resources, secure communication, and per-session access

4. **Use dynamic policy.** Decisions should consider identity, resource, action, device or workload state, and environmental conditions rather than one fixed network rule.
5. **Monitor enterprise assets.** An organization needs current evidence about the integrity and security posture of managed devices and systems.
6. **Authenticate and authorize dynamically.** Identity verification and permission checks occur before access and may be reassessed when conditions change.

7. **Use operational evidence to improve protection.** Access decisions, security events, asset information, and observed behavior should feed policy review and response.

Chapter 1: Zero Trust Big Picture

1-4 Zero Trust Tenets 4–7



Figure 1-10: Tenets 4–7: dynamic policy, asset monitoring, reassessment, and learning

These tenets are architectural goals, not a list of products to purchase. An organization may implement them gradually and with different technologies in different environments.

Protect resources, not just zones

A **resource** is the application, service, data, device, or administrative capability that a requester wants to use. Resource-centric design starts by identifying what requires protection and what actions are possible.

Suppose the operations network contains the following systems:

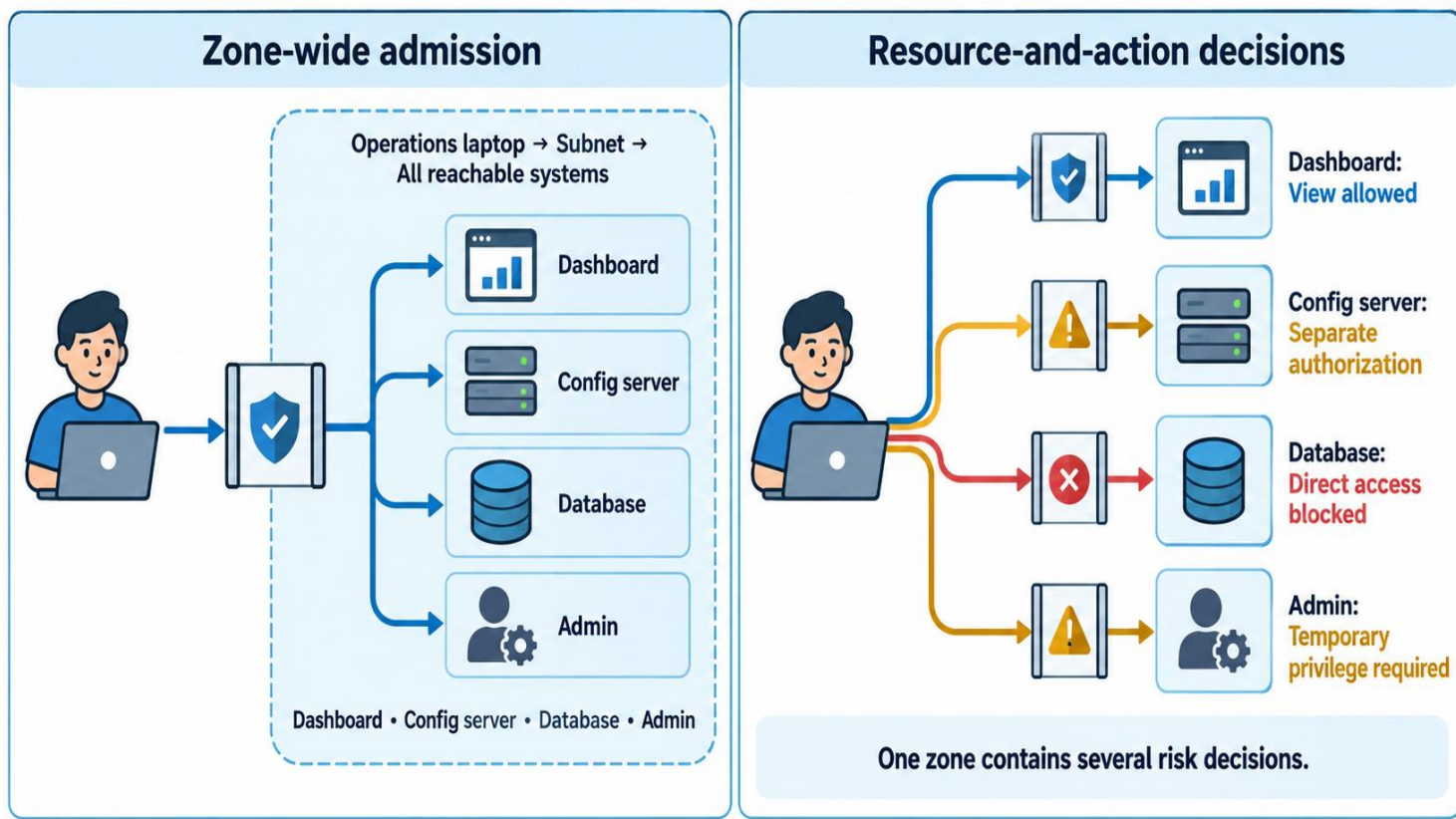
- A monitoring dashboard.
- A configuration-management server.
- A production database.
- A privileged administration interface.

A zone-based rule might allow a connected operations laptop to reach the whole subnet. A resource-centric design asks separate questions:

- May this engineer view the monitoring dashboard?
- May the same engineer edit monitoring rules?
- May the laptop initiate a connection to the configuration server?

- May an approved administrator change production data?

Chapter 1: Zero Trust Big Picture



011

Figure 1-11: Zone-wide admission versus resource-and-action decisions

This shift matters because the risk and required evidence differ for each action. Viewing health metrics is not equivalent to changing production configuration, even when both occur in the same application.

Strategy, architecture, and products are different

Telemetry is operational and security data emitted by systems, such as access decisions, device events, and application activity.

Concept	What it means	Example
Zero Trust strategy	The organization's security direction: remove implicit trust, use least privilege, and make resource-specific decisions.	A policy that internal network location alone is never sufficient authorization.
Zero Trust architecture	The designed relationships among identities, evidence sources, policy decision services, enforcement points, resources, and telemetry.	A central identity service and device-management system provide evidence to a policy service, which instructs a gateway in front of an application.
Zero Trust capability or product	A technology that performs part of the architecture.	A strong authenticator, an application proxy, a device-management platform, or a decision service.

No single product supplies every required function. A remote-access gateway may enforce application access but still depend on an identity provider, device posture service, resource inventory, and well-designed authorization policy.

Security outcomes and limits

A well-implemented Zero Trust approach seeks to:

- Reduce unauthorized access to applications, data, and administrative functions.
- Limit the damage caused by stolen credentials.
- Reduce lateral movement after a system is compromised.
- Make access decisions and exceptions more visible.
- Revoke or constrain access when identity, device, or risk conditions change.
- Improve accountability by linking actions to identifiable users and workloads.

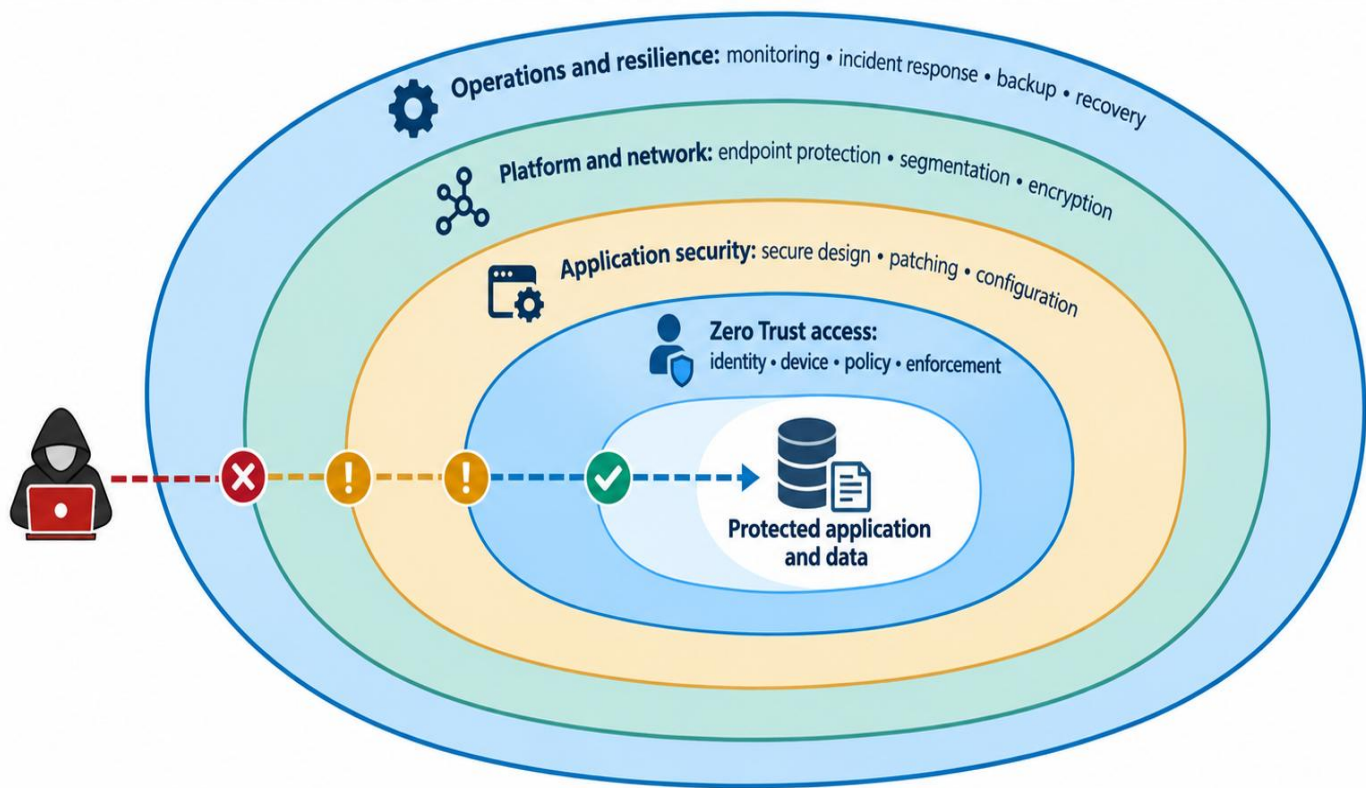
Zero Trust does not remove all risk. A correctly authorized administrator can still make a mistake. A vulnerable application can still be exploited. A malicious insider may still abuse legitimate access. An outage in a critical identity service can still affect availability.

Zero Trust therefore complements, rather than replaces:

- Secure configuration and patching.
- Endpoint protection.
- Firewalls and network controls.
- Encryption.
- Backups and recovery.
- Secure software development.
- Monitoring and incident response.

Chapter 1: Zero Trust Big Picture

1-7 Zero Trust Within Defense in Depth



012

Figure 1-12: Zero Trust within defense in depth

The broader principle is **defense in depth**, which means using multiple complementary controls so that one control failure does not determine the entire outcome.

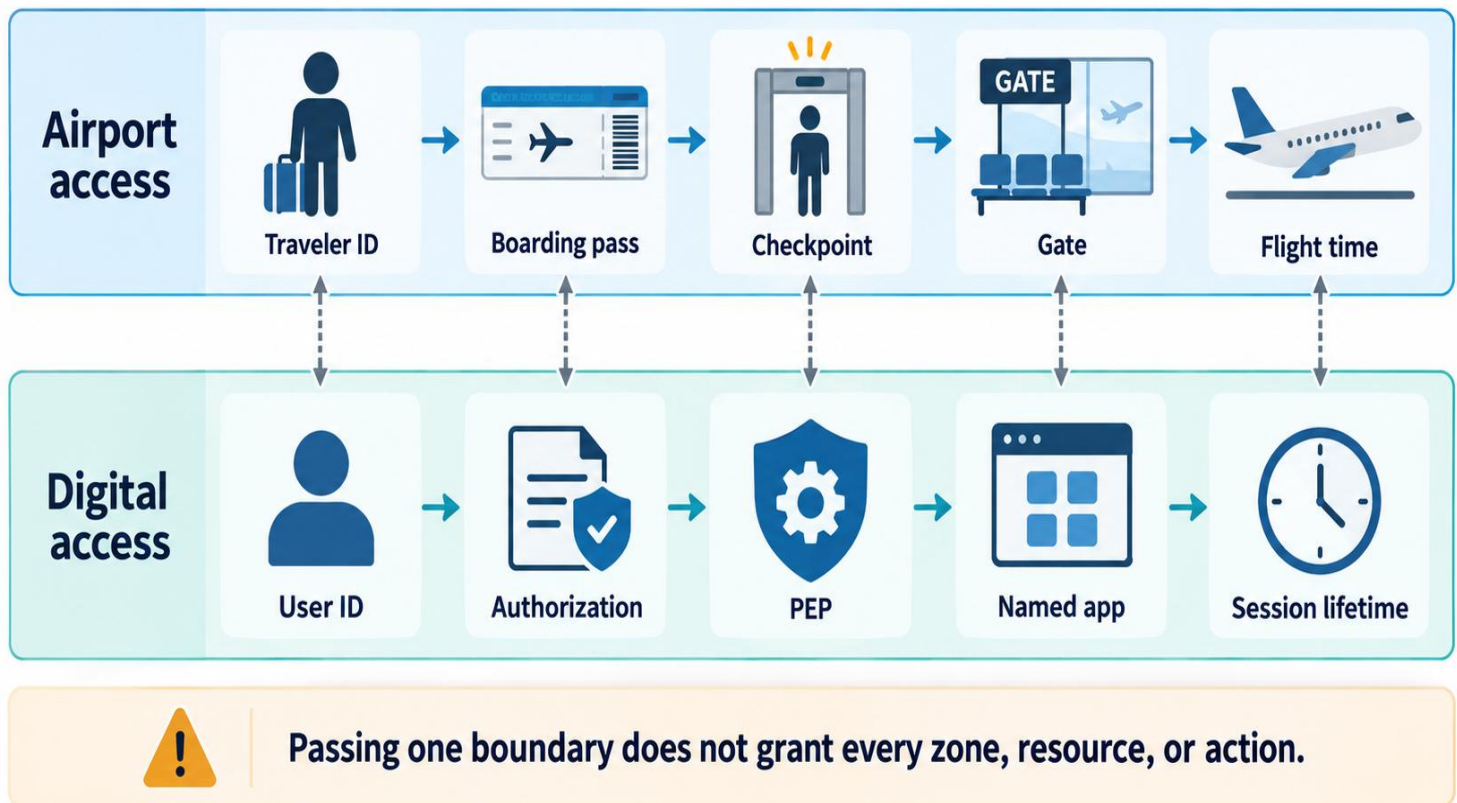
An airport analogy

An airport illustrates the resource-specific nature of access. Entering the public terminal does not grant permission to enter the baggage-handling area, an aircraft cockpit, or every departure gate. A boarding pass authorizes a particular journey. Staff areas require different credentials. Security personnel can withdraw access when conditions change.

The analogy is imperfect, but it captures three important points:

- Passing one boundary does not grant access everywhere.
- Different resources require different evidence and permission.
- Access can be limited in purpose, location, and time.

Chapter 1: Zero Trust Big Picture



013

Figure 1-13: Airport zones mapped to resource-specific digital access

A corporate network connection should be treated in the same way. It may provide a communication path, but it should not be the final proof that every action is authorized.

Misconceptions that cause design errors

“Never trust” means employees are treated as dishonest. Zero Trust addresses automatic technical trust, not personal character. It protects legitimate users by limiting what stolen identities and compromised devices can do.

Zero Trust is a replacement name for a VPN. A virtual private network creates an encrypted network connection. It can be part of a Zero Trust design, but the connection itself does not establish least-privilege authorization to each resource.

The corporate LAN is trusted. Internal location can be one contextual signal, but it is not sufficient evidence. Malware, stolen credentials, misconfiguration, and unauthorized internal movement all occur inside enterprise networks.

Purchasing a Zero Trust product completes the strategy. Products provide capabilities. The architecture still requires resource ownership, identity lifecycle controls, device evidence, authorization rules, enforcement coverage, monitoring, and operations.

What a systems engineer should retain

The foundation can be reduced to one working statement:

Do not grant broad or durable access because a request appears to come from a familiar place. Verify relevant evidence, authorize a specific action on a specific resource, enforce the decision, and remain able to change it.

Chapter Summary

Zero Trust removes automatic technical trust and replaces it with explicit, resource-specific decisions. Its purpose is to reduce unauthorized access and the effect of compromise while working as one layer of a broader defense-in-depth strategy.

Review Questions

1. Why is network location insufficient evidence for granting access to a sensitive resource?
2. How do least privilege and assume breach limit the effect of a stolen credential?
3. What is the difference between a Zero Trust strategy, an architecture, and a product capability?

Chapter 2 — The Anatomy of an Access Request

What You'll Be Able to Do

- Break an access request into subject, device or workload, resource, action, context, policy, and session.
- Trace a request from authentication through authorization, enforcement, monitoring, and possible revocation.
- Distinguish sign-in from resource access and explain why outcomes can be more precise than a simple allow-or-deny decision.

Every decision answers the same core questions

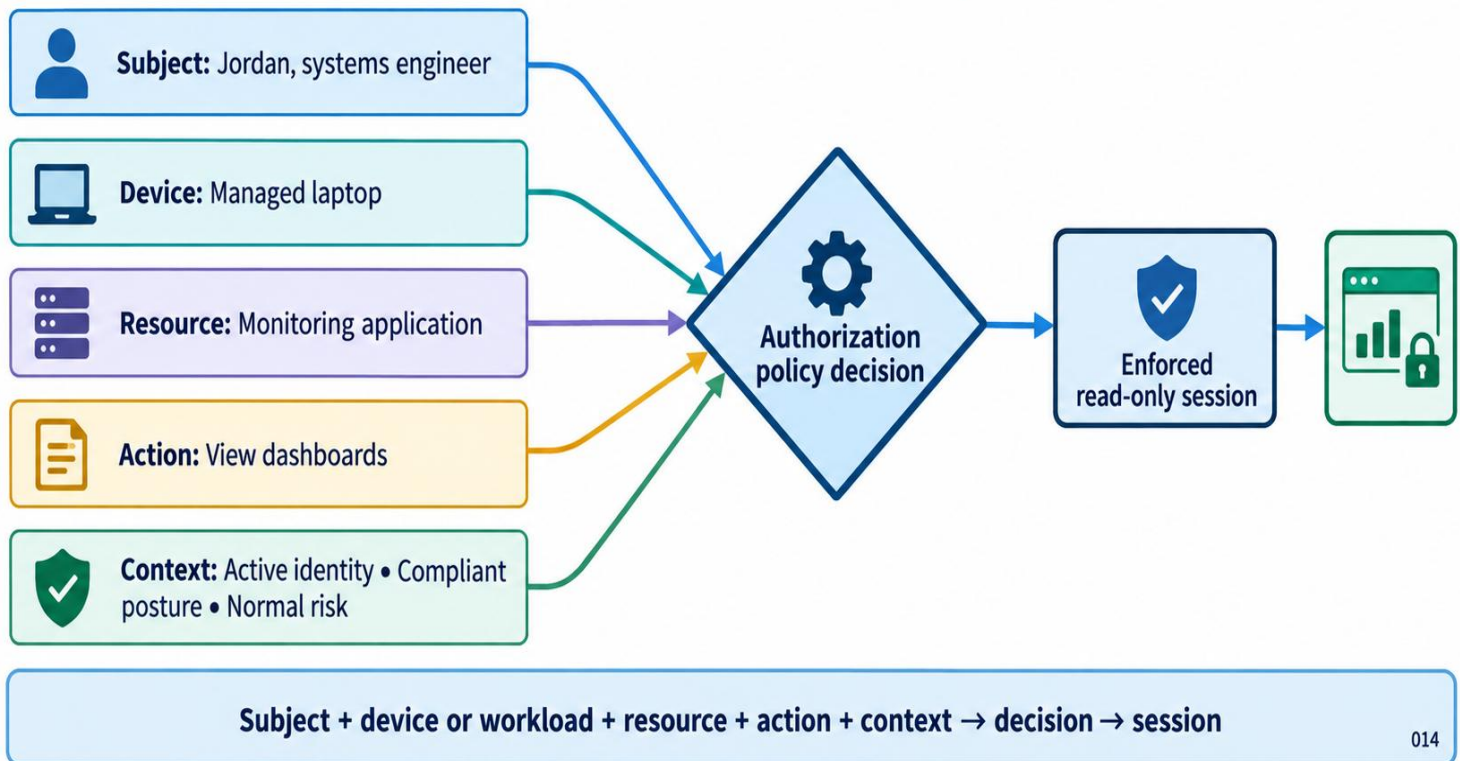
A Zero Trust access decision identifies the requester, the device or workload, the resource, the action, the current context, and the policy that applies.

A useful mental model is:

subject + device or workload + resource + action + context → policy decision → enforced session

Chapter 2: Access Requests

2-1 Anatomy of One Access Request



014

Figure 2-1: Anatomy of one access request

The terms in that model have precise roles. **Device posture** is the security-relevant state of a device, such as its management, configuration, update, and compromise status.

A **subject**, also called a **principal**, is the user, service, or workload requesting access. A person signed in as an employee is a human subject; a scheduled backup process using its own identity is a non-human subject.

A **resource** is the application, service, data, system, or administrative capability being protected.

An **action** is what the subject wants to do, such as view, download, change, approve, execute, or administer.

Authentication establishes which identity is making the request, usually by validating one or more credentials.

Authorization determines whether the authenticated identity may perform the requested action on the requested resource.

Context is additional evidence relevant to the decision, such as device compliance, authentication strength, location, time, resource sensitivity, or an active security incident.

A **session** is the bounded period during which approved access remains active under defined conditions.

These elements should not be collapsed into one question. A user can be successfully authenticated and still be unauthorized for a particular action.

The access lifecycle, step by step

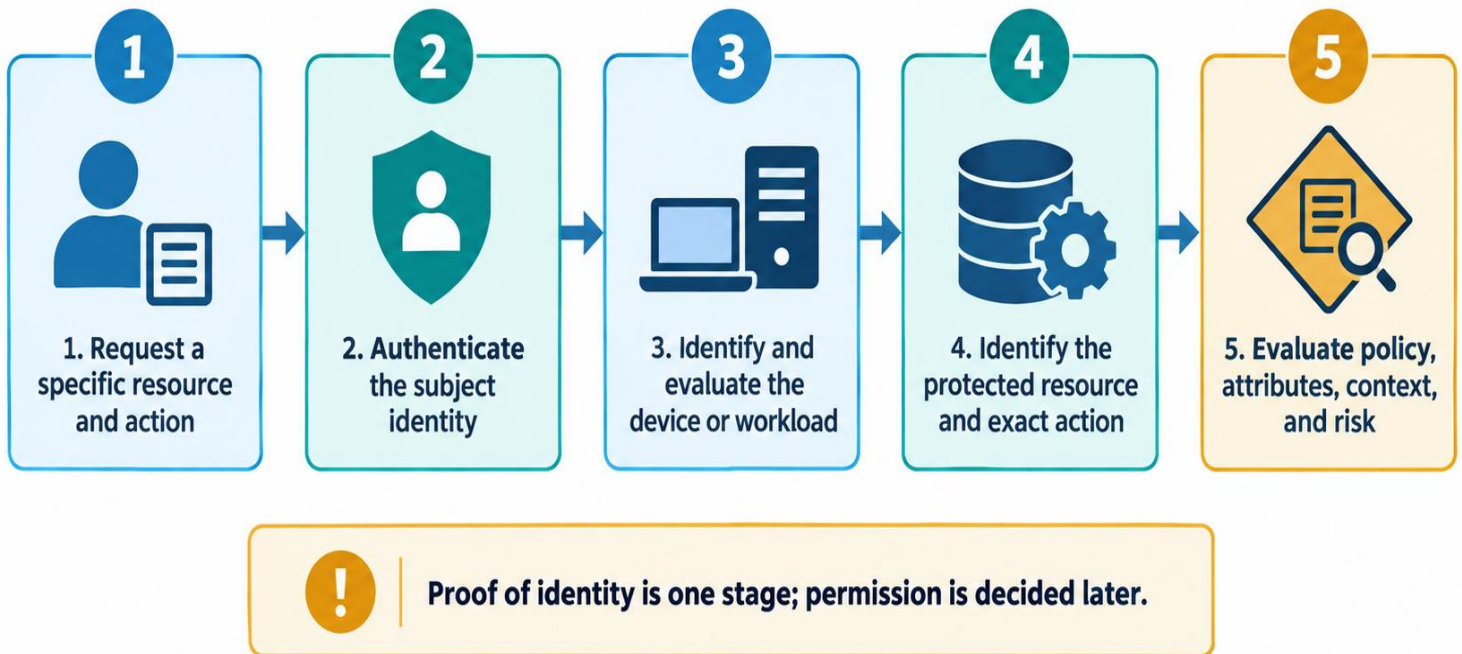
A typical access request follows eight stages. Products may combine stages, but the logic remains useful for design and troubleshooting.

1. **The subject requests a resource and action.** The request should be specific enough to distinguish, for example, viewing a dashboard from changing production settings.
2. **The subject identity is authenticated.** The identity service validates the credential and may require more than one authentication factor.
3. **The device or workload is identified and evaluated.** The system checks available evidence such as device enrollment, certificate identity, security posture, or **workload attestation**, which is supported evidence that a software instance has an expected identity or state.
4. **The resource and requested action are identified.** Policy cannot be precise when the protected object or operation is unknown.

5. **Policy evaluates identity, attributes, context, and risk.** The decision uses the evidence required for that resource and action.

Chapter 2: Access Requests

2-2 Access Lifecycle: Request Through Policy Evaluation



015

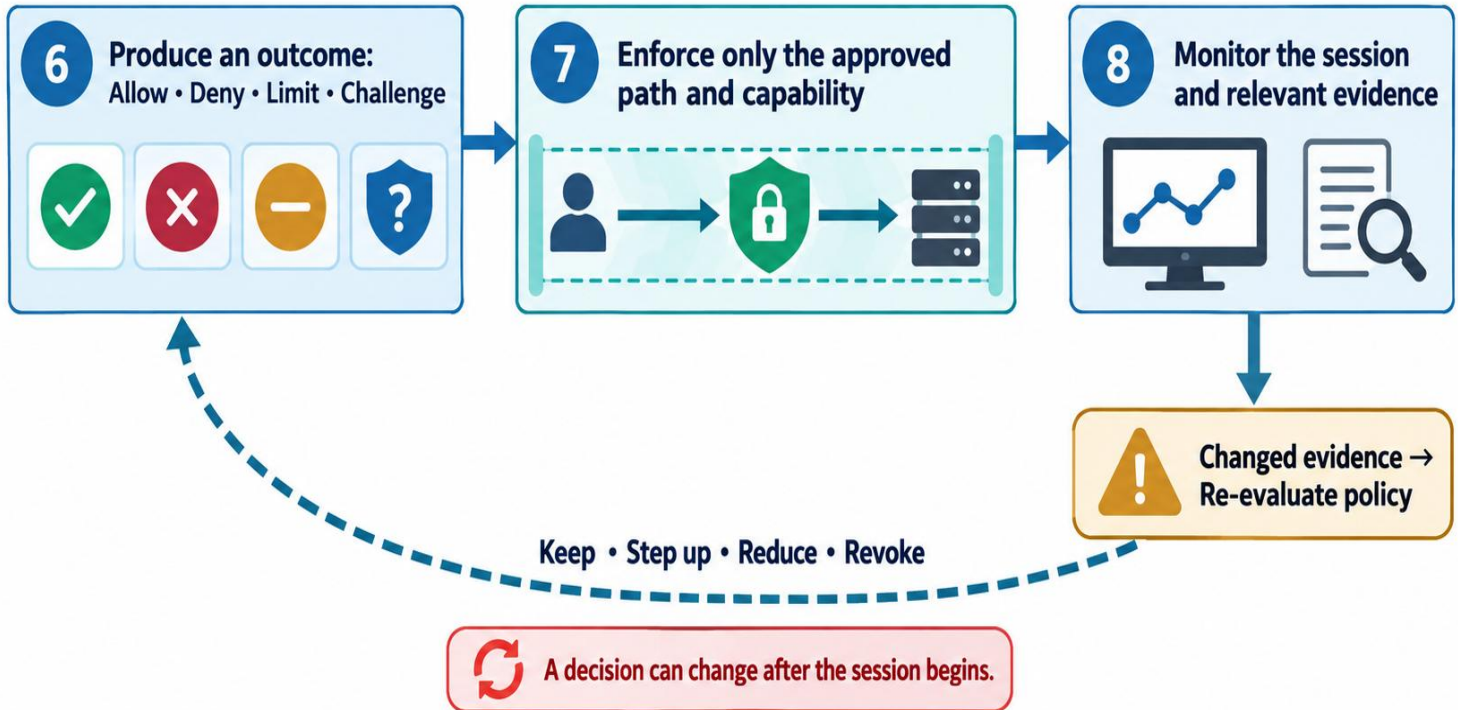
Figure 2-2: Access lifecycle stages 1–5: request through policy evaluation

6. **The system produces an outcome.** It may allow, deny, limit, delay, or require an additional condition.
7. **An enforcement control establishes the approved access.** The subject receives only the path and capability represented by the decision.

8. **The session is monitored and may be reassessed.** A significant change, such as a disabled account or compromised device, can reduce or end access.

Chapter 2: Access Requests

2-2 Access Lifecycle: Outcome, Enforcement, and Reassessment



016

Figure 2-3: Access lifecycle stages 6–8: outcome, enforcement, and reassessment

This lifecycle separates proof of identity from permission and separates permission from enforcement. That separation is important because each stage can fail independently.

Running example: opening the monitoring application

Jordan is a systems engineer working remotely from a company-managed laptop. Jordan needs to view an internal monitoring dashboard.

1. A specific request is made

Jordan opens the monitoring application in a browser. The requested resource is the monitoring application, and the requested action is initially **view dashboards**.

The browser is not asking for general membership in the internal network. It is asking for a particular application capability.

2. The user identity is authenticated

The application redirects Jordan to the organization's **identity provider**, a service that authenticates identities and issues information that other systems can rely on. Jordan completes the required authentication process.

A successful authentication establishes that the request is associated with Jordan's account to the level of confidence required by the organization. It does not yet establish that Jordan may use the application.

3. The device contributes independent evidence

The laptop presents a device identity and current posture information. The organization's device-management service reports that the laptop is enrolled, uses disk encryption, has an acceptable operating-system update level, and has a healthy endpoint-security agent.

This evidence is independent of Jordan's password or authenticator. A stolen user credential presented from an unknown device would produce a different decision context.

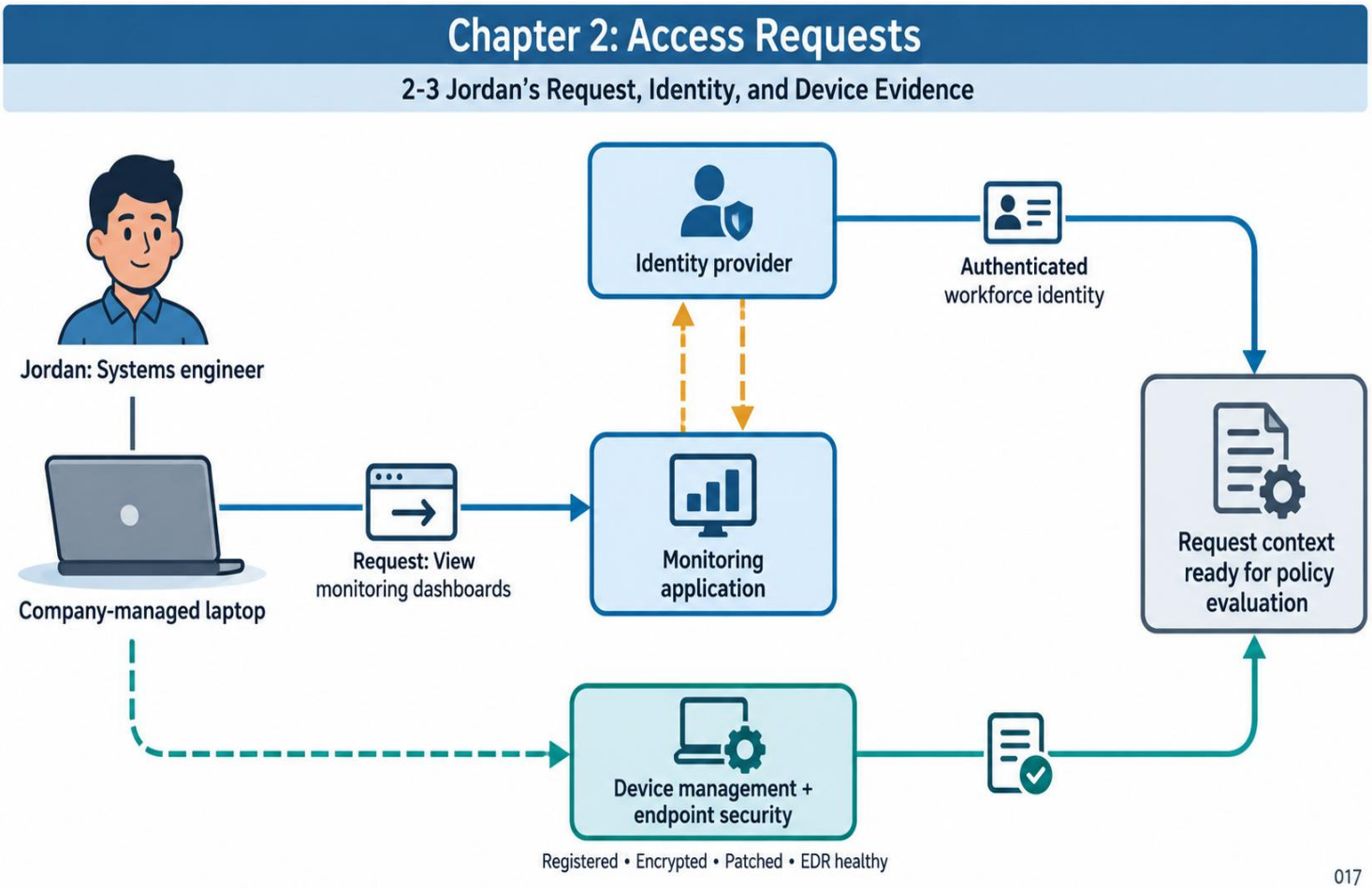


Figure 2-4: Jordan's request, identity authentication, and independent device evidence

4. The resource and action are matched to policy

The policy service identifies:

- Subject: Jordan's active workforce identity.
- Role: operations engineer.
- Device: managed and currently compliant.
- Resource: monitoring application.
- Action: view dashboards.
- Context: normal risk, permitted time, and required authentication strength.

The policy allows the request because all required conditions are present.

5. A limited session is established

An enforcement component establishes a session to the monitoring application. The session provides dashboard viewing but not production changes. The application and gateway record the decision and subsequent access events.

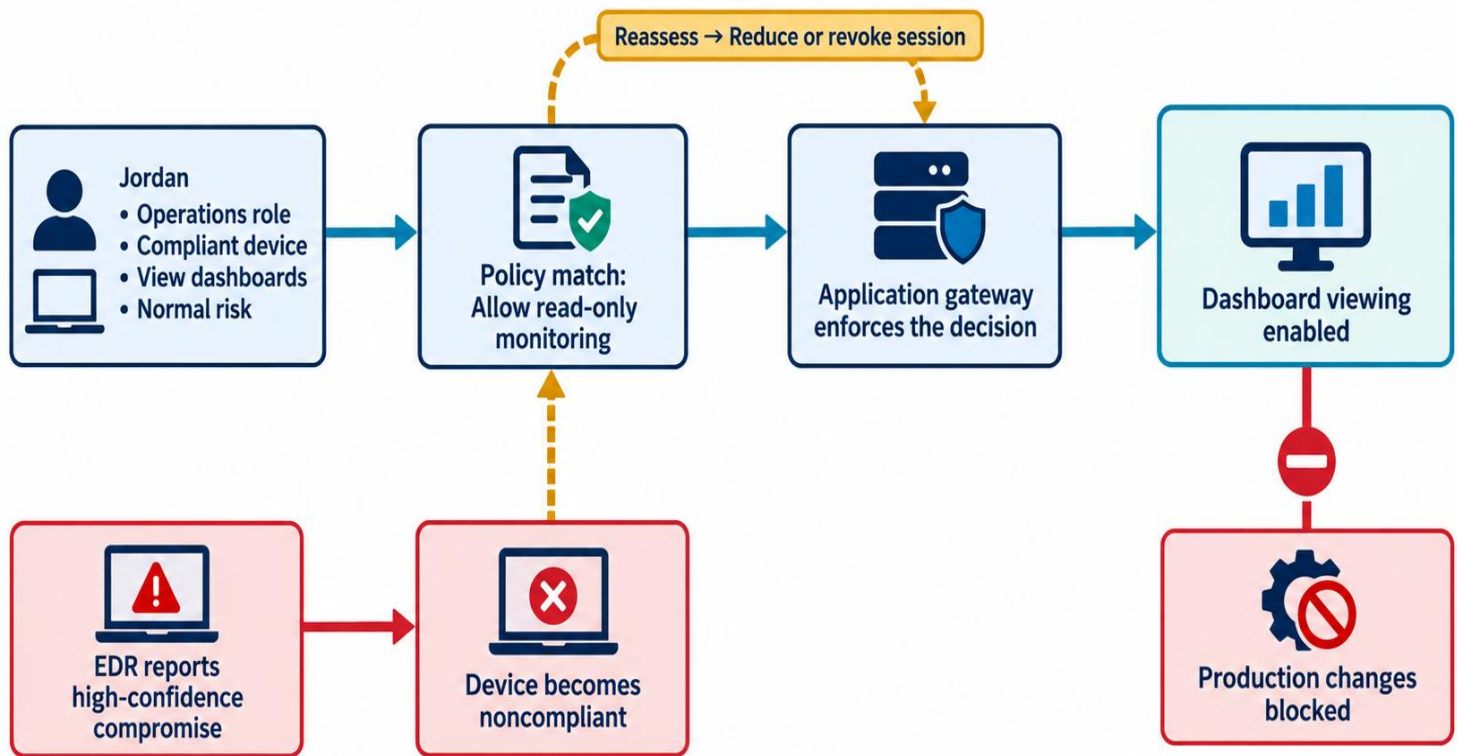
6. Changed evidence can change the outcome

Later, the endpoint-security system reports strong evidence that the laptop is compromised. The device is marked noncompliant. The application session can be revoked or reduced according to policy.

The important point is not the specific product mechanism. It is that **the original decision does not have to remain valid after its supporting evidence changes**.

Chapter 2: Access Requests

2-3 Policy Match, Limited Session, and Revocation



018

Figure 2-5: Policy match, limited session, and change-driven revocation

Login is not the same as resource access

A common design error is to treat a successful login as a universal authorization event.

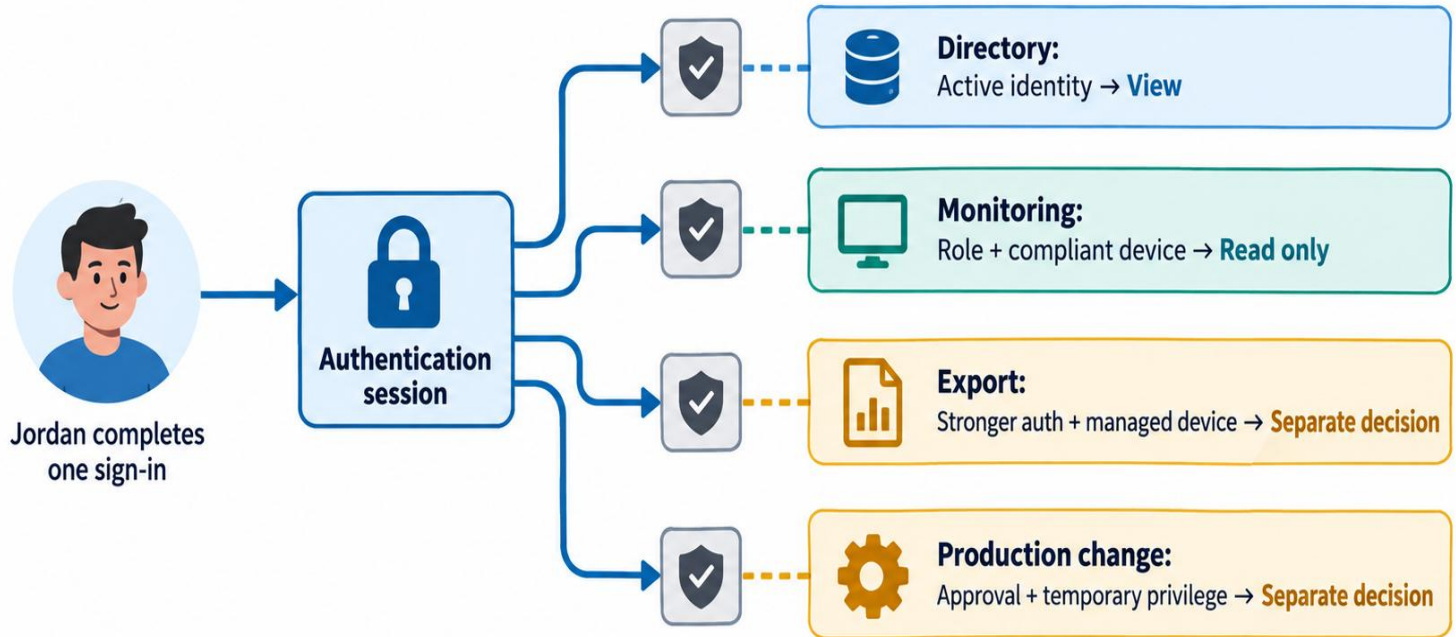
An initial sign-in can establish an identity and create an authentication session. The user may then request several resources, each with different policies:

- A staff directory may require only an active workforce account.
- The monitoring dashboard may require an operations role and a compliant device.
- Exporting operational data may require stronger authentication and a managed device.

- Changing production configuration may require temporary privilege and approval.

Chapter 2: Access Requests

2-4 One Login, Separate Resource Decisions



One sign-in does not replace resource authorization.

019

Figure 2-6: One authentication session feeding separate resource decisions

A single login can support these later decisions, but it should not replace them. **Authentication answers “Who is this?”** **Authorization answers “May this identity perform this action here and now?”**

Static and dynamic evidence

Some evidence changes slowly. Other evidence can change while a session is active.

Evidence type	Examples	Engineering implication
Relatively stable evidence	Department, employment status, assigned role, resource owner, data classification	Review it through lifecycle and governance processes; do not assume it is permanent.
Dynamic evidence	Device compliance, current risk, authentication strength, network path, active incident, time of day	Define freshness requirements and what happens when the value changes or becomes unavailable.

The distinction is useful but not absolute. A job role can change quickly during a transfer, and a device's patch level may remain stable for days. Policy should reflect the real rate of change and the cost of stale information.

More outcomes than allow or deny

A decision system can respond in several useful ways:

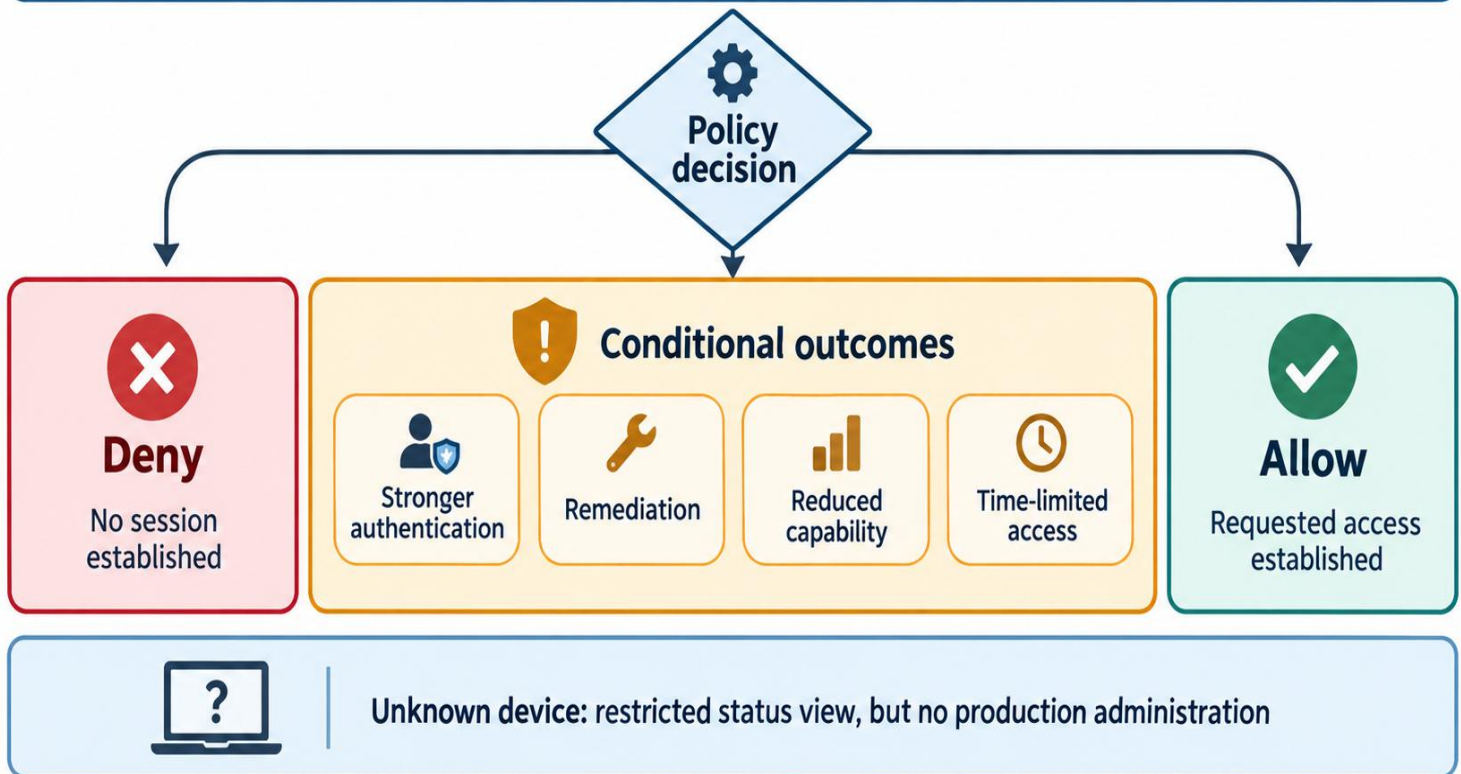
- **Allow:** establish the requested access.
- **Deny:** do not establish access.
- **Require stronger authentication:** ask for a higher-assurance authenticator before continuing.
- **Require remediation:** direct the user to update or repair the device.

- **Allow reduced capability:** permit viewing but block download, copy, or administration.
- **Grant time-limited access:** approve the action until a defined expiry time.



Chapter 2: Access Requests

2-6 More Outcomes Than Allow or Deny



020

Figure 2-7: Decision outcome spectrum

These outcomes help match controls to risk. An unknown device does not always require the same response. It may be denied access to production administration while receiving a restricted browser-only view of low-sensitivity information.

Continuous verification without constant interruption

Continuous verification means that access remains dependent on relevant evidence and policy after the initial decision. It does not mean asking the user to enter a password every few seconds.

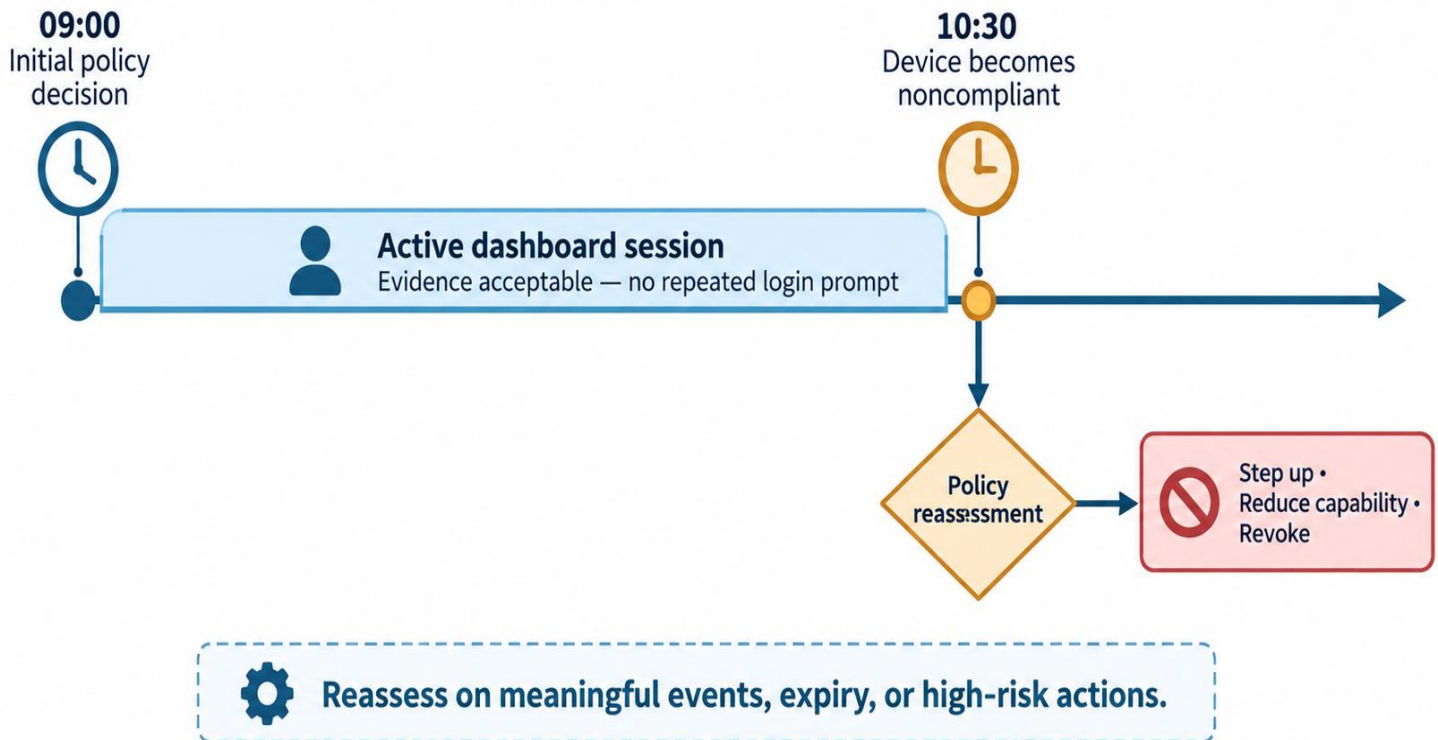
Reassessment can be event-driven. Examples include:

- The account is disabled.
- The device becomes noncompliant.
- A privileged approval expires.
- The session exceeds its maximum lifetime.

- A high-confidence security incident is associated with the identity or device.

Chapter 2: Access Requests

2-7 Event-Driven Continuous Verification



021

Figure 2-8: Session timeline with event-driven reassessment

The resulting action may be invisible to the user when risk is unchanged. When risk increases, the system may require stronger authentication, reduce capabilities, or revoke the session.

A practical tracing method

When designing or troubleshooting access, state the request in one sentence:

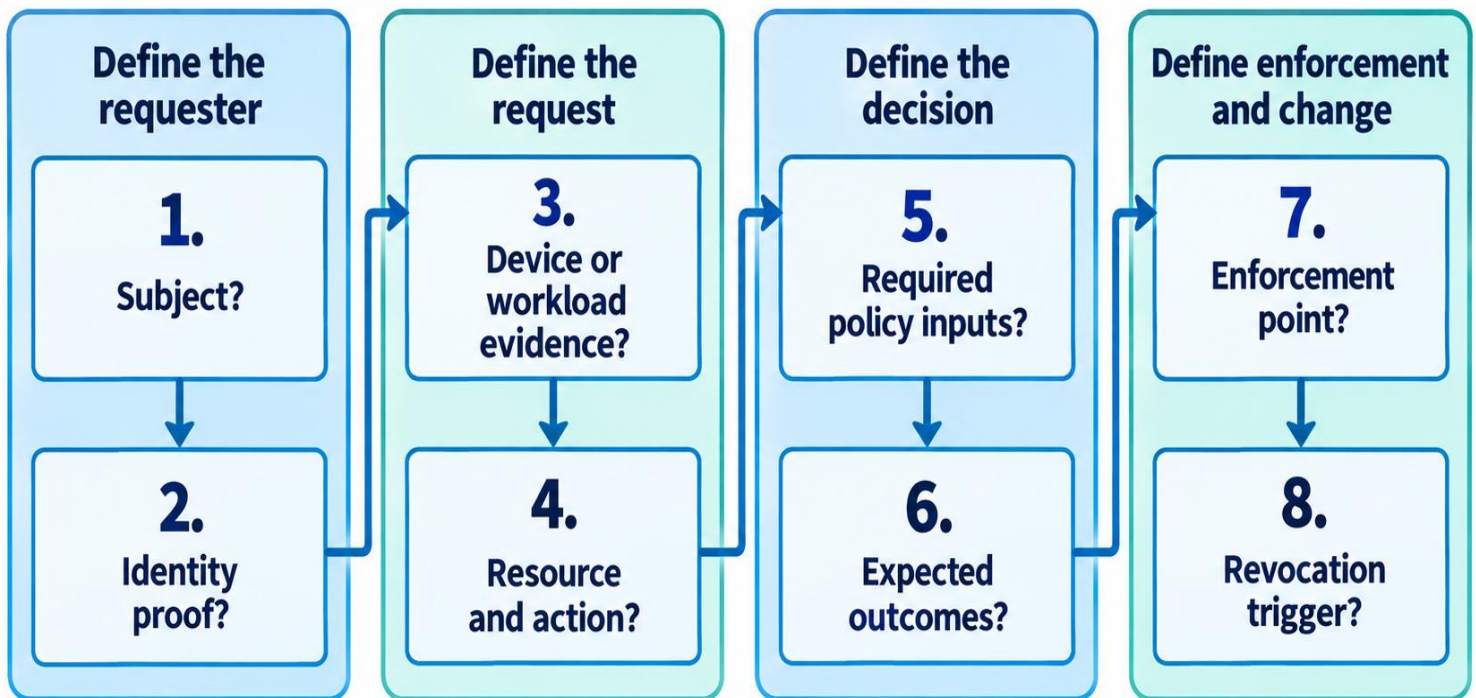
Jordan, using a currently compliant managed laptop, requests permission to view the monitoring dashboard during an active workforce session.

Then answer these questions in order:

1. **Who or what is the subject?**
2. **What proves the subject's identity?**
3. **Which device or workload is involved, and what evidence describes it?**
4. **What exact resource and action are requested?**
5. **Which policy inputs are required?**
6. **What outcome should each important case produce?**
7. **Where is the decision enforced?**

Chapter 2: Access Requests

2-8 Eight-Question Request Tracing Workflow



022

Figure 2-9: Eight-question request tracing workflow

If one answer is vague, the design is likely relying on implicit trust. “The user is internal,” “the laptop is corporate,” or “the server is behind the firewall” does not identify a precise authorization rule.

Two distinctions that prevent major mistakes

Authentication is not authorization. Proving that Jordan controls Jordan’s account does not prove that Jordan may change production systems.

Continuous verification is not continuous login prompting. It is the continued evaluation of evidence and policy, with user interaction only when a decision requires it.

The next chapter places this lifecycle into a logical architecture and separates the components that decide access from those that enforce it.

Chapter Summary

Every access request can be traced through the same decision anatomy. Authentication establishes an identity, authorization evaluates a specific action on a specific resource, enforcement creates the permitted path, and current evidence can later change the session.

Review Questions

1. Which facts must be identified before an access policy can evaluate a request?
2. Why can one successful sign-in lead to different decisions for different resources or actions?
3. Which changes during a session should cause reassessment, limitation, or revocation?