

Microsoft Intune A Visual Guide

A Practical Foundation for Device Enrollment Security and Operations

Reiner Braun

Sample Book — Chapters 1 to 2

Introduction

This book is designed for readers who are taking their first steps with Microsoft Intune, 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 on your first pass. The sequence builds the required background step by step, beginning with the Intune mental model and continuing through enrollment, configuration, security, operations, troubleshooting, automation, and migration.

At the beginning of each chapter, review **What You'll Be Able to Do**. These outcomes show what the chapter is intended to teach and help you distinguish foundational material from supporting detail.

Use the diagrams to understand relationships, processes, boundaries, and structures that are difficult to hold in mind from prose alone. Read the surrounding text as well; each diagram supports the explanation rather than replacing it.

Try the commands, scripts, and hands-on examples in a controlled test environment whenever practical. Reproducing a result yourself will make the operational sequence and the required evidence easier to remember.

Use each **Chapter Summary** and **Review Questions** section to confirm that you understand and can recall the fundamentals before moving forward. Return to the relevant section when an answer depends on guesswork rather than a clear explanation.

Advanced-topic overviews do not need to be memorized at this stage. Treat them as signposts for capabilities, dependencies, and operating concerns that may become important in later study or project work.

Table of Contents

Part I — Build the Intune Mental Model

Chapter 1 — What Intune Is and Where It Fits

- The endpoint-management problem
- MDM manages the device
- MAM protects work inside applications
- Ownership changes the design
- Identity, device, and application form one system
- The services around Intune
- What Intune does not replace
- A new employee with two devices
- Errors in the basic mental model
- Chapter Summary
- Review Questions

Chapter 2 — How Intune Works: Control Plane, Clients, and Lifecycle

- The tenant is the management boundary
- Administrative and user-facing components
- The operating system remains in control of the device
- The policy journey
- Intent, delivery, application, and evidence are different
- Why cloud state is not instantaneous
- Platform notification services prompt check-in
- The endpoint lifecycle
- Status is evidence, not an absolute truth
- Following one password policy
- The limited value of manual synchronization
- Chapter Summary
- Review Questions

Main Text

Microsoft Intune is a cloud service for turning endpoint-management requirements into repeatable controls for devices, applications, and organizational data. The essential skill is not memorizing every setting in the admin center. It is understanding the management chain: identify the user and device, choose the right enrollment and ownership model, target the correct population, deliver the intended configuration or application, and verify what actually happened.

This text builds that foundation from the service model through enrollment, configuration, connectivity, certificates, and application management, then extends it into compliance, endpoint security, updates, monitoring, troubleshooting, automation, migration, and production operations. Windows 11 is used as the main endpoint example, while Apple and Android are covered where their enrollment ecosystems require different decisions.

Part I — Build the Intune Mental Model

Chapter 1 — What Intune Is and Where It Fits

Intune is the management layer for endpoints and work applications; it is not the identity provider, the operating system, or the final access-control engine. A systems engineer should first understand that boundary. Most Intune projects become easier to design and troubleshoot when every responsibility is assigned to the correct service.

What You'll Be Able to Do

- Explain the management problems Intune is designed to solve.
- Distinguish MDM, MAM, and unified endpoint management.
- Describe how identity, devices, applications, and access controls work together.
- Choose a reasonable management boundary for corporate and personal devices.

The endpoint-management problem

An **endpoint** is a user-facing device that accesses organizational resources, such as a laptop, desktop, phone, tablet, kiosk, or handheld scanner. Endpoints create a practical engineering problem because they are numerous, mobile, and not always connected to the corporate network.

Traditional management often assumed that a computer was:

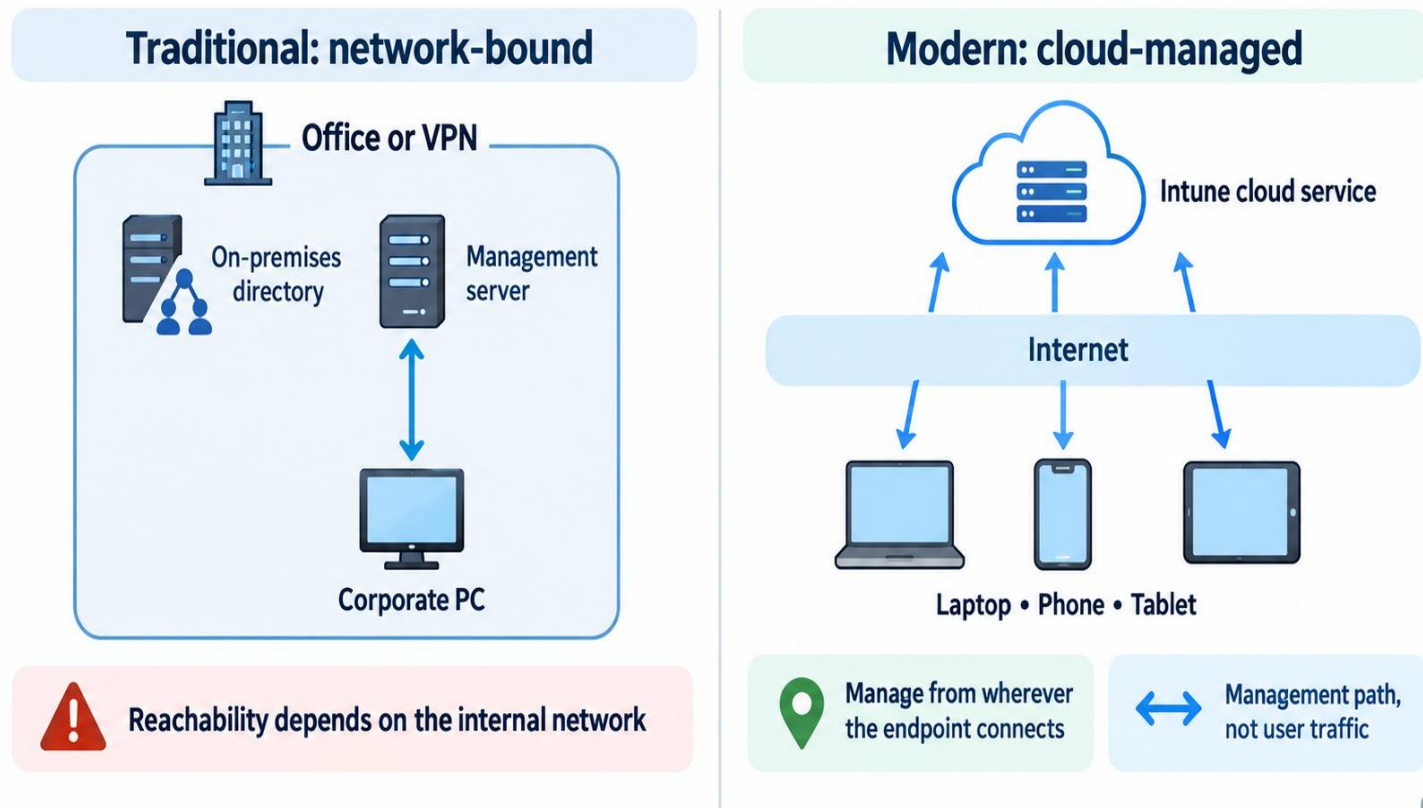
- Owned by the organization.
- Joined to an on-premises directory.
- Connected regularly to an office network or virtual private network.
- Managed with tools that could reach it directly.

Modern work breaks those assumptions. A laptop might be shipped directly to an employee, a phone might be personally owned, and both might connect only to cloud services. Intune addresses this model by delivering management from the cloud rather than requiring every endpoint to be on an internal network.

A useful analogy is a building-management system. The system can issue access badges, define which rooms are available, check whether safety rules are being followed, and revoke access when a person leaves. It does not manufacture the doors, employ the people, or replace the fire department. In the same way, Intune coordinates management but relies on identity, operating-system, networking, and security services around it.

Chapter 1: What Intune Is and Where It Fits

1-01 From Network-Bound Management to Cloud-Managed Endpoints



001

Figure 1-1: From Network-Bound Management to Cloud-Managed Endpoints.

MDM manages the device

Mobile device management (MDM) is the process of enrolling a device so that the organization can configure and manage the device itself. Despite the word *mobile*, MDM also applies to Windows and macOS computers.

After MDM enrollment, Intune can perform platform-supported tasks such as:

- Configure operating-system settings.
- Deploy applications.
- Install certificates and network profiles.
- Collect hardware and software inventory.
- Evaluate device state.
- Perform remote actions, including removing organizational management or wiping a device when appropriate.

The exact controls depend on the platform, operating-system version, enrollment method, ownership, and license. Intune cannot bypass the management framework supplied by Windows, Apple, or Android. If a platform does not expose a setting to MDM, Intune cannot reliably invent that capability.

MDM is normally the strongest fit for an organization-owned device. The organization controls the hardware, defines its acceptable use, and needs enough authority to configure and retire it safely.

MAM protects work inside applications

Mobile application management (MAM) protects organizational data inside supported applications. The device itself does not always need to be enrolled.

For example, an employee might use Outlook and Teams on a personal phone. An app-protection policy can require an application PIN, restrict copying work data into personal applications, and remove work data when the user leaves. Personal photos, messages, and unrelated applications remain outside the management boundary.

This creates a clear distinction:

Management model	Main object being managed	Typical use	Administrative reach
MDM	The device, its settings, and managed applications	Corporate laptop or corporate phone	Broad, within platform and enrollment limits
MAM	Supported work applications and their organizational data	Personal phone used for email and collaboration	Limited to the managed application boundary
MDM and MAM together	The device plus additional controls inside supported applications	Corporate phone handling sensitive data	Device-level control plus app-level data protection

MDM and MAM are complementary, not interchangeable. MDM can configure the device, while MAM controls how work data behaves inside supported applications. A project can use either one or both.

Unified endpoint management (UEM) is the broader practice of managing multiple endpoint platforms and ownership models through a common operating approach. Intune supports UEM by bringing device, application, and reporting workflows together while still respecting the different capabilities of Windows, Apple, Android, and other supported platforms.

Ownership changes the design

Device ownership is not merely an inventory label. It affects privacy, enrollment, available controls, and retirement behavior.

An **organization-owned device** is purchased or otherwise controlled by the organization. It is reasonable to apply stronger configuration, collect broader inventory, and perform a full reset when the device is reassigned or retired.

A **personally owned device**, often called a bring-your-own-device or **BYOD** device, belongs to the user. The organization should manage only what is necessary to protect work. The design must make the boundary visible and understandable to the user.

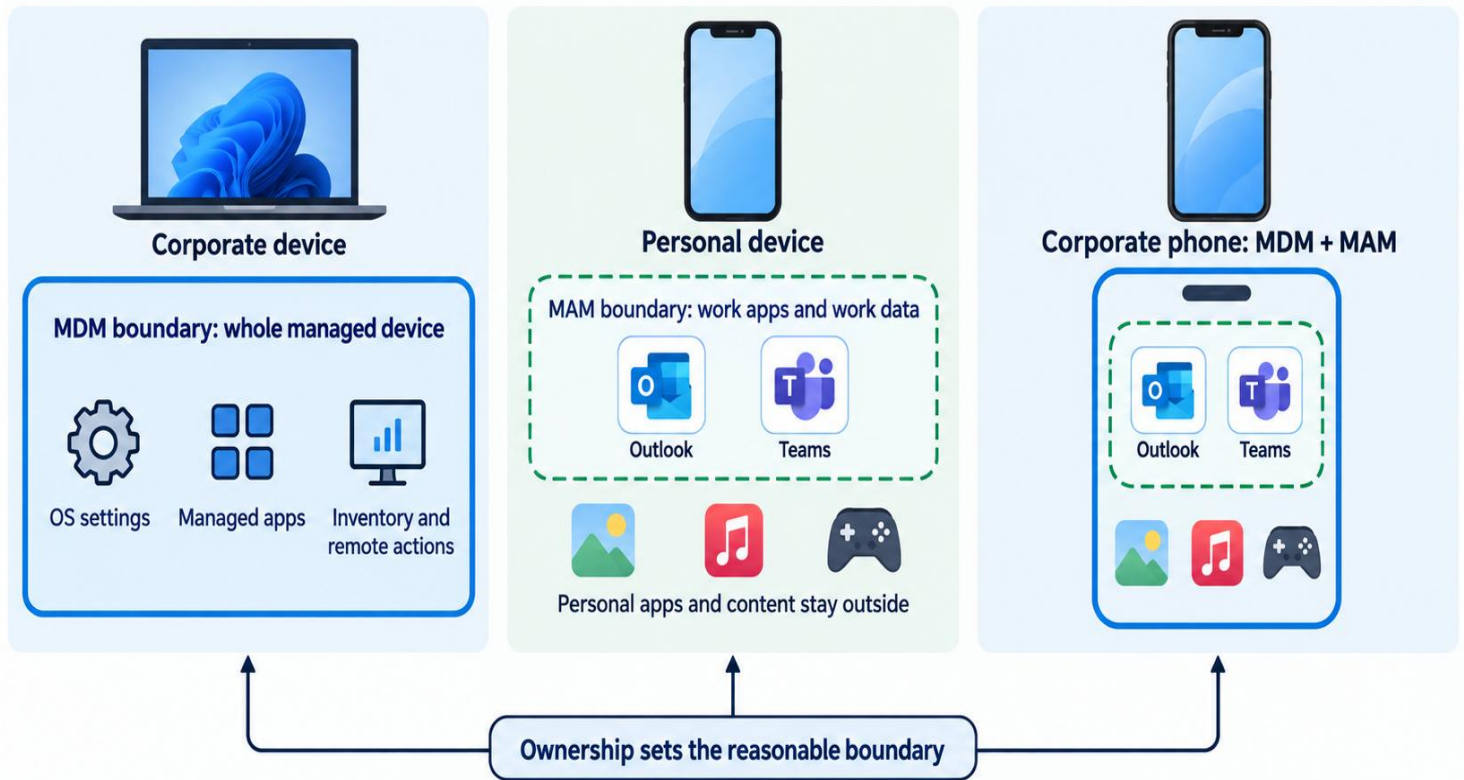
Consider two phones with the same model and operating system:

- A corporate phone may be enrolled with an automated, supervised or fully managed method. The organization can configure the whole work experience.
- A personal phone may use MAM only, or a privacy-oriented enrollment method that separates work from personal data.

Using the same control model for both devices can create either a security gap or an unnecessary privacy intrusion.

Chapter 1: What Intune Is and Where It Fits

1-02-1-04 MDM, MAM, and Ownership Control Boundaries



002

Figure 1-2: MDM, MAM, and Ownership Control Boundaries.

Identity, device, and application form one system

Intune works around three connected objects:

1. **Identity** answers who the user or administrator is.
2. **Device** answers what endpoint is being used and what its managed state is.
3. **Application** answers which software is accessing organizational data and how that data is handled.

Microsoft Entra ID is Microsoft's cloud identity and access-management service. It provides users, groups, authentication, device identities, and the identity context used by Intune.

Microsoft Entra Conditional Access evaluates sign-in context and applies access requirements. Intune can report device or application-management information, but Conditional Access is commonly the component that allows or blocks access to resources.

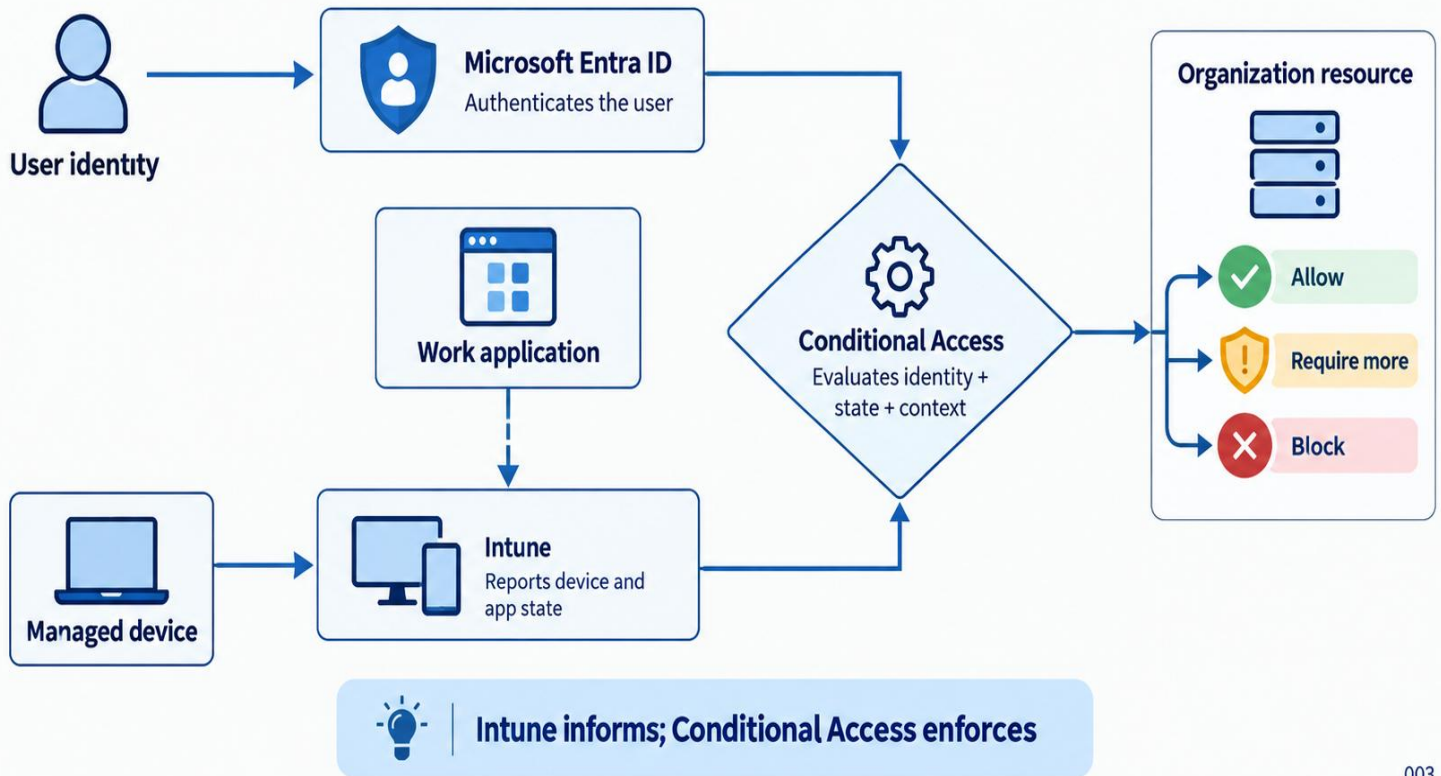
A simplified access flow looks like this:

1. A user signs in with a Microsoft Entra identity.
2. The device reports managed state to Intune.
3. Intune makes relevant device or application state available to Microsoft Entra.
4. Conditional Access evaluates the user, device, application, location, and other signals.
5. Access is allowed, blocked, or made subject to an additional requirement.

This separation matters. **Intune supplies management state; it does not authenticate the user or independently make every access decision.**

Chapter 1: What Intune Is and Where It Fits

1-05 Identity, Device, Application, and Access as One System



003

Figure 1-3: Identity, Device, Application, and Access as One System.

The services around Intune

Intune is usually part of a larger endpoint-management architecture.

Microsoft Defender for Endpoint provides endpoint detection and response, threat information, and device-risk signals. Intune can use integrations with Defender, but Defender remains a separate security service with its own administration and operating model.

Microsoft Configuration Manager is an on-premises management product used primarily for Windows clients and servers. Organizations can connect it with Intune through co-management or related cloud-attachment capabilities. This relationship is useful during migration, but Intune is not simply Configuration Manager hosted in the cloud.

Windows Autopilot and **Windows Autopilot device preparation** support cloud-based Windows provisioning. They help associate a new Windows device with the organization and guide it through setup, while Intune supplies the policies and applications used to make it work-ready.

Windows Autopatch is a managed update service for eligible Windows environments. It works with Intune but is not the same service as Intune.

Apple management depends on Apple services. **Apple Business Manager** connects organization-owned Apple devices and app licenses to management. The **Apple Push Notification service**, commonly abbreviated as APNs, provides the notification channel that prompts Apple devices to contact their MDM service.

Android Enterprise management depends on Google services. **Managed Google Play** is the managed application catalog and service connection used to approve, configure, and distribute Android Enterprise applications.

These dependencies explain why endpoint management often crosses team boundaries. Identity, networking, security, application packaging, procurement, and platform administration all contribute to a successful Intune service.

What Intune does not replace

A correct mental model also includes explicit limits.

Intune is not a backup product. A device wipe or application removal cannot be reversed by Intune, and user data must be protected through appropriate storage and backup services.

Intune is not a general-purpose remote desktop tool. Some remote-help capabilities can integrate with the environment, but routine interactive remote control is a separate function.

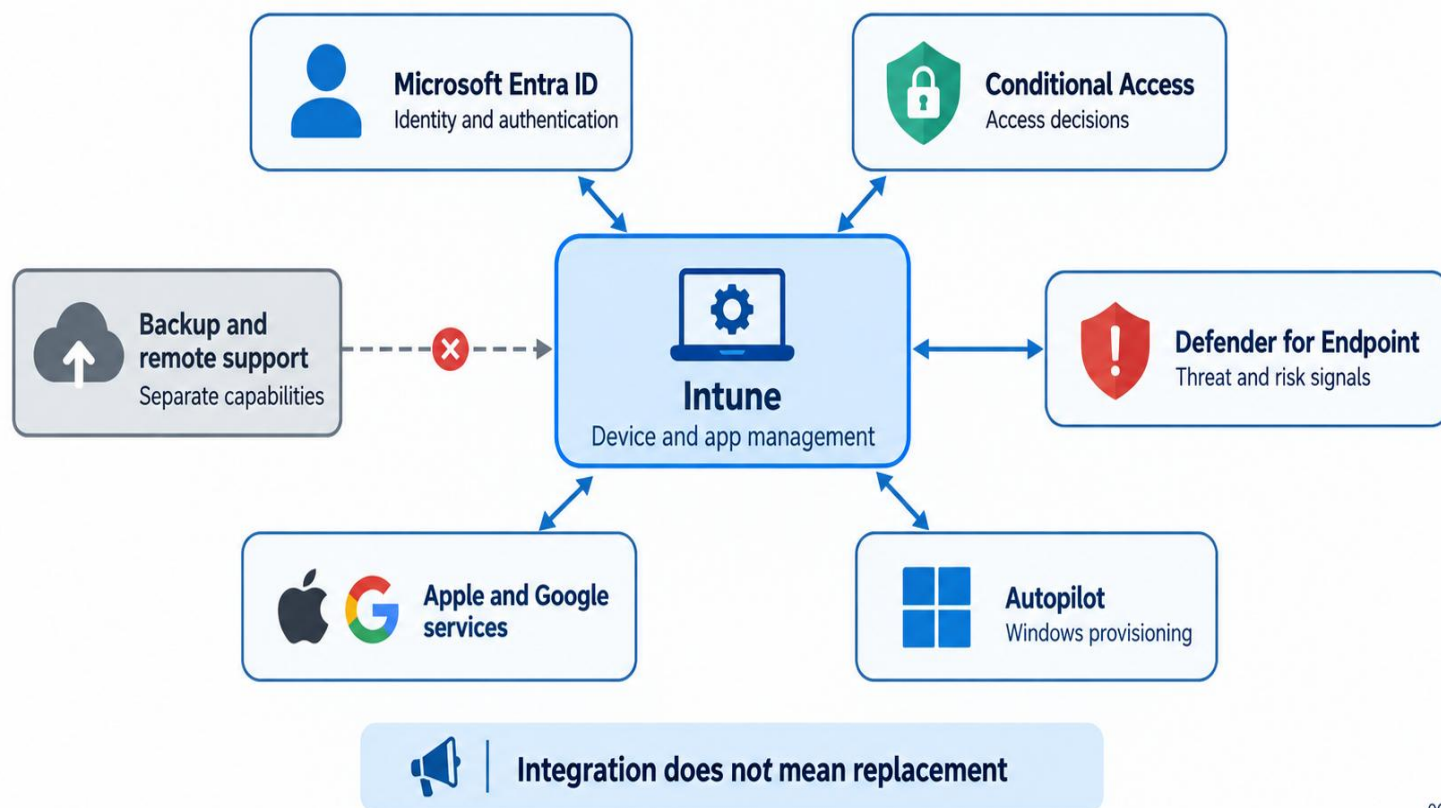
Intune is not a complete identity system. It relies on Microsoft Entra ID for users, groups, authentication, and device identity.

Intune is not a universal security product. It configures many security controls and exchanges signals with security services, but it does not replace every antivirus, detection, incident-response, data-governance, or network-security capability.

Intune is also not a guarantee of a secure endpoint. Enrollment creates a management relationship. Security depends on the policies selected, whether they are applicable, whether they process successfully, and how access is enforced.

Chapter 1: What Intune Is and Where It Fits

1-06–1-07 Intune Ecosystem and Explicit Service Boundaries



004

Figure 1-4: Intune Ecosystem and Explicit Service Boundaries.

A new employee with two devices

A practical example shows how the pieces fit together.

A new employee receives a corporate Windows 11 laptop and wants to read work email on a personal phone.

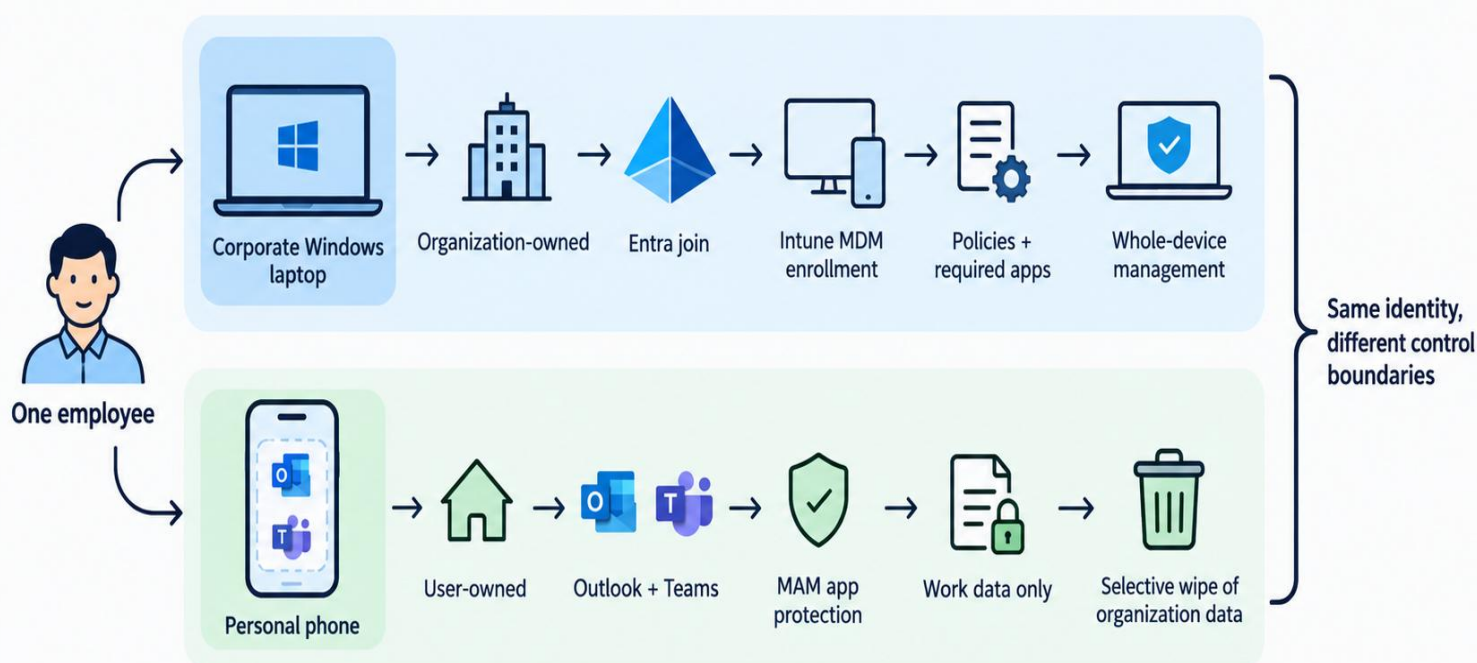
The laptop is organization-owned. During setup, the user signs in with a Microsoft Entra identity. The laptop joins Microsoft Entra ID, enrolls in Intune, receives configuration profiles and required applications, and later reports its state. The organization can manage the whole device because it owns the hardware.

The phone is personal. Instead of enrolling the entire phone, the organization allows Outlook and Teams under MAM. Work data is kept inside managed applications, and the organization can remove that data without resetting the phone.

Both devices use the same user identity, but their management boundaries differ. The laptop is managed as a device. The phone's work applications are managed as protected containers. Conditional Access can apply different requirements to each scenario.

Chapter 1: What Intune Is and Where It Fits

1-08 One Employee, a Corporate Laptop, and a Personal Phone



005

Figure 1-5: One Employee, a Corporate Laptop, and a Personal Phone.

Errors in the basic mental model

Three misconceptions cause many early design problems.

Enrollment is not the same as security. Enrollment establishes management. Security requires relevant configuration, successful processing, monitoring, and access enforcement.

MDM and MAM are not two names for the same control. MDM manages a device; MAM manages work data inside supported applications.

Intune is not the final access gate. In a common Microsoft architecture, Intune provides state and Microsoft Entra Conditional Access enforces access decisions.

A systems engineer who keeps these boundaries clear can reason about most later Intune features without treating the admin center as a collection of unrelated menus.

Chapter Summary

- Intune manages endpoints, supported applications, and organizational data through platform-supported controls.
- MDM manages the device; MAM protects work data inside compatible applications; the two can be used separately or together.
- Microsoft Entra ID, Conditional Access, security services, and platform ecosystems retain distinct responsibilities around Intune.

Review Questions

1. What is the practical difference between MDM and MAM?
2. Why should a personally owned phone usually have a narrower management boundary than a corporate phone?
3. Which service commonly authenticates the user, and which service commonly enforces an access decision?
4. Name one capability that Intune does not replace and explain why.

Chapter 2 — How Intune Works: Control Plane, Clients, and Lifecycle

Intune operates as a cloud control plane: administrators define desired outcomes, endpoints receive applicable instructions through platform management channels, and status returns as evidence. Saving a policy creates intent. It does not prove that a device received, understood, applied, or retained that intent.

What You'll Be Able to Do

- Describe the Intune tenant, control plane, user-facing tools, and endpoint components.
- Trace a policy from creation and assignment through client processing and reporting.
- Explain why cloud management state is eventually consistent rather than instantaneous.
- Interpret reported status as time-stamped evidence instead of absolute real-time truth.

The tenant is the management boundary

A **tenant** is an organization's isolated cloud-service boundary for identities, devices, applications, policies, and administrative configuration. A tenant is not a server that the organization owns. It is a logical service boundary operated in Microsoft's cloud.

Within the tenant, an administrator creates objects such as:

- Configuration profiles.
- Enrollment profiles.
- Applications.
- Assignment filters.
- Administrative role assignments.

These objects are stored in the cloud control plane and associated with Microsoft Entra users or groups. Endpoints contact the service over the internet to discover work that applies to them.

Administrative and user-facing components

The main components have distinct roles.

Component	Main purpose
Microsoft Intune admin center	Web console for creating policies, assigning applications, viewing status, and initiating device actions
Microsoft Graph	Programmatic interface used by the admin center and available for automation
Company Portal	User-facing application and website for enrollment, available applications, device status, and support information
Operating-system MDM channel	Built-in platform mechanism that processes supported management commands
Intune Management Extension	Windows agent that provides capabilities beyond the built-in MDM channel, including Win32 application delivery and selected scripting functions

Microsoft Graph is Microsoft's API surface for accessing cloud-service data and operations. An API, or application programming interface, allows software to perform operations that a person might otherwise perform through a user interface.

The admin center is therefore not a separate management engine. It is a human-friendly front end to service operations. This becomes important later when an engineer needs repeatable exports, reports, or automation.

The **Company Portal** is not the administrative console. It is the user-facing surface. Depending on platform and enrollment method, it can guide enrollment, show available applications, display device-management information, initiate a sync, and present support contacts.

The operating system remains in control of the device

Intune delivers policy through management frameworks provided by each operating system.

Windows includes an MDM client and a set of **configuration service providers (CSPs)**, which are operating-system interfaces for reading or changing managed settings. Apple devices use Apple's MDM protocols and platform services. Android Enterprise uses Android's enterprise-management framework and Google services where applicable.

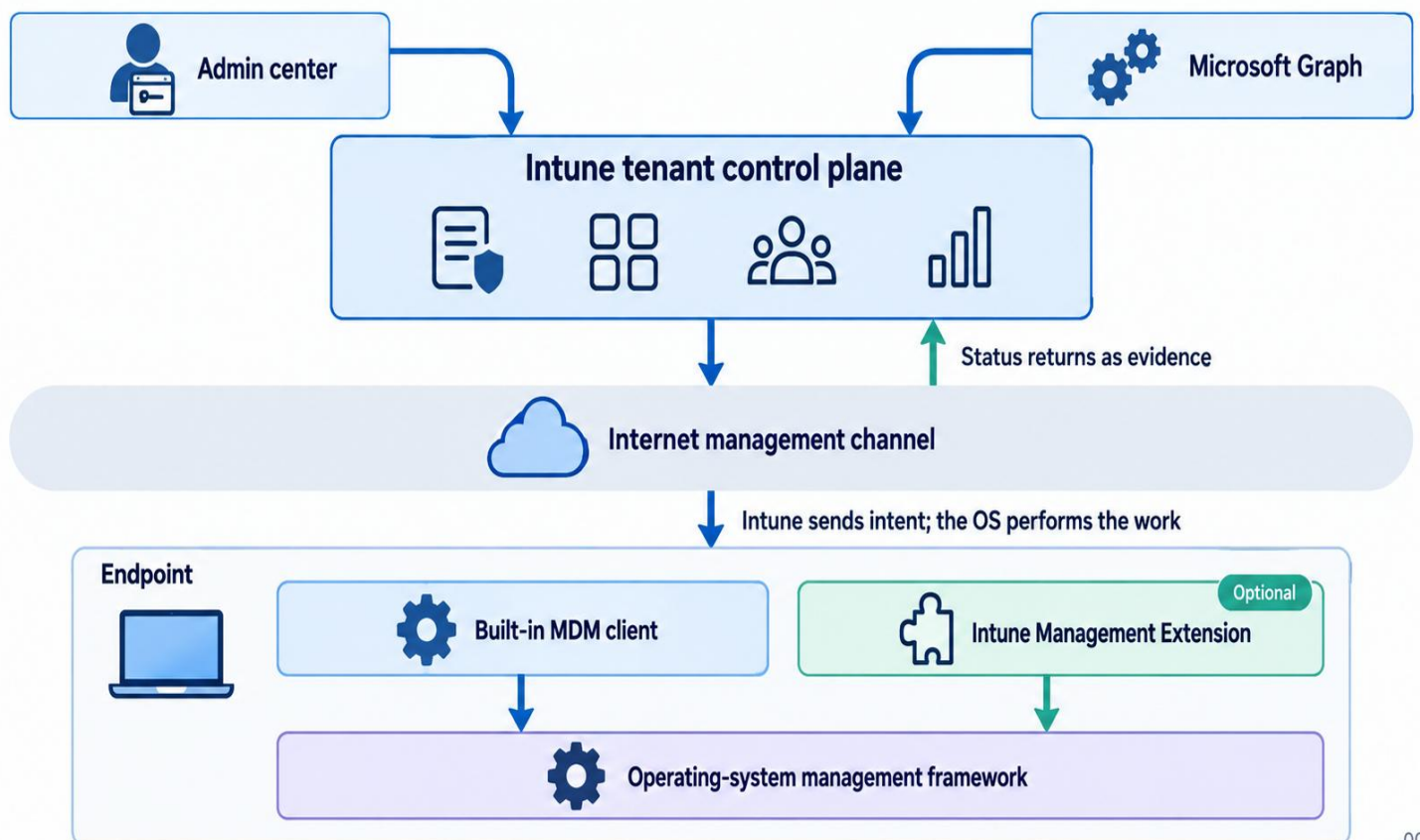
This means that the device is not a passive remote-control target. It receives a management command, evaluates whether the command is valid and applicable, applies it through the operating system, and reports a result.

On Windows, the built-in MDM channel does not cover every practical management task. The **Intune Management Extension (IME)** is an additional Windows agent used for capabilities such as Win32 application deployment and selected scripts. A device can therefore process some work through the built-in MDM client and other work through IME.

A useful analogy is a delivery company. Intune creates and addresses the package, but the local operating system is the receiving office. The office checks the address, decides which department should handle the package, performs the work, and sends a receipt. A delivered package can still fail because the contents were invalid or the receiving department could not use them.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-01–2-03 Tenant Control Plane, Clients, and Operating-System Authority



006

Figure 2-1: Tenant Control Plane, Management Clients, and Operating-System Authority.

The policy journey

A management object normally passes through several stages.

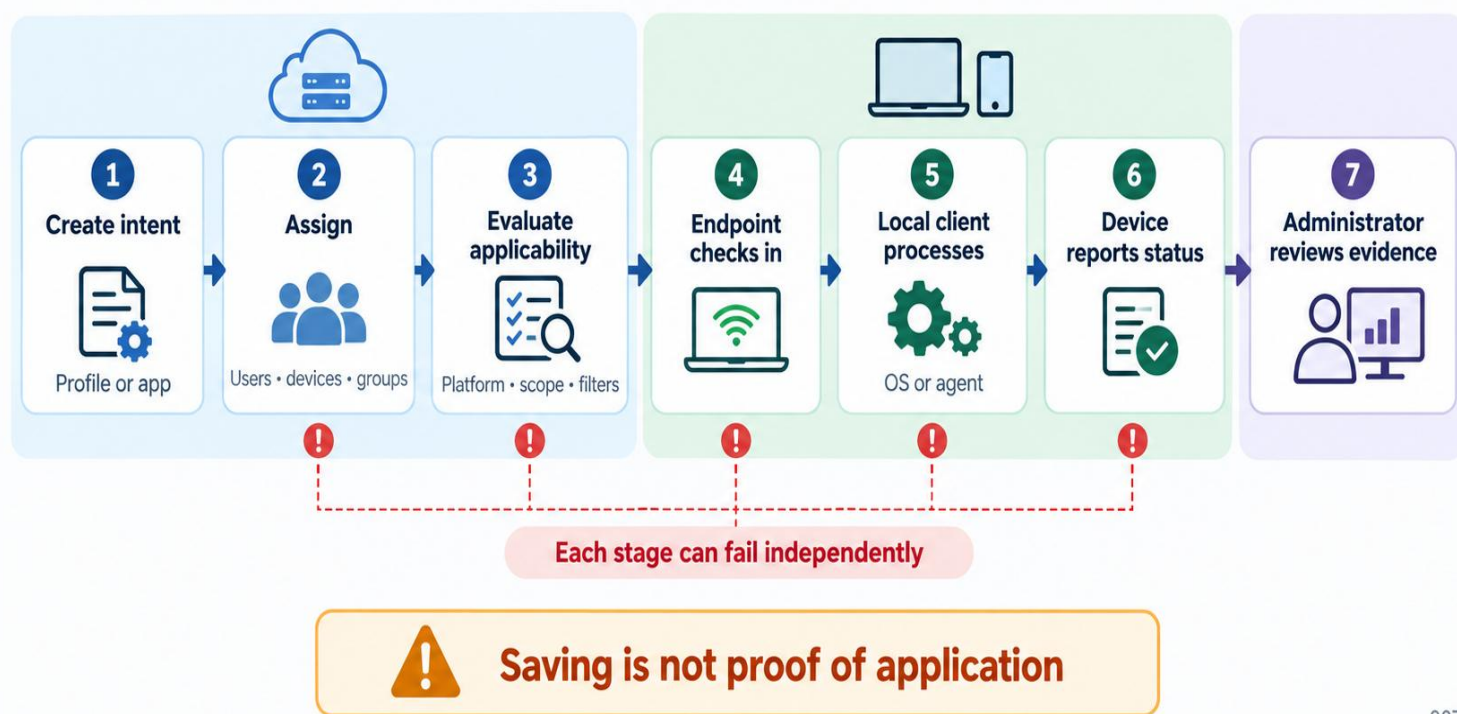
1. **An administrator creates intent.** A profile, application, or other object defines what should happen.
2. **The object is assigned.** The administrator targets users, devices, or groups and may apply filters or exclusions.
3. **The service evaluates applicability.** Intune determines whether the assignment and platform conditions match the endpoint.
4. **The endpoint checks in.** The device contacts the service on its normal schedule or after a supported sync request.

5. **The local client processes the instruction.** The operating-system MDM client, Company Portal, IME, or another platform component performs the work.
6. **The endpoint reports status.** Results return to Intune and become visible in reports.
7. **The service presents evidence.** Administrators review per-policy, per-setting, per-user, or per-device status.

Each stage can succeed or fail independently. A policy can be valid but targeted incorrectly. It can be targeted correctly but not applicable to the operating-system edition. It can reach the device but conflict with another control. It can apply locally but take time to appear in cloud reporting.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-04 The End-to-End Policy Journey



007

Figure 2-2: The End-to-End Policy Journey.

Intent, delivery, application, and evidence are different

Systems engineers should separate four questions:

What did we intend? This is the configuration stored in Intune.

Who or what was targeted? This is the effective assignment after group membership, exclusions, and filters are considered.

What did the endpoint process? This is the local result from the operating-system management client or agent.

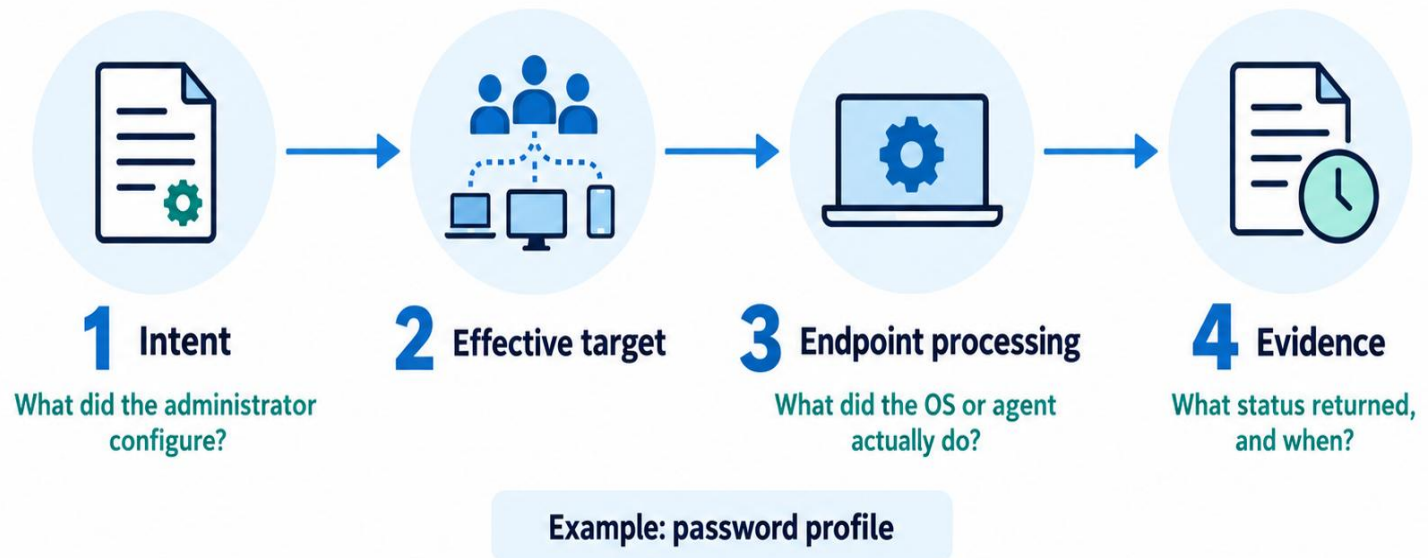
What evidence has returned? This is the status visible in the service at a particular time.

The distinction prevents a common reasoning error: treating the presence of an object in the admin center as proof of endpoint state.

For example, an administrator creates a Windows password setting and selects **Save**. At that point, only the cloud object exists. The device still needs to be in scope, check in, support the setting, process it in the correct context, and report the result.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-05 Intent, Target, Endpoint Processing, and Evidence



Saved ≠ targeted ≠ applied ≠ reported

008

Figure 2-3: Intent, Target, Endpoint Processing, and Evidence.

Why cloud state is not instantaneous

Eventual consistency means that changes made in one part of a distributed cloud system can take time to appear in all other parts. Intune depends on group evaluation, service processing, network connectivity, device check-in, local operating-system work, and status upload.

A short delay can therefore be normal. The correct response is not to assume either immediate success or immediate failure.

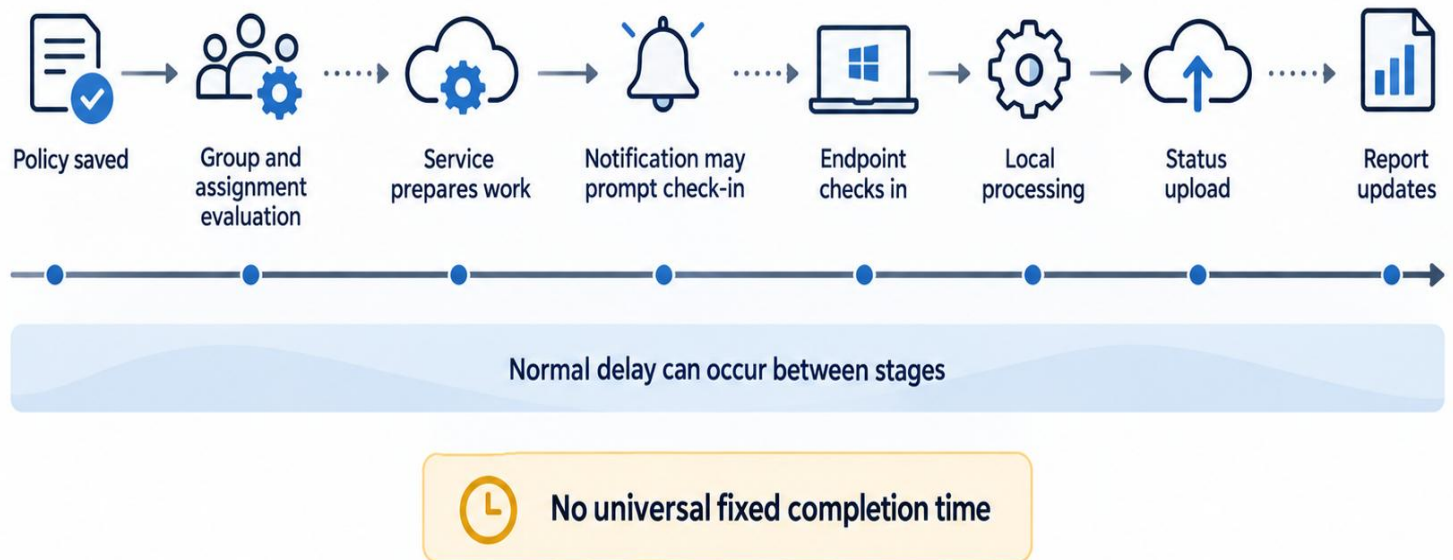
An engineer should ask:

- Was the assignment created and saved?
- Is the user or device actually in scope?
- Has group membership finished processing?
- Is the setting applicable to the platform and enrollment type?
- Has the device checked in since the change?
- Is local processing waiting on another prerequisite?
- Has the reporting service received the latest result?

This sequence is more useful than repeatedly selecting sync without identifying the stage at which progress stopped.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-06 Why Cloud Management State Is Not Instantaneous



009

Figure 2-4: Why Cloud Management State Is Not Instantaneous.

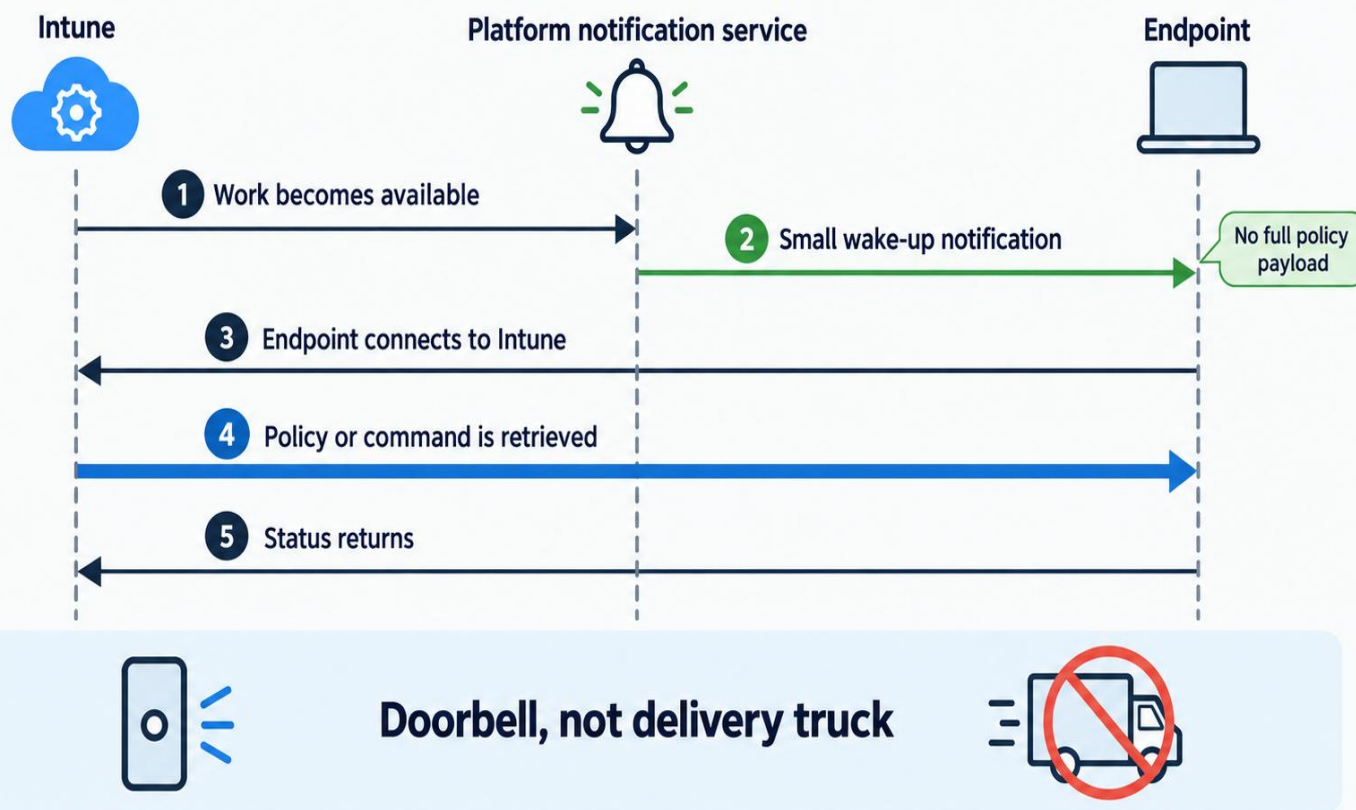
Platform notification services prompt check-in

Apple and Android ecosystems use platform notification services as part of management communication. A push notification generally tells a device that work is available; the device then contacts the management service through its supported channel.

The notification does not normally contain the full policy. It is more like a doorbell than a delivery truck. If the device cannot reach the management service, if a required certificate has expired, or if the platform connection is misconfigured, ringing the doorbell does not complete the work.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-07 Notification Services Prompt Check-In Rather Than Carry Full Policy



010

Figure 2-5: Notification Services Prompt Check-In Rather Than Carry Full Policy.

The endpoint lifecycle

Intune should be designed as a lifecycle service rather than an enrollment project.

Plan. Define ownership, user scenarios, platform support, identity dependencies, applications, and success criteria.

Enroll. Establish the management relationship using the correct platform method.

Configure. Apply settings, resource-access profiles, and application configuration.

Secure. Apply appropriate controls and produce state that can support access decisions.

Operate. Maintain applications, policies, certificates, tokens, and user support.

Update. Keep the operating system and applications within supported versions.

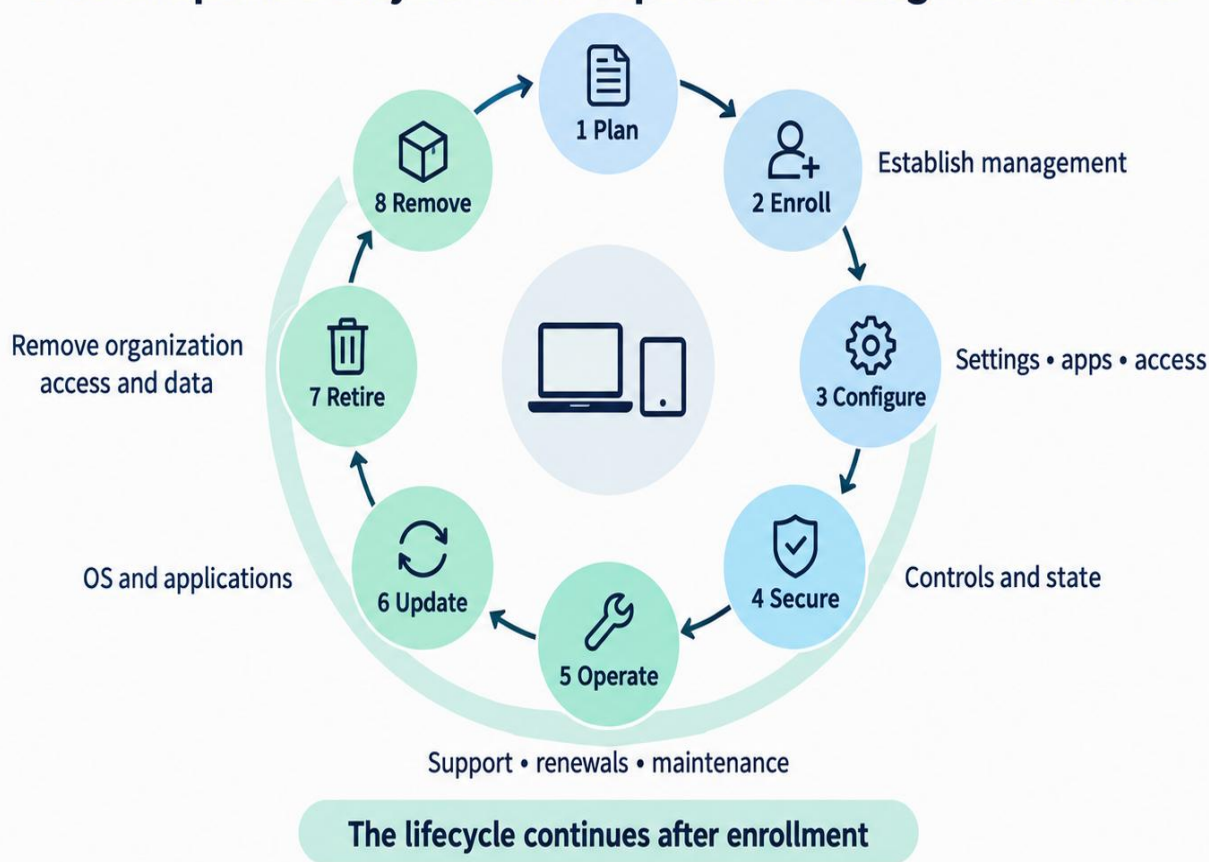
Retire. Remove organizational access and data in a way appropriate to ownership.

Remove. Clean up stale records and release the device from connected enrollment systems when the hardware leaves service.

Skipping the final stages creates common operational debt. A device record can remain in Intune, Microsoft Entra ID, Windows Autopilot, Apple Business Manager, or another platform service even after the physical device has been replaced. Each record represents a different relationship and may require a different cleanup action.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-08 Endpoint Lifecycle from Preparation Through Retirement



011

Figure 2-6: Endpoint Lifecycle from Preparation Through Retirement.

Status is evidence, not an absolute truth

A status value is an observation produced by a particular component at a particular time. It should be interpreted in context.

A device can report **Succeeded** for a configuration profile while the application that depends on that setting still fails for an unrelated reason. A device can appear **Pending** because it has not checked in. A setting can be **Not applicable** because the operating system, edition, enrollment mode, or user/device context does not support it. A **Conflict** indicates competing managed values, not necessarily which value finally controls the local behavior.

The engineer's task is to connect service evidence with endpoint evidence. Later operational work often uses both cloud reports and local logs, but the foundational point is simple: **the portal shows reported management state, not omniscient real-time truth.**

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-09 Reported Status as Time-Stamped Evidence

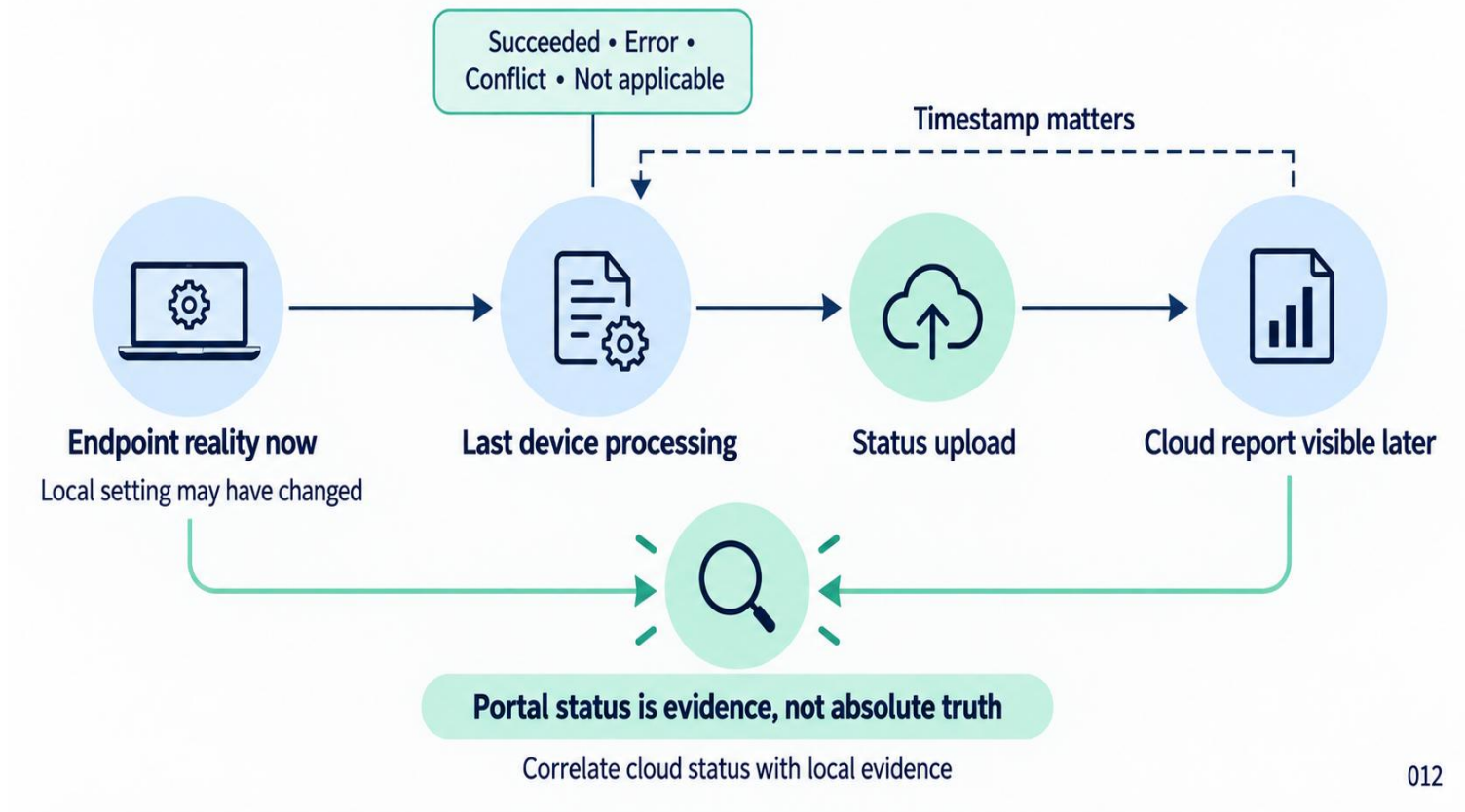


Figure 2-7: Reported Status as Time-Stamped Evidence.

Following one password policy

Assume a team creates a Windows configuration profile that requires a minimum password length for a pilot group.

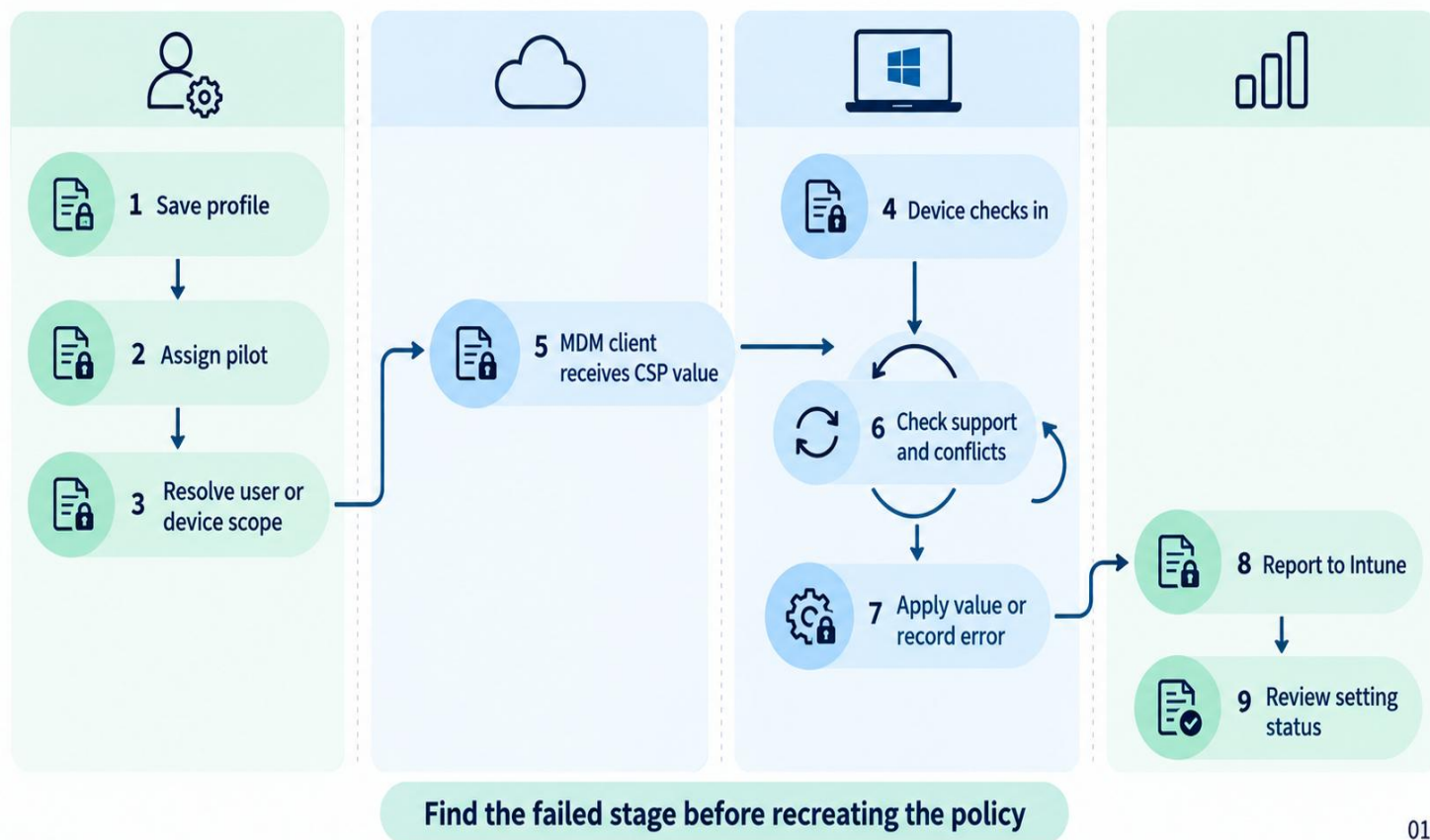
The sequence is:

1. The profile is saved in the tenant.
2. The pilot group is assigned.
3. Intune evaluates whether a device or its signed-in user is in scope.
4. The Windows device checks in.
5. The Windows MDM client receives the setting through the appropriate CSP.
6. Windows evaluates whether the setting is supported and whether another authority conflicts with it.
7. The device applies the value or records a local error.
8. The result is reported to Intune.
9. The administrator reviews both the overall profile and the individual setting status.

If the device remains unchanged, the team should not begin by recreating the policy. It should identify the failed stage. A wrong group, unsupported edition, conflicting policy, or stale check-in each requires a different correction.

Chapter 2: How Intune Works: Control Plane, Clients, and Lifecycle

2-10 Password Policy: End-to-End Path



013

Figure 2-8: Tracing One Password Policy Through the System.

The limited value of manual synchronization

A manual sync requests that the endpoint contact Intune sooner than it otherwise might. This can be useful after a known-good assignment change or while validating a pilot.

A sync does not repair:

- An invalid group assignment.
- An unsupported setting.
- A missing license or platform prerequisite.
- A broken installer command.
- An expired Apple certificate.
- A policy conflict.

Synchronization moves valid work through the pipeline; it does not make invalid work valid. That principle is central to efficient Intune troubleshooting.

Chapter Summary

- Intune expresses management intent in the cloud, while the operating system and local agents process supported instructions on the endpoint.
- Assignment, applicability, check-in, local processing, and reporting are separate stages that can succeed or fail independently.

- Synchronization can request work, but it cannot repair an incorrect target, unsupported setting, or failed local prerequisite.

Review Questions

1. What is the difference between the Intune control plane and the endpoint management client?
2. What stages occur between assigning a policy and seeing its result in a report?
3. Why can a correct policy take time to appear on a device?
4. What does a successful manual synchronization prove, and what does it not prove?