Azure Governance Approach

Audience: Primary: cloud platform team leads and enterprise architects; Secondary: security/compliance, finance (FinOps), and app teams; intermediate Azure maturity. **Job ID:** 69d5474c-53ba-4278-957a-3114cfaf614c **Generated:** 2026-02-03

Table of Contents

1. Executive Summary and Governance Principles

1.1 Purpose

You align stakeholders on **why** you govern Azure, **what outcomes** you expect, and **how** you will balance speed with risk, cost discipline, and operational sustainability. This section is the executive-level, normative baseline; detailed mechanisms live in later sections.

1.2 Scope

This applies to all Azure resources and teams operating in your **Landing Zones** (a pre-configured Azure environment with baseline identity, network, security, policy, and management capabilities for hosting workloads).

Governance domains covered at a principle level:

- **Resource organization**: Management Group and subscription strategy (details: s05).
- **Identity and access**: Entra ID, RBAC, PIM, break-glass accounts (details: s04).
- **Policy enforcement**: Azure Policy, policy initiatives (initiative), policy exemptions, progressive enforcement (details: s06).
- **Security posture**: Defender for Cloud, secure baselines, data protection (details: s07).
- **Networking**: hub-spoke, Virtual WAN (vWAN), private endpoints, DNS (details: s08).
- **Observability and operations**: diagnostic settings, Log Analytics workspace strategy, alerting, backup/DR (details: s09).
- **Cost governance**: FinOps and Chargeback/Showback (details: s10).

- **Platform delivery**: golden paths and landing zone productization (details: s11).
- **Audit and evidence**: control mapping and evidence retention (details: s12).

1.3 Decisions

- You **MUST** centralize governance control points at **Management Group** scope wherever Azure supports inheritance, to minimize drift and reduce per-subscription variance.
- You **MUST** treat a **policy exemption** as a time-bound risk acceptance with an owner, rationale, expiry, and remediation plan, to avoid accumulating **exemption debt** (exemptions that are past due or repeatedly renewed without progress).
- You **MUST** implement least privilege using **RBAC**, and you **MUST** gate privileged elevation with **PIM** for in-scope privileged roles (Azure RBAC roles and, where applicable, Entra ID directory roles).
- You **SHOULD** provide a supported **golden path** (preferred, supported deployment pattern balancing speed and governance) so teams can comply by default rather than by exception.
- You **SHOULD** use progressive policy rollout (audit → DeployIfNotExists → deny) for controls that can be remediated deterministically, while reserving deny policies for non-negotiable, high-blast-radius risks.

1.4 Standards/Controls

- **Normative keywords**: MUST/SHOULD/MAY are used as requirement levels for governance controls.
- **Logging and auditability (baseline)**:
    - You **MUST** centralize operational logs by enforcing **diagnostic settings** for all **in-scope resource types that support diagnostic settings**, routing to a governed **Log Analytics Workspace** (or a security-approved equivalent meeting retention, access control, and query requirements).
    - You **MUST** retain logs for a minimum of **180 days** online unless regulatory needs require longer (details and tiering: s09, s12).
- **Change traceability**:
    - The platform team **MUST** manage baseline governance artifacts (Management Group hierarchy, policy assignments, RBAC baselines) as **Infrastructure as Code (IaC)** with approvals and audit trails (details: s05, s06).

- **Privileged access safeguards**:
  - The security team **MUST** define which privileged roles are in-scope for PIM across Azure RBAC and Entra ID directory roles, including emergency access constraints (details: s04).

1.5 Implementation Notes

- Keep this section “thin”: you set intent and non-negotiables here; you operationalize in domain sections (s04–s12).
- Use explicit subject transitions to avoid blurred accountability:
  - “You MUST …” for organization-wide requirements.
  - “The platform team MUST …” or “The security team MUST …” for role-owned requirements.
- Workspace strategy and “approved equivalent” criteria are defined in s09; do not negotiate logging exceptions ad hoc in project delivery.

1.6 Metrics

These KPIs are defined as **enterprise success measures**; detailed KPI definitions, data sources, and targets are maintained in your roadmap/metrics section (s13).

KPI	Definition (normalized)	Target	Data Source	Reporting Cadence	Primary Owner
Policy compliance rate	Compliant resources ÷ total in-scope resources (excluding approved policy exemptions)	≥ 90% (maturity ramp permitted)	Azure Policy compliance	Monthly	Security team (A), platform team (R)
Drift rate	Count of unmanaged/manual changes to governed resources per period	Trending down	IaC pipeline + change audit	Monthly	Platform team
Time-to-landing-zone	Request submitted → subscription and	≤ 5 business days	Vending workflow + pipeline logs	Monthly	Platform team

KPI	Definition (normalized)	Target	Data Source	Reporting Cadence	Primary Owner
	baseline ready for workload onboarding				
Exemption debt	# past-due policy exemptions and % renewed without remediation evidence	0 past-due	Exemption registry	Monthly	Security team
Cost variance	Actual vs forecast by subscription/ cost center/app	≤ ±10%	Cost management + tags	Monthly	FinOps team

2. Organization Context, Assumptions, and Target Operating Model

2.1 Purpose

Enable your teams to run Azure governance as a repeatable operating model (not ad-hoc decisions) with clear decision rights, controlled change flow, and measurable outcomes across Management Groups, subscriptions, and Landing Zones.

2.2 Scope

- Organizational context for the in-scope governance domains defined in **s01** (identity/access, networking, policy/compliance, security operations, observability, and FinOps).
- Decision rights across roles: platform team, security/compliance, enterprise architecture, FinOps, and app teams.
- How governance artifacts change over time: intake, review, implementation via Infrastructure as Code (IaC), rollout, and measurement.
- Escalation and exception handling model (process view only; the detailed policy exemption mechanics belong in the policy-as-code section).

2.3 Decisions

- You **MUST** treat governance as a product with a managed backlog, versioned releases, and defined ownership to reduce drift and improve auditability.
- You **MUST** centralize baseline governance decisions at **Management Group** scope and operationalize reduced autonomy through a controlled exception path (policy exemption) and a supported golden path.
- You **MUST** distinguish privileged access across planes:
    - Azure **role-based access control (RBAC)** privileged roles at Management Group/subscription/resource scopes.
    - **Entra ID** directory roles (identity plane) where applicable. The security/compliance team is accountable for the privileged access model across both planes.
- You **SHOULD** standardize a predictable governance release cadence (biweekly or monthly) for non-breaking changes; you **MUST** document and review emergency changes.

2.4 Standards/Controls

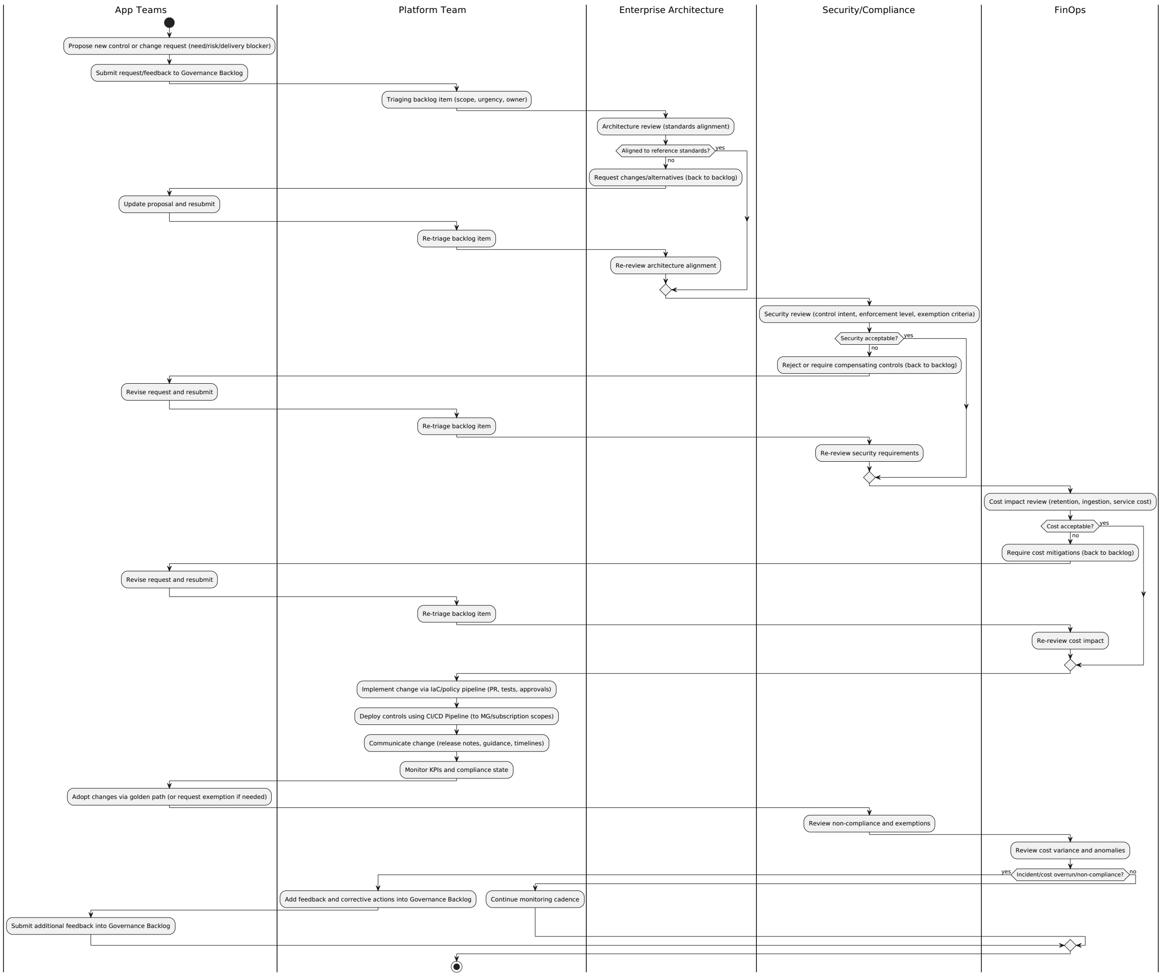
- Change control
    - All governance changes **MUST** be captured as backlog items with: owner, rationale, risk impact, cost impact, scope, rollout plan, and success metrics.
    - Governance artifacts (policy assignments, initiatives, RBAC assignments, diagnostic settings baselines, Landing Zone templates) **MUST** be changed via versioned IaC with approvals and traceability.
- Review gates
    - Enterprise architecture **MUST** confirm alignment to reference architectures/standards before implementation.
    - Security/compliance **MUST** approve control intent, enforcement level (audit vs DeployIfNotExists vs deny), and exception criteria.
    - FinOps **MUST** approve material cost-impacting changes (e.g., logging retention increases, new central services).
- Exception handling
    - Exceptions **MUST** be explicit, time-bound, and reviewable; they **MUST** have an owner, reason, and expiration date and be tracked to prevent accumulating exemption debt.

2.5 Implementation Notes

2.5.1 Decision flow checklist (Intake → Review → Implement → Rollout → Measure)

Stage	Minimum inputs	Output artifact	Primary accountable role
Intake	Problem statement, proposed change, scope, urgency	Backlog item	Platform team
Review	Architecture fit, security requirements, cost impact	Approved design + enforcement approach	Enterprise architecture / Security/ Compliance / FinOps
Implement	IaC change + tests + rollback plan	Pull request + pipeline run	Platform team
Rollout	Comms + staged deployment plan	Release notes + change record	Platform team
Measure	KPI impact + incidents/ non-compliance feedback	KPI report + backlog updates	Platform team / Security/Compliance / FinOps

2.5.2 Governance Operating Model (Roles and Decision Flow)


App Teams
Platform Team
Enterprise Architecture
Security/Compliance
FinOps
Propose new control or change request (need/risk/delivery blocker)
Submit request/feedback to Governance Backlog
Triaging backlog item (scope, urgency, owner)
Architecture review (standards alignment)
Aligned to reference standards?
yes
no
Request changes/alternatives (back to backlog)
Update proposal and resubmit
Re-triage backlog item
Re-review architecture alignment
Security review (control intent, enforcement level, exemption criteria)
Security acceptable?
yes
no
Reject or require compensating controls (back to backlog)
Revise request and resubmit
Re-triage backlog item
Re-review security requirements
Cost impact review (retention, ingestion, service cost)
Cost acceptable?
yes
no
Require cost mitigations (back to backlog)
Revise request and resubmit
Re-triage backlog item
Re-review cost impact
Implement change via IaC/policy pipeline (PR, tests, approvals)
Deploy controls using CI/CD Pipeline (to MG/subscription scopes)
Communicate change (release notes, guidance, timelines)
Monitor KPIs and compliance state
Adopt changes via golden path (or request exemption if needed)
Review non-compliance and exemptions
Review cost variance and anomalies
Incident/cost overrun/non-compliance?
yes
no
Add feedback and corrective actions into Governance Backlog
Continue monitoring cadence
Submit additional feedback into Governance Backlog

2.6 Metrics

KPI	Definition (denominator rules)	Target	Data source	Reporting cadence	Primary owner
Policy compliance rate	Compliant resources / total in-scope resources (exclude approved policy exemptions)	≥ 90%	Azure Policy compliance	Monthly	Security/ Compliance
Drift trend	Count of unauthorized portal/manual	Downward trend	IaC pipeline logs +	Monthly	Platform team

KPI	Definition (denominator rules)	Target	Data source	Reporting cadence	Primary owner
	changes vs IaC-deployed changes (window = 30 days)		change audit logs		
Time-to-Landing Zone	Median business days from approved request → subscription ready under correct Management Group with baseline controls applied	≤ 5 business days	Vending workflow system + pipeline timestamps	Monthly	Platform team
Exemption debt	Number of expired/past-due policy exemptions	0 past-due	Exemption registry	Weekly	Security/ Compliance
Privileged access hygiene	% privileged assignments gated by PIM (Azure RBAC and in-scope Entra ID roles)	≥ 95%	PIM reports + RBAC audit	Monthly	Security/ Compliance
Cost variance	Monthly actual vs budget per subscription/team	≤ ±10%	Cost management + budgets	Monthly	FinOps

3. s03 — Governance Scope and Architecture Overview (CAF-aligned)

3.1 Purpose

You align Azure governance to reduce risk while enabling delivery speed through a repeatable, auditable operating model across **Management Groups**, **subscriptions**, and **Landing Zones**. This section provides a single end-to-end mental model your teams can use to understand *who governs what* and *how control signals flow*.

3.2 Scope

You cover the governance capability areas that shape your Landing Zones and workload operations:

- **Identity** (Entra ID) and privileged access foundations that enable safe delegation (see s04).
- **Resource organization** (Management Groups/subscriptions) that defines inheritance and isolation boundaries (see s05).
- **Policy & compliance** (Azure Policy, policy initiative (initiative), exemptions) that enforce guardrails (see s06).
- **Security posture** (Defender for Cloud) for baseline posture and workload protections (see s07).
- **Networking** (hub-spoke, Virtual WAN (vWAN), Private Endpoint/Private DNS Zone patterns) that constrain connectivity risk (see s08).
- **Observability** (Azure Monitor/Log Analytics workspace, diagnostic settings) for operational and audit evidence (see s09).
- **Cost governance** (FinOps, Chargeback/Showback, tagging) for allocation and optimization (see s10).

Out of scope: detailed implementation standards for each capability area; those are defined in the referenced sections.

3.3 Decisions

- You **MUST** centralize baseline governance at **Management Group** scope to maximize inheritance and reduce drift vectors; autonomy is provided via a time-bound **policy exemption** workflow and a supported **golden path** (see s06, s11).
- You **MUST** treat **Landing Zones** as products: the platform team owns the baseline lifecycle; workload teams consume via **Infrastructure as Code (IaC)** (see s11).
- You **MUST** prefer preventive controls where blast radius is high:
  - **MUST** use **deny policy** for non-negotiables.
  - **SHOULD** use **audit policy** during adoption phases.
  - **SHOULD** use **DeployIfNotExists** when deterministic remediation is available (see s06).

- You **MUST** make exceptions explicit, time-bound, and reviewable to control exemption debt (see s06).

3.3.1 CAF alignment and deliberate deviations (rationale)

- You align to Azure CAF concepts of platform foundations and Landing Zones, but you **deviate intentionally** in these areas:
    - **Stricter Management Group centralization** than some federated CAF variants to reduce control drift and audit variance.
    - **Progressive enforcement as a standard release pattern** (audit → DeployIfNotExists → deny) to balance delivery speed with operational sustainability.
    - **Formal exemption governance** (explicit owner/reason/expiration) to prevent “exceptions by default” operating models.

3.4 Standards/Controls

- Identity foundation:
    - You **MUST** use **Entra ID** as the authentication/authorization foundation.
    - You **MUST** enforce least privilege using **RBAC** and gate privileged elevation using **PIM** for in-scope privileged roles (see s04).
    - You **MUST** maintain at least one monitored and tested **break-glass account** (see s04).
- Policy and auditability:
    - You **MUST** manage standards using **Azure Policy** and **policy initiatives (initiatives)**, assigned primarily at **Management Group** scope (see s06).
    - You **MUST** keep policy exemptions time-bound and reviewable (owner/reason/scope/expiration) (see s06).
- Logging and evidence:
    - You **MUST** centralize operational logs by enforcing **diagnostic settings** for all in-scope resource types that support them, routing to a governed **Log Analytics workspace** (or a security-approved equivalent meeting retention, access control, and query requirements) (see s09).
- Shared responsibility:
    - The platform team is accountable for baseline controls in the management plane; workload teams are accountable for workload configuration and data handling within guardrails.

3.5 Implementation Notes

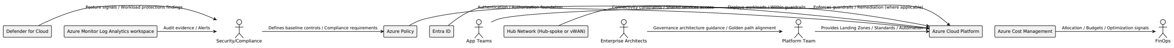
- Use this section as the “map”; treat each capability area as a product with:
  - a control catalog (policies/standards),
  - an operating cadence (review, rollout, exception handling),
  - measurable outcomes (KPIs).
- Keep the organization model simple initially:
  - fewer Management Groups and subscriptions, stronger inheritance, and a controlled exemption workflow generally outperform highly customized trees with inconsistent controls.

3.6 Metrics

These KPIs operationalize the success criteria for the governance model; detailed KPI definitions SHOULD be centralized in your KPI catalog (see s13 if present).

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Policy compliance rate	Compliant resources ÷ total in-scope resources for assigned initiatives at Management Group + subscription scopes	≥ 90%	Azure Policy compliance	Monthly	security team (A), platform team (R)
Drift trend	Count of out-of-process changes (not via approved IaC pipeline) over time	Downward trend	Activity logs + IaC pipeline audit	Monthly	platform team
Time-to-landing-zone	Median business days from request intake to subscription ready with baseline controls	≤ 5 business days	Vending workflow + change records	Monthly	platform team
		0 past-due		Monthly	

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Exemption debt	# exemptions past `expiresOn` (no renewal/closure)		Exemption registry (policy exemptions)		security team
Privileged access hygiene	% privileged assignments that are PIM-gated and time-bound	≥ 95%	PIM reports + RBAC assignments	Monthly	security team
Cost variance	Actual spend vs budget per subscription/team/cost center	≤ ±10%	Azure Cost Management	Monthly	FinOps team


Defender for Cloud
Posture signals / Workload protections findings
Azure Monitor Log Analytics workspace
Audit evidence / Alerts
Security/Compliance
Defines baseline controls / Compliance requirements
Azure Policy
Entra ID
Authentication / Authorization foundation
App Teams
Hub Network (Hub-spoke or vWAN)
Connectivity constraints / Shared services access
Deploys workloads / Within guardrails
Enforces guardrails / Remediation (where applicable)
Enterprise Architects
Governance architecture guidance / Golden path alignment
Platform Team
Provides Landing Zones / Standards / Automation
Azure Cloud Platform
Azure Cost Management
Allocation / Budgets / Optimization signals
FinOps

4. s04 — Identity and Access Governance (Entra ID, RBAC, PIM)

4.1 Purpose

You standardize identity and privileged access so your teams can operate Landing Zones safely with least privilege, auditable elevation, and predictable scope boundaries. This reduces role sprawl, limits blast radius, and improves operational sustainability through repeatable access patterns.

4.2 Scope

You govern:

- **Entra ID** tenant-level identity lifecycle and access governance for Azure access (users, groups, service principals, managed identities).
- **Azure RBAC** assignments at **Management Group**, **subscription**, **resource group**, and **resource** scope.
- **Privileged Identity Management (PIM)** for privileged roles (Azure RBAC roles and, where applicable, Entra ID directory roles).

Out of scope:

- Workload application authorization models inside the application (handled by workload teams' application security standards).
- Network access controls (see networking governance) and policy-as-code mechanics (see policy governance).

Dependencies:

- Management Group and subscription boundaries (resource organization) because they define RBAC scopes.
- Policy-as-code and exemptions (policy governance) because enforcement and exemptions influence privileged operations.

4.3 Decisions

- You **MUST** enforce **least privilege** by default, using Azure RBAC and group-based assignments to reduce direct user permissions and improve auditability.
- You **MUST** gate privileged access through **PIM** for in-scope privileged roles to eliminate standing administrative access and support time-bound elevation.
- You **SHOULD** centralize baseline role assignments at **Management Group scope** where inheritance is intended; you **MAY** use subscription/resource group scopes for workload-specific delegation.
- You **MUST** treat **break-glass account** access as an emergency-only pathway with heightened monitoring and explicit operational controls.

4.4 Standards/Controls

4.4.1 Identity and access baseline

- You **MUST** use **Entra ID** security groups (not individual users) as the primary principal for Azure RBAC assignments where feasible.
- You **MUST** define a small, approved catalog of role bundles per operator persona (e.g., platform operator, security operator, workload operator) to prevent role sprawl.
- You **SHOULD** minimize custom RBAC roles; if you create them, you **MUST** review custom roles at least **quarterly** for necessity and permissions creep.

4.4.2 Privileged access (PIM)

- You **MUST** use **PIM** for privileged Azure RBAC roles at **Management Group** and **subscription** scopes (e.g., Owner, Contributor, User Access Administrator, and security-sensitive roles).
- You **MUST** configure PIM activations to be **time-bound** and require **justification**.
- You **SHOULD** require approval for high-risk roles and production scopes; you **MAY** allow self-activation for lower-risk roles in non-production with justification and short durations.
- You **SHOULD** require Conditional Access protections for privileged activation; you **MUST** require them for production scopes unless you have a documented compensating control.

4.4.3 Break-glass access

- You **MUST** maintain at least **one** break-glass account.
- You **SHOULD** maintain **two** break-glass accounts to reduce lockout risk and operational single points of failure.
- Break-glass credentials **MUST** be stored and accessed via a controlled process and **MUST** trigger high-priority monitoring and incident handling when used.

4.4.4 Service principals and managed identities

- Workload teams **SHOULD** prefer managed identities over service principals when supported by the service.
- Service principals **MUST** have credential hygiene controls (rotation/expiration, minimal scopes, and no shared credentials across environments).
- You **MUST** prohibit “shadow” service principals (created ad hoc without ownership metadata and lifecycle controls).

4.4.5 Access reviews and attestation

- You **MUST** perform access reviews for privileged groups and privileged role eligibility at least **quarterly**.
- Access reviews **MUST** produce evidence (review outcome, approver, date, and removals) suitable for audit.

4.5 Implementation Notes

4.5.1 RBAC scope model (practical guidance)

- Prefer **Management Group** scope for baseline access patterns that should be consistent and inherited (platform team, security team).
- Prefer **subscription** scope for workload team delegation aligned to the subscription as an isolation unit.
- Use **resource group** scope for finer delegation when a workload team owns only a subset of resources within a subscription and you want explicit separation of duties.

4.5.2 Minimal role set (recommended starting point)

- Platform operators: Contributor at platform subscriptions; limited Owner only via PIM for break-fix.
- Security operators: Security Reader / Security Admin patterns as needed; avoid broad Owner.
- Workload operators: Contributor scoped to workload subscriptions or workload resource groups; no standing Owner.

4.5.3 Evidence and auditability

- Your teams **MUST** ensure privileged operations produce:
    - Entra ID sign-in/audit evidence for activation and authentication events
    - Azure Activity Log evidence for ARM actions
    - PIM activation logs and access review records

4.6 Operationalization

4.6.1 Ownership

- **Platform team**: Accountable for RBAC design patterns, group strategy, and default role bundles at Management Group/subscription scopes.
- **Security team**: Accountable for PIM policy, break-glass governance, and access review standards.
- **Workload teams**: Responsible for requesting access via the approved process and maintaining workload group membership and ownership metadata.

4.6.2 Cadence

- **Quarterly**: Access reviews for privileged eligibility and privileged group membership; custom role review (if any).
- **Monthly**: Review privileged activation trends (top roles, top scopes, unusual patterns).
- **After incident/emergency**: Post-use break-glass review within **2 business days** with remediation actions.

4.6.3 Tooling

- PIM for role eligibility/activation and access reviews.
- Central log platform (governed Log Analytics Workspace and/or security-approved equivalent) as the evidence store for access and activity logs.
- IaC pipelines for RBAC assignments where feasible to reduce drift and enable change review.

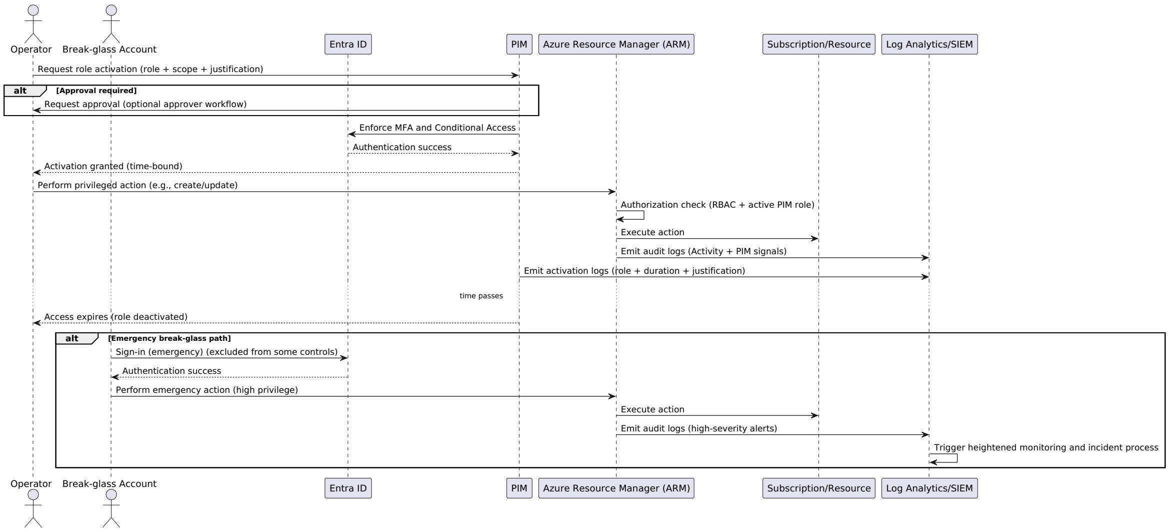
4.7 Anti-patterns

- Standing Owner/Contributor assignments for operators in production subscriptions.
- Direct user RBAC assignments instead of group-based assignments without documented justification.
- Untracked service principals with long-lived secrets and unclear ownership.
- Break-glass accounts used for convenience or routine operations.

4.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Privileged access hygiene	% of privileged Azure RBAC role assignments at MG/subscription scopes that are PIM-gated (eligible, not permanent)	≥ 95%	PIM role assignments + Azure RBAC inventory	Monthly	Security team
	Count of permanent		Azure RBAC inventory	Monthly	Platform team

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Standing privilege count	privileged role assignments for human identities at MG/subscription scope	Downward trend; goal 0			
Access review completion	% of required quarterly access reviews completed on time with recorded outcomes	100%	PIM/access review records	Quarterly	Security team
Break-glass readiness	# of break-glass accounts tested within the last quarter with evidence	≥ 1 (SHOULD be 2)	Runbook evidence + sign-in logs	Quarterly	Security team


Operator
Break-glass Account
Entra ID
PIM
Azure Resource Manager (ARM)
Subscription/Resource
Log Analytics/SIEM
Request role activation (role + scope + justification)
alt
[Approval required]
Request approval (optional approver workflow)
Enforce MFA and Conditional Access
Authentication success
Activation granted (time-bound)
Perform privileged action (e.g., create/update)
Authorization check (RBAC + active PIM role)
Execute action
Emit audit logs (Activity + PIM signals)
Emit activation logs (role + duration + justification)
time passes
Access expires (role deactivated)
alt
[Emergency break-glass path]
Sign-in (emergency) (excluded from some controls)
Authentication success
Perform emergency action (high privilege)
Execute action
Emit audit logs (high-severity alerts)
Trigger heightened monitoring and incident process
Operator
Break-glass Account
Entra ID
PIM
Azure Resource Manager (ARM)
Subscription/Resource
Log Analytics/SIEM

5. s05 — Resource Organization: Management Groups, Subscriptions, and Naming

5.1 Purpose

You standardize how your teams organize Azure so governance inherits cleanly, ownership is unambiguous, and scaling does not multiply drift. This section defines the reference **Management Group** hierarchy, subscription model, and the naming/tagging/lifecycle standards that make operations and cost allocation sustainable.

5.2 Scope

You apply this to:

- All Azure **Management Groups**, **subscriptions**, resource groups, and resources created in your **Landing Zones**
- Platform and workload subscriptions across **dev**, **test**, **prod**, and **sandbox**
- All resource types that support **Azure Policy**, **RBAC**, and tagging (with documented exceptions where a service cannot support a control)

Dependencies:

- s04 (Identity and Access Governance: **Entra ID**, **RBAC**, **PIM**, break-glass)
- s06 (Policy-as-Code: **Azure Policy**, policy initiative (initiative), exemptions, rollout patterns)

5.3 Decisions

- You **MUST** centralize baseline governance (RBAC guardrails and policy initiatives) at **Management Group** scope to maximize inheritance and minimize drift; autonomy is provided via a time-bound **policy exemption** workflow and a supported **golden path**.
- You **MUST** separate **platform** subscriptions from **workload** subscriptions to reduce blast radius and create clear operational/accounting boundaries.
- You **SHOULD** use environment isolation as the default: **prod** separated from non-prod unless a documented risk decision says otherwise.

- You **MAY** provide **sandbox** subscriptions for experimentation, but only with explicit non-negotiable controls (defined below) and explicit relaxations (also defined below).

5.4 Standards/Controls

5.4.1 Management Group hierarchy and placement (reference model)

Purpose
Provide a consistent hierarchy so policy/RBAC inheritance is predictable and reporting is coherent.

Scope
All Management Groups and subscriptions in Landing Zones.

Decisions
Baseline controls are assigned once at Management Group scope; overlays are used only for true deltas.

Standards/Controls

- You **MUST** implement a consistent hierarchy with at least:
    - Root Management Group
    - Platform Management Group
    - Landing Zones Management Group
    - Sandbox Management Group
- You **MUST** assign baseline policy initiatives at **Management Group** scope and rely on inheritance to subscriptions.
- You **MAY** assign **subscription-level overlay** initiatives only for deltas that cannot be expressed safely at Management Group scope (e.g., environment/workload-class differences). Overlay initiatives **MUST NOT** duplicate baseline initiatives.

Implementation Notes

- Keep the hierarchy shallow. Add additional Management Group levels only when you need a real boundary (compliance scope, RBAC delegation model, or policy variance).

Metrics

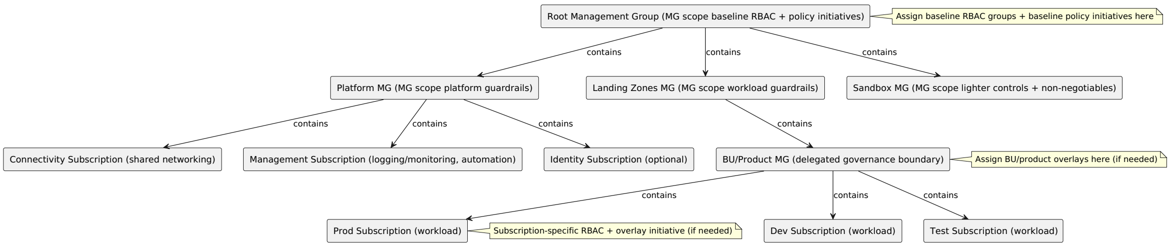
- See section Metrics.

Operationalization

- See section Operationalization.

Anti-patterns

- Adding Management Group layers “because we might need them later.”
- Assigning the same initiatives at both Management Group and subscription scopes (duplicate/conflicting evaluation).


Root Management Group (MG scope baseline RBAC + policy initiatives)
Assign baseline RBAC groups + baseline policy initiatives here
contains
contains
contains
Platform MG (MG scope platform guardrails)
Landing Zones MG (MG scope workload guardrails)
Sandbox MG (MG scope lighter controls + non-negotiables)
contains
contains
contains
contains
Connectivity Subscription (shared networking)
Management Subscription (logging/monitoring, automation)
Identity Subscription (optional)
BU/Product MG (delegated governance boundary)
Assign BU/product overlays here (if needed)
contains
contains
contains
Prod Subscription (workload)
Subscription-specific RBAC + overlay initiative (if needed)
Dev Subscription (workload)
Test Subscription (workload)

5.4.2 Subscription strategy (platform vs workload vs sandbox)

Purpose

Create clear operational boundaries, reduce blast radius, and enable cost governance.

Scope

All platform, workload, and sandbox subscriptions.

Decisions

Platform subscriptions host shared services; workloads are isolated by environment by default.

Standards/Controls

- You **MUST** maintain, at minimum, these platform subscriptions:
  - **Connectivity subscription** for shared networking
  - **Management subscription** for central monitoring/automation components
  - **Identity subscription** is **MAY** (only if your architecture requires identity hosting patterns)
- You **MUST** define a subscription vending workflow that captures required metadata and places subscriptions into the correct **Management Group**.

5.4.2.1 Subscription vending (minimum metadata + minimum controls)

- Subscription requests **MUST** capture at minimum:
  - `Environment` (dev/test/prod/sandbox)
  - `App` (service/application identifier)
  - `Owner` (team group/alias)
  - `CostCenter`
  - Requested `Regions` (primary + optional secondary)
  - Data classification intent (`DataClassification` target)
  - Connectivity model (hub-spoke vs vWAN) and on-prem requirement (yes/no)
- Vending automation **MUST** apply at creation time:
  - Correct Management Group placement
  - Baseline policy initiative inheritance validation
  - Baseline RBAC groups assignment (per s04)
  - Budget + alert configuration (per s10)
  - Diagnostic settings routing for subscription/activity logs to the governed **Log Analytics Workspace** (per s09)

5.4.2.2 Sandbox policy posture (non-negotiables + relaxations)

- Sandbox subscriptions **MUST** enforce the following non-negotiables:
  - **Entra ID** authentication and multi-factor authentication per s04.
  - Diagnostic settings routing of required operational logs to the governed **Log Analytics Workspace** (or approved equivalent) per s09.
  - **PIM**-gated privileged access per s04.
  - Budgets/alerts per s10.
- Sandbox subscriptions **SHOULD** relax controls explicitly, rather than via ad-hoc exemptions. Approved relaxations **MAY** include:
  - Fewer deny policies (retain only “high blast-radius” denies)
  - Shorter log retention where it does not violate audit requirements
  - Broader SKU allowance (but still within approved regions/providers)
  - No direct connectivity to production spokes by default

Implementation Notes

- Use a single intake path for subscription requests; block ad-hoc creation outside the vending workflow except for documented break-glass processes (see s04/s06).

Metrics

- See section Metrics.

Operationalization

- See section Operationalization.

Anti-patterns

- Creating subscriptions directly in the portal without vending metadata and placement controls.
- Allowing sandbox to become “shadow prod” (no budgets, no logging, standing privileged access).

5.4.3 Naming standard (automation-friendly and deterministic)

Purpose
Ensure names are predictable for operations, automation, and incident triage.

Scope
All subscriptions, resource groups, and resources where naming is under your control (not system-generated).

Decisions
Deterministic names are preferred; globally-unique constraints are handled with deterministic hashing.

Standards/Controls

- You **MUST** use deterministic naming that is consistent, discoverable, and compatible with automation.
- Default token pattern (adjust only where Azure resource naming rules require it):
    - `org-app-env-region-instance[-role]`
- Token definitions (default):
    - `org`: short organization/tenant identifier (lowercase alphanumeric)
    - `app`: application/product identifier (lowercase alphanumeric, no personal identifiers)
    - `env`: `dev|test|prod|sandbox`
    - `region`: approved short region code list maintained by the platform team (e.g., `eus2`, `wus3`, `weu`) and version-controlled
    - `instance`: `01-99` (two digits) unless the resource type requires another format

    - `role`: optional functional suffix (e.g., `api`, `web`, `sql`, `agw`)
- Global uniqueness rule:
    - Where Azure enforces global uniqueness and length constraints, you **MAY** append a short **deterministic hash** derived from stable inputs (e.g., `org-app-env-region-instance`) rather than random strings.

Implementation Notes

- If deterministic hashing is used, the hash algorithm/length **MUST** be standardized by the platform team so names remain reproducible.

Metrics

- See section Metrics.

Operationalization

- Implement naming via golden path modules and CI checks; enforce via policy only where technically feasible.

Anti-patterns

- Using random name suffixes "because it's easier" when deterministic hashing would meet uniqueness.

5.4.3.1 Checklist: naming validation (recommended)

- Names **SHOULD** encode `env` and `region` for operational triage.
- Names **MUST** avoid embedding personal identifiers.

5.4.4 Tagging schema (required keys)

Purpose
Use tags as the system-of-record for operational ownership, cost attribution, and classification.

Scope
All taggable resource groups and resources in Landing Zones (documented exceptions allowed where Azure does not support tags).

Decisions
Tag enforcement rolls out progressively; production enforcement becomes deny once a working golden path exists.

Standards/Controls

- You use tags as the system-of-record for operational ownership, cost attribution dimensions, and classification. Even if subscription is the primary cost boundary, tags remain mandatory for routing, automation, and reporting consistency.
- Rollout model for tag enforcement:
    - Phase 1: **Audit** for required tags broadly to measure gaps
    - Phase 2: **Deny** for production **resource group** creation/update when required tags are missing/invalid
    - Phase 3: Expand deny to additional scopes where operationally safe

Tag Key	Required?	Allowed Values / Format	Enforcement Level (Target by Phase)	Primary Owner	Notes
Owner	MUST	Team alias/ group (no individuals)	Phase 1: Audit; Phase 2+: Deny for prod RGs	Workload team	Drives incident routing and operational ownership
CostCenter	MUST	Finance master list code	Phase 1: Audit; Phase 2+: Deny for prod RGs	FinOps team	Drives Chargeback/ Showback
App	MUST	App/ product identifier	Phase 1: Audit; Phase 2+: Deny for prod RGs	Workload team	Align to service catalog/ CMDB if used
Environment	MUST	`dev	test	prod	sandbox`
DataClassification	MUST	`public	internal	confidential	restricted`
ManagedBy	SHOULD	`IaC	Portal	Other`	Audit

Implementation Notes

- Maintain an allow-list of resource types that cannot support tags and document compensating controls (e.g., enforce at resource group level; rely on subscription metadata).

Metrics

- See section Metrics.

Operationalization

- Implement tag policy assignments and remediation tasks via s06; validate tag values against approved lists where feasible.

Anti-patterns

- Treating tags as optional in prod, then attempting to reconstruct ownership/cost attribution later.

5.4.5 Resource lifecycle (create, transfer, decommission)

Purpose
Prevent orphaned spend, unknown ownership, and unmanaged risk through consistent lifecycle handling.

Scope
Resource and subscription lifecycle actions across all environments.

Decisions
Production entry requires ownership + cost attribution; decommissioning is a controlled process.

Standards/Controls

- You **MUST** require an identified `Owner` and `CostCenter` before resources enter **prod**.
- You **MUST** define a decommission standard:
    - Owner confirms data retention requirements and handoff
    - Resources are deleted or archived per policy (see s09/s12 for retention evidence)
    - Subscription closure follows a documented runbook to prevent orphaned spend and access
- You **SHOULD** automate lifecycle actions (e.g., subscription close-out checks, stale resource reporting) using your standard IaC and automation tooling.

Implementation Notes

- Treat subscription closure as a change-controlled activity with explicit approvals (platform + FinOps + security if regulated data exists).

Metrics

- See section Metrics.

Operationalization

- Use a standard runbook and checklist; require evidence capture for retention and access removal.

Anti-patterns

- Decommissioning by “stop paying attention” instead of a formal closure process.

5.5 Implementation Notes

- Use a “baseline + overlay” model: baseline at Management Group; small overlays only when needed. This reduces duplicate assignments and reporting confusion.
- Treat naming/tagging as enforceable controls: start in **audit**, then move to **deny** for **prod** once the golden path exists.

5.6 Operationalization

5.6.1 Purpose

You operationalize hierarchy, subscription vending, naming, and tagging so changes are repeatable, reviewable, and measurable.

5.6.2 Scope

Covers ownership, cadences, and tooling for:

- Management Group hierarchy changes
- Subscription vending/placement
- Naming/tagging enforcement and exception handling
- Lifecycle governance (transfer/decommission)

5.6.3 Decisions

- The platform team is **Accountable** for hierarchy and subscription placement standards; workload teams are **Responsible** for applying naming/tagging within their scopes.
- The security team is **Accountable** for classification requirements and approval of high-risk exceptions.

5.6.4 Standards/Controls

- Governance artifacts (hierarchy definitions, subscription vending configuration, policy assignments for tags) **MUST** be maintained as versioned **IaC** with approvals.
- Exceptions to tagging/naming controls **MUST** follow the **policy exemption** workflow defined in s06 (time-bound, owned, and tracked).

5.6.4.1 RACI: subscription vending and placement

Activity	**Platform team**	**Security team**	**FinOps team**	**Workload team**
Define MG hierarchy standard	A/R	C	I	I
Implement vending automation (IaC)	A/R	C	C	I
Approve subscription requests (standard)	A	C	C	R
Approve high-risk deviations/ exemptions	C	A/R	C	R
Configure budgets/alerts	C	I	A/R	I
Verify diagnostics routing (subscription)	A/R	C	I	I

5.6.5 Implementation Notes

- Keep an allow-list of resources that cannot support tags or certain naming patterns, and document compensating controls.

5.6.6 Metrics

- See section Metrics.

5.7 Anti-patterns

- Creating subscriptions directly in the portal without vending metadata and placement controls.
- Assigning the same policy initiatives at both Management Group and subscription scope (duplicate/conflicting evaluation).
- Using random name suffixes “because it’s easier” when deterministic hashing would meet uniqueness.
- Treating tags as optional in prod, then attempting to reconstruct ownership/cost attribution later.
- Allowing sandbox to become “shadow prod” (no budgets, no logging, standing privileged access).

5.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Time-to-landing-zone	Business days from approved subscription request to subscription ready (placed in correct Management Group with baseline inheritance confirmed)	≤ 5 business days	Vending workflow + subscription inventory	Monthly	Platform team
Tag compliance rate (prod)	Compliant prod resource groups / total prod resource groups (required tags present and valid)	≥ 95%	Azure Policy compliance	Weekly	Platform team
Naming compliance rate (sampled)	Compliant sampled resources / total sampled resources (per naming patterns)	≥ 90%	Policy/compliance queries or inventory scripts	Monthly	Platform team
Exemption debt	Count of past-due exemptions for			Weekly	

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
(resource org)	naming/tagging controls	0 past-due	Exemption registry (s06)		Security team
Subscription budget coverage	Subscriptions with budget + alert configured / total subscriptions	100%	Cost management exports	Monthly	FinOps team

6. s06 — Policy-as-Code: Azure Policy, Initiatives, and Exemptions

6.1 Purpose

Establish a repeatable, auditable **policy-as-code** operating model so your teams can author, test, deploy, and measure **Azure Policy** controls consistently across **Management Groups** and **Subscriptions**, while maintaining delivery speed through governed, time-bound **policy exemptions**.

6.2 Scope

Covers:

- Authoring and versioning of **Azure Policy** definitions, **Initiatives (Policy Initiatives)**, and assignments
- Progressive enforcement using **audit policy**, **DeployIfNotExists**, and **deny policy**
- Testing, staged rollout, remediation tasks, and compliance reporting
- Exemption governance, renewal, and “exemption debt” tracking

Excludes:

- Identity/RBAC mechanics and privileged access operations (see s04)
- Management Group and subscription hierarchy design (see s05)

6.3 Decisions

- You **MUST** manage policy definitions, initiatives, assignments, and exemptions as **versioned Infrastructure as Code (IaC)** with peer review and auditable change history.
- You **MUST** centralize baseline policy assignments at **Management Group** scope to maximize inheritance and reduce drift.
- You **MUST** use **progressive enforcement**: start with **audit policy**, move to **DeployIfNotExists** where safe, and use **deny policy** for non-negotiable/ high blast-radius requirements.
- You **MUST** make exceptions explicit and expiring via **policy exemptions** (no indefinite exemptions); “exemption debt” (past-due exemptions) is tracked and burned down.
- You **MUST** roll out changes safely using pilots and staged assignments with clear versioning and impact assessment.

6.4 Standards/Controls

6.4.1 Policy design and enforcement

- You **MUST** document intent for each policy/initiative: control objective, scope boundaries, and expected remediation behavior.
- You **MUST** use **deny policy** only when:
    - A safe **golden path** exists for compliant deployment, and
    - A defined adoption window has completed (see “Safe rollout”).
- You **SHOULD** prefer **DeployIfNotExists** when remediation is deterministic, reversible, and ownership is clear (platform team for shared services; workload teams for workload resources).
- You **MAY** use **audit policy** long-term for advisory controls where denial would create unacceptable operational risk.

6.4.2 Initiative structure (baseline catalog)

Your teams **MUST** structure initiatives by control domain to enable targeted rollout and reporting.

Initiative (example name)	Primary Domain	Default Effect Mix	Assignment Scope	Primary Owner	Rollout Phase
					Staged

Initiative (example name)	Primary Domain	Default Effect Mix	Assignment Scope	Primary Owner	Rollout Phase
`lz-baseline-security`	Security baseline	Audit → DeployIfNotExists → Deny	Management Group	Security team (A), platform team (R)	
`lz-baseline-logging`	Diagnostic settings	Audit → DeployIfNotExists	Management Group	Platform team (A)	Staged
`lz-baseline-tagging`	Tagging/ metadata	Audit → Deny (prod)	Management Group	FinOps team (A), platform team (R)	Staged
`lz-baseline-networking`	Network guardrails	Audit → Deny	Management Group	Platform team (A)	Pilot → Staged

6.4.3 Safe rollout and versioning

- Your teams **MUST** define an **adoption window** per deny promotion. Unless otherwise approved, the default adoption window is **30 days**.
- Deny rollouts **MUST** follow:
  - **≥30 days** in audit in a pilot Management Group (or equivalent pilot scope)
  - Published release notes and migration guidance
  - Verified golden path templates/pipelines updated to comply
- Assignments **MUST** be staged (e.g., pilot → limited production → full production) to control blast radius.
- Policy artifacts **MUST** be versioned (semantic versioning **SHOULD** be used) and release-tagged in Git.
- Deny promotions (audit → deny) **SHOULD** be scheduled and communicated with a minimum **2-week notice** once the adoption window is complete.
- Emergency deny changes **MAY** bypass the full adoption window only with documented risk justification, security team approval, and a defined rollback plan.

6.4.4 Remediation and drift management

- When using **DeployIfNotExists**, your teams **MUST** define:
  - The remediation identity and required permissions
  - Remediation task execution boundaries (what it can and cannot change)
- Drift between portal state and IaC intent **MUST** be detectable and acted upon.
  - Minimum detection standard: at least weekly, enumerate current policy definitions/initiatives/assignments/exemptions at relevant scopes and compare to your Git-declared desired state (or flag unmanaged artifacts for review).

6.4.5 Exemption governance (mandatory)

- Exemptions **MUST** be time-bound and include the minimum schema below.
- Renewals **MUST** be approved before expiry and include remediation progress evidence and an updated completion plan.
- Repeated renewals **SHOULD** require escalating approval (e.g., security team leadership and enterprise architecture).
- Exemptions **MUST** be implemented as Azure Policy exemption resources deployed via IaC (not only tracked in tickets).
- Exemption expiry handling **MUST** be operationalized (e.g., alerting before `expiresOn`, and post-expiry reporting to trigger remediation).

6.4.5.1 Exemption minimum schema

Field	Requirement	Notes
`assignmentReference`	MUST	`policyAssignmentId` or `initiativeAssignmentId`
`scope`	MUST	Smallest practical scope
`owner`	MUST	Prefer group/team alias over individual
`reason`	MUST	Business + technical rationale
`ticketId`	MUST	Work item or change record
`expiresOn`	MUST	No indefinite exemptions
`compensatingControls`	SHOULD	Required when risk increases
`renewalJustification`	MUST (for renewals)	Include evidence/progress

6.4.6 Evidence and auditability notes

- Policy changes **MUST** be traceable to an approved pull request (PR) with reviewer identity and change rationale.
- Exemption records **MUST** be exportable for audit (scope, owner, reason, expiry, approver, and renewal history).
- Compliance reporting **MUST** identify denominator rules (in-scope resources only) to prevent metric disputes.
    - Minimum denominator rule: “in-scope” equals resources under the assignment scope and within supported resource types, excluding resources with valid, unexpired exemptions and excluding known unsupported/unenforceable cases documented in a capability matrix.

6.5 Implementation Notes

- Keep “baseline” initiatives small and stable; use overlays for environment or workload-class deltas to avoid policy sprawl.
- Build a resource-type capability matrix for diagnostic settings and other controls where enforcement/remediation varies by service.
- Avoid introducing new deny controls during peak delivery windows; bundle deny changes into predictable governance releases unless urgent risk requires emergency change handling.

6.6 Operationalization

6.6.1 Ownership

- Platform team: authors and operates the policy-as-code pipeline; **Responsible** for Management Group–scoped baseline assignments deployment mechanics and compliance reporting plumbing.
- Security team: **Accountable** for security control intent, denial criteria, and exemption approvals for high-risk controls.
- Workload teams: **Responsible** for workload remediation and for requesting exemptions with evidence and timelines.

6.6.2 Cadence

- Biweekly or monthly policy release train **SHOULD** be used for non-breaking changes.
- Exemption reviews **MUST** occur at least monthly; past-due exemptions are prioritized for closure.

6.6.3 Tooling

- Git repository as the system of record for policy definitions, initiatives, assignments, and exemption declarations.
- CI/CD pipeline with linting, “what-if” evaluation, and automated deployment to test and production Management Groups.
- Compliance reporting integrated into your central reporting workspace/ dashboard with alerting on regressions and exemption expiry.

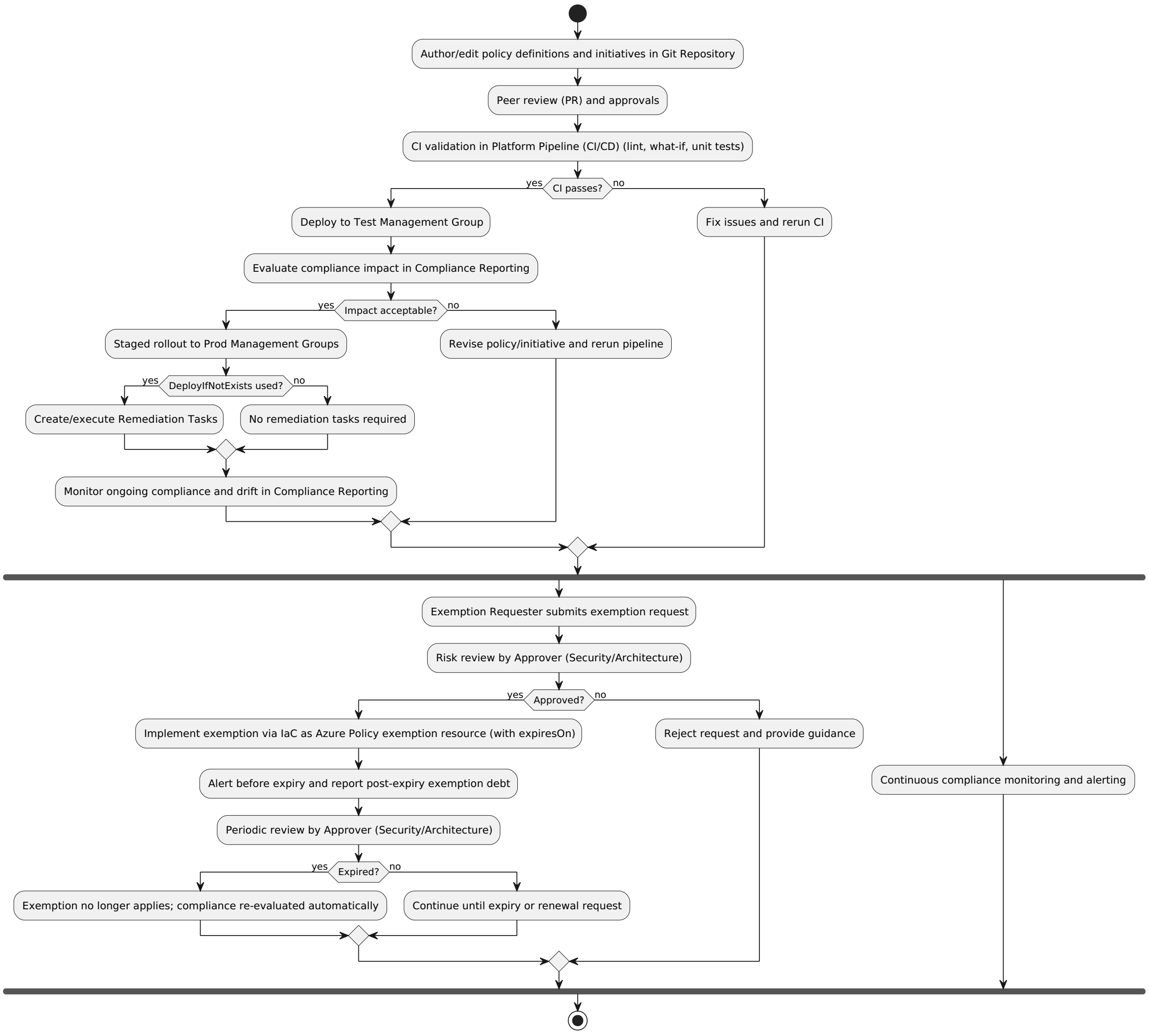
6.7 Anti-patterns

- Treating portal edits as “temporary” and never reconciling them back into IaC
- Rolling out deny effects without a pilot, adoption window, and an updated golden path
- Using exemptions as a default delivery mechanism (unbounded “exception debt”)
- Assigning overlapping initiatives at multiple scopes without a documented inheritance/override model and explicit justification
- Using **DeployIfNotExists** for changes that are not deterministic or not owned by a clear operator

6.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Policy compliance rate	Compliant in-scope resources ÷ total in-scope resources	≥90% (baseline); security initiatives SHOULD target ≥95% excluding valid exemptions	Azure Policy compliance	Monthly	Platform team (R), security team (A)
Time to policy change	Median time from PR open → production assignment complete	≤10 business days (non-emergency)	Git + pipeline logs	Monthly	Platform team (A/R)
Exemption debt	Count of expired	0 past-due	Exemption registry	Weekly	Security team (A),

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
	exemptions not closed or renewed		(deployed exemptions + export)		platform team (R)
Deny rollout hygiene	% deny promotions that met adoption-window and comms checklist	100%	Release checklist	Quarterly	Platform team (R), security team (A)


Author/edit policy definitions and initiatives in Git Repository
Peer review (PR) and approvals
CI validation in Platform Pipeline (CI/CD) (lint, what-if, unit tests)
CI passes?
yes
no
Deploy to Test Management Group
Fix issues and rerun CI
Evaluate compliance impact in Compliance Reporting
Impact acceptable?
yes
no
Staged rollout to Prod Management Groups
Revise policy/initiative and rerun pipeline
DeployIfNotExists used?
yes
no
Create/execute Remediation Tasks
No remediation tasks required
Monitor ongoing compliance and drift in Compliance Reporting
Exemption Requester submits exemption request
Risk review by Approver (Security/Architecture)
Approved?
yes
no
Implement exemption via IaC as Azure Policy exemption resource (with expiresOn)
Reject request and provide guidance
Continuous compliance monitoring and alerting
Alert before expiry and report post-expiry exemption debt
Periodic review by Approver (Security/Architecture)
Expired?
yes
no
Exemption no longer applies; compliance re-evaluated automatically
Continue until expiry or renewal request

7. s07 — Security Governance: Defender for Cloud, Secure Baselines, and Data Protection

7.1 Purpose

You standardize security outcomes across all Landing Zones so controls are repeatable, auditable, and sustainable. Your teams use a single set of minimum baselines, progressive enforcement, and clear decision rights to reduce drift and avoid “security by exception.”

7.2 Scope

You cover:

- Secure baseline requirements for Azure resources hosted in Landing Zones.
- Defender for Cloud enablement, recommendations governance, and secure score usage.
- Key Vault governance for secrets/keys/certificates, including rotation and hardware security module (HSM) considerations.
- Data classification, encryption, and key management ownership.
- Vulnerability management guardrails, including container/image controls.

You exclude:

- Network-specific security architecture (owned by the networking governance section; referenced here only where it affects data exfiltration controls and private access patterns).
- Application secure software development lifecycle (SDLC) implementation details beyond minimum governance guardrails.

Dependencies:

- s06 (policy-as-code lifecycle for Azure Policy definitions, policy initiatives (initiatives), assignments, and exemptions)
- s04 (identity guardrails for Entra ID, RBAC, and PIM)
- s05 (data classification tagging and ownership metadata)

7.3 Decisions

- You **MUST** adopt a single secure baseline and map it to enforceable controls, prioritizing high blast-radius risks.
- You **MUST** use progressive enforcement for baseline controls (**audit policy → DeployIfNotExists → deny policy**) and reserve **deny policy** for non-negotiable requirements.
- The security team **MUST** be **Accountable** for risk acceptance, including policy exemptions for high-risk controls; the platform team **MUST** be **Responsible** for implementing approved controls as code (per s06).
- You **MUST** treat key management as a shared responsibility: workload teams own data classification and required encryption posture; the platform team owns baseline key governance patterns; the security team owns minimum cryptographic standards and exceptions.

7.4 Standards/Controls

7.4.1 Secure baseline (minimum requirements)

Purpose
Ensure every subscription and workload starts from an enforceable, measurable baseline.

Scope
All in-scope resource types in Landing Zones, including platform and workload subscriptions.

Decisions
Baseline controls prioritize identity, logging, encryption, and exposure reduction.

Standards/Controls

- You **MUST** define a baseline aligned to stable external references and translate it into Azure Policy initiatives (initiatives) at **Management Group** scope (per s06).
  - Baseline references **SHOULD** include (pin versions): Microsoft Security Benchmark (MSB) for Azure and/or CIS Microsoft Azure Foundations Benchmark.
- You **MUST** implement deny policies for high blast-radius controls, including at minimum:
  - Public exposure controls where feasible (e.g., disallow public endpoints on in-scope PaaS where private access is required).

    - Encryption requirements for data services where the platform has an enforceable setting.
    - Mandatory diagnostic settings enablement where supported (see s09 for the diagnostic settings standard).
- You **SHOULD** maintain an explicit “control-to-policy” mapping so audits can trace each baseline control to: policy assignment, evidence source, and exemption path.

Implementation Notes

- Keep the baseline small enough to sustain (few initiatives, clear ownership, minimal overlaps). Expand via versioned releases.
- Where deny is not technically feasible for a resource type, you **MUST** use audit plus compensating controls and measure the gap.

Metrics

- Baseline compliance rate (excluding valid, unexpired exemptions): **target ≥ 95%**
- Past-due exemption count for baseline controls: **target 0**

Operationalization

- Ownership: security team defines baseline outcomes; platform team implements initiatives/assignments as IaC; workload teams remediate resources.
- Cadence: monthly baseline review; quarterly baseline version release (or faster for critical risks).
- Tooling: Azure Policy compliance reports; Defender for Cloud regulatory compliance dashboard; Git-based policy-as-code pipeline (per s06).

Anti-patterns

- Treating the baseline as “documentation only” with no enforceable mapping.
- Large, untested deny rollouts without an adoption window and migration guidance.
- Using subscription-local policies as the default instead of MG inheritance.

7.4.2 Defender for Cloud governance

Purpose
Use Defender for Cloud as the system-of-record for security posture management and workload protections, with clear plans and measurable outcomes.

Scope
Defender for Cloud plans, recommendations, secure score governance, and remediation workflows.

Decisions
Defender for Cloud is used for posture signals and prioritized remediation; Azure Policy remains the primary enforcement mechanism.

Standards/Controls

- You **MUST** enable Defender for Cloud at the required scopes and define which Defender plans are mandatory by workload class (platform team implements; security team approves).
- You **MUST** define a recommendation governance model:
  - Recommendations are triaged into: **must-fix**, **plan-fix**, **accept-risk** (via time-bound exemption per s06), or **not-applicable** (with evidence).
- You **SHOULD** use secure score trends to validate program direction, but you **MUST** not manage to secure score alone when it conflicts with your baseline control outcomes.

Implementation Notes

- Use a backlog model: recommendations become work items with an owner, target date, and verification criteria.
- Ensure Defender signals are available to your central monitoring model (see s09) for alerting and reporting.

Metrics

- % of “must-fix” recommendations past due: **target ≤ 5%**
- Median time to remediate “must-fix” recommendations: **target defined by the security team severity/SLA model**

Operationalization

- Ownership: security team owns recommendation policy and risk acceptance; workload teams remediate; platform team maintains Defender scope enablement.
- Cadence: weekly triage for new high-severity items; monthly posture review.
- Tooling: Defender for Cloud recommendations export; work item system integration; Azure Policy compliance overlay.

Anti-patterns

- Turning on Defender plans without defining ownership for remediation.
- Treating “not applicable” as a default classification to reduce noise.
- Using secure score as a KPI without a denominator/coverage statement.

7.4.3 Key Vault governance (secrets/keys/certificates)

Purpose
Reduce credential leakage and key misuse by standardizing storage, access, rotation, and auditability.

Scope
Azure Key Vault usage for secrets, keys, and certificates across platform and workloads.

Decisions
Key Vault is the default for cloud-hosted secrets and keys; exceptions are time-bound and reviewed.

Standards/Controls

- Workload teams **MUST** store application secrets in Key Vault (or an approved equivalent) and **MUST** avoid secrets in code, pipeline variables, or configuration files.
- You **MUST** enforce Key Vault access via least privilege using RBAC and PIM (per s04), preferring managed identities where feasible.
- You **MUST** implement rotation standards:
    - Secrets and certificates **MUST** have a defined rotation interval and owner.
    - Keys used for encryption **MUST** have an explicit rotation plan, with downtime impact assessed.
- You **SHOULD** use HSM-backed keys where regulatory requirements, high-value assets, or tenant risk warrants it; the security team **MUST** define the criteria.

Implementation Notes

- Prefer per-application Key Vault instances when you need RBAC isolation and operational independence; consolidate only when the RBAC boundary is identical.
- Ensure Key Vault diagnostic settings are enabled and routed per s09.

Metrics

- % of production workloads with secrets sourced from Key Vault: **target 100%**
- % of secrets/certificates past rotation interval: **target 0**

Operationalization

- Ownership:
  - Workload teams own secret inventories and rotation execution.
  - Platform team provides Key Vault patterns/modules.
  - Security team sets cryptographic standards and approves exceptions.
- Cadence: monthly rotation compliance review; quarterly access review for Key Vault roles (aligned to s04).
- Tooling: secret scanning in repositories; Key Vault rotation reporting; policy compliance for Key Vault settings where applicable.

Anti-patterns

- Shared “mega-vault” spanning unrelated RBAC boundaries.
- Long-lived secrets with no owner or rotation plan.
- Manual secret distribution outside managed identity patterns.

7.4.4 Data protection (classification, encryption, key ownership)

Purpose
Ensure data handling is consistent and enforceable across teams, with clear accountability for encryption and key management.

Scope
Data classification tagging, encryption requirements, and key ownership model across data stores and services.

Decisions
Classification drives required controls; workload teams own classification accuracy and adherence.

Standards/Controls

- Workload teams **MUST** classify data using your `DataClassification` tag (per s05) and treat misclassification as a security defect.
- You **MUST** encrypt data at rest for all supported services.
- Where customer-managed keys (CMK) are required by policy or regulation, workload teams **MUST** implement CMK using Key Vault-managed keys.

- The security team **MUST** define when CMK is mandatory and when platform-managed keys are acceptable.

7.4.4.1 Minimum control expectations by data classification (governance intent)

DataClassification	Minimum expectations (governance-level)
`public`	Encryption at rest where supported; baseline logging; public endpoints allowed where approved by baseline controls
`internal`	Encryption at rest; baseline logging; public endpoints limited to approved patterns (WAF where applicable)
`confidential`	Prefer private access patterns; stronger monitoring/alerting; review public exposure exceptions
`restricted`	Private access required for supported PaaS; CMK required where supported/mandated; no unmanaged public exposure; tighter egress controls per s08 where applicable

Implementation Notes

- For services with limited CMK support, you **MUST** document compensating controls and track gaps.
- Ensure key custody and break-glass procedures do not create uncontrolled access paths (coordinate with s04 break-glass controls).

Metrics

- % of restricted/confidential workloads using required encryption configuration: **target ≥ 95%** (excluding valid exemptions)

Operationalization

- Ownership:
  - Workload teams own classification accuracy and CMK implementation where required.
  - Platform team maintains reusable modules and reference architectures.
  - Security team owns policy requirements, cryptographic standards, and exception decisions.
  - CMK lifecycle responsibility **MUST** be explicit per workload: key creation, rotation execution, and access reviews must be assigned to a team (not individuals).

- Cadence: quarterly audit sampling for classification accuracy; monthly review of encryption compliance exceptions.
- Tooling: Azure Policy compliance; data service configuration reporting; Key Vault key usage logs (via s09 diagnostics).

Anti-patterns

- “CMK everywhere” mandates without criteria, causing operational fragility.
- Data classification treated as a one-time exercise with no review loop.
- Keys owned by individuals rather than teams/managed processes.

7.4.5 Vulnerability management and container guardrails

Purpose
Reduce exploitability through consistent scanning, patch posture, and enforceable container/image minimums.

Scope
Vulnerability scanning expectations, container image hygiene, and minimum guardrails enforceable via policy and pipelines.

Decisions
You enforce minimums centrally; workload teams choose implementation details within guardrails.

Standards/Controls

- Workload teams **MUST** maintain a vulnerability remediation process with defined severity tiers and target remediation windows.
- For containerized workloads:
    - You **MUST** prohibit deployment of images from untrusted sources (enforced via approved registry patterns and pipeline gates).
    - You **SHOULD** require image scanning and provenance checks for production deployments.
- Exemptions **MUST** follow the workflow defined in s06 (time-bound, least-scope, tracked as exemption debt).

Implementation Notes

- Keep governance guardrails minimal and testable; avoid prescribing a single scanning product unless your operating model requires it.
- Ensure findings are routable to owners via tags and service ownership metadata (per s05).

Metrics

- % of critical vulnerabilities past due in production: **target defined by the security team severity/SLA model (with a documented default)**
- % of production deployments using approved image sources: **target 100%**

Operationalization

- Ownership: workload teams remediate; security team defines severity windows and acceptance criteria; platform team supplies pipeline templates for golden path.
- Cadence: weekly review of critical findings; monthly governance reporting.
- Tooling: registry controls, pipeline policy gates, Defender for Cloud container recommendations where enabled.

Anti-patterns

- Scanning without remediation ownership (“findings as noise”).
- Exceptions granted for “temporary” bypasses that become permanent.
- Allowing direct production deployments outside the golden path.

7.5 Implementation Notes

- Keep s07 control outcomes stable; evolve implementation via s06 policy-as-code releases (versioned, peer-reviewed).
- Where s07 references controls that are enforced elsewhere (identity in s04, logging in s09, hierarchy in s05), you **MUST** treat those sections as the source of truth and avoid duplicating mechanisms here.

7.6 Operationalization

- Governance requires a control-to-evidence map. The security team **MUST** maintain a single mapping artifact that ties: baseline control → policy initiative/assignment → evidence source → exemption path.

7.7 Anti-patterns (section-wide)

- Using policy exemptions as an alternative to remediation (building persistent exemption debt).
- Splitting standards across documents without a single control-to-evidence map.

- Applying deny controls before providing a working golden path and adoption window.

7.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Secure baseline compliance rate	Compliant resources / total in-scope resources, excluding valid unexpired exemptions	≥ 95%	Azure Policy compliance	Monthly	Security team
Past-due exemption count (security controls)	# exemptions with expiresOn < now	0	Exemption registry (s06)	Weekly	Security team
Must-fix recommendations past due	% of must-fix items beyond SLA window	≤ 5%	Defender for Cloud export + work items	Monthly	Security team
Key rotation compliance	% secrets/keys/certs within defined rotation interval	100%	Key Vault inventory + logs	Monthly	Workload teams

8. s08 — Network Governance: Connectivity, Segmentation, and DNS

8.1 Purpose

Standardize how your teams design and operate network connectivity, segmentation, and name resolution so you reduce lateral movement risk, preserve workload autonomy within guardrails, and keep operations sustainable at scale.

8.2 Scope

Covers:

- Topology selection: hub-spoke and Azure Virtual WAN (vWAN)
- Address space governance and IP address management (IPAM) process
- Ingress/egress controls: Web Application Firewall (WAF), Network Security Groups (NSGs), User-defined routes (UDRs), and Azure Firewall
- Private Endpoint and Private DNS Zone governance
- Peering, routing, and on-prem connectivity (ExpressRoute/VPN)

Excludes:

- Workload application-layer network design (owned by workload teams)
- Non-Azure networks beyond on-prem connectivity interfaces

Dependencies:

- s05 (subscription strategy, naming/tagging, ownership boundaries)
- s06 (policy-as-code lifecycle and exemption workflow)
- s07 (security governance and risk acceptance; data classification expectations)

8.3 Decisions

- **Topology choice:** You **SHOULD** default to hub-spoke for most Landing Zones; you **MAY** use vWAN when you have many branches, complex routing, or rapid site onboarding needs that outweigh added platform complexity.
- **Centralized shared services:** The platform team **MUST** centralize shared connectivity services (gateway, firewall, DNS resolver, shared private DNS) in the hub (or vWAN hub) to reduce drift and duplicate spend.
- **Progressive enforcement:** You **MUST** apply the progressive enforcement model (audit policy → DeployIfNotExists → deny policy) for network governance controls where feasible; deny policy is reserved for non-negotiable, high blast-radius requirements.
- **Private access-first:** You **SHOULD** prefer Private Endpoint for supported PaaS connectivity to reduce public exposure; exceptions **MUST** use time-bound policy exemptions per s06.
- **Deterministic routing:** You **MUST** standardize routing so workload teams can predict egress paths (forced tunneling via firewall where required) and avoid hidden transitive connectivity.

- **Controlled environments definition:** “Controlled environments” **MUST** include, at minimum, production subscriptions and any workload handling `confidential` or `restricted` data (per s05 `DataClassification`).

8.4 Standards/Controls

8.4.1 Topology and connectivity standards

Purpose
Ensure consistent connectivity and shared service placement.

Scope
Hub-spoke and vWAN implementations across Landing Zones.

Decisions
Use approved reference topologies; centralize shared services.

Standards/Controls

- You **MUST** implement one of the approved reference topologies:
  - Hub-spoke with a dedicated Hub VNet hosting shared services, or
  - vWAN with centrally managed hubs and routing policies.
- You **MUST** place shared inbound/outbound controls (WAF/ingress tier and Azure Firewall/egress tier) under platform team ownership.
- You **MUST** restrict VNet peering to “spoke-to-hub” by default; spoke-to-spoke peering **MUST NOT** be used unless explicitly approved and documented with routing impact.

Implementation Notes

- Use separate subnets in the hub for gateway, firewall, and DNS resolver components to keep routing and permissions auditable.

Metrics

- See section Metrics.

Operationalization

- See section Operationalization.

Anti-patterns

- Creating multiple hub VNets per team without a requirement.

- Spoke-to-spoke peering as the default pattern (“it’s faster”) without routing and blast-radius review.

8.4.2 Segmentation and routing standards (NSG/UDR)

Purpose
Reduce lateral movement and enforce predictable egress.

Scope
Spoke subnet design, NSGs, and UDR patterns.

Decisions
Forced tunneling and controlled peering.

Standards/Controls

- Workload teams **MUST** segment spokes at minimum into `app` and `data` subnets (or equivalent) with NSGs applied.
- In controlled environments, you **MUST** use UDRs where required to force egress via Azure Firewall (or approved equivalent), with documented allowed destinations.
- You **MUST** deny “any-to-any” east-west rules across subnets unless justified and reviewed.

Implementation Notes

- Not all segmentation outcomes are enforceable purely via Azure Policy. Where policy cannot enforce intent, you **MUST** enforce via:
  - approved IaC modules (golden path) and
  - architecture review gates for exceptions.
- Keep NSG rules minimal and derive them from explicit app dependency maps; avoid broad “temporary” rules.

Metrics

- See section Metrics.

Operationalization

- Use reusable NSG/route-table modules; require PR review for rule changes in controlled environments.

Anti-patterns

- “Temporary allow any” rules with no expiry.

- Egress paths that vary by team because routes are hand-configured.

8.4.3 Ingress and egress control standards

Purpose
Centralize high-risk controls and reduce exposure.

Scope
Internet ingress, outbound internet, and on-prem egress paths.

Decisions
Controlled ingress via WAF tier; egress via firewall.

Standards/Controls

- Internet-facing workload endpoints **MUST** be protected by a WAF (Application Gateway WAF or Front Door WAF) with a documented policy baseline.
- In controlled environments, outbound internet egress **MUST** traverse Azure Firewall (or approved equivalent) with logging enabled.
- Public IP assignment in workload spokes **SHOULD** be avoided; if required, it **MUST** be justified and time-bound via policy exemption where enforcement exists.

Implementation Notes

- Separate “platform ingress” components from workload spokes to preserve least privilege and simplify incident response.

Metrics

- See section Metrics.

Operationalization

- Use a change-controlled firewall/WAF policy process with owners and rollback plans.

Anti-patterns

- Allowing workloads to egress directly to the internet because firewall onboarding is “too slow”.

8.4.4 Private Endpoint and Private DNS governance

Purpose
Make private access scalable without DNS fragmentation.

Scope
Private Endpoint usage and Private DNS Zone lifecycle.

Decisions
Centralized Private DNS with governed linking.

Standards/Controls

- You **MUST** use Private Endpoint for supported PaaS where private connectivity is required by data classification or exposure policy.
- The platform team **MUST** own shared Private DNS Zones and their links to VNets; workload teams **MUST** request zone links through a standard intake.
- You **MUST** standardize DNS resolution via a hub DNS resolver pattern (central forwarding/resolution) to avoid split-brain DNS.

Implementation Notes

- Maintain a "supported private endpoint + DNS zone" matrix as part of the golden path; not all services support the same DNS patterns.

Metrics

- See section Metrics.

Operationalization

- Track zone links and private endpoint inventory centrally; review requested links for scope creep.

Anti-patterns

- Creating Private DNS Zones per workload team, causing conflicting DNS and operational fragmentation.

8.4.5 Address space governance and IPAM process

Purpose
Prevent overlap and simplify future mergers/connectivity expansion.

Scope
VNet/subnet address planning and allocation workflow.

Decisions
Central authority for allocations; workload autonomy within assigned ranges.

Standards/Controls

- The platform team **MUST** maintain an IPAM registry for all VNet address spaces and delegated allocations.
- Workload teams **MUST** request address allocations before creating VNets; ad-hoc address selection **MUST NOT** be used.
- Address space changes **MUST** follow change control with routing impact assessment.

Implementation Notes

- Use reserved blocks per environment and region to reduce re-IP events.

Metrics

- See section Metrics.

Operationalization

- Maintain a ticketed workflow and a versioned source-of-truth for allocations.

Anti-patterns

- Ad-hoc VNet address allocation without IPAM, resulting in overlaps during on-prem integration.

8.4.6 Peering, on-prem connectivity, and routing control

Purpose
Keep connectivity auditable and avoid emergent transitive paths.

Scope
Peering, ExpressRoute/VPN, route propagation, and default routes.

Decisions
Centralized gateways; controlled propagation.

Standards/Controls

- The platform team **MUST** own ExpressRoute/VPN gateways and on-prem routing integration in the hub/vWAN.
- Route propagation **SHOULD** be explicitly controlled; avoid “propagate everything everywhere” designs.
- Workload teams **MUST NOT** deploy independent gateways without exception approval.

Implementation Notes

- Validate routing changes in a non-production hub first to reduce blast radius.

Metrics

- See section Metrics.

Operationalization

- Require change records for propagation changes and peering exceptions.

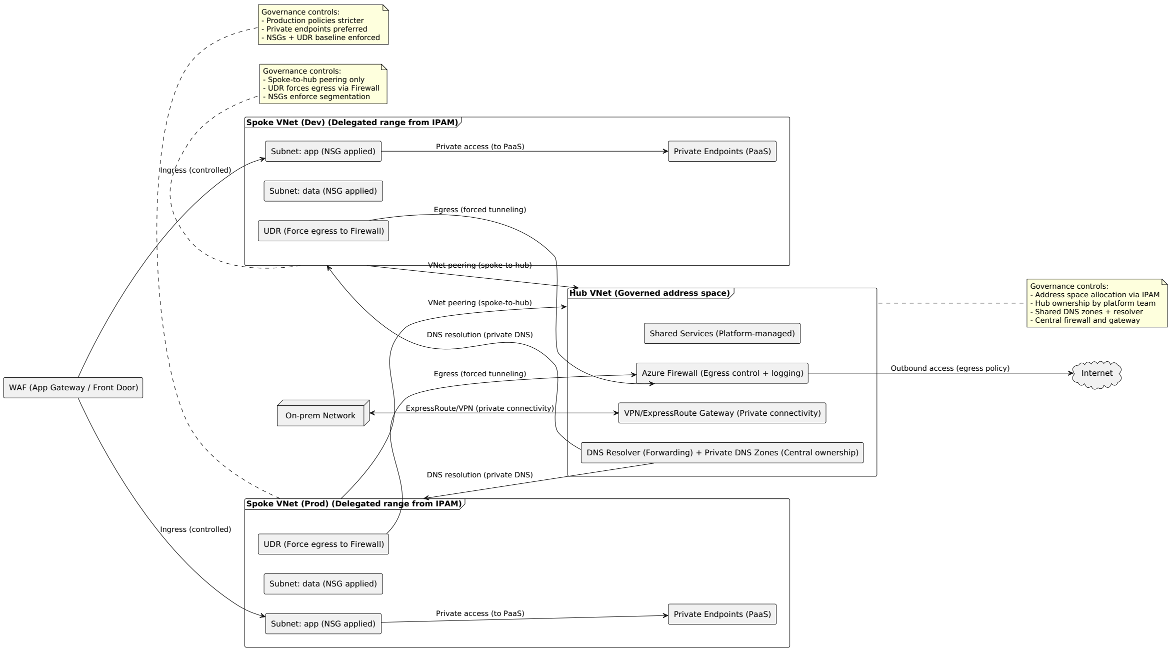
Anti-patterns

- “Propagate everything everywhere” route designs with unclear transitive impacts.

8.5 Implementation Notes

- Treat network baselines as part of your versioned Landing Zone product. Changes to hub routing, DNS, or firewall rules are platform changes and **MUST** be released with change notes and rollback plans.
- Align enforcement to risk:
    - Start with audit policy to measure impact,
    - Use DeployIfNotExists where remediation is deterministic,
    - Use deny policy for “no public exposure” or “forced egress” controls where breaking changes are unacceptable.

8.5.1 Hub-Spoke Network Governance Reference (diagram)


Governance controls:
- Production policies stricter
- Private endpoints preferred
- NSGs + UDR baseline enforced
Governance controls:
- Spoke-to-hub peering only
- UDR forces egress via Firewall
- NSGs enforce segmentation
Spoke VNet (Dev) (Delegated range from IPAM)
Subnet: app (NSG applied)
Private access (to PaaS)
Private Endpoints (PaaS)
Ingress (controlled)
Subnet: data (NSG applied)
Egress (forced tunneling)
UDR (Force egress to Firewall)
VNet peering (spoke-to-hub)
Hub VNet (Governed address space)
Governance controls:
- Address space allocation via IPAM
- Hub ownership by platform team
- Shared DNS zones + resolver
- Central firewall and gateway
VNet peering (spoke-to-hub)
Shared Services (Platform-managed)
DNS resolution (private DNS)
Egress (forced tunneling)
Azure Firewall (Egress control + logging)
Outbound access (egress policy)
Internet
WAF (App Gateway / Front Door)
On-prem Network
ExpressRoute/VPN (private connectivity)
VPN/ExpressRoute Gateway (Private connectivity)
DNS Resolver (Forwarding) + Private DNS Zones (Central ownership)
DNS resolution (private DNS)
Spoke VNet (Prod) (Delegated range from IPAM)
Ingress (controlled)
UDR (Force egress to Firewall)
Subnet: data (NSG applied)
Private access (to PaaS)
Subnet: app (NSG applied)
Private Endpoints (PaaS)

8.6 Operationalization

Purpose

Make network governance durable through clear ownership, predictable cadence, and tooling.

Scope

Network baseline operations for hub/spokes, DNS, routing, and private endpoints.

Decisions

Centralize shared services; progressive enforcement.

Standards/Controls

- Ownership:
    - Platform team **Accountable** for hub/vWAN, firewall, gateways, DNS resolver, Private DNS Zones, and baseline policies.
    - Workload teams **Responsible** for spoke VNets, subnets, NSGs, and application connectivity within assigned patterns.
- Cadence:
    - You **SHOULD** review address allocations and peering exceptions monthly.
    - You **MUST** review firewall policy/rules at least quarterly (or more frequently for regulated environments).

- Tooling:
  - You **MUST** implement network baselines using Infrastructure as Code (IaC) and enforce drift detection (per s06).
  - You **SHOULD** use a ticketed intake for: new VNet address blocks, Private DNS Zone links, and spoke-to-spoke exception requests.

Implementation Notes

- Keep a single “network golden path” repo/module set; avoid per-team forked network modules.

Metrics

- See section Metrics.

8.7 Anti-patterns

- Spoke-to-spoke peering as the default pattern (“it’s faster”) without routing and blast-radius review.
- Allowing workloads to egress directly to the internet because firewall onboarding is “too slow”.
- Creating Private DNS Zones per workload team, causing conflicting DNS and operational fragmentation.
- Ad-hoc VNet address allocation without IPAM, resulting in overlaps during on-prem integration.
- Mixing production and non-production workloads in the same spoke without segmentation and ownership clarity.

8.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Network baseline compliance	Compliant network resources / total in-scope network resources (hub, spokes, NSGs, UDRs, Private Endpoint/DNS links)	≥ 90%	Azure Policy compliance	Monthly	Platform team
	Spoke VNets with required UDR egress	≥ 95%		Monthly	Platform team

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Forced-egress coverage	via Azure Firewall / total spokes in enforced environments		Route tables + policy compliance		
Private Endpoint adoption (where required)	Workloads in “private-required” data classification using Private Endpoint / total applicable workloads	≥ 90%	Private Endpoint inventory + tagging	Quarterly	Security team
DNS governance compliance	VNets using approved DNS resolver + linked to governed Private DNS Zones / total in-scope VNets	≥ 95%	DNS resolver config + zone link inventory	Monthly	Platform team
Exception debt (network)	Count of past-due policy exemptions related to network controls	0	Exemption registry (per s06)	Monthly	Security team

9. s09 — Observability and Operations Governance (Monitoring, Logging, Backup)

9.1 Purpose

Ensure your teams have consistent, auditable observability across the **management plane** and **workload plane**, with reliable signals for incident response and sustainable operations (cost, access, retention).

9.2 Scope

Covers:

- **Azure Monitor** signals: logs, metrics, alerts, action groups, dashboards/workbooks.

- **Log Analytics workspace** strategy, access controls, and retention tiers.
- **Diagnostic settings** baselines for in-scope resource types.
- Backup and **disaster recovery (DR)** governance using **recovery point objective (RPO)** / **recovery time objective (RTO)** tiers, vault strategy, and testing evidence.
- Runbooks and automation for response and remediation.

Excludes:

- Workload-specific application performance instrumentation details (owned by workload teams; governed here only via minimum outcomes and metadata requirements).

Dependencies: **s06** (Policy-as-code and exemptions), **s08** (network governance affecting private access/log routing).

9.3 Decisions

- You **MUST** centralize observability standards at **Management Group** scope, implemented through **Azure Policy** (per s06).
- You **MUST** standardize diagnostic settings for all **in-scope resource types that support diagnostic settings**, using progressive enforcement (**audit policy** → **DeployIfNotExists** → **deny policy**).
- You **MUST** separate **platform logs** from **workload logs** when an RBAC boundary, retention boundary, or regulated-data boundary requires it; otherwise you **SHOULD** consolidate to reduce operational overhead and cost.
- You **MUST** route actionable alerts to a single **IT service management (ITSM)/incident management system** integration path for traceability.
- You **MUST** define backup/DR tiers using **RPO/RTO** and test restores on a defined cadence with retained evidence.

9.4 Standards/Controls

9.4.1 Log Analytics workspace strategy (routing, access, retention)

- Your teams **MUST** use governed **Log Analytics workspace**(s) with:
    - RBAC aligned to least privilege (platform team administers platform workspace; workload teams administer workload workspace access for their teams).

    - Standard retention tiers (minimum defaults below).
- Workspace boundary rules:
    - You **MUST** use separate workspaces when **any** of the following differ: RBAC boundary, retention requirement, regulated-data boundary.
    - You **SHOULD** otherwise consolidate by environment/region pair to reduce query fragmentation and ingestion cost.

Retention tiers (default minimums)

- You **MUST** retain operational logs in Log Analytics for **≥ 180 days** unless regulatory requirements mandate longer.
- Long-term retention for audit evidence **MAY** be implemented via **Azure Monitor Logs archive and/or export to Storage** (implementation depends on service capability and audit needs). You **MUST** document the chosen method per data class and workload tier.

9.4.2 Diagnostic settings baseline (policy-driven)

- Diagnostic settings **MUST** be configured (where supported) to send:
    - Logs to the governed Log Analytics workspace (platform or workload as appropriate).
    - Security-relevant streams **SHOULD** also be forwarded to **security information and event management (SIEM)** via Event Hub where required by the security team.
    - Long-term retention **MAY** also export to Storage for evidence retention.
- Because not all Azure services support the same sinks simultaneously, the platform team **MUST** maintain a **resource-type capability matrix** (per golden path) defining required destinations by resource type.

9.4.3 Alerting and SLO governance

- Workload teams **MUST** define service level objectives (**SLOs**) for production workloads and map them to alerts.
- Alert rules **MUST** include standard metadata: `serviceName`, `ownerTeam`, `severity`, `runbookLink`, `customerImpact`, and `routingTarget` (ITSM queue/service).
- The platform team **MUST** provide baseline platform alerts (identity, policy, network, shared services) with documented runbooks.

9.4.4 Backup and DR governance

- The platform team **MUST** publish standard **RPO/RTO tiers** (e.g., Tier 0/1/2/3) and approved backup mechanisms per workload type.
- Workload teams **MUST** select a tier, implement backups accordingly, and meet restore-test cadence.
- Restore testing **MUST** produce evidence (ticket IDs, logs, timestamps, and outcome) retained per audit requirements.

9.5 Implementation Notes

- Map **platform Log Analytics workspace** to the management/operations subscription; map **workload Log Analytics workspace** to workload subscriptions where RBAC separation is needed (align to s05 subscription strategy).
- Enforce diagnostic settings with **DeployIfNotExists** where deterministic; use **deny policy** for high blast-radius omissions (e.g., required logging on production-critical resource types), after an adoption window per s06.
- Put ingestion cost guardrails in place early: per-workspace budgets/alerts and retention caps by environment.

9.6 Operationalization

9.6.1 Ownership

- Platform team: workspace governance, baseline diagnostic settings and alerting, runbook library, platform DR standards.
- Security team: SIEM forwarding requirements, alert severity model alignment, approval for exceptions affecting security telemetry.
- Workload teams: SLOs, workload alert rules, backup/restore execution, and evidence capture.

9.6.2 Cadence

- **Weekly:** review new high-severity alerts/noise reduction backlog.
- **Monthly:** workspace cost/ingestion review; retention and table usage review.
- **Quarterly:** DR restore testing (minimum) for production tiers; validate alert metadata/runbook links.
- **On change:** update capability matrix when new resource types are introduced into the golden path.

9.6.3 Tooling

- Azure Policy (assignments/initiatives/exemptions managed as IaC in Git per s06).
- Azure Monitor (alerts/action groups), Log Analytics workspaces.
- Automation (Automation Account/Functions/Logic Apps) for remediation/runbooks and evidence capture into ITSM.

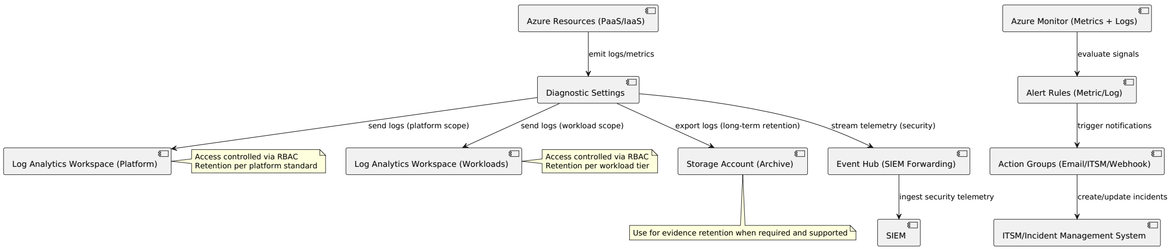
9.7 Anti-patterns

- Shipping logs to unmanaged team-owned workspaces with inconsistent retention/RBAC.
- Alerting without an owner, runbook link, or routing to ITSM (alerts become noise).
- Enabling “everything everywhere” diagnostics without cost guardrails or table-level review.
- Skipping restore tests and treating backup configuration as proof of recoverability.
- Using indefinite observability exemptions instead of time-bound policy exemptions (creates exception debt).

9.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Diagnostic settings coverage	Compliant resources / total in-scope resources that support diagnostic settings	≥ 90% (ramp to ≥ 95%)	Azure Policy compliance	Monthly	Platform team
Alert actionability rate	Alerts with required metadata fields / total production alerts	≥ 95%	Alert rule inventory + metadata checks	Monthly	Platform team
Mean time to acknowledge (MTTA)	Time from alert firing to incident	Defined per	ITSM timestamps	Monthly	Operations (platform/ workload)

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
	acknowledged in ITSM	severity standard			
Restore test compliance	Successful restore tests completed / required restore tests by tier	100%	ITSM + test evidence store	Quarterly	Workload teams
Observability exception debt	Past-due policy exemptions related to logging/ monitoring/ backup	0 past-due	Exemption registry (s06)	Monthly	Security team


Azure Resources (PaaS/IaaS)
emit logs/metrics
Diagnostic Settings
send logs (platform scope)
send logs (workload scope)
export logs (long-term retention)
stream telemetry (security)
Log Analytics Workspace (Platform)
Access controlled via RBAC
Retention per platform standard
Log Analytics Workspace (Workloads)
Access controlled via RBAC
Retention per workload tier
Storage Account (Archive)
Use for evidence retention when required and supported
Event Hub (SIEM Forwarding)
ingest security telemetry
SIEM
Azure Monitor (Metrics + Logs)
evaluate signals
Alert Rules (Metric/Log)
trigger notifications
Action Groups (Email/ITSM/Webhook)
create/update incidents
ITSM/Incident Management System

10. s10 — Cost Governance and FinOps Integration

10.1 Purpose

You establish cost accountability and predictable spend management by integrating FinOps practices into your Azure governance model, using subscription boundaries and mandatory tags as allocation dimensions and running a repeatable optimization cycle.

10.2 Scope

You cover:

- Cost allocation across **Management Group**, **Subscription**, and workload scopes.

- Chargeback/Showback, budgets, alerts, anomaly detection, and variance reviews.
- Optimization governance (rightsizing, schedules, storage tiering, egress controls).
- Reservation/savings plan governance (ownership, approval, and reporting).

You do not cover:

- Detailed tagging schema definitions (see **s05** for required tags and allowed values).
- Central logging architecture and diagnostic settings standards (see **s09**).

10.3 Decisions

- You **MUST** allocate spend primarily by **Subscription** (accountability boundary) and secondarily by **tags** (business dimensions and ownership routing).
  Rationale: subscriptions scale and reduce ambiguity; tags enable unit-cost reporting and operational routing.
- You **MUST** enforce required tags for in-scope resources to reach tag compliance targets.
  Trade-off: tag enforcement can slow ad-hoc provisioning; you offset with a supported **golden path** (templates/pipelines).
- You **MUST** run a monthly FinOps variance and optimization review, and you **SHOULD** run near-real-time alerting for budget breaches and anomalies.
  Rationale: monthly governance controls prevent drift; near-real-time signals reduce “surprise spend.”
- You **MUST** define clear ownership for reservations/savings plans and optimization actions (FinOps drives analysis; platform/workload teams execute changes).
  Trade-off: centralized purchasing increases leverage but can reduce product autonomy; you balance this with transparent approval and showback.

10.4 Standards/Controls

- **Tagging and allocation**

  ◦ Your teams **MUST** meet required tag standards defined in **s05**, including `Owner`, `CostCenter`, `App`, `Environment`, `DataClassification`.

    - Tag enforcement **MUST** be implemented via **Azure Policy** at **Management Group** scope where feasible, using progressive enforcement (**Audit** → **DeployIfNotExists** → **Deny**) for production readiness controls.
      Note: FinOps does not “enforce budgets” via Azure Policy; budget controls are implemented via automation and governance cadence.

- **Budgets, alerts, and anomaly detection**

    - Every in-scope subscription **MUST** have:
        - A budget aligned to forecasted spend for the environment and workload class.
        - Alerting to the responsible `Owner`/`CostCenter` distribution and the FinOps function.
    - Anomaly detection **SHOULD** be enabled and triaged within an agreed response window (define in Operationalization).

- **Optimization governance**

    - Optimization actions **MUST** be tracked as work items with owner, expected savings, risk/impact notes, and implementation date.
    - High-impact changes (e.g., right-sizing production compute, storage lifecycle changes affecting retention, egress routing changes) **MUST** follow your change control and rollback standards.

- **Reservations/savings plans**

    - Reservations/savings plans **MUST** have a named accountable owner (FinOps or platform team, per your operating model).
    - Purchase and scope changes **MUST** have documented decision rationale (forecast, utilization assumptions, and break-even horizon).

- **Evidence / auditability notes**

    - You **MUST** be able to evidence: tag compliance, budget coverage, alert routing, monthly variance review outputs, and realized savings vs forecast.

10.5 Implementation Notes

- Align all reporting dimensions to **s05** tag allowed values; do not create a parallel FinOps taxonomy.

- Start with showback; move to chargeback only after tag compliance and budget coverage stabilize (to avoid cost disputes driven by bad metadata).
- Use separate action backlogs for:
    - “No-regrets” optimizations (schedules in non-prod, stale resource cleanup).
    - “Risk-managed” optimizations (production right-sizing, storage tier changes).

10.6 Operationalization

10.6.1 Ownership

- **FinOps Analyst:** Responsible for reporting, anomaly triage, recommendations, and facilitation of monthly reviews.
- **Product/BU Owner:** Accountable for budget adherence and approving material optimization actions for their spend.
- **Platform team:** Responsible for platform-level optimizations (shared services, reservations administration where centralized, guardrail updates).
- **Tagging/Subscription Standards (s05 owners):** Consulted for schema changes and enforcement rollout.

10.6.2 Cadence

- Near-real-time: budget alerts/anomaly triage (daily business hours, or per on-call model).
- Monthly: variance review, optimization decision log, and forecast updates.
- Quarterly: reservation/savings plan strategy review and coverage targets.

10.6.3 Tooling

- Source of truth: **Azure Cost Management** exports/reports.
- Enforcement: **Azure Policy** for tag compliance; automation (pipelines/runbooks) for budget creation and alert routing.
- Work tracking: a ticket/work item system for recommendations → approvals → implementation → measured savings.

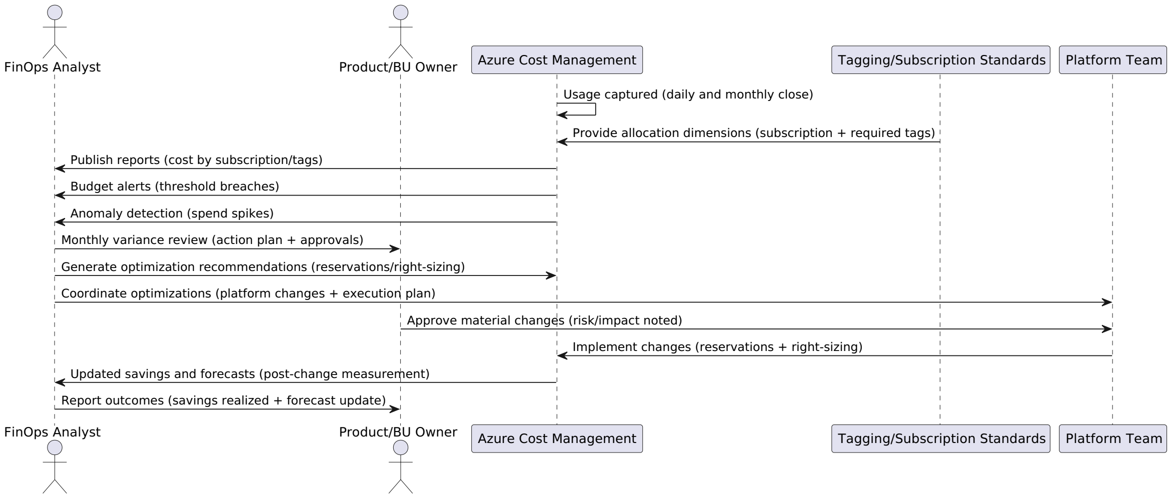
10.7 Anti-patterns

- Treating tags as “optional metadata” and then attempting chargeback/showback.

- Buying reservations/savings plans without ownership, utilization targets, and a review cadence.
- Optimizing only during incidents (“panic right-sizing”) instead of running a managed monthly cycle.
- Allowing inconsistent environment taxonomies (e.g., redefining `Environment` values outside **s05**) that break reporting.

10.8 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Tag compliance	Compliant resources / total in-scope resources (per required tags in s05)	≥95%	Azure Policy compliance + CM exports	Weekly + monthly	Platform team
Cost variance	(Actual spend - forecast) / forecast by CostCenter and App	≤±10%	Azure Cost Management	Monthly	FinOps Analyst
Budget coverage	Subscriptions with budgets and alerts / total in-scope subscriptions	100%	Azure Cost Management	Monthly	FinOps Analyst
Reservation/ savings plan coverage	% of eligible spend covered by reservations/ savings plans (org-defined eligibility rules)	Org-defined	Azure Cost Management	Monthly + quarterly	FinOps Analyst
Waste	Estimated waste spend / total spend (idle, orphaned, unused)	Downward trend	Azure Cost Management + inventory	Monthly	Product/ BU Owner


FinOps Analyst
Product/BU Owner
Azure Cost Management
Tagging/Subscription Standards
Platform Team
Usage captured (daily and monthly close)
Provide allocation dimensions (subscription + required tags)
Publish reports (cost by subscription/tags)
Budget alerts (threshold breaches)
Anomaly detection (spend spikes)
Monthly variance review (action plan + approvals)
Generate optimization recommendations (reservations/right-sizing)
Coordinate optimizations (platform changes + execution plan)
Approve material changes (risk/impact noted)
Implement changes (reservations + right-sizing)
Updated savings and forecasts (post-change measurement)
Report outcomes (savings realized + forecast update)
FinOps Analyst
Product/BU Owner
Azure Cost Management
Tagging/Subscription Standards
Platform Team

11. Platform Engineering: Landing Zones, Templates, and Guardrailed Self-Service

11.1 Purpose

You productize Landing Zones as repeatable, versioned platform capabilities so your teams can onboard workloads quickly while keeping governance centralized, auditable, and sustainable.

11.2 Scope

You cover:

- Landing Zone baseline (identity, network, management, security, policy) as a versioned product owned by the platform team
- Infrastructure as Code (IaC) modules, template standards, and a module registry
- GitOps workflow for platform changes (policy, identity/RBAC, shared services)
- Guardrailed self-service: subscription vending, service catalog, and optional approvals
- Exception handling via time-bound policy exemptions (see policy exemption definition in the glossary)

Out of scope:

- Workload application architecture and SDLC pipelines beyond the supported “golden path” interfaces

11.3 Decisions

- You **MUST** treat the Landing Zone as a **versioned product**: the platform team owns baseline changes; workload teams consume via IaC.
- You **MUST** centralize baseline governance at **Management Group** scope for inheritance and reduced drift; subscription-level policy is limited to *overlays* for workload-class/environment deltas.
- You **MUST** provide guardrailed self-service to reduce lead time while preserving non-negotiable controls (deny policy where blast radius is high).
- You **SHOULD** implement progressive enforcement for new controls (audit policy → DeployIfNotExists → deny policy) with an adoption window and migration guidance before deny.

11.4 Standards/Controls

- Subscription vending
  - You **MUST** provision subscriptions only through an approved service catalog/request portal and automation (no ad-hoc manual creation).
  - Provisioning **MUST** place subscriptions under the correct **Management Group** and apply inherited baseline policy initiatives from Management Group scope.
  - Provisioning **MUST** apply only approved **subscription overlay** policy initiatives (when required) and document the rationale (workload class/ environment delta).
- IaC and module standards
  - The platform team **MUST** maintain a module registry (internal) with versioning and deprecation policy.
  - Workload teams **SHOULD** use the platform-provided modules and templates for supported patterns (golden paths).
  - Workload teams **MAY** use custom modules only when a supported pattern does not exist and the platform team approves the deviation.

- GitOps controls for platform changes
    - Platform changes **MUST** be managed as versioned IaC in Git with peer review, approvals, and traceability (“who/what/when/why”).
    - The platform team **MUST** publish release notes and a migration guide for baseline changes that impact workload teams.
    - Emergency changes **MUST** have a post-change review and be reconciled back into code to reduce drift.
- Exceptions
    - Policy exemptions **MUST** be time-bound and include, at minimum: `owner`, `reason`, `scope`, `expiresOn`, and `ticketId`.
    - Exemptions **MUST NOT** be indefinite; past-due exemptions are “exception debt” and **MUST** be driven to zero.

11.4.1 Anti-patterns

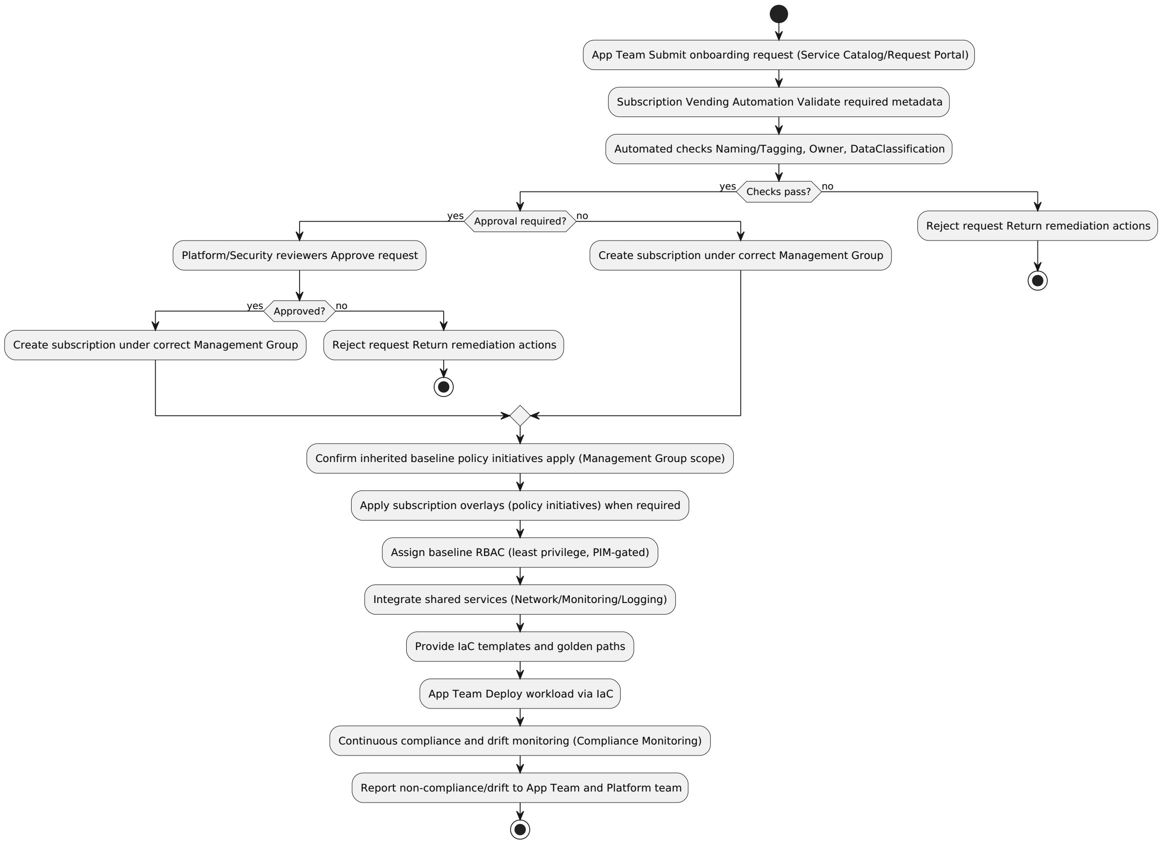
- Treating Landing Zones as one-off projects rather than versioned products
- Creating subscriptions outside the vending workflow “to save time”
- Assigning baseline policy initiatives at subscription scope instead of inheriting from Management Group scope
- Allowing permanent policy exemptions or “temporary” exemptions with no expiration
- Maintaining multiple competing golden paths with unclear support boundaries

11.5 Implementation Notes

- Define “golden paths” as supported end-to-end templates (subscription onboarding + baseline integrations + workload deployment skeleton) rather than as isolated snippets.
- Keep overlays minimal: only apply subscription-level deltas that cannot be expressed through inheritance or that represent a distinct compliance boundary.
- Maintain a per-resource-type diagnostic settings capability matrix (some services support multiple sinks; others do not) and ensure your templates reflect reality.

11.6 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Time-to-landing-zone	Business days from approved request submission to subscription ready (placed under Management Group, baseline inherited, RBAC applied, shared services integrated)	≤ 5 business days	Service catalog + automation logs	Monthly	Platform team
Policy compliance rate	Compliant resources / total in-scope resources for baseline initiatives at Management Group scope (exclude valid exemptions)	≥ 90%	Azure Policy compliance	Weekly	Security team
Exception debt	Count of past-due policy exemptions (current date > `expiresOn`)	0 past-due	Exemption registry (IaC + inventory)	Weekly	Security team
Drift trend	Count of out-of-band changes to platform-managed resources not reconciled to Git within SLA	Downward trend	Change logs + drift detection	Monthly	Platform team


App Team Submit onboarding request (Service Catalog/Request Portal)
Subscription Vending Automation Validate required metadata
Automated checks Naming/Tagging, Owner, DataClassification
yes
Checks pass?
no
yes
Approval required?
no
Reject request Return remediation actions
Platform/Security reviewers Approve request
Create subscription under correct Management Group
yes
Approved?
no
Create subscription under correct Management Group
Reject request Return remediation actions
Confirm inherited baseline policy initiatives apply (Management Group scope)
Apply subscription overlays (policy initiatives) when required
Assign baseline RBAC (least privilege, PIM-gated)
Integrate shared services (Network/Monitoring/Logging)
Provide IaC templates and golden paths
App Team Deploy workload via IaC
Continuous compliance and drift monitoring (Compliance Monitoring)
Report non-compliance/drift to App Team and Platform team

12. s12 — Compliance, Audit, and Risk Management

12.1 Purpose

You make governance controls provable by mapping requirements to technical controls, automating evidence collection, and sustaining audit readiness with repeatable testing and remediation workflows.

12.2 Scope

You cover:

- Control mapping across Management Group, subscription, and Landing Zone scopes.
- Evidence sources: Azure Policy (and policy initiative assignments), Entra ID, Azure RBAC, PIM, Defender for Cloud, Azure Activity Log, diagnostic settings, and Log Analytics Workspace data.

- Audit operations: evidence retention, access to evidence, sampling, control testing, remediation tracking, and third-party risk inputs.

You exclude:

- Detailed implementation of policy-as-code pipelines (see s06).
- Identity implementation mechanics (see s07).
- Logging/retention architecture patterns (see s09).

12.3 Decisions

- You **MUST** operate compliance as “controls + evidence,” not as periodic manual attestation.
- You **MUST** treat the Management Group as the primary governance scope for assignable controls to maximize inheritance and reduce drift; workload autonomy is via time-bound policy exemptions.
- You **MUST** use progressive enforcement (audit policy → DeployIfNotExists → deny policy) and reserve deny policy for non-negotiable/high blast-radius controls.
- You **MUST** implement evidence automation and retention baselines with explicit owners and periodic control testing.
- You **MUST** manage “exception debt” (time-bound policy exemptions) with a target of **0 past-due**.

12.4 Standards/Controls

12.4.1 Control mapping standard (minimum)

You **MUST** maintain a control catalog that maps:

- Requirement/control objective → implementation control (Azure Policy / configuration baseline / process control)
- Scope (Management Group/subscription/resource group/resource)
- Enforcement mode (audit policy / DeployIfNotExists / deny policy / procedural)
- Evidence source(s) and query method
- Owner, test cadence, and remediation SLA
- Approved exemption path (must reference the s06 exemption workflow)

You **SHOULD** align the catalog structure to Azure CAF governance and your internal control framework (e.g., ISO 27001, SOC 2), but the mapping **MUST** remain implementation-verifiable.

12.4.2 Evidence automation and retention (minimum baseline)

You **MUST** define and centrally publish minimum retention baselines (unless regulatory requires longer):

- **Azure Activity Log: MUST** be retained **≥ 365 days** in a governed destination.
- **Entra ID sign-in/audit logs: MUST** be retained **≥ 180 days** in a governed destination.
- **Resource diagnostic logs (via diagnostic settings): MUST** be retained **≥ 180 days** online/queryable in a governed Log Analytics Workspace (or security-approved equivalent meeting access control, retention, and queryability requirements).

You **MUST** document where each evidence type is stored, who can access it (least privilege), and how access is audited.

12.4.3 Audit readiness controls

- Policy and configuration controls:
  - You **MUST** manage Azure Policy definitions, policy initiatives (initiatives), assignments, and exemptions as versioned Infrastructure as Code (IaC) with peer review and traceability (see s06).
  - Policy exemptions **MUST** be explicit and expiring; indefinite exemptions are prohibited (see s06).
- Access controls:
  - Privileged access **MUST** be least privilege using Azure RBAC with PIM for privileged roles (see s07).
  - At least one break-glass account **MUST** exist, be monitored, and be periodically tested with documented procedures (see s07).
- Posture controls:
  - Defender for Cloud recommendations and secure score trends **SHOULD** be used as supporting evidence, but you **MUST** not treat them as the sole compliance proof.

12.4.4 Third-party risk and shared responsibility

- You **MUST** document shared responsibility boundaries for each workload class (platform-managed vs workload-managed controls).
- You **MUST** require third parties with access to your Azure environments to comply with your access model (Entra ID integration, RBAC, and PIM where applicable) and provide auditable evidence of controls they operate.

12.4.5 Anti-patterns

- Treating compliance as an annual spreadsheet exercise rather than continuously collected evidence.
- Allowing policy exemptions without expiry (“permanent exceptions”).
- Storing audit evidence in team-owned locations without controlled access, retention, or immutability controls.
- Passing audits via manual screenshots instead of reproducible queries and immutable logs.

12.5 Implementation Notes

12.5.1 Control catalog (policy catalog) table

You **MUST** maintain a minimum policy/control catalog similar to the table below (extend per framework):

Control ID	Requirement / Objective	Implementation Control	Scope	Enforcement	Evidence Source
GOV-POL-001	Required tags present (`Owner`, `CostCenter`, `App`, `Environment`, `DataClassification`)	Azure Policy initiative assignment	Management Group	deny policy (prod), audit policy (non-prod)	Azure Policy compliance + resource graph queries
GOV-LOG-001	Diagnostic settings to governed Log Analytics Workspace	Azure Policy initiative assignment	Management Group	DeployIfNotExists then deny policy for in-scope types	Policy compliance + Log Analytics queries
GOV-IAM-001	Privileged access is PIM-gated	PIM configuration + access reviews	Tenant + Management Group	procedural + monitoring	PIM logs + Entra ID audit logs
GOV-EXC-001	Exemptions time-bound and tracked	Exemption workflow (IaC)	All scopes	procedural + reporting	Git history + exemption registry

12.5.2 Control testing and remediation tracking

- You **MUST** maintain a remediation backlog with:
  - control ID, affected scope, owner, due date, risk rating, and status.
- You **MUST** link remediation items to the underlying evidence (policy non-compliance, log queries, access review results).
- You **SHOULD** define SLAs by risk tier (e.g., critical controls remediated faster), and enforce escalations when past due.

12.5.3 Data residency and cross-border considerations

- You **MUST** document regions used for Log Analytics Workspace(s) and long-term evidence storage and justify cross-region/cross-border transfers where applicable.
- You **MAY** implement regional evidence segregation when required by regulatory boundary (retention/access rules differ), but you must preserve centralized reporting and consistent control mapping.

12.6 Metrics

12.6.1 KPI definitions

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Audit evidence automation coverage	% of in-scope controls with automated evidence queries and documented evidence location	≥ 90% (then ramp)	Control catalog + evidence run logs	Monthly	compliance/ audit team
Policy compliance rate	Compliant resources / total in-scope resources for assigned initiatives (excluding	≥ 90% (domain-dependent)	Azure Policy compliance	Weekly	platform team

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
	approved exemptions)				
Exemption debt (past-due)	Count of exemptions past `expiresOn`	0	Exemption registry (IaC + reporting)	Weekly	security team
Control test completion	Completed control tests / scheduled control tests	≥ 95%	GRC or work tracking system	Monthly	compliance/ audit team
Time to close audit findings	Median days from finding opened to closed	Trend down	GRC or work tracking system	Monthly	owning control team

13. Implementation Roadmap and Metrics

13.1 Purpose

You execute governance as a phased rollout that reduces risk early, preserves delivery speed through a supported golden path, and sustains outcomes via a measurable KPI system and continuous improvement.

13.2 Scope

Covers:

- Phased rollout plan, milestones, and dependencies across Management Groups, subscriptions, and Landing Zones.
- Backlog intake and prioritization model (risk-based + value-based).
- KPI catalog, definitions, and reporting cadence for governance sustainability.

Depends on:

- s01 (principles and outcomes), s02 (target operating model), s11 (Landing Zones/product delivery), s12 (compliance evidence/retention).

13.3 Decisions

- You **MUST** deliver governance in phases: **Foundation → Baseline controls → Optimization → Advanced capabilities** to avoid “big-bang” deny controls that block delivery.
- You **MUST** use a single governance backlog with transparent prioritization to prevent fragmented control ownership and inconsistent enforcement.
- You **MUST** treat KPIs as a controlled artifact: versioned definitions, stable denominators, and explicit measurement sources to keep results comparable over time.
- You **SHOULD** ramp enforcement progressively (**audit → DeployIfNotExists → deny**) with documented adoption windows and migration guidance (per your policy-as-code standards).
- You **MUST** time-box and burn down policy exemption debt; exemptions are risk acceptances, not permanent architecture.

13.4 Standards/Controls

13.4.1 Roadmap phases and exit criteria (minimum)

Phase	Outcomes (what “done” means)	Key dependencies	Exit criteria (MUST be met)	Primary owner (A)
Foundation	Governance scaffolding exists and is operable	s02, s05, s11	Management Group hierarchy live; subscription vending path defined; baseline RBAC/PIM model in place; central Log Analytics Workspace pattern agreed	Platform team

Phase	Outcomes (what “done” means)	Key dependencies	Exit criteria (MUST be met)	Primary owner (A)
Baseline controls	Non-negotiables enforced; core visibility established	s04, s06, s09, s10	Deny for high blast-radius controls (limited set); diagnostic settings coverage meets baseline; tag compliance program running; break-glass tested	Security team (controls) / Platform team (implementation)
Optimization	Drift reduces; provisioning becomes predictable	s06, s11	Median time-to-landing-zone meets target; remediation automation in place; exemption renewal discipline operating; cost variance in control	Platform team / FinOps team
Advanced capabilities	Cross-domain governance and resilience are measurable	s07, s08, s12	SLO-based alerting standard adopted; evidence automation for audits; maturity targets per domain sustained for 2+ reporting cycles	Security team / Compliance/Audit team

13.4.2 Backlog prioritization model

You **MUST** score items using both risk and value so delivery teams understand trade-offs.

Dimension	Definition	Scale	Notes
Risk reduction		1–5	

Dimension	Definition	Scale	Notes
	Blast radius and likelihood of control failure		Prefer preventive controls where blast radius is high
Compliance impact	Audit/regulatory exposure reduced	1–5	Tie to s12 control mappings where possible
Delivery enablement	Unblocks teams via golden path or automation	1–5	Value includes reduced lead time and reduced toil
Cost impact	Expected cost avoidance/ optimization	1–5	Include operational overhead (logging ingestion, tooling)
Effort	Engineering + change-management effort	1–5	Used to compute WSJF-style ordering

Standard: You **MUST** publish the scoring rubric and keep it stable for at least one quarter to avoid perceived manipulation.

13.4.3 KPI catalog (authoritative definitions)

All KPIs **MUST** specify denominator rules, source-of-truth, reporting cadence, and a single Accountable owner.

KPI	Definition (MUST be unambiguous)	Target	Data source (system of record)	Reporting cadence	Accountable (A)
Policy compliance rate	**Compliant in-scope resources / total in-scope resources**, excluding approved policy exemptions	≥ 90% baseline (ramp by domain)	Azure Policy compliance state	Weekly + monthly rollup	Security team
Drift trend	Count of manual configuration changes to governed resources **outside IaC pipelines**, per week	Downward trend	Azure Activity Log + change pipeline logs	Weekly	Platform team
				Monthly	

KPI	Definition (MUST be unambiguous)	Target	Data source (system of record)	Reporting cadence	Accountable (A)
Time-to-landing-zone (median)	Median business days from **approved request** to **subscription ready for workload deployment**	≤ 5 business days	Service desk/ workflow + vending pipeline timestamps		Platform team
Exemption debt (past-due)	Number of policy exemptions with `expiresOn` in the past	0	Exemption registry (policy-as-code repo + Azure exemptions)	Weekly	Security team
Privileged access hygiene	% of privileged assignments that are **PIM-gated** (Azure RBAC and in-scope Entra ID roles)	≥ 95%	PIM reports + role assignment exports	Monthly	Security team
Cost variance	Actual vs budgeted cost per subscription/ team **for the month**	≤ ±10%	Cost Management + budget baseline	Monthly	FinOps team
Incident MTTR	Mean time to restore (MTTR) for Sev1/Sev2 incidents affecting platform baseline services	Downward trend	ITSM/ incident management system	Monthly	Platform team

Standard: You **MUST** document KPI maturity ramps where targets differ by domain (e.g., observability starts at 90% then increases) to avoid conflicting incentives during rollout.

13.5 Implementation Notes

13.5.1 Milestones and dependencies (practical sequencing)

- Start with **Management Group hierarchy + subscription vending** (otherwise policy inheritance and chargeback/showback stay inconsistent).
- Establish **policy-as-code pipeline + exemption workflow** before broad deny rollouts (so you can change safely and recover quickly).
- Implement **diagnostic settings baseline + workspace strategy** early to make governance measurable and auditable (you cannot prove controls you cannot observe).
- Roll out **tagging enforcement** before cost variance targets are treated as actionable; without tag compliance, FinOps analysis becomes disputed.

13.5.2 Communication and enablement

- You **MUST** publish release notes for baseline changes (policies, initiatives, deny changes, tagging schema changes) with:
    - impact summary, effective date, migration steps, and exemption guidance.
- You **SHOULD** run enablement as a product practice:
    - short playbooks (golden path), office hours, and a change FAQ per rollout wave.

13.5.3 Continuous improvement loop (versioning and deprecation)

- You **MUST** version governance artifacts (policies/initiatives/templates) and maintain a deprecation policy:
    - announce → adoption window → enforce → deprecate old pattern.
- You **MUST** track “governance debt” explicitly (past-due exemptions, unmanaged subscriptions, non-standard workspaces) and prioritize it alongside new features.

13.6 Metrics

This section is the **authoritative KPI dictionary** for reporting. Other sections **SHOULD** reference these KPI names/definitions rather than redefining them.

- **Weekly governance scorecard:** compliance, exemptions past-due, drift trend, critical logging coverage signals.
- **Monthly governance review:** KPI rollup, backlog outcomes, enforcement changes, and cost variance actions.
- **Quarterly maturity review:** target adjustments, control gaps, and roadmap reprioritization based on risk and operational sustainability.

14. s14 — Appendices: Standards, Templates, and Reference Configurations

14.1 Purpose

You provide concrete, reusable standards and reference artifacts that reduce interpretation risk, accelerate delivery via a supported golden path, and improve auditability.

14.2 Scope

You include:

- Reference standards tables (naming, tagging, RBAC groups/roles).
- Baseline **policy initiative (initiative)** catalog and parameter guidance.
- Network pattern references (hub-spoke, Virtual WAN (vWAN), private endpoints, Private DNS Zone patterns).
- Operational checklists (go-live, disaster recovery (DR) test, cost review).

You exclude:

- Full implementation code repositories (you link to them from your internal tooling).
- Service-by-service deep dives unless they are a recurring governance control surface.

14.3 Decisions

- You **MUST** treat these appendices as controlled artifacts: versioned, reviewed, and released on a governance cadence.
- You **MUST** use a single canonical taxonomy for tags, naming tokens, and environment values across all sections and policies.
- You **SHOULD** keep the appendix “reference-grade”: minimal prose, maximal testable standards (tables/templates).
- You **MAY** maintain multiple reference configurations when required by compliance boundaries (e.g., regulated vs non-regulated), but you **MUST** declare the decision rule.

14.4 Standards/Controls

- All standards in this appendix **MUST** be enforceable and/or reportable (Azure Policy, automation, or periodic control checks).
- All templates **MUST** declare ownership, intended scope (Management Group/subscription/resource group), and lifecycle (version, deprecation date, support window).
- Any “approved equivalent” (e.g., to Log Analytics Workspace) **MUST** meet minimum equivalency: access control model, retention controls, queryability, and exportability; approval authority is the security team.

14.5 Implementation Notes

- Keep tables “automation-friendly” (keys aligned to policy parameters and reporting fields).
- Prefer deterministic standards; where Azure requires global uniqueness, allow deterministic hash suffixes derived from stable inputs (not random strings).

14.6 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Standards drift	# of appendix artifacts changed outside Git release process	0	Repo + change management logs	Monthly	Platform team

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Golden path adoption	% of new workloads using reference templates	≥ 80%	Pipeline telemetry	Monthly	Platform team
Policy parameter alignment	% of initiative parameters matching appendix defaults	≥ 95%	Azure Policy assignments export	Monthly	Security team

14.7 Appendix A — Naming Standard (Reference)

14.8 Purpose

You ensure naming is deterministic, automatable, and supports operations at scale.

14.9 Scope

Applies to subscriptions, resource groups, and Azure resources created in Landing Zones.

14.10 Decisions

- You **MUST** use deterministic names for resources and resource groups.
- You **MAY** use a deterministic hash suffix only where Azure global uniqueness/name-length constraints require it.

14.11 Standards/Controls

14.11.1 Naming tokens (canonical)

- `org`: short org identifier (e.g., `contoso`)
- `app`: application/product identifier (from your app registry)
- `env`: `dev|test|prod|sandbox` (canonical set)
- `region`: Azure region short code (e.g., `weu`, `eun`)
- `instance`: zero-padded integer (`001`…)
- `role`: optional workload role (`api`, `web`, `db`)

14.11.2 Subscription naming

```
sub-<org>-<platform|app>-<env>-<region>-<instance>
```

14.11.3 Resource group naming

```
rg-<org>-<app>-<env>-<region>-<instance>-<role>
```

14.11.4 Deterministic uniqueness rule (when required)

If the service enforces global uniqueness, you **MAY** append `-<hash>` where:

- `hash` = 6–8 chars derived from `org-app-env-region-instance` (stable input set)
- Hash algorithm selection is implementation-specific but **MUST** be deterministic and documented.

14.12 Implementation Notes

Maintain a region code map in a single file used by pipelines and validation rules.

14.13 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Naming compliance	Compliant resources / total in-scope resources	≥ 95%	Resource inventory + policy compliance	Monthly	Platform team

14.14 Appendix B — Tagging Standard (Reference)

14.15 Purpose

You enable unambiguous ownership, cost allocation (Chargeback/Showback), and security classification.

14.16 Scope

Applies to all resource groups and resources in all Landing Zones.

14.17 Decisions

- Tags are mandatory even when subscriptions are the primary allocation boundary because they drive ownership routing, automation, and security classification.

14.18 Standards/Controls

Tag Key	Required	Allowed Values / Format	Enforcement Level	Primary Owner	Notes
`Owner`	MUST	Team alias (no individuals)	Deny (Prod), Audit (Non-prod)	Workload teams	Ops routing
`CostCenter`	MUST	Finance master list code	Deny (Prod), Audit (Non-prod)	FinOps team	Allocation
`App`	MUST	App registry ID	Deny (Prod), Audit (Non-prod)	Workload teams	CMDB/ app registry alignment
`Environment`	MUST	`dev	test	prod	sandbox`
`DataClassification`	MUST	`public	internal	confidential	restricted`
`ManagedBy`	SHOULD	`IaC	Portal	Other`	Audit

14.19 Implementation Notes

Enforce tags at resource group scope first; inherit to resources via policy where feasible.

14.20 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Tag compliance	Tagged resources / total in-scope resources	≥ 95%	Azure Policy compliance	Monthly	FinOps team

---

14.21 Appendix C — Baseline Policy Initiative Catalog (Reference)

14.22 Purpose

You standardize control sets as initiatives assigned at Management Group scope for maximum inheritance and minimal drift.

14.23 Scope

Applies to all Landing Zone subscriptions under governed Management Groups.

14.24 Decisions

- You follow progressive enforcement: **Audit** → **DeployIfNotExists** → **Deny**.
- You assign baselines at **Management Group** scope; you use subscription overlays only for environment/workload deltas.

14.25 Standards/Controls

Initiative Name	Primary Scope	Default Effect Mode	Rollout Phase	Primary Owner	Exemptions Allowed
`lz-identity-baseline`	Management Group	Deny (for high-risk), Audit (else)	Mature	Security team	Limited; requires security approval
`lz-logging-baseline`	Management Group	DeployIfNotExists then Deny (where supported)	Ramp	Platform team	Allowed with compensating controls

Initiative Name	Primary Scope	Default Effect Mode	Rollout Phase	Primary Owner	Exemptions Allowed
`lz-tagging-baseline`	Management Group	Audit then Deny (Prod)	Ramp	FinOps team	Rare; time-bound only
`lz-network-baseline`	Management Group	Deny for public exposure controls	Mature	Security team	Limited; time-bound only

14.26 Implementation Notes

Maintain a per-resource-type diagnostic settings capability matrix (supported sinks/effects) alongside initiative code to avoid “unenforceable MUSTs”.

14.27 Metrics

See s13 for the canonical KPI catalog; this appendix supplies the control inventory.

14.28 Appendix D — RBAC Role Catalog and Group Naming (Reference)

14.29 Purpose

You standardize access patterns that support least privilege and PIM.

14.30 Scope

Covers Azure RBAC roles and Entra ID privileged role handling where in-scope.

14.31 Decisions

- Privileged access **MUST** be group-based and PIM-gated.
- Direct user assignments for privileged roles **MUST NOT** be used except break-glass scenarios.

14.32 Standards/Controls

14.32.1 Group naming pattern

```
grp-az-<scopeType>-<scopeName>-<role>-<env>
```

Examples:

- `grp-az-sub-app1-prod-owner-prod`
- `grp-az-mg-platform-contributor-prod`

14.32.2 Role catalog (minimum)

Role Name	Typical Scope	Intended Use	Assignment Rule	Primary Owner
Owner	Subscription (rare)	Break-glass + subscription administration	PIM only; group-based	Security team
Contributor	Subscription/RG	Workload engineering	Group-based; PIM for prod elevated tasks	Platform team
Reader	Subscription/RG	Audit/visibility	Group-based	Platform team
Security Reader	Management Group/ Subscription	Security visibility	Group-based	Security team

14.33 Implementation Notes

Define a clear split between Azure RBAC roles and Entra ID directory roles in your access reviews (directory roles require separate governance).

14.34 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Privileged access hygiene	PIM-gated privileged assignments / total	≥ 95%	PIM reports + RBAC	Monthly	Security team

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
	privileged assignments		assignment export		

14.35 Appendix E — Network Patterns Reference (Hub-Spoke / vWAN / Private Endpoints)

14.36 Purpose

You provide supported, repeatable network reference configurations that align with centralized governance and private access patterns.

14.37 Scope

Covers connectivity topology, DNS for private endpoints, and routing patterns referenced by s08.

14.38 Decisions

- You **SHOULD** default to hub-spoke; you **MAY** use vWAN when routing/connectivity scale requires it.
- You **MUST** treat Private DNS Zones as centrally governed shared assets.

14.39 Standards/Controls

- Private endpoints **MUST** integrate with centrally managed Private DNS Zones; zone linking is approved and tracked.
- Egress controls **MUST** be explicit (documented UDRs, firewall/NVA pattern, or managed egress).

14.40 Implementation Notes

Maintain a “DNS onboarding request” template: requested zone, spokes to link, approver, implementation owner, and expected lead time.

14.41 Metrics

KPI	Definition	Target	Data Source	Reporting Cadence	Primary Owner
Private endpoint DNS compliance	Endpoints resolving via approved Private DNS Zones / total private endpoints	≥ 95%	DNS zone links + endpoint inventory	Monthly	Platform team

14.42 Appendix F — Operational Checklists

14.43 Purpose

You standardize recurring operational controls to reduce outages, audit gaps, and unmanaged cost growth.

14.44 Scope

Go-live readiness, DR test readiness, and monthly cost review.

14.45 Decisions

- Checklists are “definition of done” gates for production workloads.
- Evidence artifacts **MUST** be stored centrally and referenced by workload ID.

14.46 Standards/Controls

14.46.1.1 Go-live checklist (minimum)

- You **MUST** verify: required tags, diagnostic settings, RBAC/PIM, backup tier (RPO/RTO), runbooks, and exemption status (no past-due).
- You **MUST** record: workload owner group, escalation path, and operational SLOs (service level objectives).

14.46.1.2 DR test checklist (minimum)

- You **MUST** test restores/failover per tier and capture evidence.
- You **MUST** validate monitoring and incident routing during the exercise.

14.46.1.3 Monthly cost review checklist (minimum)

- You **MUST** review: cost variance (≤ ±10%), top cost drivers, and untagged spend.
- You **MUST** create a remediation backlog with owners and due dates.

14.47 Implementation Notes

Automate checklist evidence capture where possible (pipeline outputs, policy compliance exports, PIM reports).

14.48 Metrics

See s13 for the canonical KPI catalog; these checklists operationalize those KPIs.