INVARDIA
PIM MANAGER PRO

TECHNICAL ARCHITECTURE AND SECURITY DESIGN

PIM Manager 2.4.634

Why it matters, what it delivers, and how every component, identity, permission, data flow and control works

For architects, security officers and CISOs · the full target design

WHY · THE PROBLEM

Admin rights are standing, scattered and hard to prove

Standing access

- Admin rights stay assigned long after the task is done
- Every standing admin account is a target

Sprawl

- Privilege hides in nested groups, portal clicks and policies nobody re-checks
- Entra ID, Azure, Intune and Defender each grant it their own way

No single source of truth

- Nobody can say for certain who may do what, or how they got it
- Every audit becomes a project

WHAT · THE VALUE

Privileged access that is earned, temporary and proven

Least privilege by design

- Access comes from tiered groups and is activated just in time, never left standing
- One model for Entra ID, Azure, Intune and Defender XDR

Every change under control

- Staged, approved, guarded and journaled
- Reversible in one step

Proof on demand

- Who can do what, and how they got it
- Every change with before and after, plus the database's own audit

Yours, and unattended

- Runs in your own Azure tenant on managed identities
- Signed, approved updates apply themselves for years

WHY IT IS SAFE TO RUN

Built to be the least risky thing in your tenant

Nothing to steal

- No passwords and no sign-in secret anywhere
- Every component its own managed identity and its own database role

Nothing gets in

- The engine has no listener; the vendor has no access unless you grant it
- The portal reads; it never writes to your directory

Nothing slips through

- Guards hold what cannot heal, and tighten for 24 hours after every engine change
- Customer releases only after an approved, evidenced rollout

The rest of this deck shows how: component by component, permission by permission.

SECURITY AT A GLANCE

Twelve controls, one slide

Tenancy

- Everything in your subscription and region
- No shared database, no vendor-hosted runtime

Identities

- One managed identity per component
- No passwords, no client secrets at runtime or sign-in

Least privilege

- The portal reads only; only the engine writes
- Azure write is opt-in

Database

- Entra-only; one custom role per component
- Rights change only through one signed procedure

Sign-in

- Sign-in front: OIDC + PKCE, no secret
- 8-hour session; your Conditional Access

Change control

- Staged, approved, journaled, reversible
- Second approver and break-glass protection

Guards

- Mass changes held for the tenant's SuperAdmin
- Limits halved for 24 h after an engine change

Audit

- Append-only trail with before and after
- SQL audit in your workspace, 90 days

Releases

- Signed, built from source in your registry
- Customer rings only after an approved rollout

Network

- No inbound to the engine
- Private mode: private endpoints only

Vendor access

- None unless you create the Support app
- Every action recorded, fail-closed

Five years unattended

- Updates need no admin and no contact
- Nothing to rotate or renew

CONTENTS

18 chapters

#	Chapter	In short
1	Overview	PIM Manager keeps privileged access in Entra ID, Azure, Intune and Defender XDR matching one desired state that your administrators manage, and proves it
2	Architecture	Everything runs in your Azure subscription
3	Identities and permissions	Every identity the product uses, every permission it holds, the scope, and why
4	Authentication and authorization	Who signs in, how, with which credential, and what each role may do
5	Change control	What PIM Manager can change in your tenant, and the flow from a click to an API call
6	Guards and guard modes	Every write the engine makes passes the guards first
7	Database access	One user and one custom role per component, no passwords anywhere, and rights that change only through one signed procedure
8	Data	What is stored, where, how it is encrypted, how long it is kept, and how it is deleted
9	Logging and audit	Four independent records
10	Release process	From code to your installation
11	Updates and supply chain	Your installation pulls its ring's signed release at night, verifies it, builds it from source in your own registry, and rolls forward only
12	Network	Public with sign-in, or private only with private endpoints
13	Vendor access	Invardia has no access to your installation unless you create the Invardia Support app, at a level you choose, for as long as you choose
14	Support team	Three levels, one process
15	Threats and compliance	The top risks for a privileged-access product, the controls that answer each one, and how the controls support DORA, NIS2 and ISO 27001
16	Incidents and hardening	How incidents and vulnerabilities are handled, and the hardening you apply on day one
17	Naming conventions	Every name PIM Manager creates follows your convention
18	On-premises Active Directory (optional)	A Tier-0 worker next to your domain controllers

CHAPTER 1

INVARDIA
PIM MANAGER PRO

Overview

PIM Manager keeps privileged access in Entra ID, Azure, Intune and Defender XDR matching **one desired state** that your administrators manage, and proves it.

Describe
Who may be privileged, in which role, scope and tier, in the portal

Store
The desired state in your own Azure SQL database

Apply
The engine makes the tenant match, within minutes

Prove
Every change journaled with before and after

OVERVIEW · THE IDEA

Describe who may be privileged; the engine makes the tenant match

1 · DEFINE

Manager GUI

Admins, departments, roles, groups, delegations, policies

2 · STORE

Desired state in SQL

Every change journaled, backed up, undoable

3 · APPLY

Engine, right after each commit

Starts within about a minute and applies to Entra, Azure, Defender, Intune, AD. A 5-minute schedule and a daily reconcile catch the rest.

Admin accounts + TAP · Entra roles · PIM for Groups · Admin units

Azure resource roles + policies · Defender XDR roles · Intune roles

Access reviews · offboarding · on-premises AD (hybrid worker)

OVERVIEW · WHAT CHANGED IN 2.4.632

Fewer secrets, narrower rights, safer releases

Topic	Now, in PIM Manager 2.4.632
Sign-in	A sign-in front with OpenID Connect and PKCE; the GUI's managed identity is its credential, so no client secret exists. Sessions end after a fixed 8 hours
Database access	One database user per component; the image-pull identity has no database access; the GUI cannot change the schema
Database admin group	Only the Invardia Support app, if you create one; otherwise empty. Updates never need it
SQL auditing	Every database statement to your own Log Analytics workspace, kept 90 days
Strict guard mode	For 24 hours after an update that changed engine code, every safety limit is halved (never below 1)
Release control	Rings, named waves and recorded evidence; customer rings only get a version the release owner approved
Mail subjects	[Environment] Failed / Warning / Action needed: what happened [ticket] [PIM Manager]
Browser hardening	Strict content security policy, HSTS, no framing, no sniffing, same-origin referrer on every page

OVERVIEW · SCOPE

What it manages, and what it never reads or does

Manages

- Dedicated admin accounts and their first credential (Temporary Access Pass)
- Entra roles, PIM for Groups, administrative units
- Azure roles and Azure PIM policies (write is opt-in)
- Intune and Defender XDR roles, access reviews, offboarding
- On-premises AD admin accounts (optional worker)

Never reads

- Sign-in logs, audit logs, risky users
- Conditional Access policies
- Mailbox content, files

Never does

- Touch an object the store does not define
- Act on an empty or unreadable store
- Delete an account it does not manage
- Change a PIM policy without your SuperAdmin's approval
- Sit in the activation path: users activate in native Entra PIM

CHAPTER 2

INVARDIA PIM MANAGER PRO

Architecture

Everything runs in **your** Azure subscription. Three trust zones: your tenant, your subscription, and Invardia, which only ever receives outbound calls.

No vendor runtime
Invardia hosts no data and no runtime

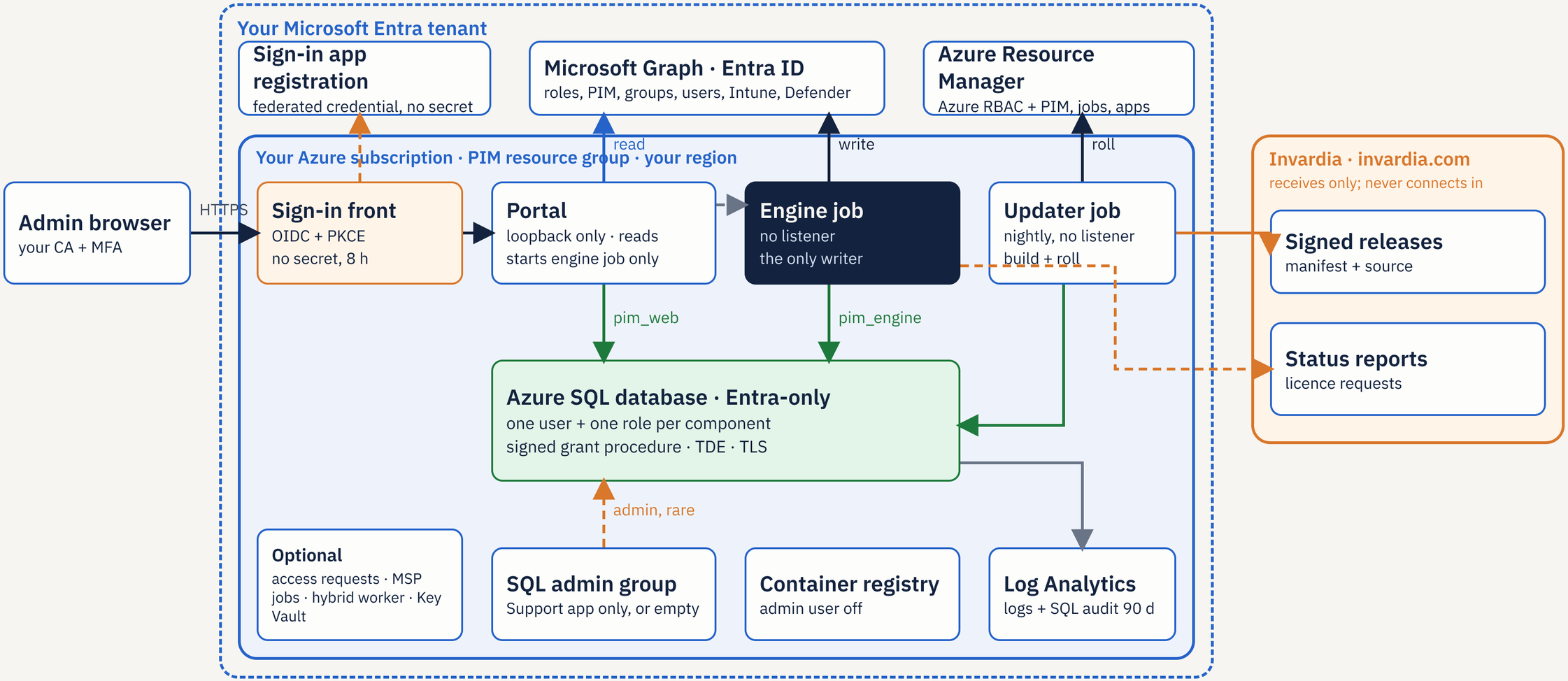
No inbound
The engine and jobs have no listener

One store
Azure SQL, Entra-only

Built locally
Images built from source in your registry

ARCHITECTURE · TRUST BOUNDARIES

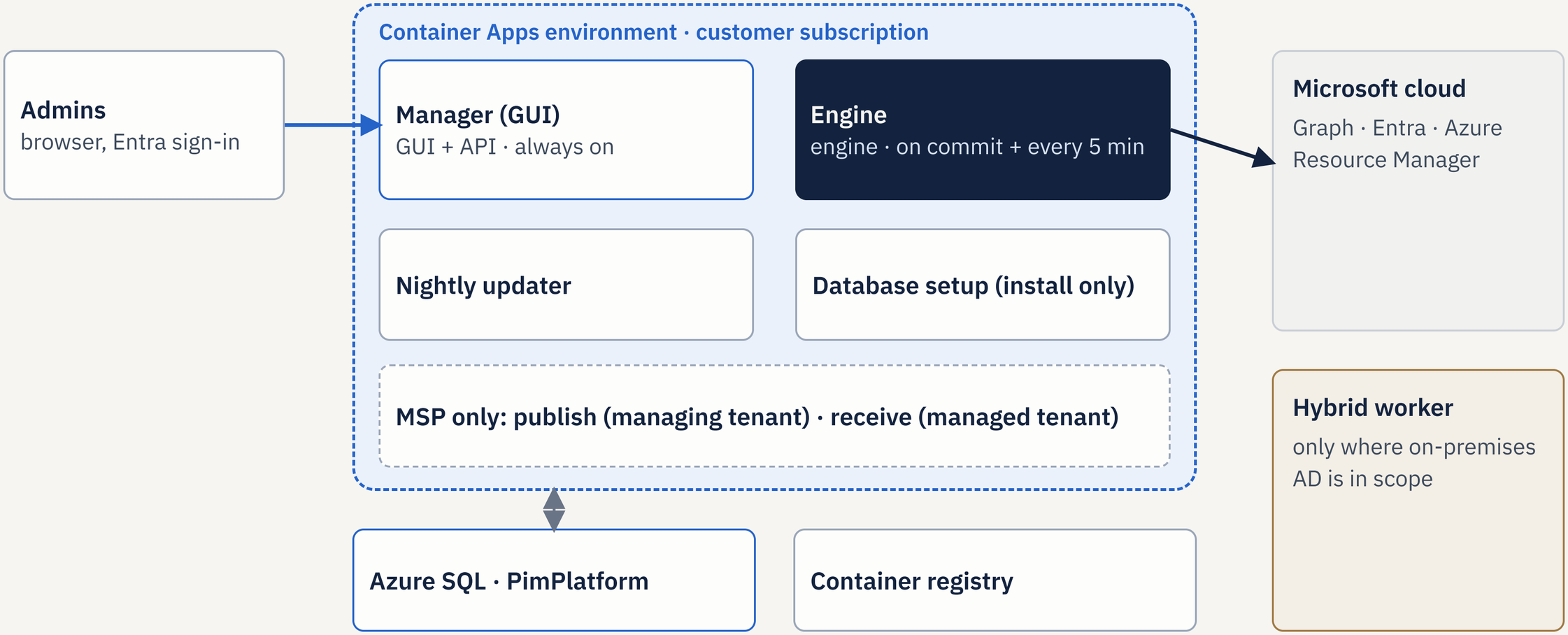
Every component, every path, three trust zones



INVARDIA PIM MANAGER PRO

ARCHITECTURE · COMPONENT MAP

Everything runs in one Container Apps environment



ARCHITECTURE · EACH COMPONENT, ITS TRIGGER AND ITS IDENTITY

Least privilege per component

Component	Does	Runs	Identity and rights
Manager	GUI + API behind the sign-in front; writes SQL, starts the tick	Always on	Own MI (also the sign-in credential, no secret) · 15 read-only Graph roles · Reader on the subscription · Jobs Operator on the tick (where granted)
Engine	Applies desired state, runs jobs	On commit + every 5 min	Own MI · 28 Graph roles · Reader at the tenant root · Azure read-only
Updater	Fetch, verify, build, schema, roll	03:00 UTC	Own MI · Contributor on PIM's resource group · database role pim_updater (schema only; may only run the signed grant procedure) · no Graph roles
Database setup	Creates SQL users from the inside	On deploy	User-assigned MI · SQL Entra admin
Image pull	Pulls the image	On start	User-assigned MI · AcrPull only, no database access
MSP publish	Signs + uploads the bundle	Self-gated	Own MI · Key Vault sign · SELECT-only SQL
MSP receive	Pulls, verifies, writes rows	Self-gated	Own MI in the managed tenant · the engine's 28 Graph roles

ARCHITECTURE · AZURE RESOURCES

What the installer creates in your resource group

Resource	Default	Purpose	Exposure
Container App: portal	Consumption, scales to zero	Sign-in front + portal (two containers)	Public with sign-in, or private IP only
Container Apps job: engine	Every 5 minutes + on commit	Applies the desired state	No ingress
Container Apps job: updater	Nightly 03:00 UTC (Pro)	Fetch, verify, build, schema, roll	No ingress
Container Apps job: as-built	Started by hand	Configuration PDF for support	No ingress
Azure SQL database	Standard S0 minimum, 250 GB max	The single store	Entra-only; private endpoint in private mode
Container registry	Basic; Premium in private mode	Images built from source	Admin user off; private endpoint in private mode
Log Analytics workspace	Workspace default retention	Container logs + SQL security audit (90 days)	n/a
User-assigned identity	One	Image pull only (AcrPull)	No database access
Key Vault	Only when a feature needs one	SMTP secret, emergency passphrase, MSP signing key	RBAC, purge protection

ARCHITECTURE · THE DATA MODEL

One database holds intent, settings and evidence

Intent

- pim.Rows: admins, definitions, assignments
- pim.ChangeQueue
- pim.ManagedKeys (what PIM owns)

Settings + state

- pim.Settings: settings, feature gates, approvals, scheduler
- pim.TenantCache: snapshots, reports

Evidence

- pim.AuditEvents (append-only)
- pim.CommitJournal (append-only)
- pim.CommitApplied
- pim.ConfigBackups

Logical tables: Account-Definitions-Admins · PIM-Definitions-* (roles, tasks, services, departments, AUs) · PIM-Assignments-* (admins, groups, roles, Azure resources, workloads). MSP adds central-admin and tenant tables.

CHAPTER 3

INVARDIA PIM MANAGER PRO

Identities and permissions

Every identity the product uses, every permission it holds, the scope, and **why**. Least privilege for every identity, by design.

One per job
Each component its own managed identity

Portal reads
15 Graph permissions, all read

Engine writes
28 Graph permissions; tier-0 asset

Opt-in
Azure write (User Access Administrator)

SECURITY · IDENTITIES

No secret to steal, at runtime or at sign-in

Workloads: one identity per job

- Managed identities, nothing stored
- Manager: read-only Graph; may only start the engine job
- Image pull: pulls the image, nothing else
- Mail through one mailbox, never Mail.Send

SQL: Entra only, one user per job

- No SQL logins, no passwords anywhere
- Each component its own database user
- The Manager cannot change the schema
- Every statement audited to your workspace, 90 days

Database admin group

- Role-assignable: only a Privileged Role Administrator changes it
- Holds the Invardia Support app if you create one, else nobody
- Updates never need it; people get in just-in-time

Secrets: one, and it is optional

- Sign-in front: no secret, the GUI's managed identity is its credential
- Invardia Support app (optional, its secret held by Invardia): you create it, set its level and can remove it; troubleshooting and setup only, never runtime

IDENTITIES · INVENTORY

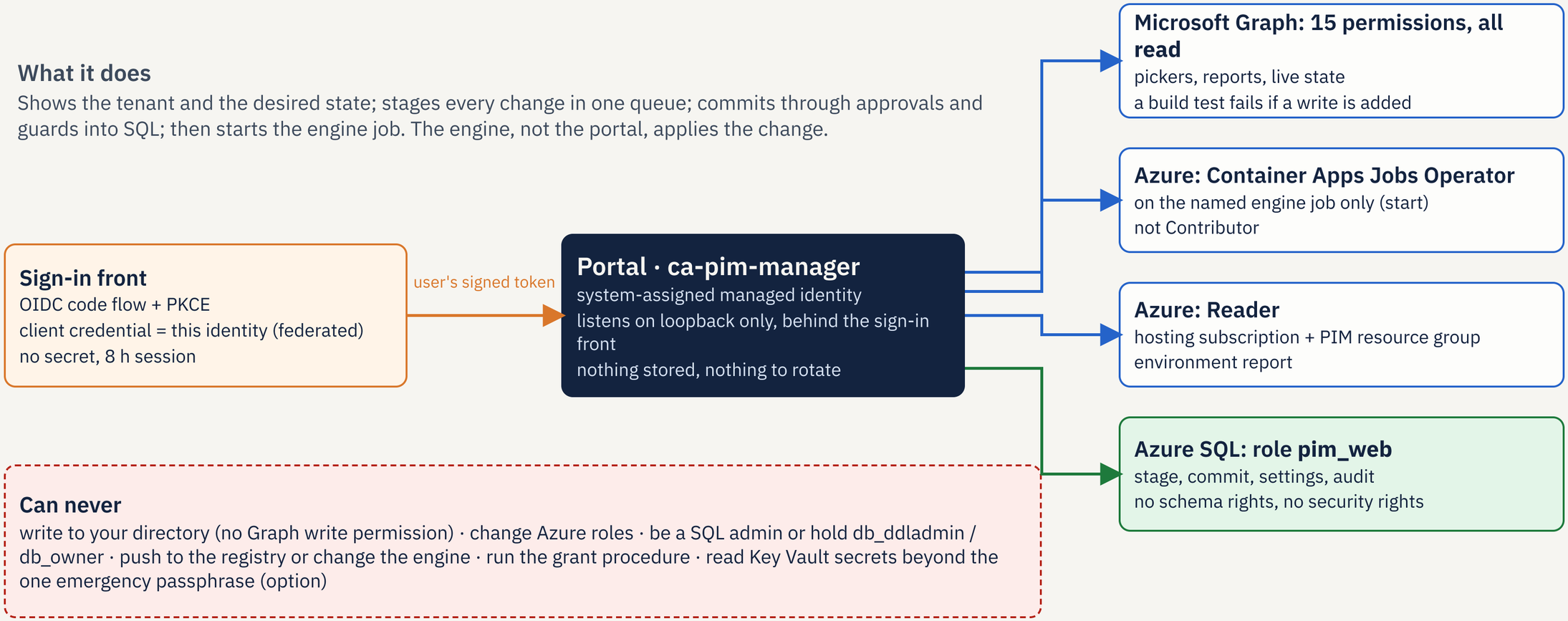
Every identity the product creates or uses

Identity	Kind	Credential	Purpose
Portal (ca-pim-manager)	System-assigned managed identity	None; also the sign-in front's federated credential	Reads for the GUI, stages changes, starts the engine
Engine (ca-pim-tick)	System-assigned managed identity	None	Reads the tenant, applies the desired state
Updater (ca-pim-update)	System-assigned managed identity	None	Builds and rolls the release, applies schema
Image pull (id-pim-...)	User-assigned managed identity	None	Pulls images; no database access
As-built job	User-assigned managed identity	None	One database view, one storage container
Manager sign-in app	App registration	**None** (federated credential)	User sign-in; assignment required
PIM Activator app (optional)	App registration, public client	None (PKCE)	Users activate their own eligible access
SQL admin group	Role-assignable security group	n/a	Server administrator: Support app only, or empty
Invardia Support app (optional)	App registration in your tenant	Client secret, 90 days, held by Invardia	Setup and troubleshooting only, never runtime

The portal's identity: what it can call, and what it can never do

What it does

Shows the tenant and the desired state; stages every change in one queue; commits through approvals and guards into SQL; then starts the engine job. The engine, not the portal, applies the change.



Portal (ca-pim-manager)

Permission	API	Scope	Why it is needed	Least privilege
15 read permissions: Directory, Domain, User, Group, AdministrativeUnit, Application, AccessReview, RoleManagement (.Read.All); RoleManagementPolicy, RoleEligibilitySchedule, RoleAssignmentSchedule (.Read.Directory); RoleManagementPolicy, PrivilegedAccess, PrivilegedEligibilitySchedule, PrivilegedAssignmentSchedule (.Read.AzureADGroup)	Graph	Tenant	Pickers, reports and live state	Read only; enforced by a build test
Container Apps Jobs Operator	ARM	The engine job	Commit and Run now start the engine	Start only, never Contributor
Reader	ARM	Subscription; PIM resource group	Azure views; environment report	Read only
pim_web role	SQL	PIM database	Stage, commit, settings, audit	No schema or security rights
Application Mail.Send (option)	Exchange RBAC	One mailbox scope	Test and approval mail	Never tenant-wide Mail.Send
Key Vault Secrets User (option)	Key Vault	One secret	Emergency passphrase check	One secret only

Engine (ca-pim-tick): Microsoft Graph

Permission	API	Scope	Why it is needed	Least privilege
Directory.Read.All	Graph	Tenant	Read users, groups, roles to compare with the desired state	Read only
Domain.Read.All	Graph	Tenant	Verified domains for admin account naming	Read only
User.ReadWrite.All	Graph	Tenant	Create and maintain the dedicated admin accounts	Needed to apply
Group.ReadWrite.All	Graph	Tenant	Create and maintain the PIM groups and membership	Needed to apply
AdministrativeUnit.ReadWrite.All	Graph	Tenant	Administrative units as delegation scopes	Needed to apply
UserAuthenticationMethod.ReadWrite.All	Graph	Tenant	Temporary Access Pass for a new admin account	Needed to apply
Policy.Read.All	Graph	Tenant	Authentication and authorisation policies used in checks	Read only
RoleManagement.Read.All	Graph	Tenant	Role definitions and assignments	Read only
RoleManagement.ReadWrite.Directory	Graph	Tenant	Role-assignable groups and their directory roles (**tier-0**)	Needed to apply
RoleEligibilitySchedule.ReadWrite.Directory	Graph	Tenant	Eligible Entra role assignments	Needed to apply
RoleEligibilitySchedule.Remove.Directory	Graph	Tenant	Remove eligible Entra role assignments on revoke	Needed to apply
RoleAssignmentSchedule.ReadWrite.Directory	Graph	Tenant	Active (time-bound) Entra role assignments	Needed to apply
RoleAssignmentSchedule.Remove.Directory	Graph	Tenant	Remove active Entra role assignments on revoke	Needed to apply
RoleManagementPolicy.Read.Directory	Graph	Tenant	Read Entra role PIM policies	Read only

Engine (ca-pim-tick): Microsoft Graph

Permission	API	Scope	Why it is needed	Least privilege
RoleManagementPolicy.ReadWrite.Directory	Graph	Tenant	Apply Entra role PIM policies (approval, MFA, duration)	Needed to apply
PrivilegedAccess.ReadWrite.AzureADGroup	Graph	Tenant	Manage PIM for Groups	Needed to apply
PrivilegedEligibilitySchedule.ReadWrite.AzureADGroup	Graph	Tenant	Eligible group memberships	Needed to apply
PrivilegedEligibilitySchedule.Remove.AzureADGroup	Graph	Tenant	Remove eligible group memberships	Needed to apply
PrivilegedAssignmentSchedule.ReadWrite.AzureADGroup	Graph	Tenant	Active group memberships	Needed to apply
PrivilegedAssignmentSchedule.Remove.AzureADGroup	Graph	Tenant	Remove active group memberships	Needed to apply
RoleManagementPolicy.Read.AzureADGroup	Graph	Tenant	Read PIM-for-Groups policies	Read only
RoleManagementPolicy.ReadWrite.AzureADGroup	Graph	Tenant	Apply PIM-for-Groups policies	Needed to apply
AppRoleAssignment.ReadWrite.All	Graph	Tenant	Enterprise-application role provider (**tier-0**)	Needed to apply
Application.Read.All	Graph	Tenant	Resolve applications and service principals	Read only
AccessReview.Read.All	Graph	Tenant	Read access reviews	Read only
AccessReview.ReadWrite.All	Graph	Tenant	Create reviews (broader than needed; no narrower one exists)	Broader than needed
RoleManagement.ReadWrite.Defender	Graph	Tenant	Defender XDR unified RBAC roles (when managed)	Needed to apply
DeviceManagementRBAC.ReadWrite.All	Graph	Tenant	Intune RBAC roles (when managed)	Needed to apply

Engine (ca-pim-tick): Azure, database, mail, secrets

Permission	API	Scope	Why it is needed	Least privilege
Reader	ARM	Tenant root management group + hosting subscription	Discover management groups, subscriptions; report Azure RBAC	Read only
User Access Administrator (opt-in)	ARM	Tenant root, or the scopes you name	Write Azure role assignments and Azure PIM	Off by default; Azure stays read-only
Reader	ARM	Its own job resource	Its own run status	Read only
pim_engine role	SQL	PIM database	Read desired state; write outcome, audit, journal	Reduced: no schema rights (was db_ddladmin)
Application Mail.Send (option)	Exchange RBAC	One mailbox scope	TAP codes, approvals, alerts	Never tenant-wide
Key Vault Secrets User (option)	Key Vault	One SMTP secret	Mail through a relay	One secret
Storage Table Data Contributor (option)	Storage	Access request store	Pull access requests	One store
Power BI security group + read-only admin API (option)	Fabric	Power BI	Read workspaces for discovery	Read only

Updater (ca-pim-update) and the image-pull identity

Permission	API	Scope	Why it is needed	Least privilege
Contributor	ARM	The PIM resource group (and the registry)	Update container apps and jobs; schedule image builds	Resource group only; remove it for manual updates
AcrPull	ARM	Your registry	Pull built images	Pull only
pim_updater role + db_ddladmin	SQL	PIM database	Additive schema changes; data migrations	Reduced: no seat in the SQL admin group
EXECUTE on the grant procedure	SQL	One procedure	Apply the release's grant list	The only security-related right it has
No Graph permissions	Graph	-	-	Cannot touch the tenant
Image pull: AcrPull	ARM	Your registry	Pull images for the portal and jobs	Pull only
Image pull: no database access	SQL	-	An image-pull identity has no reason to reach data	Removed from the admin group

Invardia Support app (optional, yours to create and remove)

Permission	API	Scope	Why it is needed	Least privilege
Troubleshoot: read Graph (directory, apps, users, groups, roles, PIM policies and schedules); Application.ReadWrite.OwnedBy	Graph	Tenant	Read for support; renew its own secret	Read only
Troubleshoot: Reader, Log Analytics Reader	ARM	The resource groups you name	See the installation and its logs	Read only
Troubleshoot: read access	SQL	PIM database	Read the store when you ask for support	Read only
As-built: start one job, read one container	ARM	One job, one container	Configuration PDF (deleted after 24 h)	Minimal
Setup: Application.ReadWrite.All, AppRoleAssignment.ReadWrite.All, DelegatedPermissionGrant.ReadWrite.All, Group.Create, GroupMember.ReadWrite.All, RoleManagement.ReadWrite.Directory	Graph	Tenant	Install and repair on your behalf	**Tier-0 equivalent:** lower after use
Setup: Contributor + User Access Administrator	ARM	Subscriptions you name (optionally root)	New install, repair	Tier-0 equivalent
Member of the SQL admin group	SQL	The SQL server	Create component users; recovery	The group's only standing member

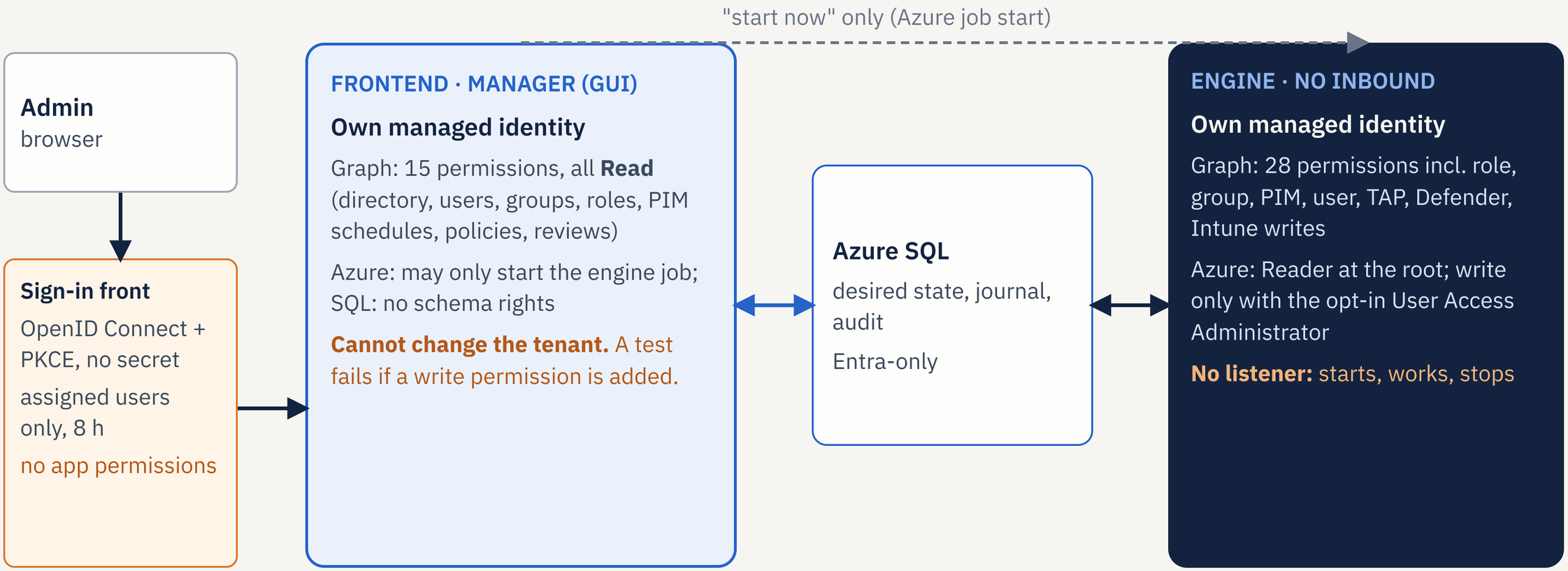
PERMISSIONS · PIM ACTIVATOR (OPTIONAL)

PIM Activator browser extension: delegated only

Permission	API	Scope	Why it is needed	Least privilege
PrivilegedAccess.ReadWrite.AzureADGroup	Graph (delegated)	The signed-in user	Activate and deactivate the user's OWN eligible groups	Bounded by the user's own eligibility
RoleManagement.ReadWrite.Directory	Graph (delegated)	The signed-in user	Activate the user's own eligible Entra roles	Bounded by the user's own eligibility
Group.Read.All, User.Read, RoleManagement.Read.Directory, AdministrativeUnit.Read.All, Application.Read.All	Graph (delegated)	The signed-in user	Show what the user may activate	Read
user_impersonation	Azure Service Management (delegated)	The signed-in user	Activate the user's own eligible Azure roles	Bounded by the user's own eligibility
No application permissions, no secret	-	-	Public client with PKCE	Nothing stored

SECURITY · FRONTEND VERSUS ENGINE

The GUI can only read; the engine writes but has no door



Someone who takes over the GUI can read and propose changes; every change still passes roles, approvals, guards and the journal before the engine acts.

CHAPTER 4

INVARDIA PIM MANAGER PRO

Authentication and authorization

Who signs in, how, with which credential, and what each role may do. **No passwords and no client secrets** in the product's own sign-in or runtime.

Sign-in front
OIDC + PKCE, no secret

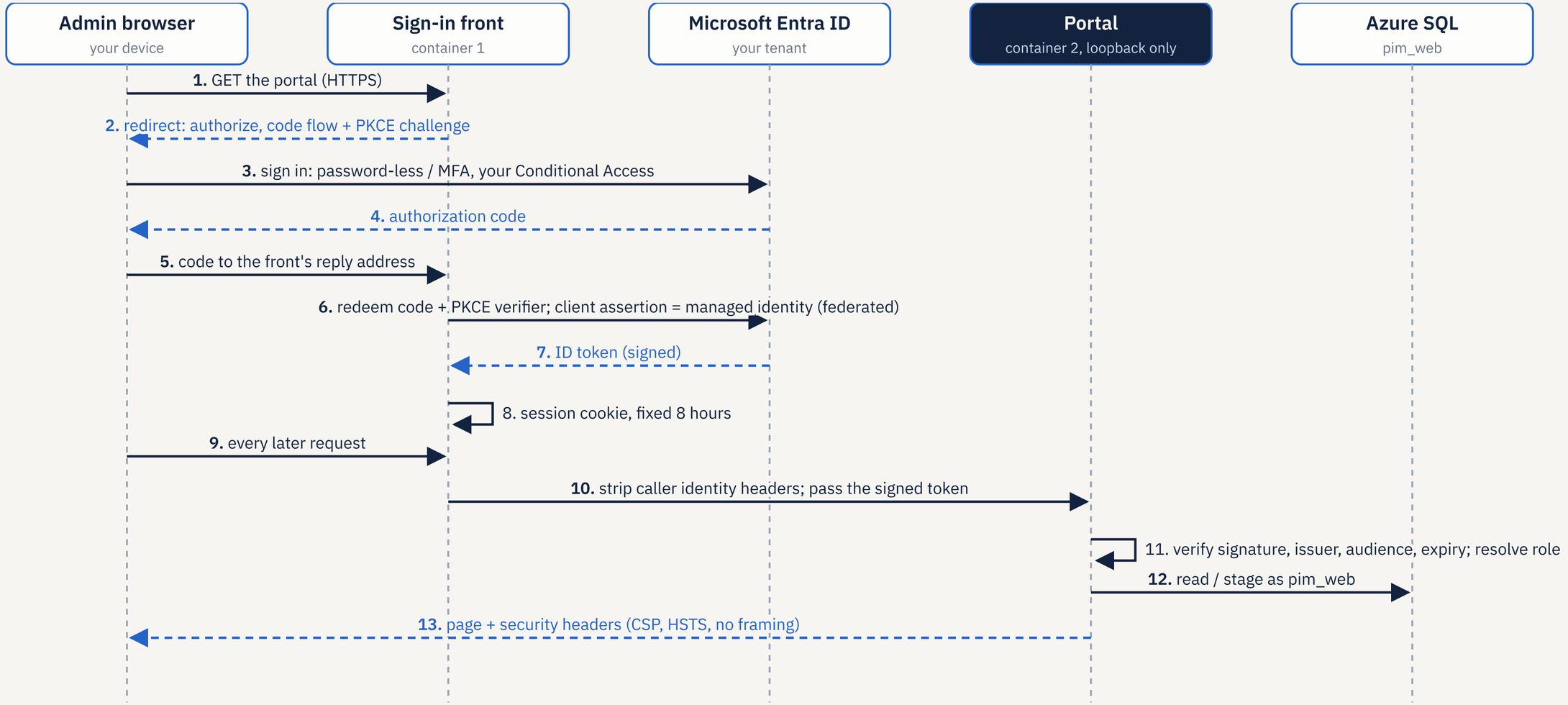
Session
Fixed 8 hours; your Conditional Access

Roles
Reader, Delegated, Admin, SuperAdmin

Separation
Second approver; nobody approves own change

SEQUENCE · USER SIGN-IN

Signing in: no secret anywhere, a signed token on every request



SECURITY · THE SIGN-IN FRONT, FOR THE CISO

Sign-in without a secret: nothing to steal, expire or rotate

What it is

- The only part of the GUI the network reaches
- OpenID Connect, code flow with PKCE
- Your tenant only; assignment required
- Asks only for sign-in: name, e-mail
- No API permissions, no Azure roles

How it proves itself

- With the GUI's own managed identity, as a federated credential
- No client secret, no certificate
- The PIM Manager GUI listens only inside the app, behind the front

On every request

- Identity headers from a caller are removed
- The user's signed token is verified
- Session ends after a fixed 8 hours
- Your MFA + Conditional Access apply at each sign-in

Browser hardening on every page: a strict content security policy, HSTS, no framing by other sites, no content-type sniffing, a same-origin referrer policy.

SECURITY · SIGN-IN: WHAT AN ATTACKER WOULD NEED

There is no sign-in secret to leak

Scenario	What they would need	Result
Steal the sign-in app's secret	There is none: the front proves itself with a managed identity	Nothing to steal
Call the GUI directly, around the front	The GUI listens only inside the app	Not reachable
Forge a user with a header	The front removes caller headers; the GUI verifies the signed token	Refused
Sign in as someone	That person's own sign-in: password, MFA, your Conditional Access	Not possible without the user
Keep a stolen session	Sessions end after a fixed 8 hours	Bounded

Evidence · Microsoft

A federated identity credential lets a managed identity act as an app's client credential (Microsoft Learn: workload identity federation)

Evidence · PIM Manager

A test fails the build if any route answers without a signed-in user

You keep

Assignment required, your Conditional Access and the Entra sign-in logs of the sign-in app

SIGN-IN · SESSION AND HEADERS

Sent with every page

Header / setting	Value
Session	Fixed 8 hours; Conditional Access and MFA apply at each sign-in
Content-Security-Policy	default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'; img-src 'self' data: blob:; font-src 'self' data:; connect-src 'self'; frame-src 'self'; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'none'
Strict-Transport-Security	max-age=31536000
X-Frame-Options	DENY
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin
No identity	401; no identity is ever given a default role

AUTHENTICATION · CREDENTIALS AND KEYS

Every credential and key: where it lives, and how it is rotated

Who / what	Credential	Stored	Rotation
Administrators	Their own Entra sign-in; MFA by your Conditional Access	Entra ID	Yours
Sign-in front	Managed identity as a federated credential	Nothing stored	Not needed
Portal, engine, updater, jobs	Managed-identity tokens per API audience	Nothing stored	Azure, automatically
Database users	Entra tokens; user created from the client id	Nothing stored	Not needed
Grant procedure certificate	In-database certificate, private key removed	Inside the database	None: expires 2099-12-31
Update signing key	RSA 4096, held by Invardia	Invardia; public key built into the product	With a release
Licence	Signed RS256 file, checked offline	Your database	Renewed by request; proves the current licence
Pro install key	Shared key for status reports and updates	Container Apps secret, never a plain setting	Renewed with the licence
MSP signing key (option)	Non-exportable Key Vault key (HSM with Premium)	Key Vault	Managing tenant's SuperAdmins; revocable
Invardia Support app	Client secret	Invardia's support Key Vault	90 days; old secrets removed

AUTHORIZATION · ROLES AND SEPARATION OF DUTIES

Who may do what, and who approves

Control	Rule
Reader	Sees everything, changes nothing; any signed-in user named nowhere
Delegated (Pro)	Manages admins and delegations only inside its profile: services, scopes, maximum level and tier; a level-2 operator cannot touch a level-1 or level-0 admin
Admin	Stages and commits changes; undoes own commits; deletes PIM-managed admin accounts at or below its level
SuperAdmin	Settings, restore, guard release, policy approval, undo any commit, the access model; the model refuses to be left without a SuperAdmin
Second approver (Pro)	Sensitive changes (privileged roles, Tier 0/1, guests, offboarding) wait for a colleague; self-approval refused
Break-glass	Listed accounts are never removed, disabled or revoked; changing the list needs a second SuperAdmin (not a setting); an unreadable list protects every account
Policy and guard approvals	Only a SuperAdmin of your own tenant, bound to the exact change set, with a reason; Invardia can approve nothing
Just in time	Admins hold eligible access and activate it in native Entra PIM; PIM Manager configures policies but is not in the activation path
Unknown role	A role that cannot be determined drops the user to read-only

SECURITY · WHO MAY SIGN IN, AND WHAT THEY MAY DO

Entra sign-in, assigned users only, four roles

Entra sign-in at the sign-in front, before PIM sees a request; no secret

Assignment required: members of one group, no guests

No identity = 401, never a default role

Role	May
Reader	See everything, change nothing (any unknown signed-in user)
Delegated	Manage admins and delegations inside a profile: services, scopes, maximum level and tier
Admin	Stage and commit changes; undo own commits; delete admins
SuperAdmin	Settings, restore, guard release, undo any commit, access model

A delegated level-2 helpdesk operator cannot touch or delete a level-1 or level-0 admin; an unreadable level is refused.

CHAPTER 5

INVARDIA PIM MANAGER PRO

Change control

What PIM Manager can change in your tenant, and the flow from a click to an API call: staged, approved, committed, journaled.

Staged
Nothing reaches the tenant without a commit

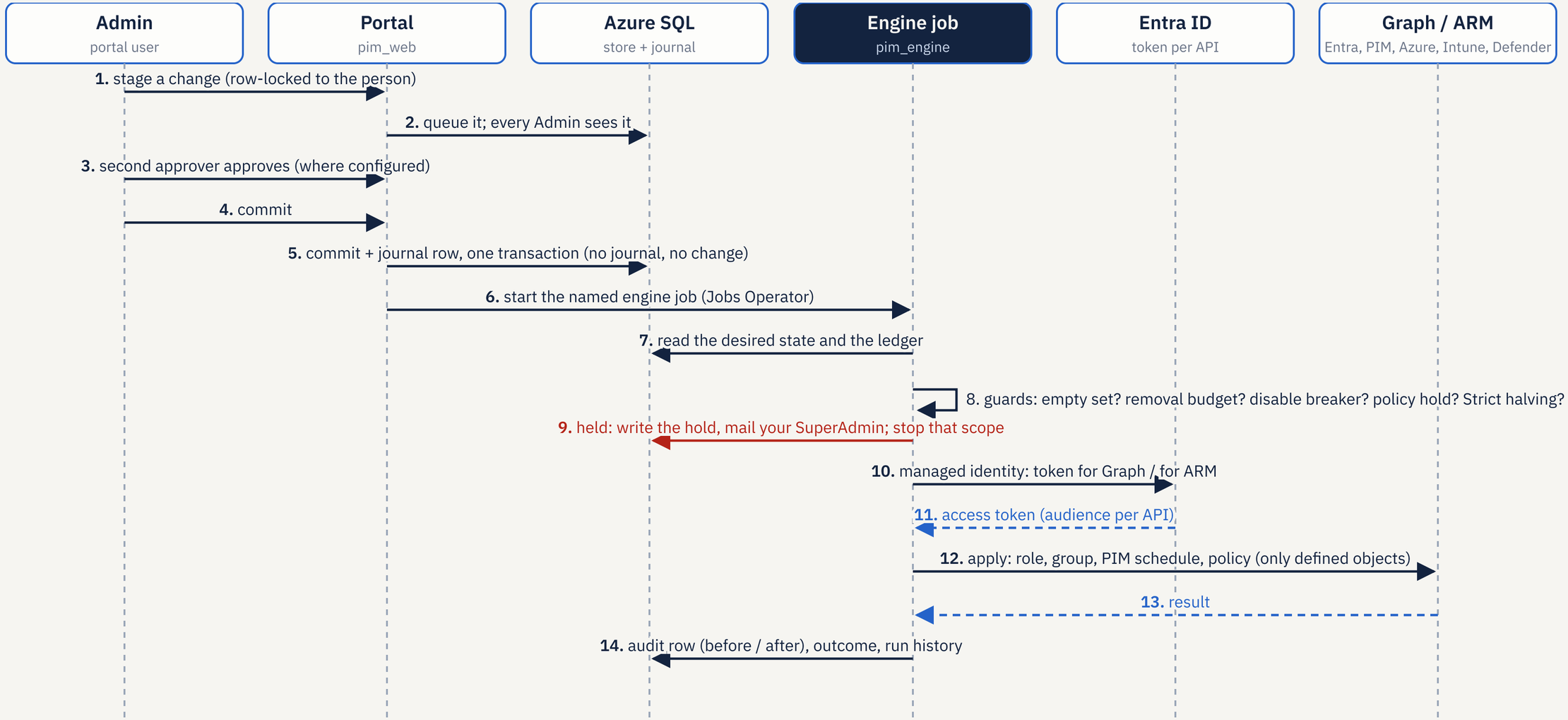
Guarded
Held for your SuperAdmin; halved after an engine change

Journaled
Before and after, same transaction

One delete path
Operator-confirmed, at most 5 per run

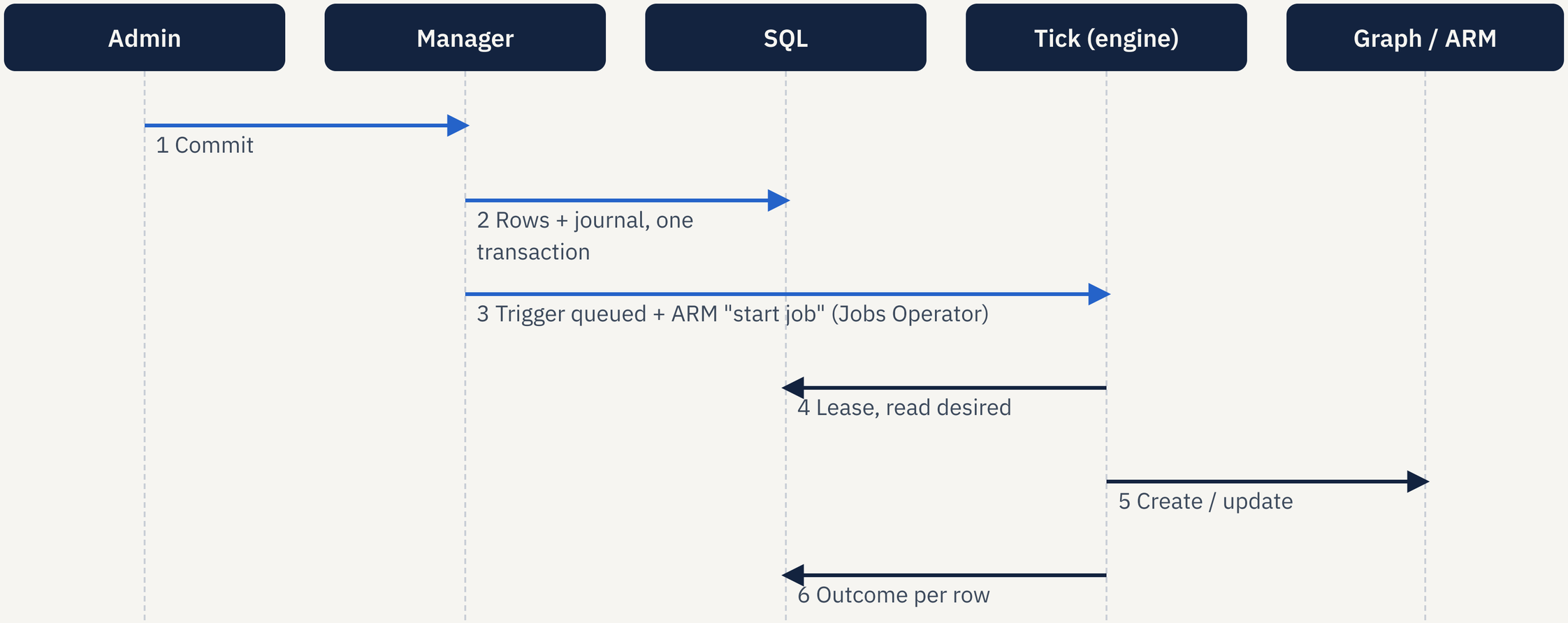
SEQUENCE · THE ENGINE ACCESS FLOW

From a staged change to an API call: every step and every check



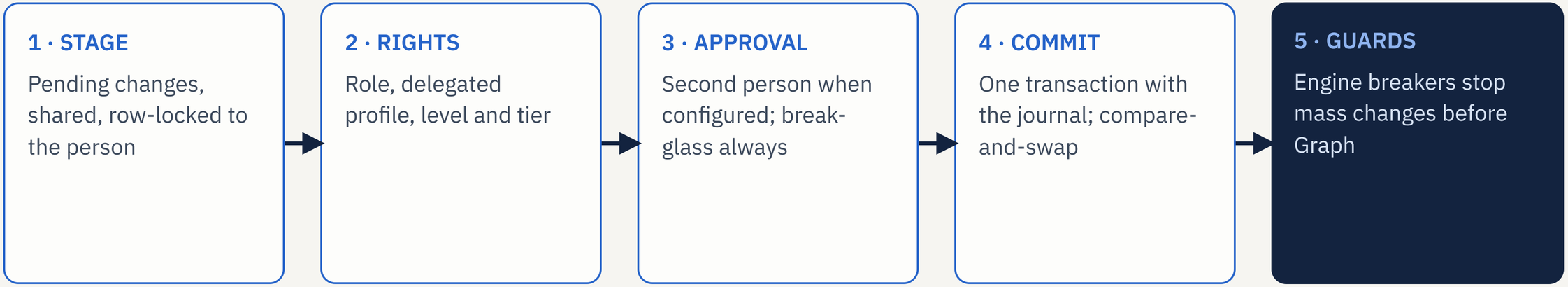
ENGINE · FROM COMMIT TO TENANT

A commit reaches the tenant in about a minute



SECURITY · CHANGE SAFETY

Every change passes five gates before it touches the tenant



Only what is defined

The engine never touches objects the store does not define

Admins are never pruned

Deleting an admin is only a person's action in GUI or API

Removal needs proof

Five rules + the change journal before reconcile removes

CHANGE CONTROL · WHAT PIM MAY CHANGE

Every kind of change, and its control

Change	Allowed?	Control
Create admin accounts the store defines	Yes	Commit; TAP issued once, never stored
Add / remove role and group assignments	Yes	Commit; removal budget per scope; held lists need your SuperAdmin
Change PIM policies	Yes, by choice	Only where you chose a value; every change set held for your SuperAdmin; safety brake on weakening
Disable admin accounts	Opt-in	Off by default; mass-disable breaker
Delete an account	One path	Operator's Delete account on a PIM-managed admin, typing DELETE; sessions revoked first; at most 5 per run; restorable 30 days; can be switched off
Azure role assignments	Opt-in	Only with User Access Administrator, which is off by default
Objects the store does not define	**Never**	Reported, never removed (managed-object ledger)
Break-glass accounts	**Never**	Excluded; no release can override
Act on an empty or unreadable store	**Never**	Guard G1: touches nothing
Accounts it does not manage	**Never deleted**	A prune or reconcile only reports

CHAPTER 6

INVARDIA PIM MANAGER PRO

Guards and guard modes

Every write the engine makes passes the guards first. A guard **holds** what cannot heal by itself and waits for a SuperAdmin of **your** tenant; after an engine change every limit is halved for 24 hours.

Counts, not shares
Five settable limits with a floor and a ceiling

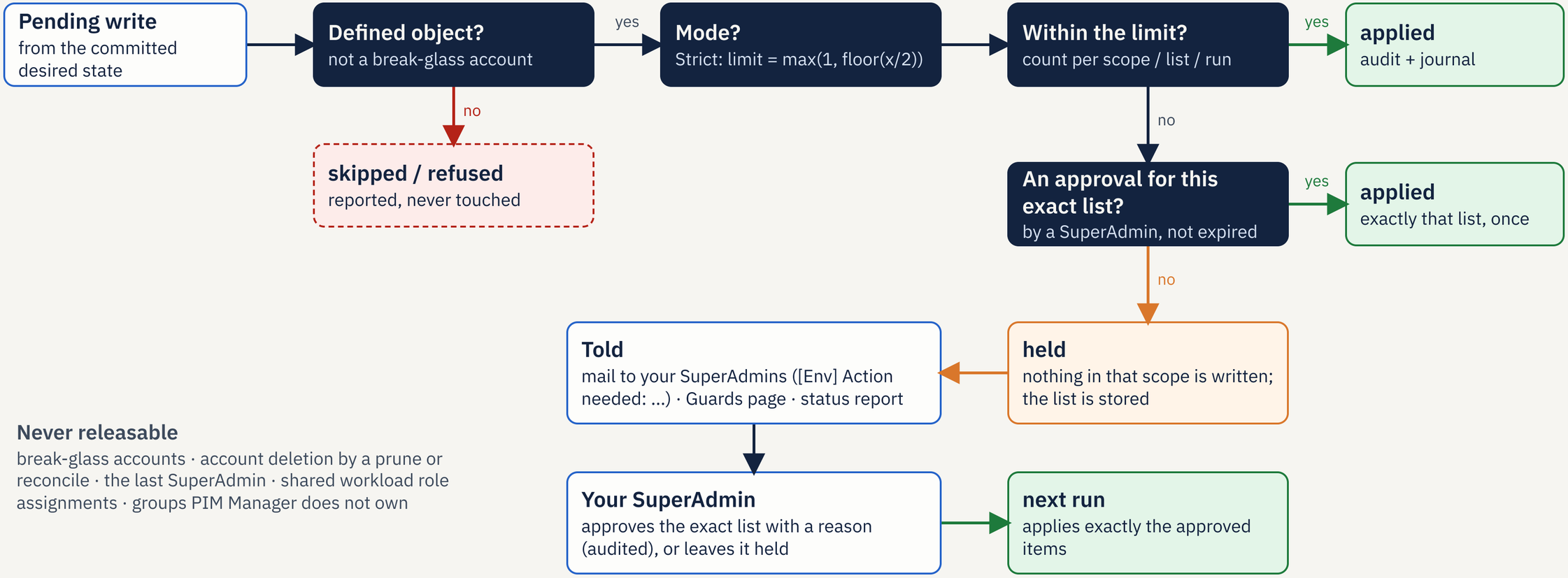
Normal / Strict
Halved for 24 h after an engine change

Your SuperAdmin
Approves the exact held list; Invardia cannot

Told at once
Mail, the Guards page, a status report

GUARDS · DECISION FLOW PER WRITE

What happens before every change the engine writes



Never releasable

break-glass accounts · account deletion by a prune or reconcile · the last SuperAdmin · shared workload role assignments · groups PIM Manager does not own

GUARDS · EVERY GUARD

Guard, limit, Strict limit, outcome, and who is told

Guard	Protects against	Default (min-max)	Strict	Outcome	Who may change / is told
manager.commit-removal	A commit that takes many rows out of one list, or keeps none	25 (5-200)	12	held in the commit dialog	SuperAdmin sets; a SuperAdmin approves with a reason
engine.remove-budget	Mass removal: assignments, prune, reconcile, retire, group delete, offboarding, MSP withdrawal	5 per scope (1-25)	2	held	SuperAdmin sets; SuperAdmins mailed; approval 48 h
engine.disable-guard	Mass disabling of admin accounts	3 per run (1-25)	1	held	SuperAdmin sets; SuperAdmins mailed; released for one run
manager.bulk-revoke	A large revoke in one batch	5 (1-25)	2	held for approval	SuperAdmin sets; approved on the Approvals page
engine.admin-delete	Too many account deletes in one run	5 per run (1-25)	2	the rest wait	SuperAdmin sets; typing DELETE approves each
Policy change hold	Any PIM policy change set, whatever its size	every set	every set	held	Only your tenant's SuperAdmin approves (24 h); never downgraded to a warning
Empty desired set	Acting on an empty or unreadable configuration	never acts	-	halted	SuperAdmins mailed
Last Global Administrator	Leaving the tenant without one	held	-	held	A SuperAdmin release for that plan
Last SuperAdmin; break-glass	Locking yourself out; touching emergency accounts	always	always	refused / skipped	Nobody can release

Limits are set by a SuperAdmin on the Guards page within the floor and ceiling; a server setting can only lower them; no guard can be switched off. The hybrid server batch is halved in Strict by its own reader.

GUARDS · OUTCOMES

What each outcome does

Outcome	What happens	What you do
held	Nothing in that scope is written; the exact list is stored and shown; your SuperAdmins are mailed	Approve the exact list with a reason, or leave it held; a stale hold clears itself
halted	The run stops before writing anything (for example an empty or unreadable desired set)	Fix the cause; the next run continues
refused	A locked change is never made (last SuperAdmin, break-glass, unowned objects)	Nothing: it is never releasable
skipped	One item is left out and reported; the rest of the run goes ahead	Review the report
warning	A gap the next run can close; listed in one daily mail	Nothing urgent; the menu shows no number for a warning

SECURITY · GUARDS AND CIRCUIT BREAKERS

The engine holds what cannot heal by itself

Guard	Stops	Limit
G1 · empty desired set	Acting on an empty or unreadable configuration	Never acts
G2 · disable breaker	Mass disabling of admin accounts	3 accounts per run (1-25)
G3 · disabling opt-in	Automatic offboarding unless switched on	Off by default
G4 · removal budget	Mass removal of assignments	5 per scope per run (1-25)
Last Global Administrator	A change that would leave the tenant without one	Held; released for that plan only
Policy change hold	Every PIM policy change set, whatever its size	Held until a SuperAdmin of your tenant approves it (24 h)
Last SuperAdmin	Leaving PIM Manager without a SuperAdmin	Always refused
Break-glass	Any change to a break-glass account	Always excluded

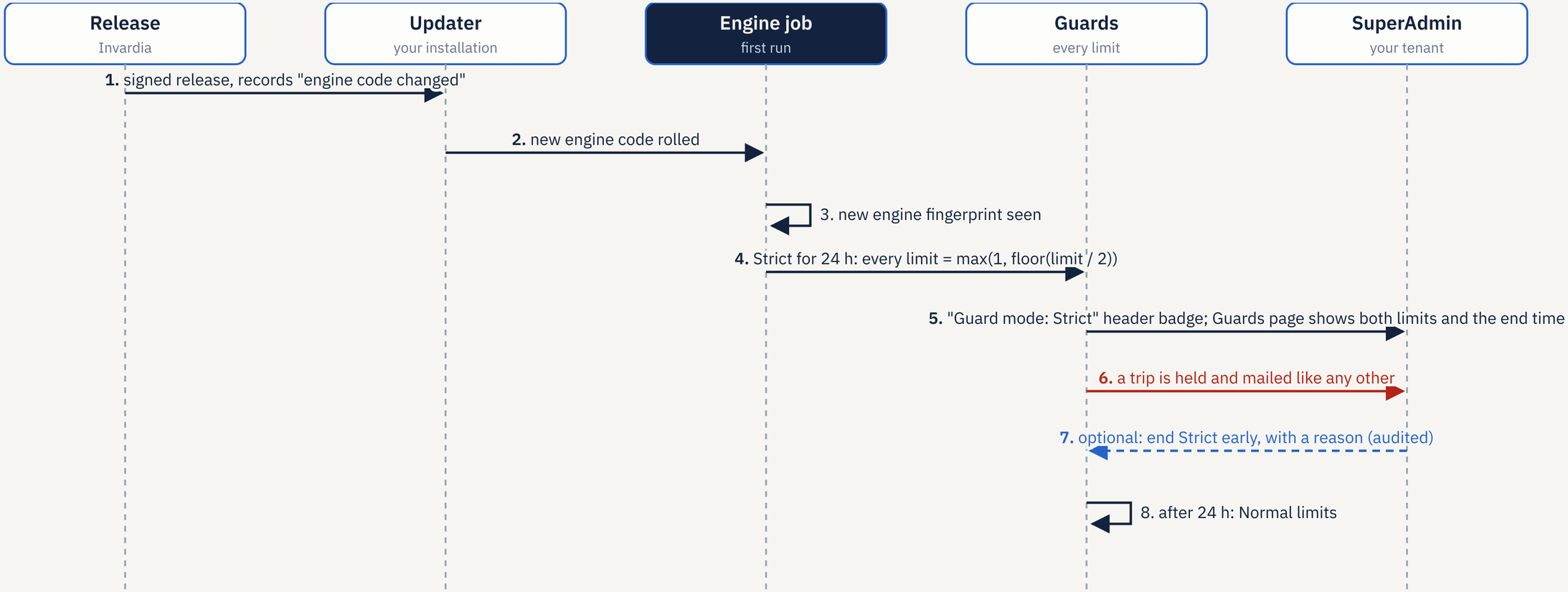
Release a guard: SuperAdmin, a reason, bound to that exact list, one run or 24 h

Strict mode: for 24 h after an update that changed engine code, every limit is halved (never below 1)

Never releasable: break-glass, admin deletion by prune, the last SuperAdmin, shared workload roles, unowned groups

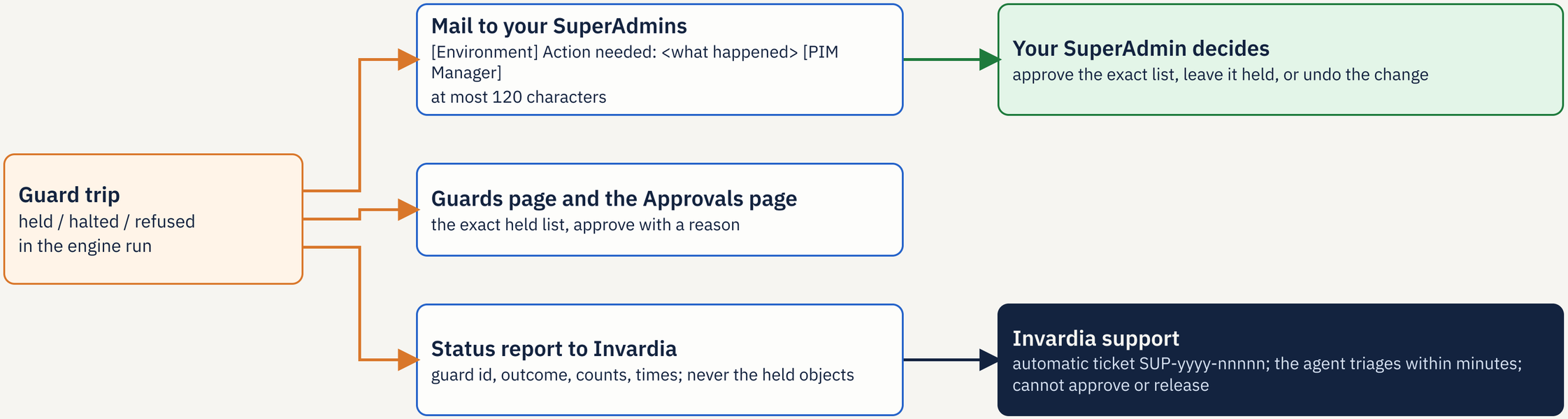
GUARDS · NORMAL AND STRICT MODE

A safer first day after every engine change



GUARDS · WHO IS TOLD

A trip reaches your SuperAdmins and Invardia within minutes



Never downgraded

A mass policy change is held for the tenant's own SuperAdmin; it is never turned into a warning, and neither Invardia nor a support engineer can approve it.

CHAPTER 7

INVARDIA PIM MANAGER PRO

Database access

One user and one custom role per component, **no passwords anywhere**, and rights that change only through one signed procedure. Designed to run five years with nobody in the admin group.

Entra-only
No SQL logins, no passwords

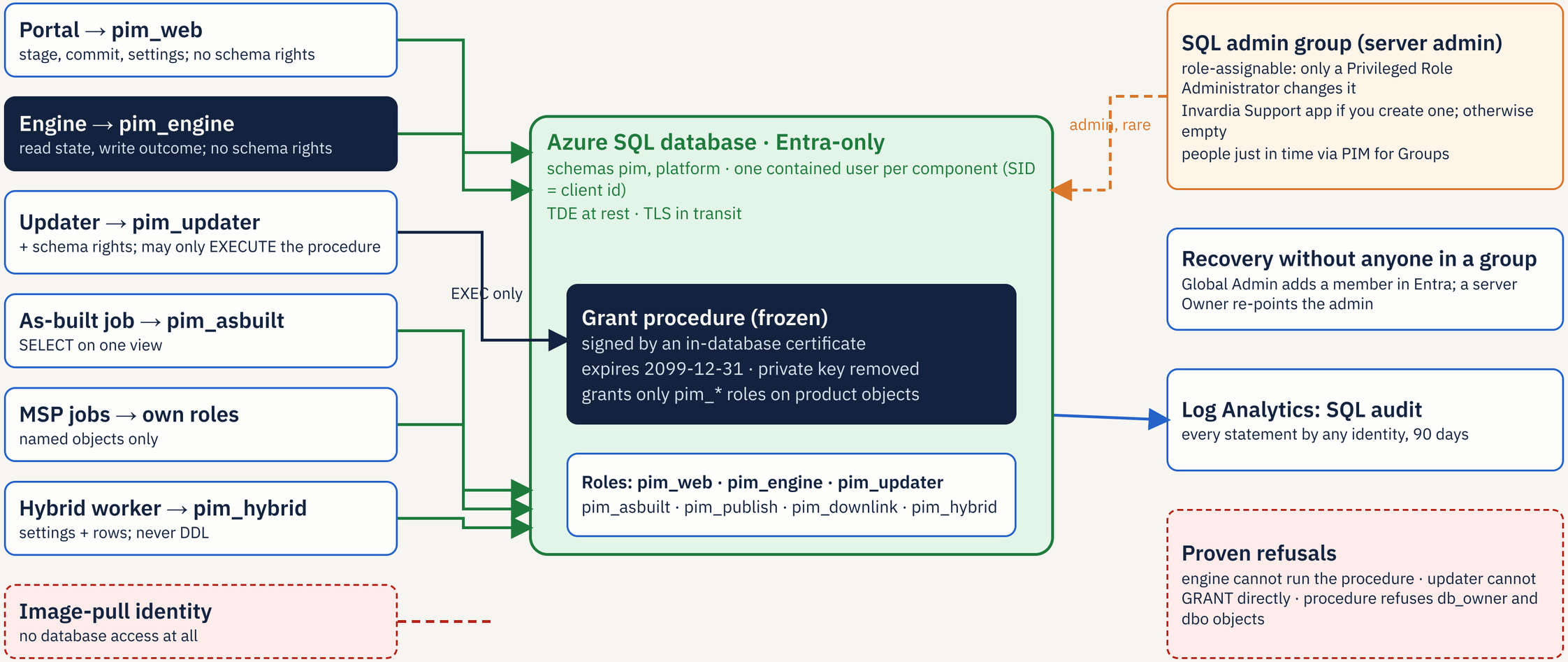
Per-component roles
pim_web, pim_engine, pim_updater, …

Signed procedure
The only way rights change

Admin group
Support app only, or empty

DATABASE · WHO REACHES IT, WITH WHICH RIGHTS

Every identity, its role, and what it can never do



DATABASE · IDENTITY BY ROLE

Every identity and its database role

Identity	Database user	Role	May	Never
Portal	ca-pim-manager	pim_web	Read; stage and commit; settings; audit rows	Schema, security, the procedure
Engine	ca-pim-tick	pim_engine	Read the state; write outcome, audit, journal	Schema, security, the procedure
Updater	ca-pim-update	pim_updater + db_ddladmin	Additive schema; data migrations; EXECUTE the procedure	GRANT directly; admin group seat
As-built job	caj-pim-asbuilt	pim_asbuilt	SELECT on one view	pim.Settings, any write
MSP publish	ca-pim-publish	pim_publish	SELECT named objects; its own last-run view	Any other write
MSP downlink	own	pim_downlink	SELECT, INSERT, UPDATE on pim.Settings	Delete, schema
Hybrid worker	own	pim_hybrid	Settings; read rows	Delete, schema
Image pull	-	-	-	Any database access
Invardia Support app	own (optional)	Admin group member	Setup, repair, troubleshooting	Runtime; it is never required

Least privilege for every product database

#	Principle	What it means
P1	Entra-only	No SQL logins or passwords on any server
P2	One user per component	Mapped from its own managed identity; an image-pull identity has no database access
P3	One custom role per component	Never db_datareader / db_datawriter / db_owner on the whole database; schema-level where it owns the data, object-level elsewhere, plus per-component views and procedures over the shared tables; no DELETE where it never deletes
P4	No standing security right at runtime	Grants after a release only through ONE certificate-signed procedure; the updater may only run it
P5	Admin group = Support app only	People get in just in time through PIM for Groups; setup identities only for the length of an install
P7	Audited and tested	SQL auditing to your workspace (90 days) + a static test that fails when code touches a table outside its role
P8	Scripted, audited cut-over	Existing installations move through the Support app, one at a time, plan first, with a rollback record
P9	Five years without contact	Every update applies its own schema and grants; the admin group may be empty for years

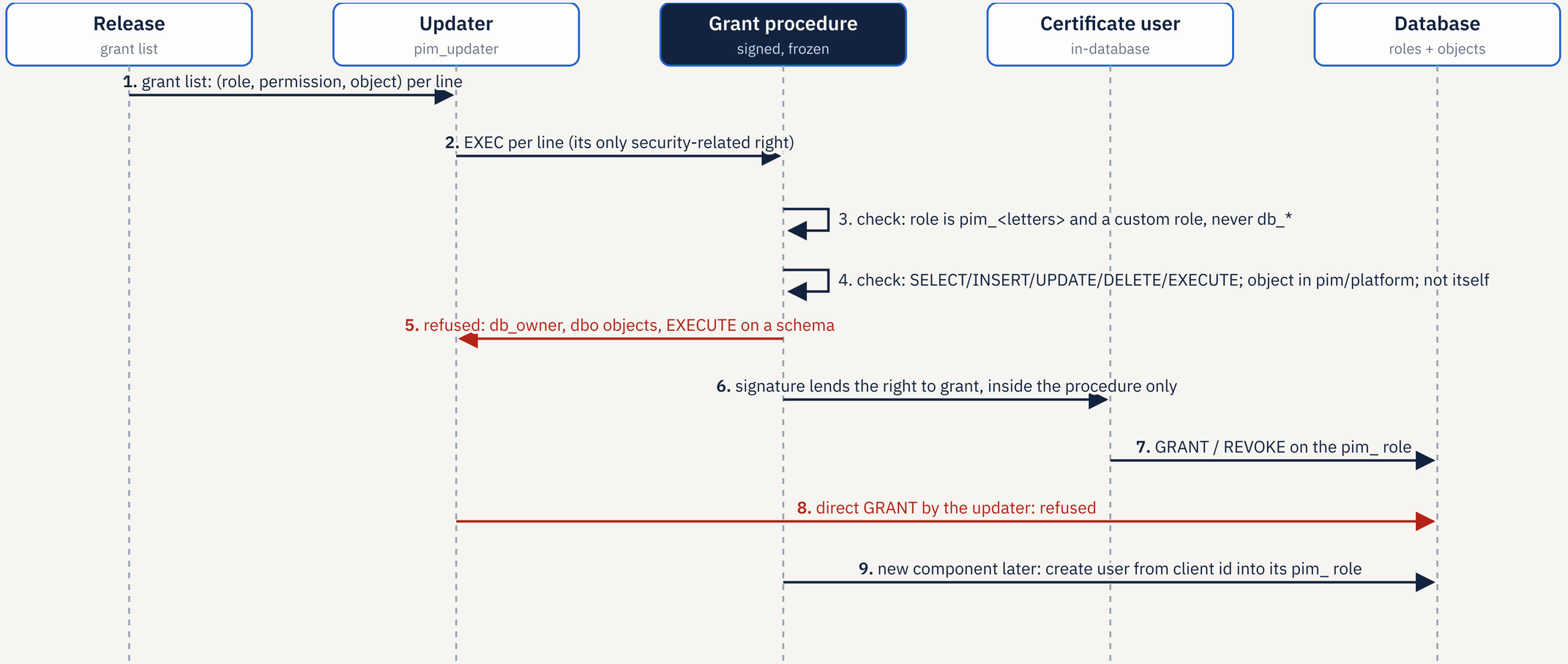
DATABASE · PER-COMPONENT ROLES

What each role may do

Role	Grants	Why
pim_web	SELECT, INSERT, UPDATE, DELETE on schema pim; SELECT on schema platform; INSERT, UPDATE on platform.Tenants	Stage and commit changes; MSP tenant list
pim_engine	SELECT, INSERT, UPDATE, DELETE on schema pim; SELECT on schema platform	Read desired state, write outcome, audit, journal
pim_updater	SELECT, INSERT, UPDATE, DELETE on schema pim; SELECT, INSERT, UPDATE on schema platform; schema rights; EXECUTE on the grant procedure	Schema and data migrations, grants
pim_asbuilt	SELECT on pim.vw_AsBuiltSettings only	The as-built PDF; never pim.Settings
pim_publish	SELECT on four named tables + two views; INSERT, UPDATE on its own last-run view	Build the MSP bundle
pim_downlink	SELECT, INSERT, UPDATE on pim.Settings	Store the pulled MSP bundle state
pim_hybrid	SELECT, INSERT, UPDATE on pim.Settings; SELECT on pim.Rows	Hybrid AD results

No role holds a fixed database role, CONTROL, ALTER or a security right. Per-component views narrow the shared tables further. A static test compares each component's SQL text with its role.

Rights change only through one frozen, signed procedure



DATABASE · THE CERTIFICATE

A certificate inside the database, with nothing to maintain

Created once

- By the product's own setup, or the audited Support-app cut-over
- Inside the database: never a file, never in Key Vault
- Its password is generated inside SQL and never shown

Signs, then loses its key

- ADD SIGNATURE to the grant procedure
- REMOVE PRIVATE KEY right after
- Nobody can sign anything else with it

Never expires in practice

- Expires 2099-12-31
- Nothing to rotate, renew or distribute
- A changed procedure (rare) gets a new certificate from the same script

No passwords anywhere: component users sign in with Entra tokens; the certificate's user cannot sign in at all; the certificate's password unlocks nothing once its private key is gone.

DATABASE · PROVEN ON A LIVE INSTALLATION

What we tested, and what SQL Server answered

Test	Result
Which certificate signs the procedure, and does it have a private key?	In-database certificate, expiry 2099-12-31, **no private key**
The updater runs GRANT directly	**Refused** by SQL Server
The updater grants through the procedure	Applied, for pim_ roles on product objects only
The engine runs the procedure	**Refused**: it has no EXECUTE right
The procedure is asked to grant to db_owner	**Refused** by the procedure
The procedure is asked to grant on a dbo object	**Refused** by the procedure
How are database users created?	From the identity's client id (SID), Entra type; **no passwords anywhere**

DATABASE · FIVE YEARS WITHOUT CONTACT

Nothing routine needs the admin group, or anyone in it

Every update, unattended

- Schema changes by the updater's own user
- Grants through the signed procedure
- New component users through the same procedure
- No admin, no person, no Support app

The admin group

- Role-assignable: only a Privileged Role Administrator changes it
- Holds the Invardia Support app if you create one
- Otherwise empty, for years if need be
- People just in time through PIM for Groups

Recovery always exists

- A Global Administrator adds a member in Entra: no database access needed
- An Owner of the SQL server can point the administrator at another group
- The Support app is created only on request

DATABASE · AUDITED CUT-OVER

How an existing installation moves to the model

Step	How	Proof
1 Plan	The script reads the installation and prints every change; nothing is written	The plan
2 Add	Run 1: roles, the signed procedure and its certificate; users join their roles	Read-back
3 Prove	A green pipeline: engine runs since, 0 failed; one real update through the procedure	Run history
4 Remove	Run 2: the broad roles and the updater's admin-group seat go	Read-back
5 Record	A rollback record of every value before it changed	The record

Every SQL statement runs through Invardia's audited support route, which records the statement and its reason before it runs, and appears in the installation's SQL audit. Order: the vendor's own installation, the test tenants, then customers in batches, each customer told first.

CHAPTER 8

INVARDIA PIM MANAGER PRO

Data

What is stored, **where**, how it is encrypted, how long it is kept, and how it is deleted. Your data stays in your subscription and region.

Location
Your subscription, your region

At rest
Azure SQL TDE

In transit
TLS everywhere, certificate-validated

Leaves
Only status reports, no directory data

DATA · LOCATION, ENCRYPTION, RETENTION

Every kind of data: where, how protected, how long

Data	Location	Encryption	Retention	Tamper protection
Desired state, definitions (names, UPNs, manager e-mail)	Your Azure SQL, your region	TDE; TLS	Until changed	Journal
Settings, ledger, tenant cache	Your Azure SQL	TDE; TLS	Until changed	Journal
Audit trail (before / after)	Your Azure SQL	TDE; TLS	Forever by default; 13 months minimum	Append-only trigger
Commit journal	Your Azure SQL	TDE; TLS	Never purged	Append-only, same transaction
Configuration backups	Your Azure SQL	TDE; no secrets in them	90 days default; pinned kept	-
SQL security audit	Your Log Analytics	Azure platform	90 days	Outside the database
Container logs	Your Log Analytics	Azure platform	Workspace default (30 days)	Outside the database
Secrets (SMTP, passphrase, MSP key)	Your Key Vault	Key Vault; HSM option	Until removed; purge protection	RBAC
Status reports	Invardia	HTTPS	Per Invardia's policy	No directory data
Never stored	-	-	TAP codes and generated passwords shown once; API keys only as SHA-256 hashes	-

DATA · FLOWS

Every flow, its direction and how it is authenticated

From	To	Protocol	Authentication	Content
Admin browser	Sign-in front	HTTPS	Entra sign-in (OIDC + PKCE)	GUI and API requests
Sign-in front	Portal (loopback only)	Inside the app	User's signed token, verified	The request, caller headers removed
Engine, portal	Microsoft Graph	HTTPS	Managed-identity token	Directory, roles, PIM, Intune, Defender, mail
Engine, portal, updater	Azure Resource Manager	HTTPS	Managed-identity token	Azure roles, start the engine, roll containers
All components	Azure SQL	TDS over TLS	Entra token; own user and role	Desired state, settings, queue, audit
Updater	invardia.com	HTTPS	Licence + install key	Signed manifest, source archive
Engine	invardia.com	HTTPS	Install key, or anonymous	Status reports, licence request
Azure SQL	Log Analytics	Azure diagnostics	Platform	Security audit events
Invardia	Your environment	-	-	**Nothing connects in**

DATA · WHAT REACHES INVARDIA

Status reports carry no directory data

Record	Fields	Mode
Every record	Product, version, ring, last seen	Both
Heartbeat (4 h)	Edition, hosting, runtime identity type, licence state; identified: environment name, guard mode	Both
Run report (per job)	Job, outcome, duration, error class, count	Both
Guard record	Guard id, outcome, severity, numbers and limits, held count, times; never the held objects	Both
Anonymous mode adds	A random install id; counts bucketed	No install key
Identified mode adds	Tenant id, host, error text with tokens, keys, names and UPNs removed	Pro install key
Licence request	Product, version, tenant id, install id, licence id and end; proof of the current licence	On by default; can be off
Opt-out	One final record, then nothing	Both

DATA · ENCRYPTION AND SIGNATURES

At rest, in transit, and integrity

Control	How	Note
At rest	Azure SQL transparent data encryption, service-managed keys	Customer-managed keys not configured by the product
In transit	HTTPS to every API; SQL with Encrypt=True and certificate validation	Set minimum TLS 1.2 on the server (checklist)
Updates	Manifest RS256 (RSA 4096), key built in; sequence, SHA-256, size, version checked	Invardia holds the signing key
Licence	RS256, checked offline, bound to the engine identity's own tenant	A copied licence unlocks nothing
MSP bundles	Non-exportable Key Vault key; managed tenant pins the key id	Expiry, anti-rollback, revocation
Grant procedure	In-database certificate, private key removed, expiry 2099	Altering the procedure drops the signature

DATA · RESIDENCY AND DELETION

Where it lives, and how it leaves

Topic	How
Region	Your choice; setup checks every service it needs exists there before anything is created
Hosting	Every resource in your subscription; Invardia hosts no data and no runtime
Backups	Azure SQL point-in-time restore (your settings); every restore goes to a new database
Exit	Export the configuration; delete the resource group: apps, jobs, database (all stored data), registry, workspace, identities
Key Vault	Soft-deleted for 90 days, then purged; purge protection on the product's own vaults
Left behind (remove by hand)	Role assignments at the tenant root or other subscriptions; app registrations (sign-in, access request, Activator); the admin accounts and groups, which are your access model and keep working

CHAPTER 9

INVARDIA PIM MANAGER PRO

Logging and audit

Four independent records: the **before / after** audit trail, the commit journal, the database's own SQL audit, and your Entra and Azure logs.

Before / after
Every change, who, why

Append-only
Triggers on trail and journal

Outside the database
SQL audit 90 days

SIEM
Export + Log Analytics

SECURITY · AUDIT AND RECOVERY

Every change is recorded, and every mistake can be undone

Audit log

Who, what, target, before and after, run. Append-only by trigger. Never holds passwords or TAP codes. Kept forever unless set (minimum 13 months).

Change journal

Every row change, in the same transaction as the change: no journal, no change. Append-only, never pruned.

Undo

A commit is reversed by an inverse commit through every normal gate. Admins undo their own; SuperAdmins any.

Backups + restore

Daily, kept 90 days (configurable), secrets stripped. Restore picks rows, which are staged and committed normally.

AUDIT · WHERE EVIDENCE LIVES

Four independent records

Record	What it holds	Protection	Kept
Audit trail (pim.AuditEvents)	Actor, source, action, target, before, after, dry-run flag	Append-only trigger; purge only by the retention job, itself audited	Forever by default; 13 months minimum
Commit journal (pim.CommitJournal)	Every commit: before, after, initiator, approver	Append-only; same transaction as the change	Never purged
SQL audit (Log Analytics)	Every statement by any identity, including the Support app	Outside the database	90 days
Container logs (Log Analytics)	Every job run and request	Outside the database	Workspace default
Your Entra and Azure logs	Sign-ins and changes by every product identity	Yours	Your policy

SIEM: connect the Log Analytics workspace (SQL audit, container logs) and your Entra and Azure logs to your SIEM; export the audit trail as CSV. Direct forwarding of the audit trail itself is on the backlog. **Tamper resistance:** with the per-component roles only the updater keeps schema rights; a trigger change is recorded by the SQL audit.

CHAPTER 10

INVARDIA PIM MANAGER PRO

Release process

From code to your installation: tested offline, signed, proven on the vendor's own installation and test tenants, **approved by the release owner**, then rolled out in waves.

Ring 0 first
The vendor's own installation

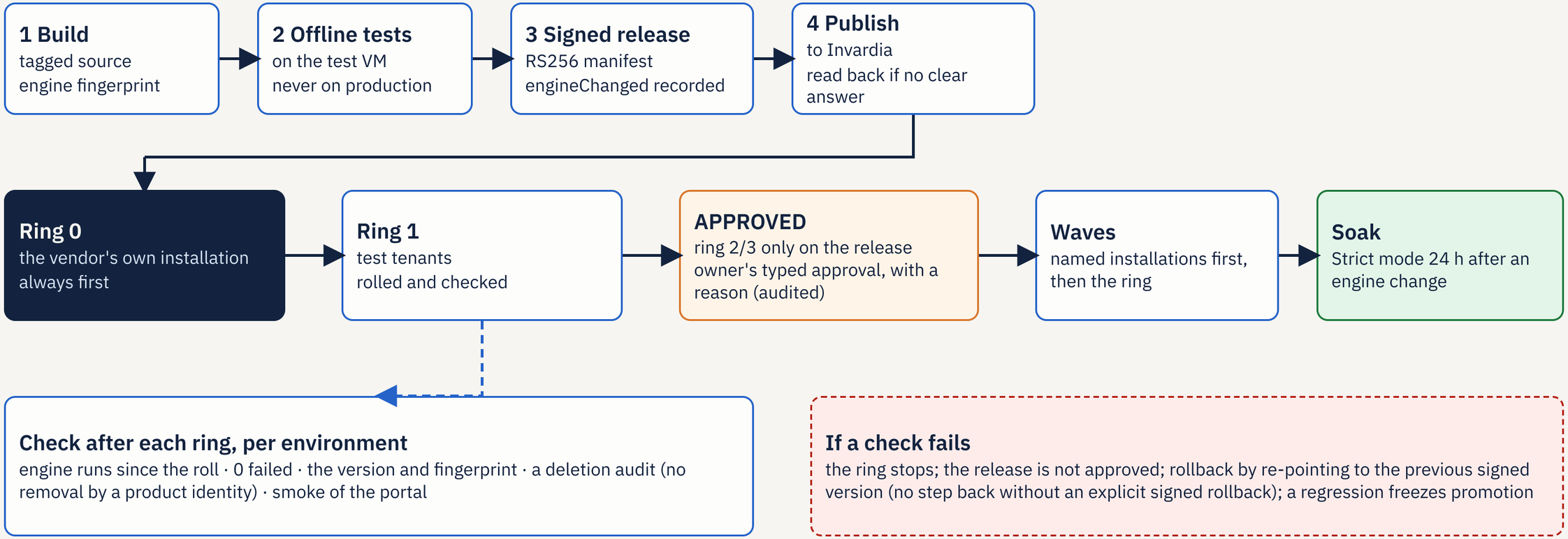
Ring 1
Test tenants, checked per environment

Approved
Ring 2/3 only on the owner's written approval

Strict
Halved limits after an engine change

RELEASE PROCESS · FROM CODE TO CUSTOMER

Every release takes the same path, and every gate has an owner



Who approves

Rings 0 and 1: the release tooling, after tests. Rings 2 and 3 and every wave: only the release owner's written approval for that exact version. Infrastructure changes to a running installation are separate: through the Support app, never through an update.

A customer ring only gets an approved version

Ring	Who	A version gets there
0	Development: the vendor's own installation	First, with every release
1	Test: the vendor's test tenants	With every normal release, after its tests; checked per environment
2	Design partners	Only when the release owner marked that exact version APPROVED, with a reason (audited, append-only)
3	Broad (new installations start here)	The same; never advanced automatically
Wave	Named installations inside a ring	Early, only after a lower ring has the version; leaving a wave never moves an installation back

Evidence for each approval: whether the engine changed; run results and guard trips after the roll; the engine's own changes; installations report more often in the first hours after an engine change.

CHAPTER 11

INVARDIA PIM MANAGER PRO

Updates and supply chain

Your installation **pulls** its ring's signed release at night, verifies it, builds it from source in your own registry, and rolls forward only.

Pull
Nothing is pushed

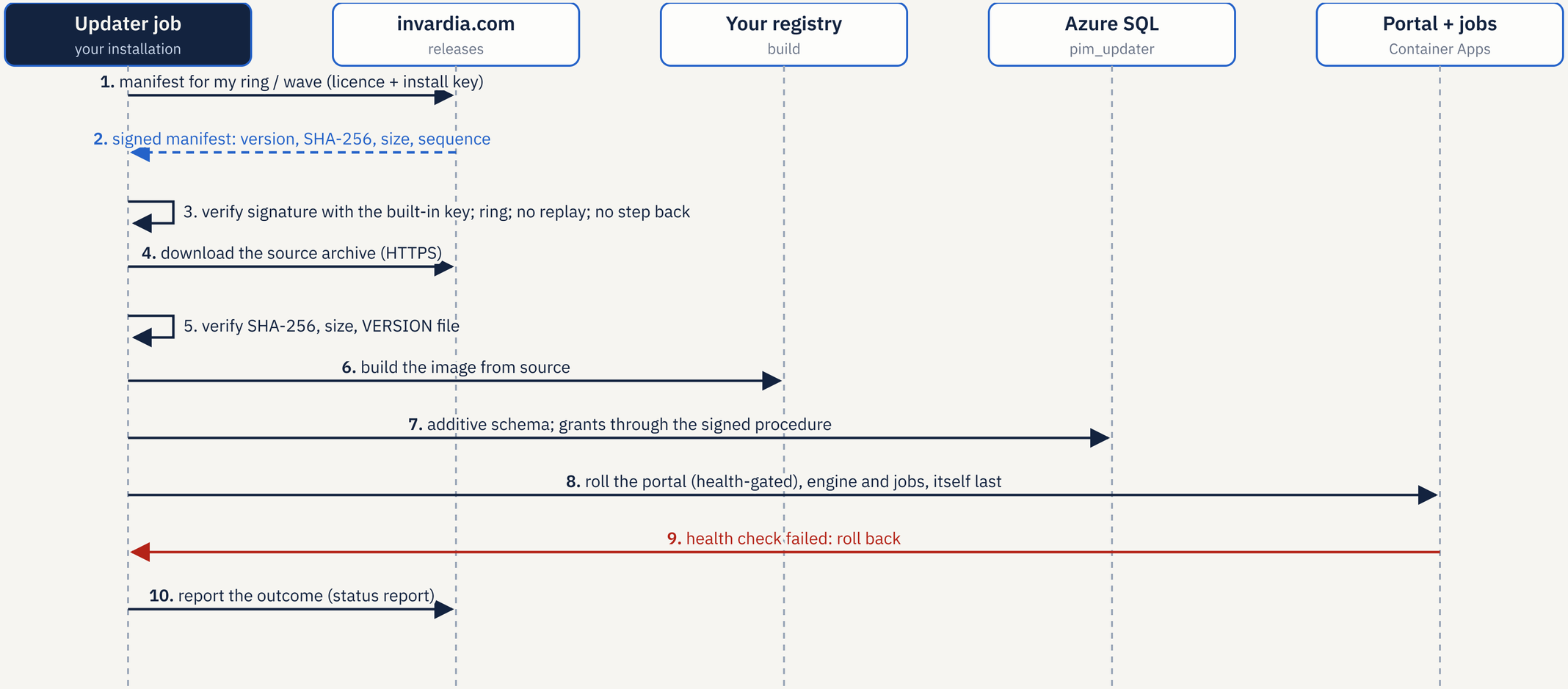
Verify
Signature, sequence, hash, version

Build locally
In your registry

No modules
REST and .NET only

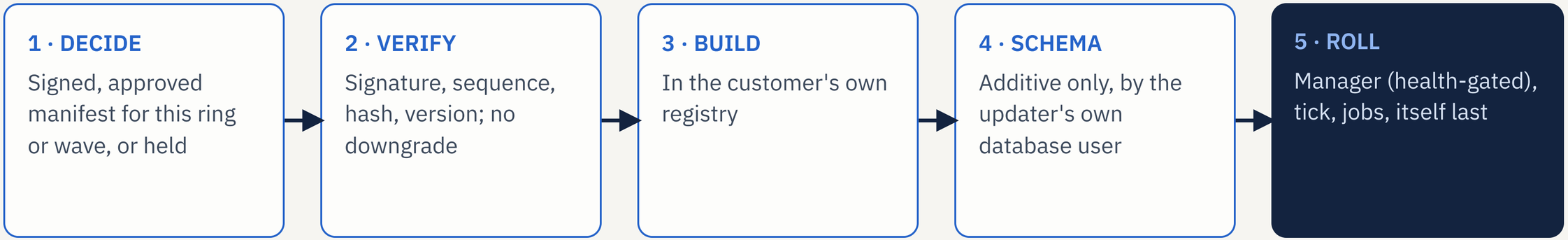
SEQUENCE · THE NIGHTLY UPDATE

Pull, verify, build, migrate, roll, check



ENGINE · THE NIGHTLY UPDATE

Each environment updates itself, and only forward



Fails safe

A failed health check rolls the Manager back; nothing half-rolled

You control

A hold switch freezes updates; customer rings get only versions the release owner approved

Reports back

Outcome per run in the ring badge; after an engine change, limits are halved for 24 h

Nothing is built until every check passes

Check	Refuses
Signature	A manifest not signed RS256 (RSA 4096) with the key built into the product; the key is never fetched at run time
Ring	A manifest for another ring
Sequence	A replay, or a step back (unless you allow a downgrade)
Archive	An archive whose SHA-256 or size differs, or not served over HTTPS
Version	A VERSION file inside the archive that differs from the manifest
Schema	A column drop or other data loss
Health	A portal that fails its health check: rolled back

SECURITY · INTEGRITY AND SUPPLY CHAIN

Three signatures protect what reaches the environment

Updates

- Invardia manifest signed (RSA-4096)
- Key built into PIM
- Sequence: no replay, no downgrade
- Archive hash + size + version
- Built in the customer's own registry
- Customer rings: approved versions only

MSP bundles

- Signed with a key that cannot leave Key Vault
- Managed tenant pins the key ID
- Expiry, anti-rollback floor, revocation
- Pull only: no inbound to the managed side

Licence

- Signed, checked offline
- Bound to the tenant of the engine's own identity
- A copied licence does not unlock Pro

CHAPTER 12

INVARDIA PIM MANAGER PRO

Network

Public with sign-in, or **private only** with private endpoints. Nothing ever connects inbound to the engine.

Public
Created closed until sign-in is verified

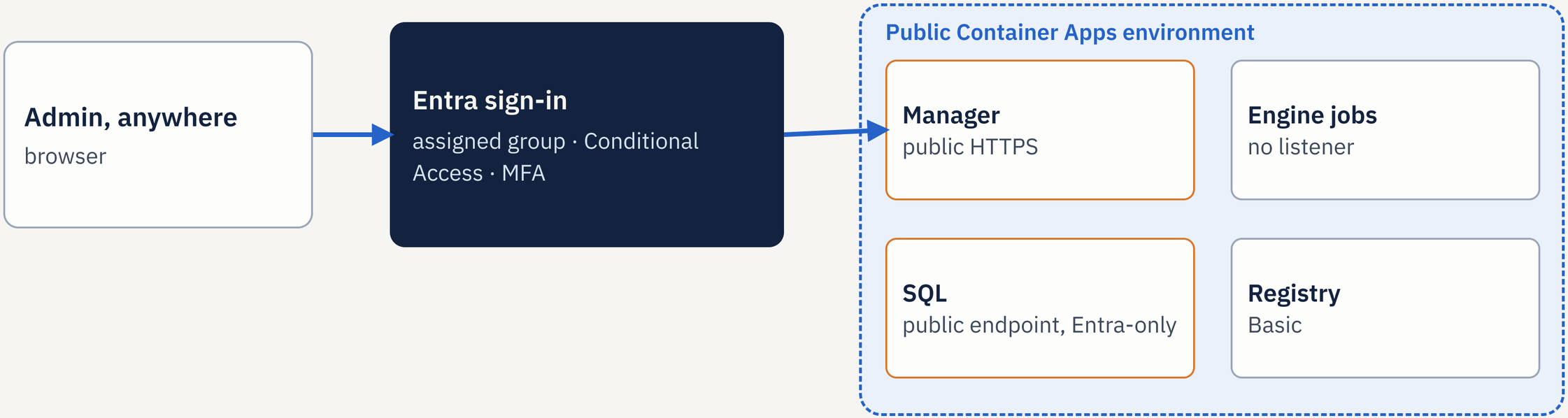
Private
Private IPs, private endpoints, public access off

Outbound only
A short egress list

Firewalling
IP allow-rules, or your hub firewall

MODE A · PUBLIC (EXTERNAL)

The GUI has a public address; identity is the wall



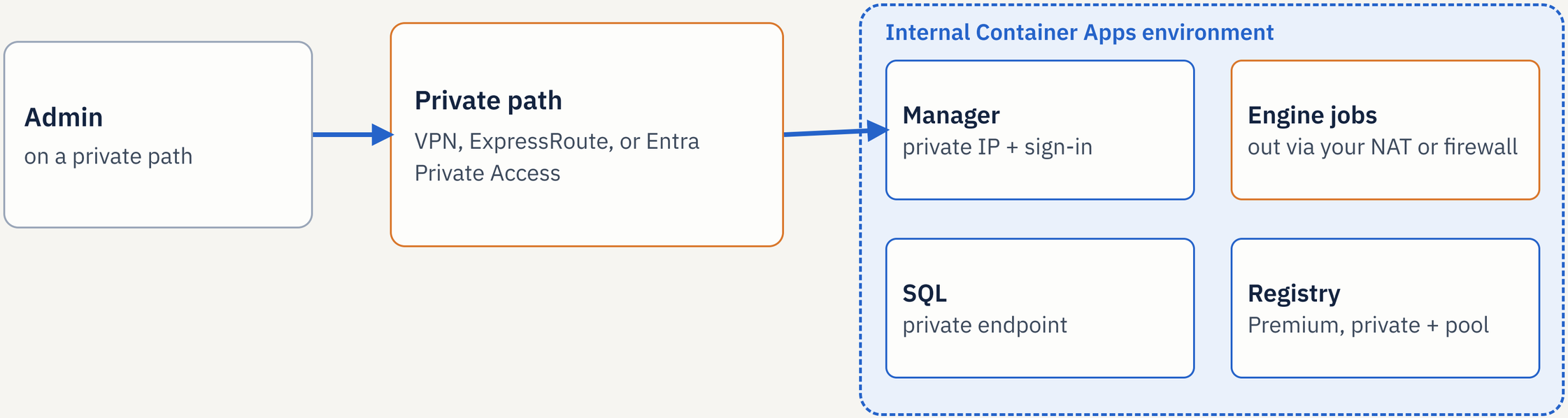
+ Works from anywhere, nothing to install, cheapest

+ Optional: IP allow-list on the Manager and SQL

– Sign-in page and names visible on the internet

MODE B · PRIVATE (INTERNAL)

No public address at all; the network is a second wall



+ Nothing listens on the internet; names resolve privately

+ Meets "no public network access" policies

– Needs a private path for admins + DNS work

MODES COMPARED

Same product, same security core; the network is the difference

	A · Public	B · Private
Manager address	Public HTTPS + Entra sign-in	Private IP + Entra sign-in
SQL	Public endpoint, Entra-only	Private endpoint, public off
Registry	Basic	Premium, private, build pool
Outbound	Default	NAT gateway (or firewall)
DNS work	None	Private zones + resolver integration
Admin access	Any internet connection	VPN / Private Access only
Identity, guards, audit, signing	Identical	Identical
Change later	Rebuild	Rebuild

NETWORK · EGRESS HOSTS

The complete outbound allow-list

Destination	Port	Used for	From
login.microsoftonline.com	443	Managed-identity and user sign-in tokens	All components
graph.microsoft.com	443	Roles, PIM, groups, users, AUs, TAP, Intune, Defender, reviews, mail	Engine (write), portal (read)
management.azure.com	443	Azure roles and policies, start the engine, roll containers	Engine, portal, updater
Azure SQL (private endpoint in private mode)	1433	Desired state, settings, audit	All components
Container registry (private endpoint in private mode)	443	Image pull, in-registry build	Platform, updater
invardia.com	443	Status reports, licence, signed updates, support tickets	Engine, updater, portal

Firewalling: in public mode add IP allow-rules on the portal; in private mode route outbound through your hub firewall or a NAT gateway and allow only this list. The SQL server's "Allow Azure services" rule is replaced by the private endpoint in private mode.

CHAPTER 13

INVARDIA PIM MANAGER PRO

Vendor access

Invardia has **no access** to your installation unless you create the Invardia Support app, at a level you choose, for as long as you choose.

Your choice
Create, lower, disable, delete

Levels
Troubleshoot, as-built, setup

Recorded
Every action, fail-closed

Never runtime
The product never needs it

VENDOR ACCESS · THE INVARDIA SUPPORT APP

No access unless you create it

Level	What it can do	Use
None	Nothing: Invardia has no access	The default
Troubleshoot	Read-only Graph; Azure Reader and Log Analytics Reader on named resource groups; read-only database access	When you ask for support
As-built	Start the one as-built job and read its PDF (deleted after 24 hours)	Configuration review
Setup	Application and role-assignment writes, Contributor + User Access Administrator on named subscriptions, SQL admin group membership (**tier-0 equivalent**)	An install or a repair, then lower it

Your approval: you create the app with Invardia's published script and choose its level; you can limit it with Conditional Access, disable or delete it at any time. **Audit:** every action is in your Entra and Azure logs and the SQL audit; Invardia records every action itself before it runs, fail-closed, and shows the record in the customer portal. Security changes to a running installation go through it with a plan, read-back and rollback record, never through an update.

CHAPTER 14

INVARDIA PIM MANAGER PRO

Support team

Three levels, one process: every event becomes a numbered ticket, is **qualified before it is routed**, and ends as an answer, a fix released through the rings, or a decision by the owner.

Level 1
Support agent, 24x7

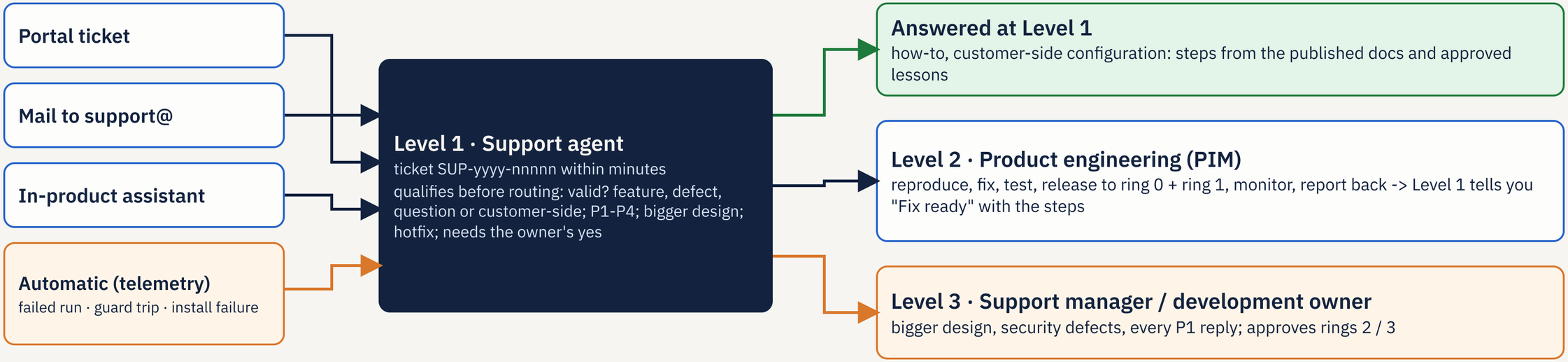
Level 2
Product engineering (PIM)

Level 3
Support manager / development owner

Gatekeeper
Every message checked before you see it

SUPPORT TEAM · FROM EVENT TO RESOLUTION

Qualified first, then routed to the level that owns it



Release rule
the product fixes and releases rings 0 + 1; the support agent proposes the candidate; the owner approves rings 2 / 3 -- never an agent

Supervisor + gatekeeper
check every message before it reaches you; every message shows its level and role

What you see, by portal and mail
Received → Acknowledged → Escalated → Fix in progress → Fix ready → Resolved

SUPPORT TEAM · PRIORITIES

How a ticket is qualified and prioritised

Priority	When	Level and handling
P1	A security defect; the product stops working; data at risk	Level 3: held for the development owner; every reply reviewed; fix through rings 0 + 1 first
P2	A feature does not work for many users; a guard holds wrongly	Level 2 hotfix; the owner's yes before it goes further; Fix ready with steps
P3	A defect with a workaround	Level 2 hotfix through rings 0 + 1; the candidate proposed for your ring
P4	A question, how-to, wish	Level 1 answers from the docs; wishes go to the backlog
Regression	Something that worked in an earlier version	Freezes promotion of that version to further rings

CHAPTER 15

INVARDIA PIM MANAGER PRO

Threats and compliance

The top risks for a privileged-access product, the controls that answer each one, and how the controls **support** DORA, NIS2 and ISO 27001.

Threat model
Nine risks, each with its controls

DORA
ICT risk and third-party controls

NIS2
Article 21 measures

ISO 27001
Annex A controls

THREATS · MODEL

The top risks and the controls that answer them

Threat	Controls	Hardening by design
Escalation through the engine identity	Managed identity; read-only portal; acts only on defined objects; guards; break-glass; journal	The most privileged directory roles are refused as engine targets; the engine is monitored as tier 0
Compromised administrator	Sign-in front, assignment required, 8 h session, signed token per request, headers, roles, staged commits, guards (halved after an engine change), undo	MFA through Conditional Access on the sign-in app; a second approver for sensitive changes; no self-approval
Update tampering	Signed manifest, built-in key, hash, size, version, ring, replay checks; built in your registry; approval before customer rings	Every update channel signature-checked, Community included; signed scripts
Vendor misuse	No access without the Support app; yours to lower, disable, remove; recorded fail-closed	Support levels scoped and Conditional-Access locked; every action in PIM's own audit trail
Audit tampering	Append-only triggers; journal in the same transaction; SQL audit outside the database; schema rights only for the updater	Any schema change is in the SQL audit, outside the database
Lateral movement	One identity, one user, one role per component; image pull without database access; portal without schema rights	Rights change only through the signed procedure
Data leaving the tenant	No directory data in status reports; anonymous mode; opt-out	Names and UPNs removed from error text before it leaves
Network exposure	Portal created closed until sign-in is verified; no ingress on jobs; private mode	Private networking as a setup option; SQL minimum TLS 1.2 set by setup
Secrets at rest	Managed identities; no sign-in secret; no storage token; registry admin off; purge protection	Install key and licence token held as secrets; salted passphrase hash

COMPLIANCE · DORA, NIS2, ISO 27001

How the controls support yours (not a certification)

Framework	Control	How PIM Manager supports it
DORA	Art. 9 Protection and prevention (access, least privilege)	Just-in-time, least-privilege delegation as desired state; staged, approved, guarded changes
DORA	Art. 10 Detection; Art. 12 Logging	Drift detection; append-only audit trail and journal; SQL audit and logs in your workspace
DORA	Art. 28-30 ICT third-party risk	Runs in your subscription; no vendor access without the Support app; every vendor action recorded; this pack
NIS2 Art. 21	(i) Access control; (j) MFA; (d) supply chain	Least-privilege JIT; MFA on activation by policy and your Conditional Access; signed, built-from-source, approved releases
ISO/IEC 27001:2022	5.15 / 5.18 / 8.2 Access control and privileged access	Dedicated admin accounts, eligible roles, PIM policies approved by a SuperAdmin
ISO/IEC 27001:2022	8.15 / 8.16 Logging and monitoring	Before / after audit trail, journal, SQL audit, guard alerts
ISO/IEC 27001:2022	8.32 Change management	Staged commits, approvals, undo, Strict guards, approved and evidenced releases
ISO/IEC 27001:2022	5.19-5.22 Supplier relationships	Vendor access only on your terms, recorded
CIS Controls v8	5.4, 6.1-6.2, 6.8, 8.2	Dedicated admin accounts; commit-based grant and revoke; RBAC definitions; audit

CHAPTER 16

INVARDIA PIM MANAGER PRO

Incidents and hardening

How incidents and vulnerabilities are handled, and the hardening you apply on day one.

Incidents
One support process, P1 first

Vulnerabilities
Fixed through the rings

Evidence
Audit, journal, SQL audit

Checklist
Ten hardening steps

Incident and vulnerability handling

Step	How
Report	A vulnerability or incident to support@invardia.com or the customer portal (portal.invardia.com); automatic tickets from telemetry for failed runs, guard trips and install failures
Triage	A security defect is P1: held for the development owner, every reply reviewed
Contain	Your controls work without Invardia: hold updates (PIM_UPDATE_HOLD), pause jobs, end Strict, disable or remove the Support app
Fix	A signed release through ring 0 and ring 1 first, checked per environment, then the approved customer release; a regression freezes further promotion
Tell	The ticket timeline in the portal and by mail; release notes on invardia.com
Evidence	Audit trail, journal, SQL audit and Invardia's record of every Support app action

HARDENING · CHECKLIST

What you can apply on day one

#	Action	Why
1	List your break-glass accounts in Settings	The engine then never touches them
2	Conditional Access (MFA, compliant device) on the sign-in app	Entra ID enforces MFA at every sign-in
3	Keep sign-in limited to named people and groups	Narrows who can get a signed token
4	Second approver on; self-approval off (Pro)	A second person for sensitive changes
5	Private mode, or IP allow-rules	Removes public exposure
6	SQL minimum TLS 1.2; alerts on the SQL audit table	Auditing is on; alerts are yours
7	User Access Administrator at the root only if Azure is managed	Least privilege
8	Support app only when needed, at troubleshoot level	Vendor access on demand
9	Monitor the engine and portal identities as tier 0	Tier-0-equivalent rights
10	PIM_UPDATE_HOLD for change windows	You decide when code changes

CHAPTER 17

INVARDIA PIM MANAGER PRO

Naming conventions

Every name PIM Manager creates follows **your** convention: seeded with defaults on a new installation, edited in Settings, never overwritten. Azure resource names are recorded per installation, so renames keep working.

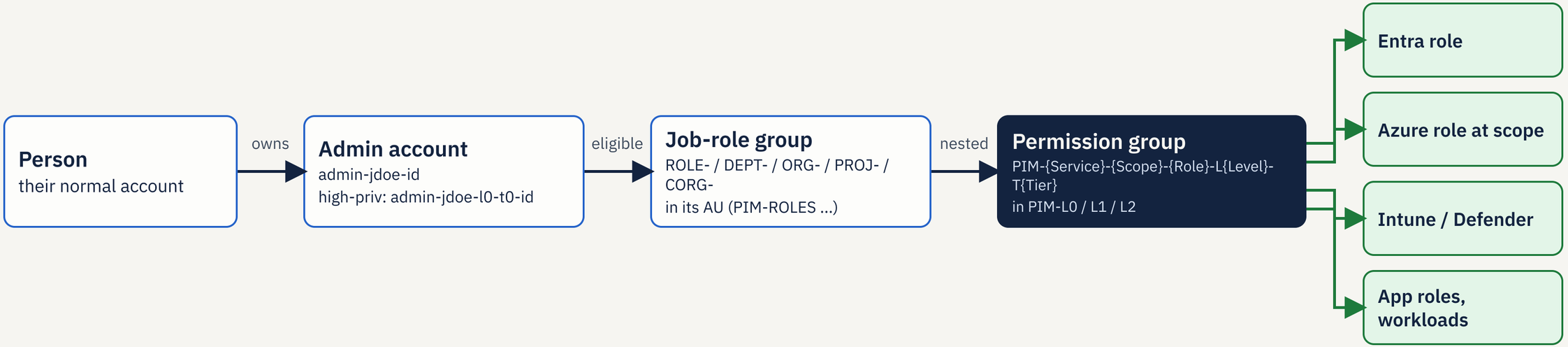
Per customer
Settings > Naming

Seed, never overwrite
Defaults only on a new store

Recorded
Install names survive renames

Hybrid AD
Fully customer-defined

People reach privilege only through nested groups, just in time



How it works

A person gets a dedicated admin account. The admin account is made eligible (PIM for Groups) in job-role groups; job-role groups are nested in permission groups; permission groups hold the roles at their scopes. Nothing is assigned to a person directly, and access is activated just in time. Tier (T0-T2) and level (L0-L5) are part of every permission group's name, so the Access map and the tier-impact report can follow every path.

NAMING · PATTERNS

Object type, pattern, example, and whether you can change it

Object	Pattern (default)	Example	Configurable
Admin account	{AdminTypePrefix}{AdminWord}-{Initial}{Platform}	admin-jdoe-id; external x-admin-jdoe-id	Yes: Settings > Naming
High-privilege admin account	{AdminWord}-{Initial}-L0-T0{Platform}	admin-jdoe-l0-t0-id	Yes
Platform suffix	entra = -ID, ad = -AD	-id	Yes
PIM group	PIM-{Role}-{Department}	PIM-HelpdeskAdmin-Finance	Yes
PIM group by AU	PIM-{Role}-AU-{AdminUnit}	PIM-UserAdmin-AU-Nordics	Yes
Permission group / Azure RG	PIM-{Workload}-{Scope}-{Permission}-L{Level}-T{Tier}-{Plane}-{Platform}	PIM-Azure-Finance-Reader-L3-T1-WDP-ID	Yes
Job-role group tags	ROLE- ORG- DEPT- PROJ- CORG- + short name	DEPT-Finance	Yes
Administrative units	PIM-ROLES, PIM-DEPARTMENTS ...; PIM-L0 / L1 / L2; PIM-Groups-{AdminUnit}	PIM-Groups-Finance	Yes
Hybrid AD names and OUs	Customer-defined	-	Yes: Settings > Hybrid Active Directory

What must stay fixed: once an estate is named, the admin-account pattern identifies the accounts PIM Manager manages; changing it on a populated estate makes existing accounts look unmanaged, so the engine refuses to run on a missing convention rather than guess.

NAMING · AZURE RESOURCES

Resource names an installation creates, recorded per installation

Resource	Default name	Recorded as	Configurable
Resource group	rg-pim	-	At install
Portal (Container App)	ca-pim-manager	web	At install; recorded
Engine job	ca-pim-tick	engine	At install; recorded
Updater job	ca-pim-update	updater	At install; recorded
As-built job + identity	caj-pim-asbuilt, id-pim-asbuilt	asbuilt, asbuiltIdentity	At install; recorded
SQL admin group	grp-pim-sql-admins	sqlAdminGroup	At install; recorded
MSP jobs	ca-pim-publish, ca-pim-downlink-<scenario>	publish, downlink	At install; recorded
SQL server, registry, identity, workspace	sql-pim-<suffix>, acrpim<suffix>, id-pim-<suffix>, log-pim-<suffix>	-	Suffix at install
Environment handle (support)	<company>-<kind>	company-prod, company-test	At onboarding

Each installation records the names of its own resources once; every component reads them from that record, so a renamed component keeps working without code changes.

CHAPTER 18

INVARDIA PIM MANAGER PRO

On-premises Active Directory (optional)

A **Tier-0** worker next to your domain controllers: no public IP, no inbound, run-command only, two group managed service accounts.

Tier 0
Treated as a domain controller

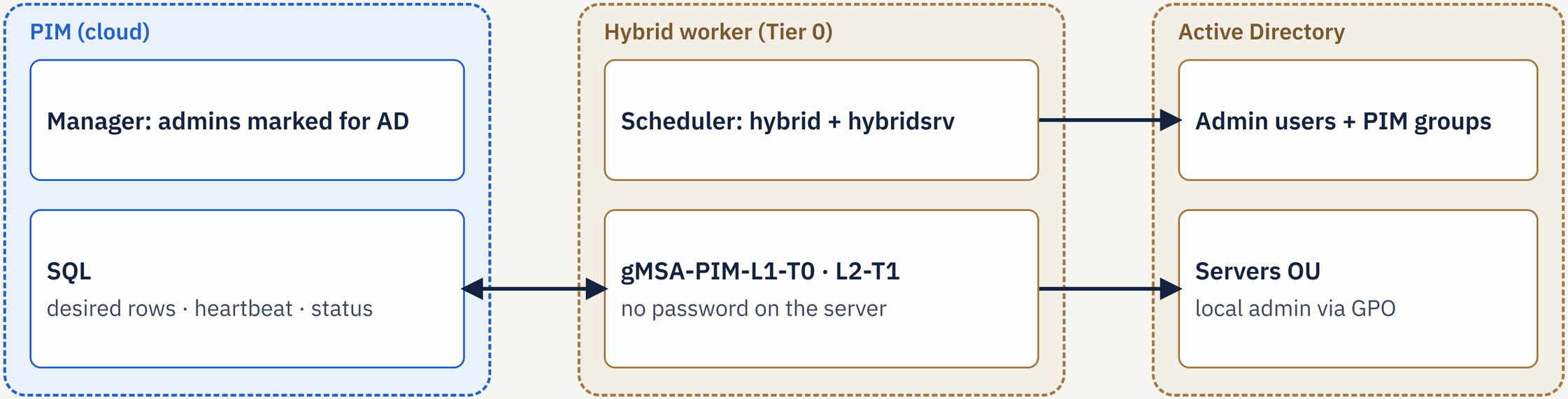
No inbound
No public IP

gMSAs
One per tier

Never deletes
Removes members only

HYBRID AD · THE FLOW

The worker reads the same SQL and writes AD with gMSAs



Apply + groups: create and update admin accounts and AD groups

Sync, continuous: active PIM elevations become AD group members, with a TTL

Status back: heartbeat per job, Hybrid AD page, watchdog

The worker can write Tier-0 groups, so it is treated as Tier 0

Network

- No public IP, ever
- All inbound denied, peers included
- Outbound through a NAT gateway only
- Managed by Azure run-command, never RDP

Identity

- One gMSA per tier; passwords held by the DCs
- Rights granted to a permission group, never the account
- Only listed computers may fetch the password
- Never a member of a just-in-time group

Identity	May
gMSA-PIM-L1-T0	Create users below the admin OU; create groups and write members in the PIM groups OU; optionally, members of protected groups
gMSA-PIM-L2-T1	Local administrator on the servers OU through a GPO (add only)