TECHNICAL ARCHITECTURE AND SECURITY

INVARDIA
SecurityInsight PRO

SecurityInsight

Why a CISO needs it, what it delivers, and exactly how it is built, signs in, stores data, updates and is supported - for architects and security teams

Value | Architecture | Identities and SQL | Network and data | Release and guards | Support

WHY · THE CISO'S PROBLEM

You have more security data than ever - and still cannot say what to fix first

TOO MUCH, SORTED WRONG

Thousands of findings, sorted by severity

Defender, Entra and Azure each list their own recommendations. A "high" on a test laptop ranks the same as a "high" on a domain controller.

ATTACKERS THINK IN PATHS

An attacker sorts by opportunity

They look for the weak asset that leads to Tier 0: an identity with a path to privileged roles, a server with an exposed public IP, an unpatched management host.

THE BOARD ASKS

"Are we getting safer?"

Regulators (NIS2, DORA) and the board want evidence: an asset inventory, a risk classification, who accepted which risk, and a trend - not screenshots.

THE GAP

The data to answer these questions already sits in your Microsoft tenant. What is missing is a model that connects it: which asset matters, what is wrong with it, how an attacker would use it, and who owns the fix.

WHAT · THE VALUE

One ranked list of what to fix first - scored by business impact, in your own Azure

- **Risk, not noise.** Every finding is scored by consequence, probability (the Tier 0-3 criticality of the asset) and risk factors. The small set of fixes that most reduces lateral movement and tenant takeover comes first.
- **Tiers from real permissions.** Identities are tiered from the permissions they actually hold, including nested groups, and every tier shows the signal that set it.
- **Your crown jewels weigh more.** Connect a CMDB and the same finding ranks higher on a payment server than on a sandbox.
- **Governed, auditable decisions.** Accepted risk is requested, approved, time-limited and reviewed; every change has a reason and an audit row.
- **Closed loop.** A vulnerability closes only after complete runs prove it gone, and reopens if it returns.

126
Risk Analysis reports (63 summary + 63 detailed)

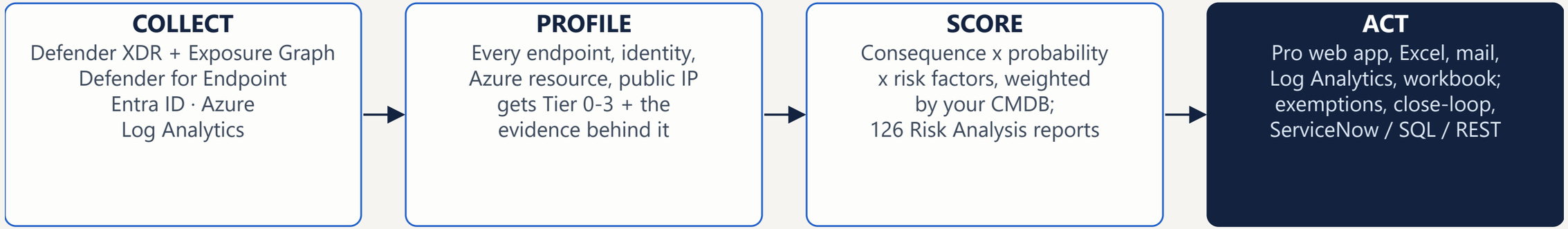
0
agents installed; collection is read-only

540+
detection rules for server roles, applications and appliances

100%
in your tenant and subscription; Invardia never hosts your data

Never green on missing data. Every run proves its own completeness; a short or partial result is shown as such, never as a clean bill of health.

Collect, profile, score, act - every day, read-only, on your own identities



Read-only managed identity · results stay in your Log Analytics workspace and your Azure SQL database · nothing connects inbound

Pro is the product in focus here: the management web app, the Pro database, correlation across sources, governance, connectors and automatic signed updates.

Community runs the same profilers and Risk Analysis as scripts you run yourself, with no web app; settings in files, results as Excel, mail and Log Analytics tables.

Hosting: Azure Container Apps jobs in your subscription. Pro is container-only.

Security at a glance

Where it runs
Your tenant, your subscription, your region. Invardia hosts nothing of yours.

Identities
Managed identities for every component. No sign-in secret (federated credential, PKCE); sessions end after 8 hours.

Least privilege
Collection is read-only. One write-capable job (tagging), opt-in, own identity. Per-component SQL roles.

SQL
Entra-only, no SQL passwords. Admin group holds only the Support app; people just-in-time. Statement auditing to your workspace, 90 days.

Network
Outbound only; nothing connects in. Public behind Entra, or fully private: setup builds the VNet, private endpoints and DNS.

Data
TLS 1.2+ in transit; encryption at rest with Microsoft-managed or your own keys. Retention you control.

Supply chain
Signed releases with an SBOM, SHA-256 verified, built from source in your own registry. No PowerShell modules.

Release safety
Rings 0-3, approval before customer rings, waves, Strict guards for 24 h after an engine change, rollback.

Damage brakes
Guards hold mass changes for a SuperAdmin; a failed or partial run never looks green.

Audit
Append-only audit trail with before/after; run records; SQL audit; every support statement recorded.

Support access
Only through an app you create, own, scope and can switch off. Every action audited.

Compliance
Evidence that supports NIS2, DORA and ISO 27001 controls. Not a certification.

CONTENTS

What this deck covers

1 **Architecture and components**
Overview, trust boundaries, jobs and pipeline

2 **Data sources, flow and storage**
What is read, where it is stored, retention

3 **Identities and permissions**
Every identity, every permission, sign-in and access flows

4 **The SQL access design**
Per-component roles, admin group, signed grants, auditing

5 **Network, encryption and secrets**
Public and private, every egress host, keys

6 **Authorization and change control**
Roles, separation of duties, approvals

7 **Logging, audit and telemetry**
What is logged, where, how long; what reaches Invardia

8 **Release process and supply chain**
Signed releases, rings, waves, approval, rollback

9 **Guards and guard modes**
Damage brakes, Normal and Strict

10 **Data health monitor and health model**
Completeness proof, fail-closed hand-off, health uplink

11 **Support model and flow**
Support app, triage, fixes, escalation, SLAs

12 **Naming, threats and controls**
Naming conventions, threat model, DORA / NIS2 / ISO 27001, incident handling

CHAPTER 1 · SUMMARY

Architecture: everything runs in your subscription, on your identities

- **Engines** run as Azure Container Apps jobs in a daily pipeline: collect, correlate, analyse, deliver.
- **Pro web app** and **MCP server** are Container Apps with their own managed identities and Entra sign-in.
- **Pro database** (Azure SQL) holds settings, decisions, audit trail and copies of findings; **Log Analytics** holds the collected profiles and results.
- **Updater** pulls signed releases from Invardia and builds the images from source in your own registry.
- **Nothing connects inbound.** Every connection to Invardia is made by your install, outbound, and is listed in chapter 5.

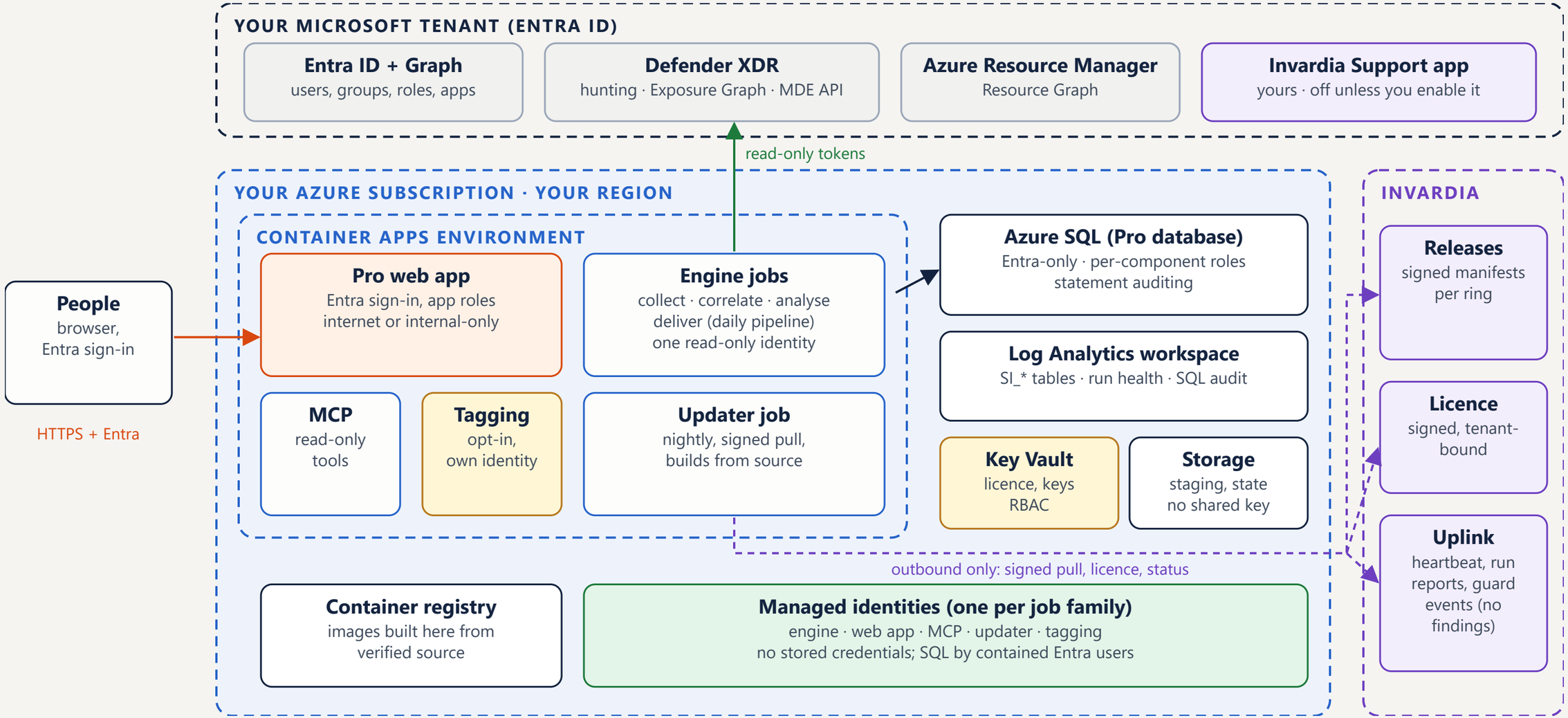
TRUST BOUNDARIES

1 · Your Microsoft tenant: the data source.
2 · Your Azure subscription: where SecurityInsight runs and stores everything.
3 · Invardia: signs releases and licences, receives status reports. Holds none of your security data.

SUPPORT

Invardia reaches an install only through the Invardia Support app you create and own (chapter 11).

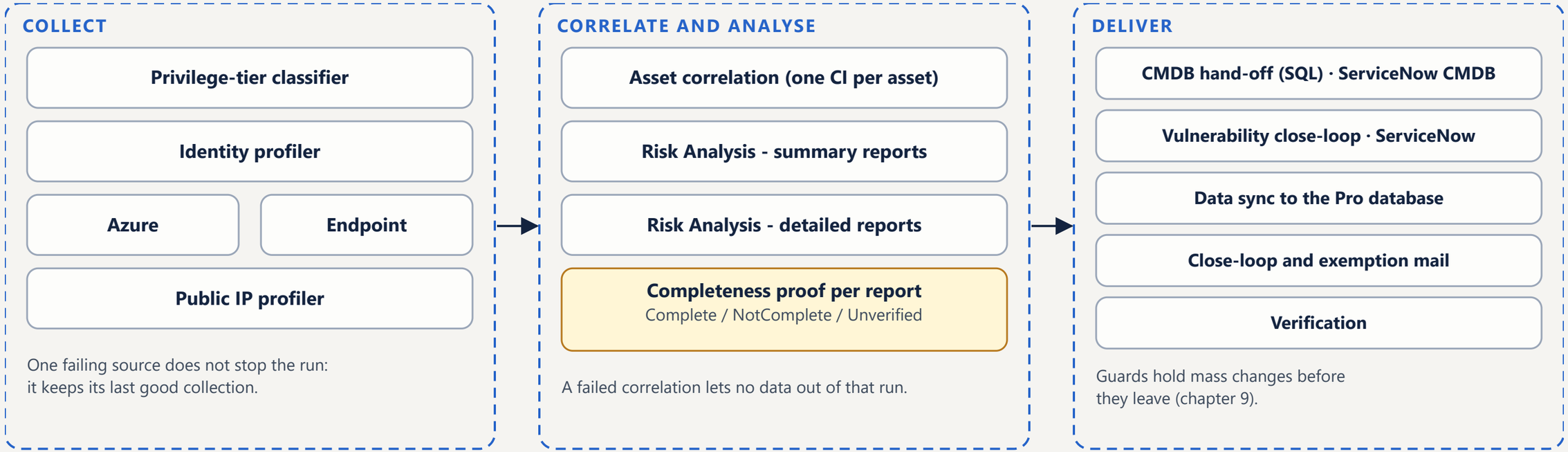
Components, trust boundaries and network paths



What each component does, how it runs and who it runs as

Component	Runs as	Identity	What it does
Engine jobs (collection, correlation, Risk Analysis, delivery)	Container Apps jobs, started by a flow job (every 5 min, pipeline nightly)	Engine managed identity (shared by the engine jobs)	Read Defender XDR, Entra, Azure, Log Analytics; build profiles, the cross-source CI model and 126 reports; deliver to SQL, ServiceNow, mail
Asset tagging	Own Container Apps job; report-only by default	Own tagging identity (opt-in write grants)	Writes Defender machine tags, Azure tags and Entra custom security attributes - only if you switch it on
Pro web app	Container App (ASP.NET Core)	Web app managed identity	Findings, collection views, tier board, exemptions, settings, jobs, guards, audit trail, chat
MCP server	Separate Container App	MCP managed identity	Read-only tools for MCP clients presenting an Entra token with an app role
Updater	Container Apps job, nightly	Updater managed identity	Licence, signed releases and template packs; builds images in your registry; rolls, verifies, rolls back
Azure SQL	Pro database + identity-ledger database; Standard tier, never Basic	Entra-only; contained users per component	Settings + history, decisions, audit trail, vulnerability state, CI model, copies of findings
Log Analytics	Your workspace	-	SI_* tables: profiles, CI model, Risk Analysis, trend, run health; SQL audit events
Key Vault · Storage · Registry	Your resources (RBAC)	-	Licence and keys · staging, run state, logs, work queues · images built from source

Collect, correlate, deliver - one pipeline, in a fixed order



Trigger: a flow job every 5 minutes starts the pipeline at the install's chosen time (default 22:00 local).

Updater: its own timer at night; never a pipeline step; waits while an engine run is active.

Separate jobs: asset tagging, template releases, configuration backup, weekly schema discovery.

Stale source: after 7 days with no good collection, delivery from that source is held.

CHAPTER 2 · SUMMARY

Data: read from Microsoft APIs, stored only in your own resources

- **Sources:** Defender XDR advanced hunting and Exposure Graph, the Defender for Endpoint API, Entra ID (Graph), Azure Resource Manager and Resource Graph, Log Analytics; optionally Shodan for your public IPs and your ServiceNow.
- **Stores:** Log Analytics tables (SI_*), the Pro SQL database, storage for staging and run state. All in your subscription and region.
- **Classification:** security-configuration and vulnerability data about your estate, plus names of users and devices. Confidential.
- **Retention** is set per store, and you control it. Nothing is hard-deleted in the decision history: exemptions are ended, assets are retired.

WHAT NEVER LEAVES YOUR ENVIRONMENT

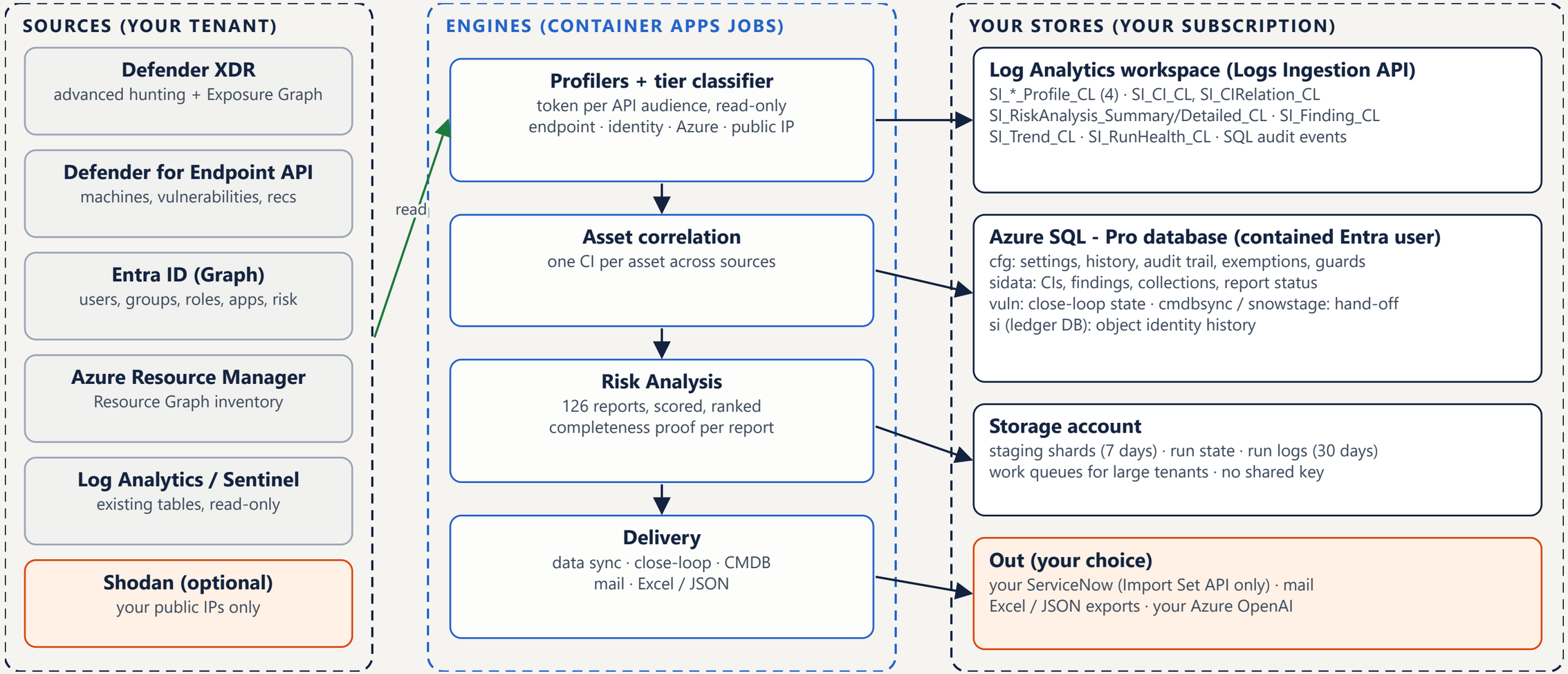
Findings, asset data, profiles and the estate inventory. Invardia receives status reports only (chapter 7).

OPTIONAL OUTBOUND DATA

Your public IPs to Shodan (only with your Shodan key). Chat and AI summary go to **your own** Azure OpenAI resource.

DATA · FLOW

From the APIs to your stores: what is read, by whom, and where it lands



Every store: what it holds, how it is protected, how long it is kept

Store	Holds	Location	Encryption	Retention
Log Analytics workspace	Profiles, CI model, Risk Analysis, trend, run health, SQL audit	Your subscription and region	TLS 1.2+ in transit; encryption at rest, Microsoft-managed or your own keys	90 days when setup creates it; an existing workspace keeps its own. Trend: 365 days recommended
Pro SQL: sidata	Copies of collections, findings, daily figures, report status	Your subscription	TLS 1.2 minimum; Transparent Data Encryption (service-managed or your key)	90 days (setting), pruned by the data sync
Pro SQL: cfg audit trail	Who, when, why, before / after	Your subscription	As above; append-only for every runtime identity	Kept; optional retention of 13 months or more, by an audited step
Pro SQL: vuln, ledger, settings history, exemptions	Decisions and lifecycle evidence	Your subscription	As above	Kept until you delete them
SQL security audit	Every statement by every identity	Your workspace	As Log Analytics	90 days
Storage	Staging shards, run state, run logs, queues	Your subscription	TLS 1.2, HTTPS only, no shared key, no public blob access; encrypted at rest (Microsoft-managed or your key)	Staging 7 days; run logs 30 days
Key Vault	Licence, install key, connector secrets, optional API keys	Your subscription	RBAC; purge protection on	Until removed
Exports, backups	Excel / JSON, configuration backups, SQL backups	Your subscription	As the store	Exports: not pruned · config backups 90 days · SQL: Azure defaults

Encryption at rest uses Microsoft-managed keys by default; customer-managed keys are an option for SQL, storage and Log Analytics. Data is classified Confidential: it describes your security posture.

CHAPTER 3 · SUMMARY

Identities: one per job family, no stored credentials, every right explained

- **Five managed identities:** engine (collection), asset tagging, web app, MCP server, updater. Each holds only its own rights.
- **Collection is read-only** on your tenant: Graph read permissions, Defender read permissions, Azure Reader at the scope you choose.
- **One write-capable job** - asset tagging - with its own identity; its three write grants are opt-in and it stays report-only until you switch it on.
- **Users sign in** with Entra ID to a single-tenant app registration whose only credential is a federated credential of the web app's managed identity: no client secret, no certificate. Your MFA and Conditional Access apply.
- **Setup checks itself:** each run reads its own token and rights and warns about anything missing or extra.

NO SECRETS TO STEAL

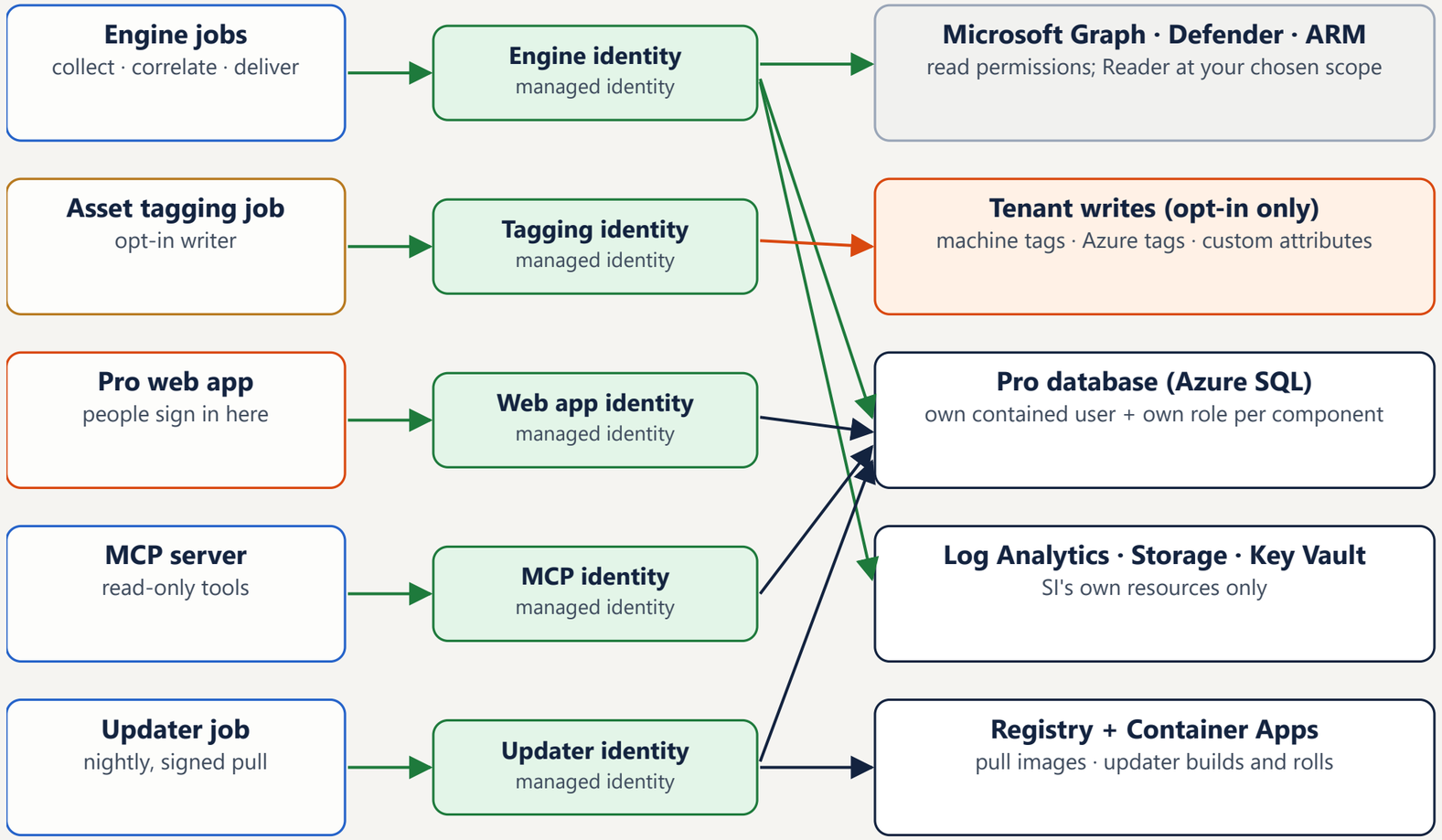
No component stores a password, client secret or certificate. Tokens are issued per run by Entra to the managed identity, per API audience, and expire on their own.

INVARDIA SUPPORT APP

A separate app you create and own. It is never the identity the product runs as. Chapter 11.

IDENTITIES · MAP

Who runs as what, and what each identity can reach



SIGN-IN (PEOPLE)

App registration "SecurityInsight"
single-tenant · assignment required
credential: federated (web app MI)
no secret, no certificate

App roles
SuperAdmin · Admin · Reader
Wallboard · Department
+ in-app Delegated scopes
checked on the server

INVARDIA SUPPORT APP

Created and owned by you
Troubleshoot: read-only
Setup: repair rights on SI resources
standing member of the SQL admin group
every action recorded

Engine identity: read-only permissions on your tenant

Permission	API	Scope	Why it is needed	Least privilege
User, Group, Directory, Device, Application, AuditLog, Policy .Read.All	Graph (application)	Tenant	Identity profiles: users, groups, devices, apps, sign-in and policy context	**READ** only
RoleManagement.Read.All / .Read.Directory, RoleManagementPolicy, RoleEligibilitySchedule, RoleAssignmentSchedule .Read.Directory, PrivilegedAccess.Read.AzureADGroup	Graph (application)	Tenant	Privilege tiers from real role assignments, eligible (PIM) and group-based	**READ** only
IdentityRiskyUser, IdentityRiskEvent, IdentityRiskyServicePrincipal .Read.All; UserAuthenticationMethod.Read.All	Graph (application)	Tenant	Risk factors: risky identities, MFA methods	**READ** only
ThreatHunting.Read.All, SecurityAlert.Read.All, AccessReview.Read.All, DeviceManagementManagedDevices / Configuration .Read.All	Graph (application)	Tenant	Advanced hunting + Exposure Graph; alerts; access reviews; Intune context	**READ** only
AdvancedHunting.Read.All	Microsoft Threat Protection	Tenant	Defender XDR advanced hunting	**READ** only
Machine.Read.All, Vulnerability.Read.All, SecurityRecommendation.Read.All	Defender for Endpoint	Tenant	Devices, vulnerabilities, recommendations. Never Machine.ReadWrite	**READ** only

22 Microsoft Graph application permissions, all read. No directory role, no Graph write permission. Taken from the product's locked permission catalog.

Engine identity: Azure rights - Reader where you choose, write only on SecurityInsight's own resources

Permission	API	Scope	Why it is needed	Least privilege
Reader	Azure RBAC	One subscription (default), a management group or the root - you choose	Azure inventory via Resource Graph	READ only
Log Analytics Reader	Azure RBAC	SI workspace (+ your Defender / Sentinel workspace)	Risk Analysis and correlation queries	READ only
Monitoring Metrics Publisher; Monitoring / Log Analytics Contributor	Azure RBAC	SI resource group / SI workspace	Ingest into SI's own tables; create its own collection rules and tables	WRITE SI's own
Storage Blob / Queue / Table Data Contributor; AcrPull; Key Vault Secrets User	Azure RBAC	SI storage / registry / vault	Run state, staging, queues; pull image; read licence and keys	WRITE SI's own
Container Apps Jobs Contributor	Azure RBAC	SI resource group	Only the flow job: starts the other pipeline jobs	WRITE SI jobs
si_engine role	SQL	Pro database	Engine data; no DDL, no security rights (chapter 4)	Per-component role

The collection plan holds no write grant on your security systems; a test enforces it, and setup refuses to give the tagging rights to this identity. No Owner, no User Access Administrator.

Tagging identity: the only writer into your tenant - opt-in, three grants

Permission	API	Scope	Why it is needed	Least privilege
ThreatHunting.Read.All, User.Read.All, Application.Read.All	Graph (application)	Tenant	Evaluate the tagging rules	READ only
CustomSecAttributeAssignment.ReadWrite.All	Graph (application)	Tenant	Set custom security attributes on users and service principals	WRITE opt-in
Machine.ReadWrite.All	Defender for Endpoint	Tenant	Set Defender machine tags	WRITE opt-in
Tag Contributor	Azure RBAC	Tagging scope (default: your Reader scope)	Set Azure resource tags. Tags only - no other resource change	WRITE opt-in
Reader; Log Analytics Reader	Azure RBAC	Reader scope; SI workspace	Resource Graph rules; read profiles	READ only
Monitoring Metrics Publisher; Storage Blob Data Contributor; AcrPull; Key Vault Secrets User	Azure RBAC	SI resources	Write tag-activity rows and run log; pull image; licence	WRITE SI's own
si_tagging role	SQL	Pro database	Read its settings	READ only

Off by default. Setup grants the read rights and only checks the write rights. Writing needs both the grants and the setting that switches tagging from report-only to apply.

Separate from collection. The engine identity can never hold these grants; setup refuses it.

Every write is recorded in the tag-activity table with before and after values.

Web app identity: what it runs as, how people sign in, what it can and cannot do

- **Runs as** its own user-assigned managed identity in a Container App.
- **Sign-in:** OpenID Connect authorization code flow with PKCE, to a single-tenant app registration. **Assignment required**: only people and groups given an app role get in. The registration's only credential is a **federated credential** of the web app's managed identity.
- **Your MFA and Conditional Access** apply at every sign-in. Every page needs a sign-in and a role, checked on the server; only a health probe is anonymous.
- **Session:** ends after a fixed 8 hours. **Hardening:** strict CSP, HSTS, no framing, anti-forgery; Secure, HttpOnly cookies.

Cannot: read your tenant beyond names and group membership · write to your directory · start a collection with tenant rights (the jobs run as the engine identity) · change the database schema · read or write Key Vault secrets.

Permission	Scope	Why
User.ReadBasic.All, GroupMember.Read.All, Application.Read.All (Graph app)	Tenant	Names, people picker, group overage, who holds roles
Log Analytics Reader	SI workspace	Dashboards, job logs
Container Apps Jobs Operator	The named SI jobs	Jobs page: start a job; the job runs as the engine identity
AcrPull	Registry	Pull its image
Cognitive Services OpenAI User	Your Azure OpenAI (optional)	Chat, by Entra token, no API key
Key Vault Secrets User	Magic-link key (optional)	Sign magic links (off by default)
si_web role (SQL)	Pro database	Its own pages' reads and staged writes; audit rows
Exchange.Manage (delegated, of the signed-in admin)	Admin's own rights	"Set up mail" only

MCP server reads; updater installs - and neither holds tenant permissions

MCP server identity

Permission	Scope	Why
AcrPull	Registry	Pull its image
Log Analytics Reader	SI workspace	Read-only tools
si_mcp role (SQL): read only what it serves	Pro database	Read-only tools; never the web sign-in key ring
No Graph permission	-	-

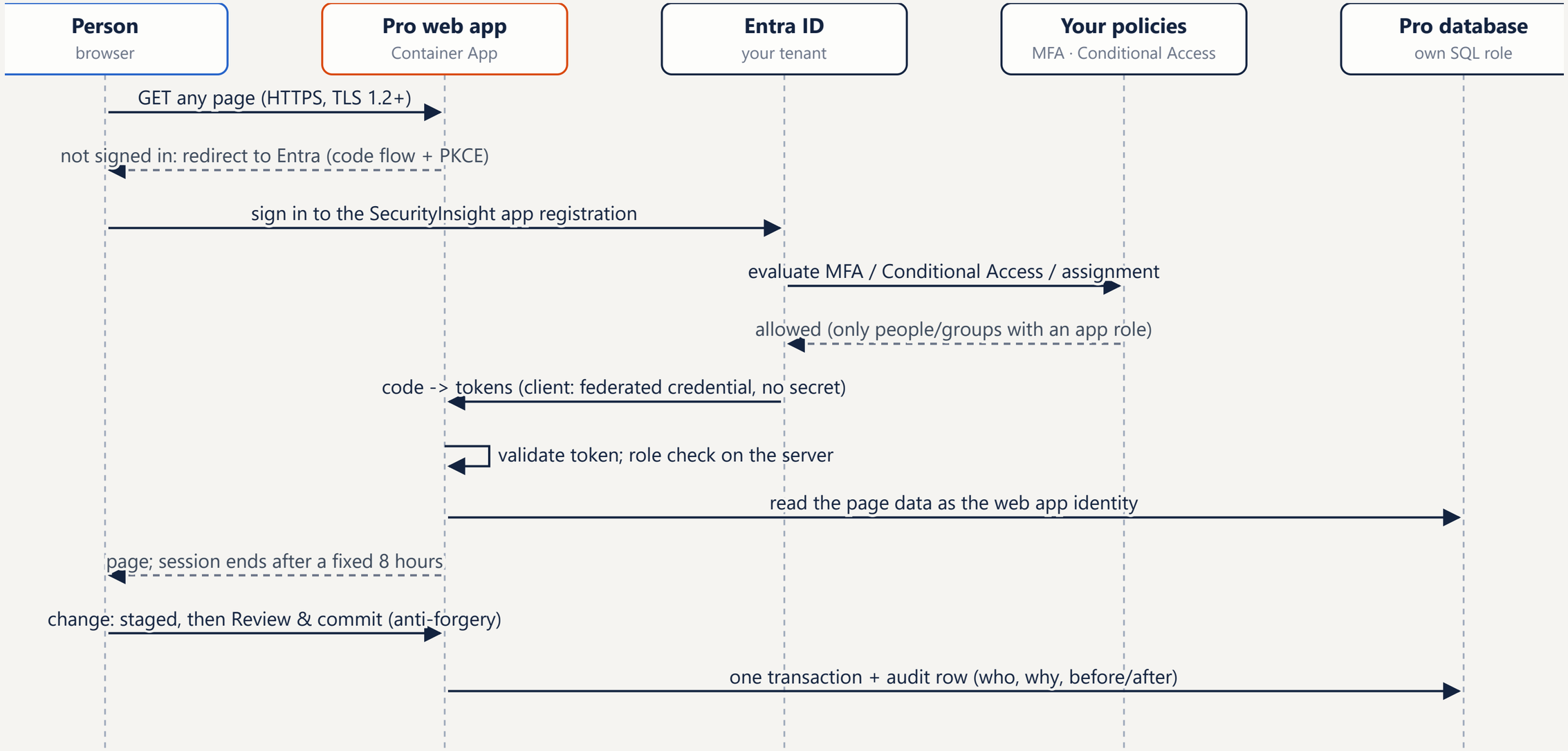
Clients present an Entra bearer token for the app with a Reader, Admin or SuperAdmin role. Settings history only for Admin and SuperAdmin. Nothing it offers can write.

Updater identity

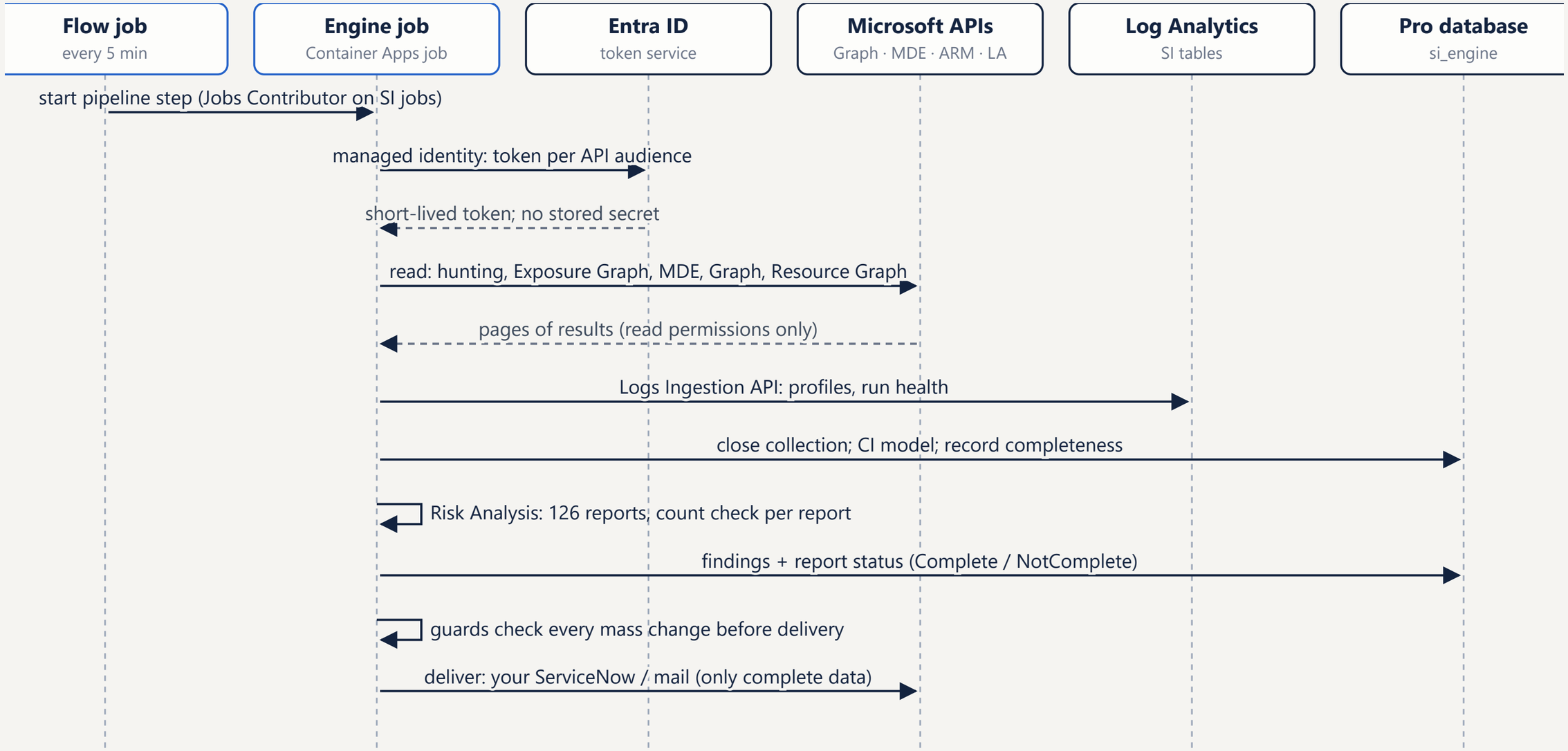
Permission	Scope	Why
Contributor	SI resource group + registry	Build images, update jobs and apps
Key Vault Secrets Officer + Crypto Officer	Install's vault	Licence, install key, update state; secret hand-over key (never exported)
AcrPull; Key Vault Secrets User	Registry; vault	Pull its image; resolve references
DDL-only database user + EXECUTE on the signed grant procedure	Pro database	Apply schema; apply the release's grant list - and nothing else
No Graph permission	-	-

The updater changes no security by hand: database rights change only through the certificate-signed procedure (chapter 4).

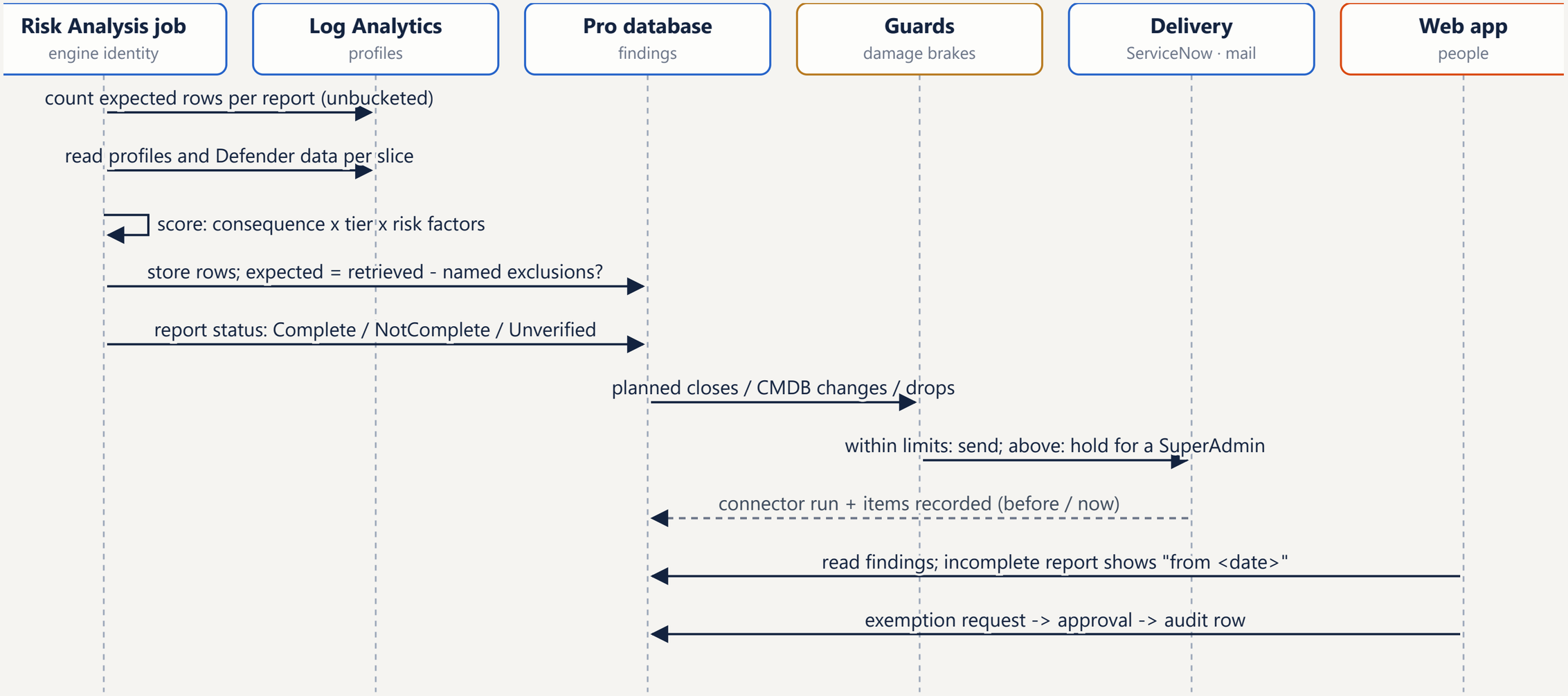
Sequence: a person signs in to the Pro web app



Sequence: the nightly collection - identity and rights at each step



Sequence: from stored data to a ranked report and a closed loop



CHAPTER 4 · SUMMARY

The SQL access design: no passwords, one role per component, rights that change only through a signed procedure

- **Entra-only sign-in.** No SQL logins, no SQL passwords. TLS 1.2 minimum.
- **One contained database user and one custom role per component** - engine, web app, MCP server, tagging, updater. Runtime identities hold no DDL and no security rights.
- **The SQL admin group is the Invardia Support app only**, standing. People get in just-in-time through PIM for Groups, with approval and a time limit. The group is role-assignable, so only a Privileged Role Administrator can change it.
- **The updater is DDL-only.** Rights change only through one certificate-signed grant procedure it may only EXECUTE.
- **Built to run five years without contact:** the certificate is valid to 2099, its private key is removed, the rules are frozen; no routine operation needs the admin group.
- **Everything is audited:** SQL auditing of every statement to your workspace for 90 days, plus a grant-list test per component.

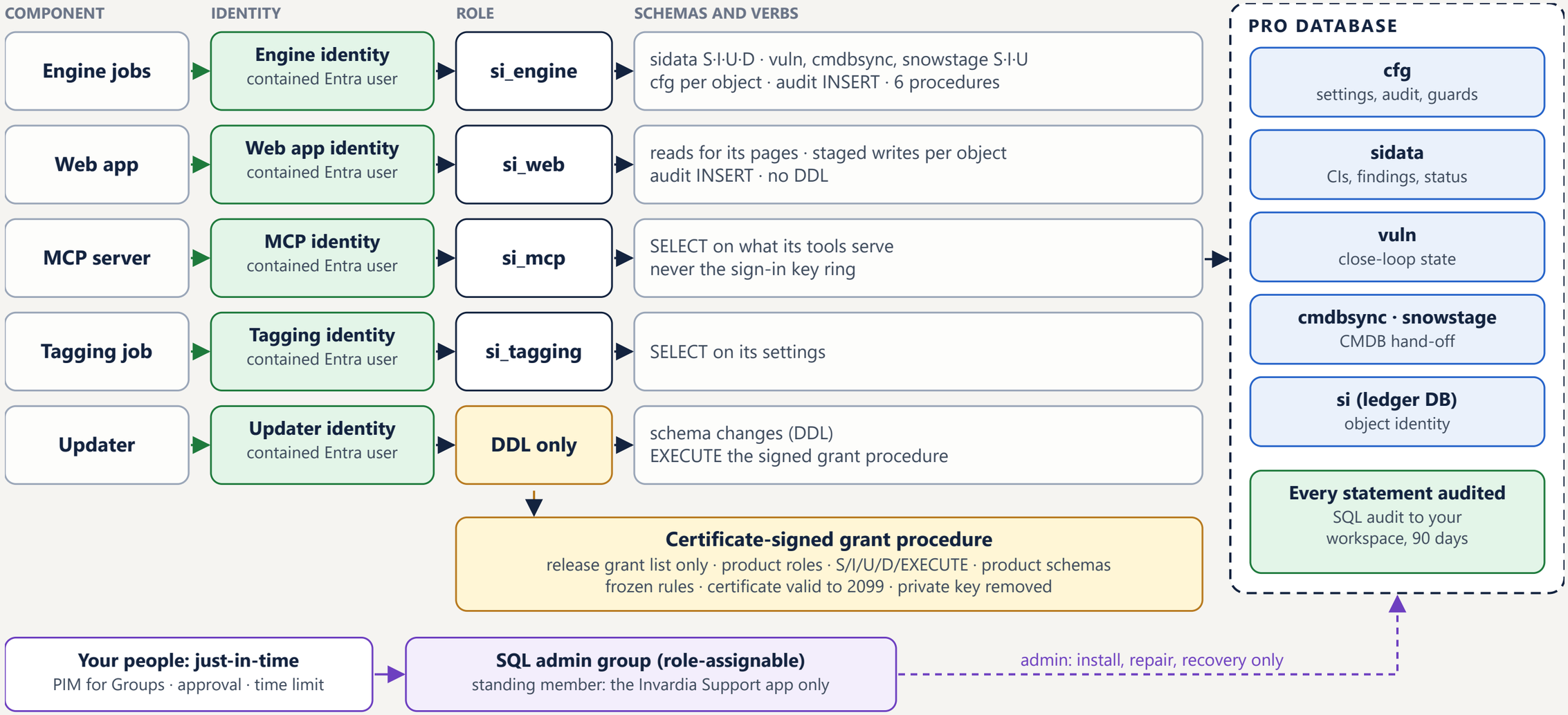
WHY IT MATTERS

A compromised runtime identity cannot change the schema, grant itself rights, or rewrite the audit trail. No person holds standing admin rights on your data. Invardia's support access is through an app you own, scope and can switch off - every action recorded.

COLLECT AND STORE KEPT APART

A separate loader role stores what collectors read, so a collector cannot alter stored decisions.

Who reaches the database, with which rights



Database users, roles and rights

Identity	Database user / role	Rights	Never
Engine (collection, correlation, Risk Analysis, delivery)	Contained Entra user; role si_engine	sidata: SELECT/INSERT/UPDATE/DELETE · vuln, cmdbsync, snowstage: SELECT/INSERT/UPDATE · cfg: ~23 objects SELECT, ~36 objects only the verbs the code uses · audit tables INSERT only · EXECUTE on 6 retention / backup / CMDB procedures · DELETE on 4 staging tables	DDL, security rights, fixed roles · web sign-in key ring, pending secrets · writes to access, settings, exemptions, overrides · UPDATE/DELETE of the audit trail
Pro web app	Contained user; role si_web	Reads for its pages · per-object INSERT/UPDATE/DELETE/EXECUTE for staged changes (tested against every write statement) · audit INSERT + SELECT	DDL · install state · schema or rights changes
MCP server	Contained user; role si_mcp	SELECT on the objects its read-only tools serve	Any write · the web sign-in key ring (explicit DENY)
Asset tagging	Contained user; role si_tagging	SELECT on its settings	Any write
Template job	Contained user; role si_templates	Template tables	Everything else
Updater	Contained user; DDL only + role si_updater	Schema changes · EXECUTE on the signed grant procedure (its only security right)	Granting rights by hand · membership of the admin group
Invardia Support app	Member of the SQL admin group	Install, repair, recovery - every statement through the audited support route	Use outside an open, recorded support session
Your administrators	Just-in-time member of the admin group (PIM for Groups)	As admin, for the approved time window	Standing access

Audit trail tables also carry DENY UPDATE, DELETE to every runtime identity. The admin group is not bound by a DENY - which is why SQL auditing records every statement, including Invardia's.

The signed grant procedure, the five-year rule and the audited cut-over

SIGNED GRANT PROCEDURE

- One procedure per product database, signed by an in-database certificate.
- Applies only the release's shipped grant list.
- Grantee must be a product role; permission one of SELECT, INSERT, UPDATE, DELETE, EXECUTE; object in a product schema. Never a fixed db_* role; never lifts a DENY.
- The updater may only EXECUTE it. Changing it needs the admin group, and a change removes the signature.

FIVE YEARS WITHOUT CONTACT

- Certificate expiry 2099; a test proves it.
- Private key removed after signing: nothing to store, steal or rotate.
- Rules frozen: a design that needs a changed procedure counts as a design error.
- Roles for future components are compiled in, so a component added in year 3 needs no admin.
- Recovery needs no standing member: a Privileged Role Administrator adds one, or an owner re-points the server admin.

AUDITED CUT-OVER PER ENVIRONMENT

- Runs only inside an open Support-app session; every SQL statement goes through the audited support route, which records server, database, SHA-256 of the statement and the reason before it runs.
- Plan first, read-only proof step, explicit named steps.
- One rollback record per run (before values), every write read back.
- One environment at a time: Invardia's own first, then test, then customers one by one, each after a green pipeline and complete reports; customers told beforehand.

CHAPTER 5 · SUMMARY

Network: outbound only, a public or a private web app, and a known list of hosts

- **Nothing connects inbound** to the engines. The engines, updater and web app make outbound HTTPS calls only.
- **Public option** (the guided install's default): the web app is reachable from the internet, always behind Entra sign-in and your Conditional Access.
- **Private option:** setup builds the virtual network, private endpoints for every data service, private DNS and an internal-only web app, reachable only from your network.
- **Encryption in transit:** TLS 1.2 minimum on SQL and storage, HTTPS only, HSTS on the web app.

DATABASE EXPOSURE

Public shape: Entra identities only, no password to guess. Private shape: a private endpoint and public network access off.

YOUR CHOICE AT SETUP

The shape is chosen when the environment is built; a public environment can be rebuilt as private later.

NETWORK · PUBLIC AND PRIVATE

Two deployment shapes, compared

Public (default)

Part	Setting
Web app	Internet ingress, Entra sign-in required, assignment required, your Conditional Access
MCP server	Entra bearer token with an app role
Azure SQL	Public endpoint, Azure services admitted, Entra-only, TLS 1.2
Storage · Key Vault · Registry	Public endpoints; identity (RBAC) only, no shared key, no admin user
Engines, updater	No ingress; outbound only
Who builds it	The guided install

Private

Part	Setting
Web app	Internal ingress only: your network, VPN or peered clients; sign-in still required
MCP server	Internal; Entra token
Azure SQL	Private endpoint; public network access off
Storage · Key Vault · Registry	Private endpoints (registry on a tier with network controls)
Engines, updater	In your virtual network; outbound through your firewall to the hosts on the next slide
Who builds it	Setup: virtual network, private endpoints, private DNS zones, internal-only web app; your team links hub peering and DNS and opens the egress list

NETWORK · EGRESS

Every outbound host the product calls

Host	Used by	Purpose	Required?
login.microsoftonline.com	All components	Entra tokens (managed identity, sign-in)	Yes
graph.microsoft.com	Engine, tagging, web app	Entra data, advanced hunting + Exposure Graph, sending mail	Yes
api.securitycenter.microsoft.com	Engine, tagging	Defender for Endpoint API	Yes
management.azure.com	Engine, web app, updater	Resource Manager, Resource Graph, job control, image builds	Yes
api.loganalytics.io	Engine, web app, MCP	Log Analytics queries	Yes
***.ingest.monitor.azure.com**	Engine	Logs Ingestion (your data collection endpoint)	Yes
***.blob / queue / table.core.windows.net**	Engine, updater	Your storage account	Yes
***.database.windows.net**	Engine, web app, MCP, updater	Your Pro database	Pro
***.vault.azure.net · *.azurecr.io**	All	Your Key Vault · your registry	Yes
invardia.com	Updater, engine	Signed releases and keys, licence, status reports (uplink)	Pro; reports can be off
mcr.microsoft.com · api.nuget.org · Ubuntu package mirrors	Image build (in your registry)	Microsoft base images and packages for the build from source	For updates
api.securityplatform.microsoft.com	Engine	Sentinel data lake queries	Optional
outlook.office365.com	Web app	"Set up mail" (delegated, by an admin)	Optional
api.shodan.io · your ServiceNow · your Azure OpenAI	Engine, web app	Public-IP enrichment · connectors · AI summary and chat	Optional

The product maintains this list, and a test keeps it in step with the code. Community installs fetch signed updates from GitHub instead of invardia.com.

Encryption, secrets and keys

Control	Setting
In transit	TLS 1.2 minimum on SQL and storage; HTTPS only; HSTS on the web app
At rest	Azure SQL Transparent Data Encryption, storage service encryption and Log Analytics encryption
Keys	Microsoft-managed by default; customer-managed keys in your Key Vault as an option for SQL, storage and Log Analytics
Storage access	Identity only; shared key off; no public blob access
Key Vault	RBAC authorization; purge protection on; secret values never returned to the web app
Registry	Admin user off; pull by managed identity

- **No stored credentials for runtime:** managed identities everywhere; the sign-in app uses a federated credential.
- **Jobs get Key Vault references,** never secret values in settings. Connectors store only the *names* of their secrets.
- **Secrets pasted in the web app** are encrypted at once (AES-256-GCM, wrapped with RSA-OAEP to an RSA 3072 Key Vault key that never leaves the vault) and handed to the updater, which stores them. The web app never writes a secret.
- **Licence and updates** are verified against public keys built into the product; new keys arrive with the product (rotation by shipping both keys).
- **Optional keys** (Shodan, SMTP, ServiceNow) live only in your Key Vault.

CHAPTER 6 · SUMMARY

Authorization: least privilege by role, separation of duties, nothing written directly

- **Roles** checked on the server for every page and action: SuperAdmin, Admin, Reader, Delegated (scoped), Department, Wallboard.
- **Separation of duties:** exemptions are approved by the named approvers of the security domain, not by the requester; a second SuperAdmin can be required for configuration changes; releasing a guard needs a SuperAdmin and a reason.
- **Nothing written directly.** Changes are staged and written on Review & commit - one transaction with an audit row. Nothing is deleted; exemptions are ended.
- **Just-in-time:** nobody holds standing database admin rights; support access is enabled per need.

GOVERNED EXEMPTIONS

Every accepted risk has a reason, an approver and an expiry (3, 6, 12 months or a date). Owners get a monthly review mail. Expired exemptions stay flagged until someone acts.

Who can do what

Role	Can	Cannot
SuperAdmin	Everything an Admin can, plus roles and scopes, licence, updates and template releases, support access, guard releases, ending Strict mode (with a reason)	- (whoever runs setup is the first SuperAdmin)
Admin	Everything in the install: settings (reviewed), exemptions, overrides, jobs, connectors, chat incl. settings history	Roles, licence, updates, template releases, support access
Delegated	As Reader, plus exemption requests, overrides, tier moves and reviews inside the assigned scope (domain, asset kinds, tiers, department)	Anything outside the scope; a bulk action is refused whole if one row is outside
Reader	Every view, export and the chat	Any change
Department member	Its own department's assets and findings	Other departments
Wallboard	The wall board page	Everything else
External advisor (magic link)	View, comment, recommend for one department; approve only after a second check on their own sign-in	Anything outside the department; links are per person, short-lived, revocable, audited

Exemptions: approved by the domain's named approvers or an Admin; departments' own business approvers for their findings.

Configuration: each setting validated and reviewed; optional second SuperAdmin approval.

Releases: a SuperAdmin reviews and releases template changes method by method; customer rings get only approved versions.

CHAPTER 7 · SUMMARY

Logging and audit: everything is on the record, in your own workspace

- **Audit trail** in the Pro database: every action with who, when, why, before and after. Append-only for every runtime identity; removal only by the audited retention step, never under 13 months.
- **SQL auditing:** every statement by every identity - including Invardia's - to your Log Analytics workspace, 90 days.
- **Run records:** each run and step with outcome, duration and completeness; guard events; connector runs with every item before and after.
- **SIEM:** it all sits in your Log Analytics workspace, so Microsoft Sentinel or your SIEM export reads it natively.

TELEMETRY TO INVARDIA

Status only: heartbeats, run outcomes, guard events, health counts. Never findings, asset data or the estate inventory. On by default; one setting turns it off.

What is logged, where it goes, and how long it is kept

Record	Contents	Where	Kept	Tamper resistance
Audit trail	Every user and job action: who, when, why, before / after; the person behind a later job action	Pro database (cfg)	Kept; optional >= 13 months	DENY UPDATE/DELETE to runtime identities; retention only via an audited procedure
SQL security audit	Every statement, every identity, sign-ins (successful and failed)	Your workspace	90 days	Outside the database; admins cannot edit it from SQL
Run health	Start / end per job and step, outcome, failed step, guard rows	Your workspace (SI_RunHealth_CL) + Pro database	Workspace setting	Ingested via the Logs Ingestion API
Report status	Expected, retrieved, stored rows per report and slice; verdict and reason	Pro database (sidata)	90 days	Written by the engine only
Guard events	Guard, measured value, limit, outcome, release with reason	Pro database + workspace + mail	Kept	Release is a SuperAdmin action, audited
Connector runs	Run + every item: before and now	Pro database	Kept	Append-only history
Container logs	Job transcripts	Your storage + Azure Monitor	30 days	Identity-only storage
Support actions	Every Support-app request, statement hash and reason	Your environment's audit log + SQL audit + Entra sign-in log	As above	Recorded before the action runs

What is sent to Invardia - and what never is

Record	When	Contents
Heartbeat	Daily	Product, edition, version, ring, hosting, identity kind, licence state; Pro adds tenant id, host, object counts, environment name, guard mode
Run report	Each engine run	Job, outcome, duration, failed step, error class, counts, fix kind; Pro adds shortened error text with secrets removed
Report incomplete	A run with a NotComplete report	Numbers only: reports expected, stored, incomplete. No report names
Guard event	Each trip	Guard id, outcome, measured value and limit (numbers), counts, Strict flag
Data health	Daily (when enabled)	Per source and report: sampled, found, missing, verdict; Healthy / Unhealthy

NEVER SENT

- Findings, vulnerabilities, risk scores
- Asset, user or device names, record keys, UPNs
- The estate inventory, profiles or reports
- Override reasons or any free text from your people
- Secrets of any kind

YOUR CONTROL

On by default; one setting turns it off. Community reports anonymously (random install id, counts as ranges, no tenant, host or error text). One attempt per record, 5-second timeout, no retries.

CHAPTER 8 · SUMMARY

Release process and supply chain: signed, approved, staged, reversible

- **Built from source in your registry.** The updater downloads the signed source for its ring, verifies it and builds the image in your own registry. No prebuilt SecurityInsight image is shipped.
- **Signed and verified, with an SBOM:** RS256 manifest signature against public keys built into the product; ring, product, sequence (no replay, no step back), size and SHA-256 of the download.
- **Rings 0-3:** Invardia's internal ring first, then test environments and early customers, then design partners, then everyone. Customer rings only receive a version the release owner has approved.
- **Waves and soak:** a version can go to named installations first; a wave stops if one of them turns unhealthy.
- **Strict guards for 24 hours** after any release that changes engine code.
- **Rollback:** automatic if a roll fails; a vendor rollback is a new, higher sequence.

DEPENDENCY POLICY

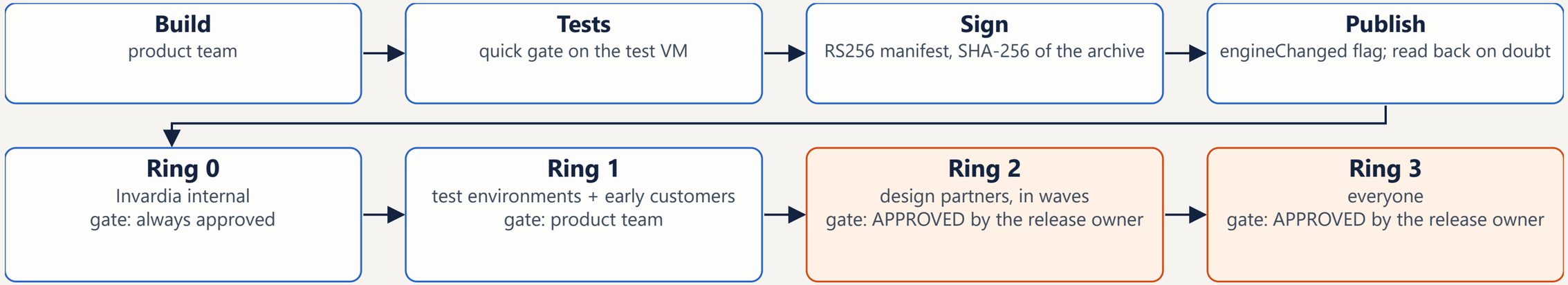
No PowerShell modules and no Azure CLI at runtime: REST and .NET only, enforced by tests. The web app and MCP server use standard .NET packages. Base images are Microsoft's.

YOUR CONTROLS

Switch automatic updates off, stay on a ring, accept manual-step releases explicitly. Nothing is pushed.

RELEASE · FROM CODE TO CUSTOMER

The release process, and who approves at each gate



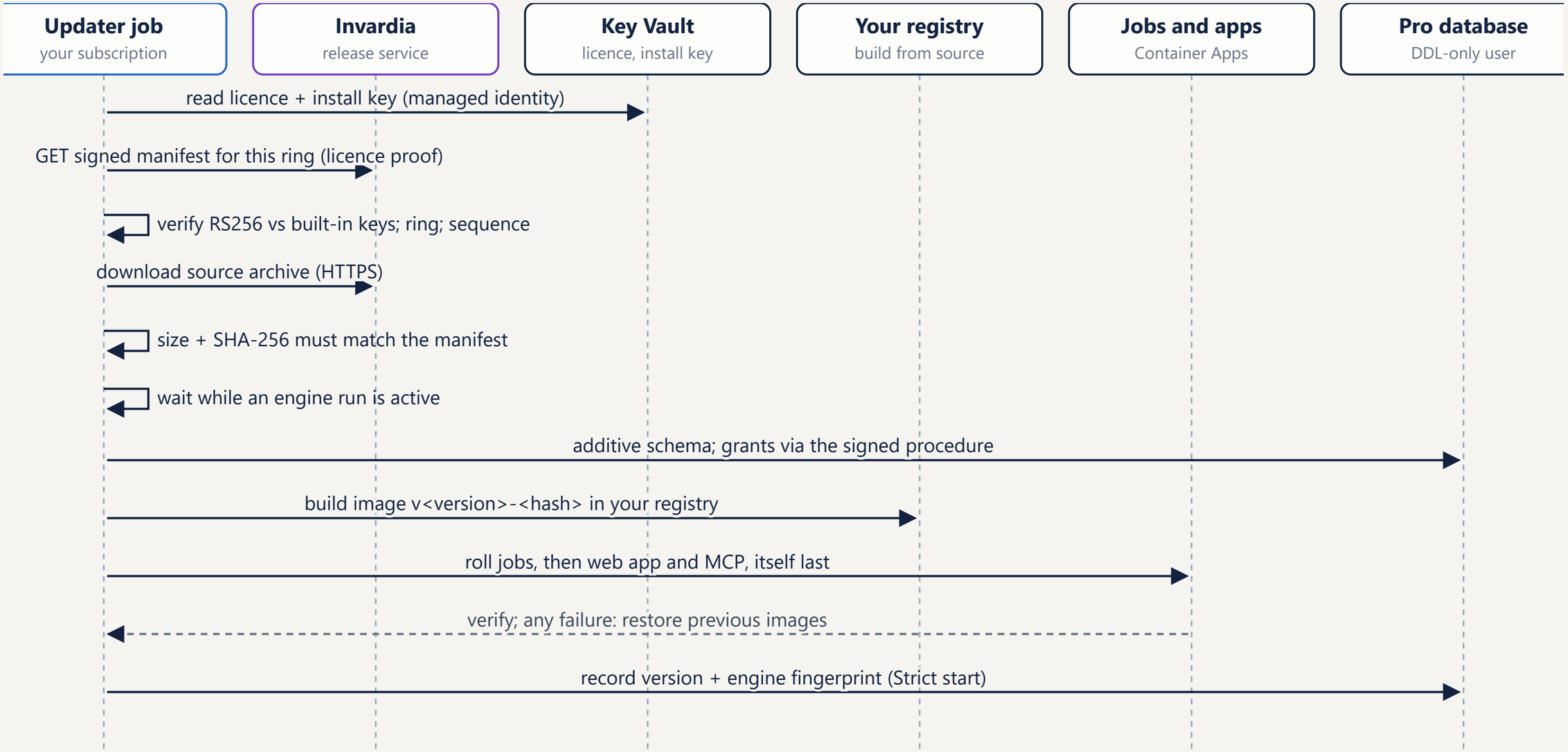
After each ring: per-environment check
runs complete, reports Complete, no new guard trips, data health Healthy - evidence collected before the next ring

Strict period
24 h after a release that changed engine code:
every numbered guard limit halved (min 1)
release mode: reports every 30 min for 12 h
SuperAdmin may end early, with a reason

Stop and roll back
a regression freezes promotion of that release
failed roll: every changed job back to its image
vendor rollback = new, higher sequence
a P1 returns affected rings to the last good version

Never published twice
if a publish gets no clear answer, the release
tooling reads the release state back first
Testing / Stable per component: Stable after
3 good daily runs with no code change

Sequence: the nightly update and its checks



Supply chain: what protects the code you run

Control	How	Stops
Signed manifests	RS256 (RSA PKCS#1 v1.5, SHA-256); public keys built into the product, chosen by key id; a key fetched over the network is never trusted	A forged or altered release
Archive integrity	Signed size and SHA-256 of the archive checked before anything is extracted	A swapped or corrupted download
Anti-replay	Sequence must not be lower than the one applied; signed ring must equal the install's ring	An old or wrong-ring release
Build from source	Image built in your registry from the verified source; tag carries the version and a content hash	Tampered prebuilt images
Licence proof on renewal	The renewal request carries the current licence's SHA-256 and the install key	Someone requesting your licence
Approval before customer rings	Ring 2 / 3 refused without a recorded approval; waves; per-environment evidence	A bad release reaching everyone
Dependency policy	No PowerShell modules, no Azure CLI in the runtime (tests enforce); one pinned PDF library verified by SHA-512	Unreviewed third-party code
SBOM	A software bill of materials (CycloneDX / SPDX) with every signed release	Unknown components in what you run
Signing key	Update-signing key held as a non-exportable key	Theft of the signing key
Strict guards + rollback	24 h halved limits after an engine change; automatic image rollback	Damage from a faulty release

CHAPTER 9 · SUMMARY

Guards: damage brakes that stop a mass change before it leaves

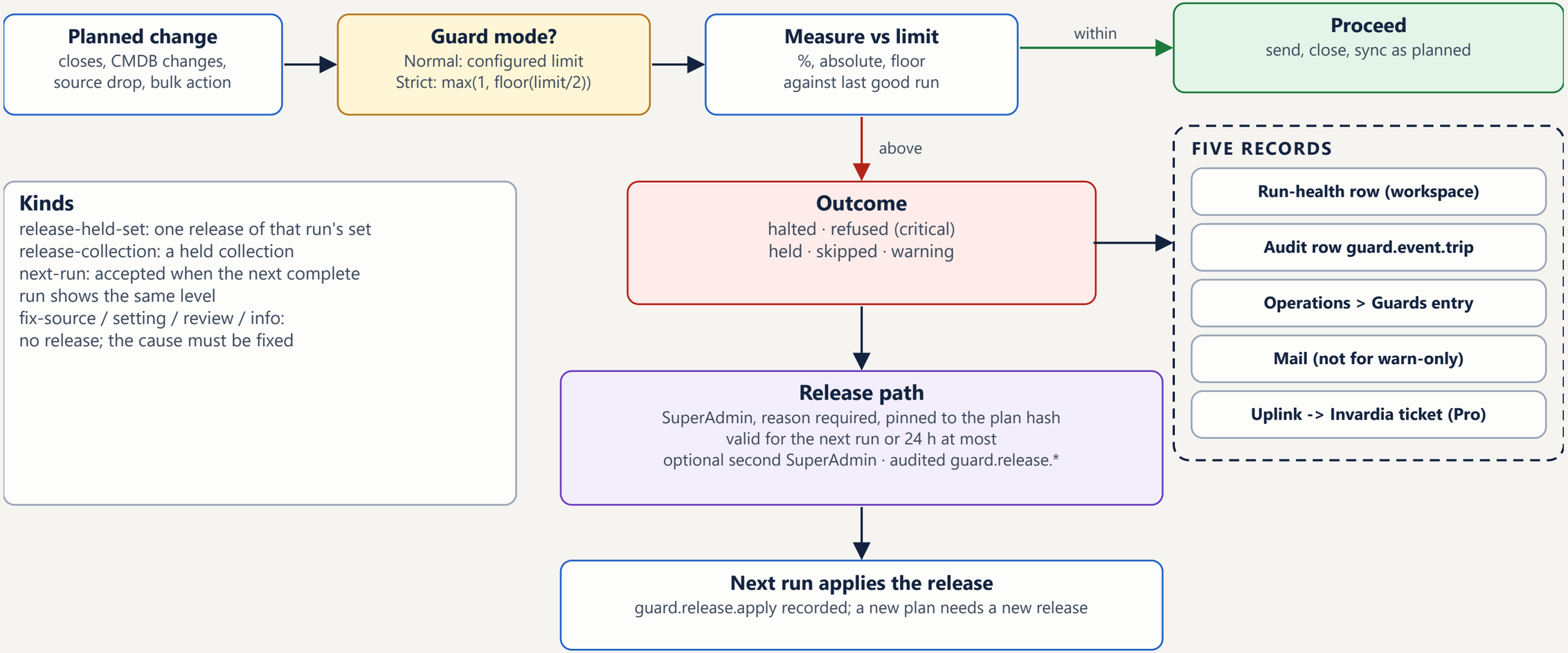
- **40 guards** in one locked catalog, each with what it protects, a limit, an outcome and a release path.
- **Outcomes:** halted or refused (critical) · held, skipped or warning.
- **Release:** a SuperAdmin, with a reason, pinned to the exact plan they saw, for the next run or at most 24 hours; optionally a second SuperAdmin.
- **Every trip** writes a run-health row, an audit row and a Guards-page entry, sends a mail (never for warn-only) and reports to Invardia, which opens a support ticket for Pro.
- **Guard mode:** Normal, or Strict for 24 hours after an engine-changing release - limits halved, never below 1, floors unchanged.

INTEGRITY STOPS ARE NEVER SWITCHED OFF

Schema gates, incomplete snapshots and short reads stop the step even when a guard's event is switched off. Only the mail and event stop.

GUARDS · DECISION FLOW

How a guard decides



Guard, limit, outcome, who is told

Guard	Protects	Limit (default setting)	Strict	Outcome	Who is told
vuln.connector.close-ceiling	Open vulnerability tickets from mass close	10 % of open (and an optional absolute cap, 0 = none)	5 %	Halted; SuperAdmin releases the held set	Guards page, mail, Invardia
vuln.connector.collapse	Close-loop from a collapsed source	Floor 0.5 (half the previous level)	Floor unchanged	Halted; never releasable	Guards page, mail, Invardia
cmdb.handoff.mass-change	CIs your ServiceNow imports	10 % changed	5 %	Held; SuperAdmin release	Guards page, mail, Invardia
cmdb.sql.collapse	The CMDB hand-off database	Below 50 % of the previous set	Floor unchanged	Refused	Guards page, mail, Invardia
asset.properties.collection-change	Asset metadata handed to CMDB / vulnerability systems	10 % changed	5 %	Held for that source	Guards page, mail, Invardia
collection.source.asset-drop	CIs and everything derived	30 % fewer records than the last good collection	15 %	Last good kept; accepted when the next complete run confirms	Guards page, mail, Invardia
exemption.bulk.refused	Findings from mass exemption	500 per action	250	Refused in the web app	User, audit
datasync.collection.held / refused	Days shown in the web app	Incomplete collection / rows without key columns	-	Held / refused	Guards page, Invardia
ra.output.row-drop · column-emptied	Risk Analysis output	More than 80 % fall, or a column emptied	-	Warning; report marked NotComplete	Guards page (no mail)

40 guards in the catalog; six are on by default (CMDB collapse, CMDB mass change, vulnerability target incomplete, asset drop, concurrent CI build, Risk Analysis bucket count jump). Each has its own on/off and mail switch.

Strict mode: extra caution for 24 hours after an engine change

- **Trigger:** the updater records a SHA-256 fingerprint of the engine code after a successful roll. A changed fingerprint starts Strict; the first fingerprint on an install starts nothing.
- **Rule:** effective limit = max(1, floor(configured x 0.5)), never above the configured value. "No limit" stays no limit. Collapse floors are not changed.
- **Applies to** the numbered brakes: close %, close absolute, asset-properties change, CMDB hand-off change, asset drop, bulk exemption.
- **Early end:** a SuperAdmin ends Strict with a reason; audited; it ends only that period.
- **Visible:** the Guards page shows configured and effective limits and when Strict ends; mails say "limit 5 % ... normally 10 %".

ON THE HEARTBEAT (IDENTIFIED INSTALLS)

```
guardMode: {
  mode:          "strict" | "normal",
  until:         "<ISO time>",
  startedBy:     "<release>",
  version:       "<version>",
  endedEarlyAt: "<ISO time>" | null,
  endedBy:       "<role, never a person>"
}
```

A trip in Strict mode reports Strict, the configured and the effective limit to Invardia, so a release's evidence shows whether its first day was calm.

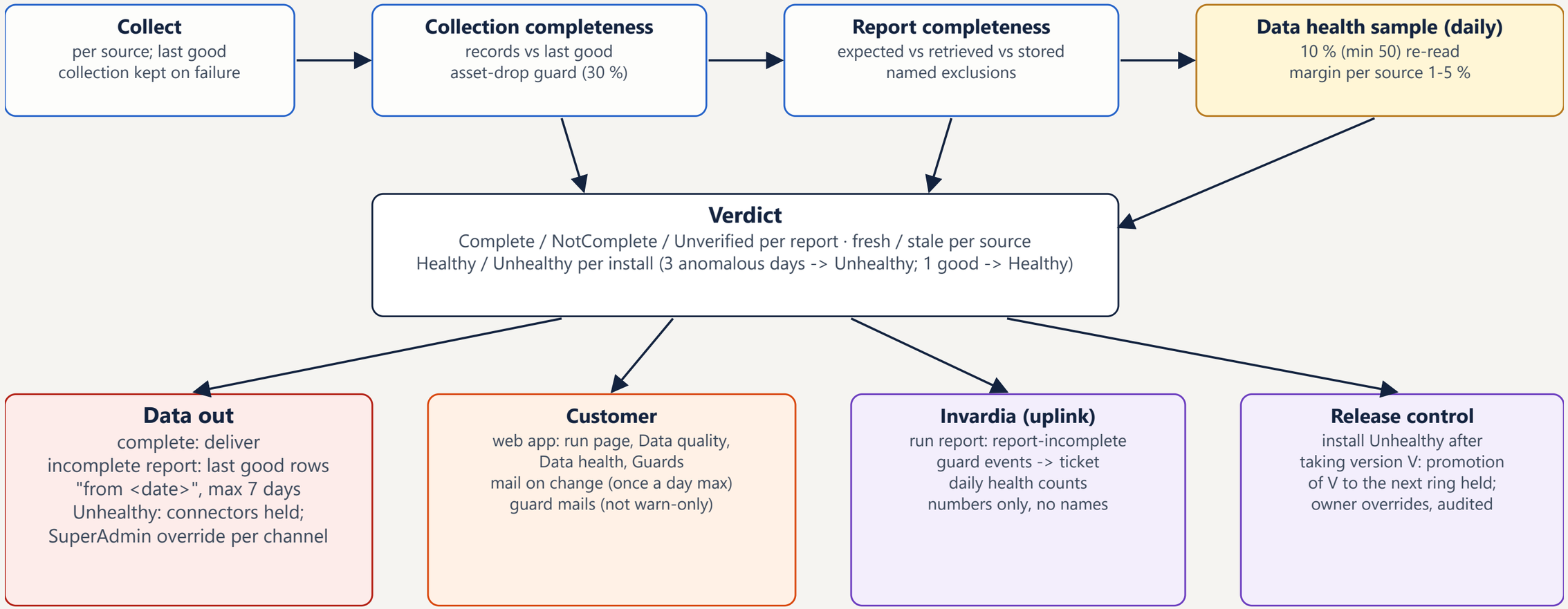
Data health monitor: never green on missing data

- **Every report proves its completeness:** expected rows (an unbucketed count at the source) vs retrieved vs stored, per report and per slice. Verdict Complete, NotComplete or Unverified.
- **Differences are named,** never silent: out of scope, default exclusion rules, duplicates. Any unexplained row makes the report NotComplete.
- **Example from Invardia's own install:** 65 of 65 summary and 65 of 65 detailed reports Complete; detailed 27,711 expected, 27,666 stored, the difference named as exclusions.
- **Incomplete means held:** nothing closes, no ticket changes, no mail goes out from a run that is not complete. A report that fails keeps its last good rows, marked "from <date>", for at most 7 days.
- **Fail-closed hand-off:** a lookup a report needs that cannot be read makes that report NotComplete - never "complete with gaps".

DATA HEALTH CHECK

Every day a random 10 % sample (at least 50) of each source is re-read from the source and compared. Three anomalous days make the install Unhealthy: data-out is held and ring promotion stops.

The monitoring loop

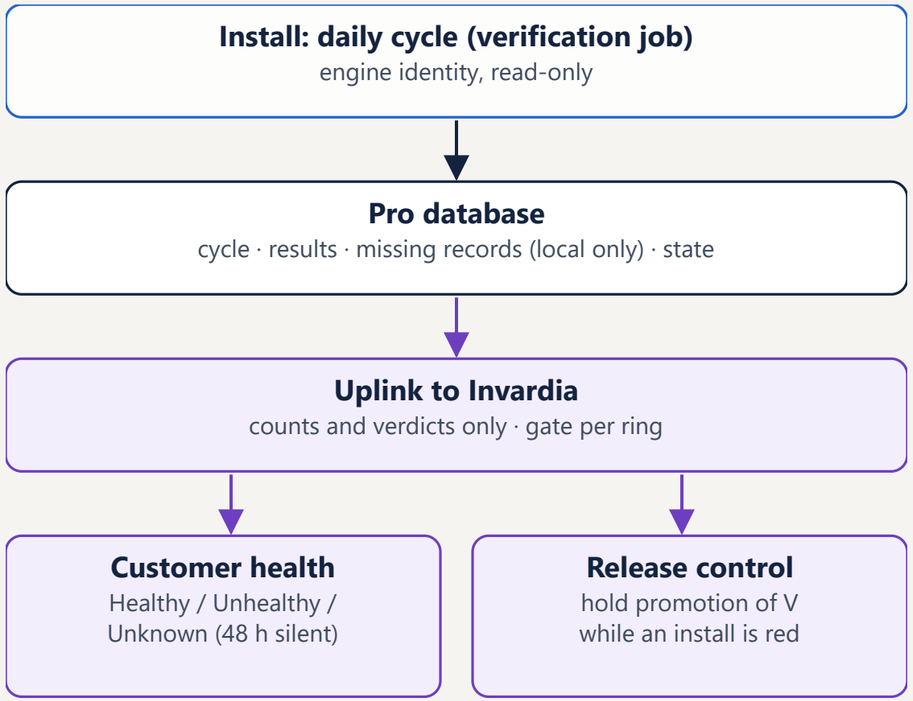


Check, threshold, action, who is told

Check	Threshold (default)	Action on breach	Who is told
Report completeness	Expected = retrieved; stored = retrieved minus named exclusions	NotComplete: run partial; nothing downstream closes on it	Run page; Invardia (report-incomplete, numbers only)
Slice completeness	Each slice vs its own count; one retry	Recorded loss -> report NotComplete	Run page
Fail-closed lookup / hand-off	Any lookup a report needs cannot be read	That report NotComplete, never retried	Run page; Invardia
Per-report loading	Last complete rows at most 7 days old	Carried rows marked "from <date>"; older shows as missing	Web app
Collection completeness (asset drop)	30 % fewer records than last good (15 % in Strict)	Last good kept; accepted when the next complete run confirms	Guards page, mail, Invardia
Freshness / staleness	No good collection for 7 days	Delivery from that source waits; calculation goes on	Run text, web app
Failed or partial run	A failed step or an incomplete collection	Data sync holds it; no closes, tickets or mail	Guards page; Invardia run report
Data-quality trends	Vs median of last 3 runs: assets -30 %, report -50 % or 0 rows, fill -30 pts, tiers 20 %	Warning only, never blocks	Web app; mail when the set changes
Data health sample	10 % (min 50) per source; margin 1-5 %, at least 5 records	3 anomalous days: Unhealthy -> data-out held, promotion held	Web app; Invardia (counts)

What the daily health report contains, and where it goes

Layer	Field	Meaning
Envelope	schema, kind = health, product, version, install, tenant, sent	Identified (Pro) installs only
State	state (Healthy | Unhealthy), since	Unknown is set by Invardia after 48 h without a report
Cycle	id, started, finished, anomalous, consecutive anomalous, complete, not checked	One daily cycle
Sources[]	source, sampled, found, missing, field mismatches, margin {%, minimum}, verdict	Entra users and apps, devices, Azure, Defender for Cloud, vulnerabilities
Reports[]	report, sampled, matched, missing, extra, margin, verdict	Up to 200; trimmed over 32 KB
Data out	held; overrides[] {channel, role, since, reason = "given"}	Never the reason text



Never sent: record keys, names, UPNs, hosts, people, reason text, any security data.

CHAPTER 11 · SUMMARY

Support: customer-controlled access, a fixed triage, and a fix path through the rings

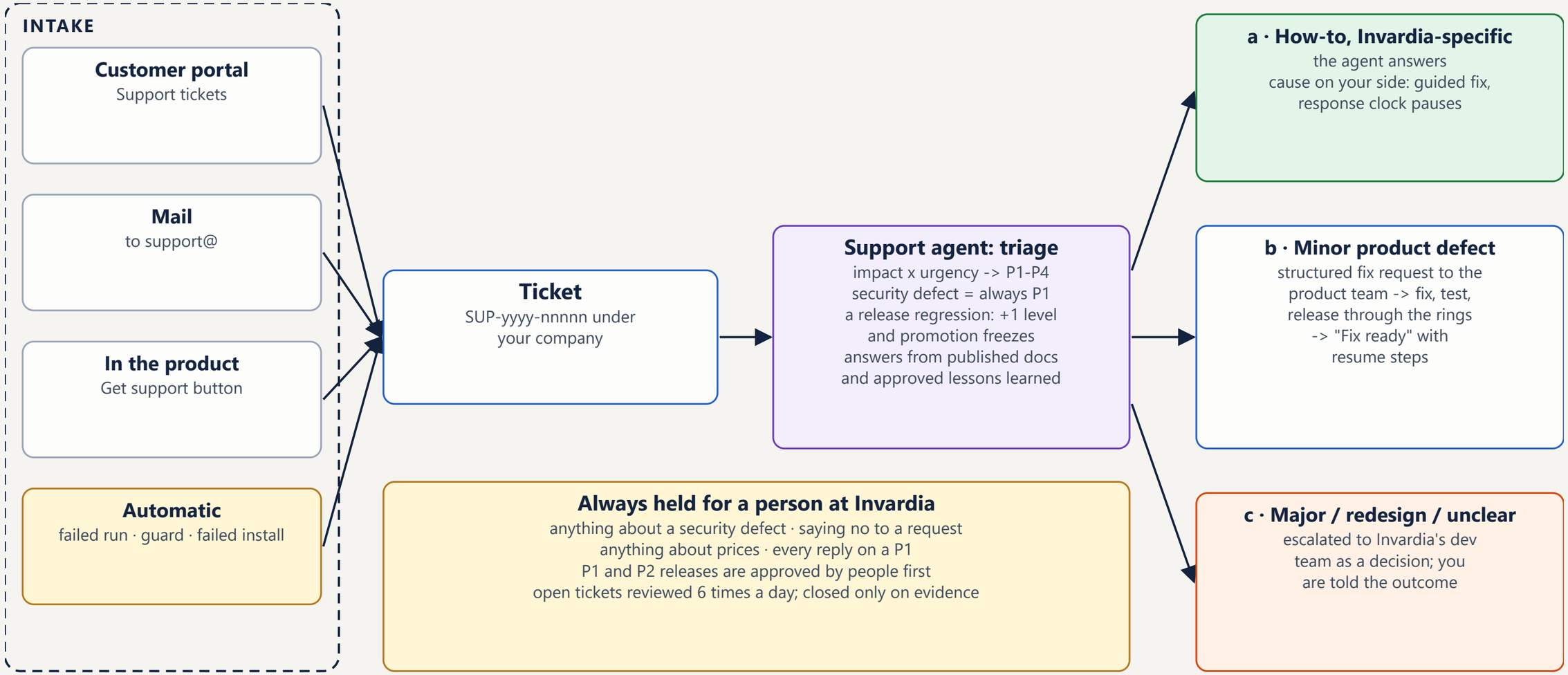
- **The Invardia Support app is yours:** you create it in your tenant with Invardia's script, scope it, and can disable or delete it at any time. It is optional - you get support without it.
- **Two levels:** Troubleshoot (read-only; Key Vault metadata without secret values) and Setup (repair rights on SecurityInsight's own resources, membership of the SQL admin group). Grant Setup for an install or repair, then lower it.
- **Customer control:** enable or disable per environment, secret lifetime 1-730 days (90 by default), renewal automatic, approval-required or off.
- **Audited:** every request goes through a recorded support session with a reason; every SQL statement is hashed and logged before it runs, and appears in your SQL audit and your Entra sign-in logs.

IT CAN NEVER

- be the identity the product runs as
- act outside the tenant and subscription the session names
- run a cut-over script without an open, recorded session
- read secret values at Troubleshoot level

SUPPORT · THE FLOW

From a request - or an automatic signal - to a fix



Priority, what it triggers, and the response targets

Impact \ urgency	High	Medium	Low
Critical (several or all customers)	P1	P1	P2
High (one whole customer)	P1	P2	P3
Medium (part of one customer)	P2	P3	P4
Low (one user, cosmetic, how-to)	P3	P4	P4

Urgency: High = blocked or blind, no workaround, spreading · Medium = a workaround exists, data going stale · Low = can wait for the next release. The same problem at a second customer becomes one critical incident.

Priority	First response	Then
P1	2 hours in support hours; best effort outside	Promotion stops; affected rings back to the last good version the same day; hotfix through ring 1, a short ring 2 soak, then ring 3; affected customers told directly
P2	1 business day	Next promotion waits; fix in the next patch
P3	2 business days	Fix in the next release
P4	5 business days	Backlog and roadmap

Support hours: Monday-Friday 08:00-16:00 Danish time, excluding Danish public holidays, 24 and 31 December. These are Invardia's published working targets; the agreement describes support as best effort.

Naming: what an install creates, and what you may change

Object	Pattern	Example	Configurable?
Resource group	rg-si-<label>	rg-si-prod-pro-ca	Yes - your own name at setup
Container Apps environment · Log Analytics · DCE	cae- / log- / dce-si-<label>	log-si-prod-pro-ca	Yes
Data collection rules	dcr-si-<stream>	dcr-si-endpoint-profile	No (one per table)
SQL server · Key Vault	sql- / kv-si-<label>-<suffix>	sql-si-prod-pro-ca-666ff6	Yes; suffix = 6 hex of a hash of the subscription id
Storage · Registry	stsi / acrsi<label><suffix>	stsiprodproca666ff6	Yes (global names)
Databases	fixed	sqldb-si-pro · sqldb-si-identity	No
Jobs · apps	caj-si-<step> · ca-si-web, ca-si-mcp	caj-si-risk-analysis	No
Managed identities (new installs)	id-si-<component>	id-si-engine · id-si-web · id-si-mcp · id-si-updater · id-si-tagging	No
App registration (sign-in)	product name	"SecurityInsight"	No
SQL admin group	grp-si-<purpose>	grp-si-sql-admins	Yes
Log Analytics tables · SQL schemas	SI_<Area>_<Kind>_CL · cfg, sidata, si, vuln, cmdbsync, snowstage	SI_Endpoint_Profile_CL	No (contract)
Reports	<Area>_<Topic>_<Summary|Detailed>	Identity_..._Summary	No (contract)
Environment handle (support)	<company>-<kind>	contoso-prod	Agreed with Invardia

<label> = <ring>-<edition>-<hosting> on Azure resources. Rule: no ring or environment in any directory object name (identities, groups, app registrations) - the resource group and its tags carry the environment. Each install records its own names, so a renamed or customer-named resource keeps working.

ASSURANCE · THREAT MODEL

Threats and the controls that answer them

Threat	Controls
Stolen user session or account	Entra sign-in with your MFA and Conditional Access; PKCE; sessions end after a fixed 8 hours; assignment required; server-side role checks; staged changes with reason; audit trail
Compromised engine identity	Read-only tenant permissions; writes only to SI's own resources; si_engine has no DDL, no security rights, cannot alter the audit trail
Compromised web app	No tenant write; no secrets; no DDL; jobs run as the engine identity; CSP, HSTS, anti-forgery
Tampered or malicious update	RS256 signature vs built-in keys; SHA-256 of the archive; anti-replay; build from source; SBOM; non-exportable signing key; approval before customer rings; Strict guards; rollback
Invardia compromised	No inbound path; status data only; Support app is yours, can be disabled; every support statement audited in your workspace
Insider mass change or mistake	Guards hold mass closes, CMDB changes, bulk exemptions; second SuperAdmin option; append-only audit
Privilege escalation in SQL	Per-component roles; grants only through the frozen signed procedure; no standing human admins; SQL audit of every statement
Silent data loss	Per-report completeness proof; fail-closed hand-off; incomplete never green; data health sample
Prompt injection via the chat	Chat calls only fixed, read-only tools; shows the queries behind every answer; your own Azure OpenAI
Data exfiltration to third parties	Optional outbound only (Shodan: public IPs; your ServiceNow; your Azure OpenAI); telemetry carries no findings or names

ASSURANCE · REGULATORY MAPPING

How SecurityInsight supports DORA, NIS2 and ISO 27001 controls

Capability	DORA (Reg. 2022/2554)	NIS2 (Art. 21(2))	ISO/IEC 27001:2022 Annex A
Asset inventory with Tier 0-3 classification	Art. 8 identification of ICT assets and their criticality	(a) risk analysis · (i) asset management	5.9 inventory · 5.12 classification
Risk-ranked vulnerabilities and close-loop	Art. 9 protection and prevention (patching)	(e) vulnerability handling · (g) cyber hygiene	8.8 technical vulnerabilities
Privilege tiers from real permissions	Art. 9(4)(c) access rights, least privilege	(i) access control · (j) MFA context	5.15, 5.18 access · 8.2 privileged access
Governed exemptions, change history, audit trail	Art. 9(4)(e) change management	(f) assessing effectiveness	8.32 change management · 5.33 records
Trend and management KPI	Art. 13 learning and evolving	(f) assessing effectiveness	9.1 monitoring and measurement
Logging, SQL audit, run records	Art. 10 detection	(b) incident handling	8.15 logging · 8.16 monitoring
Signed supply chain, rings, rollback	Art. 28-30 ICT third-party risk (evidence)	(d) supply chain security	5.19-5.22 supplier · 8.25-8.28 secure development
Encryption in transit and at rest	Art. 9(2) data security	(h) cryptography	8.24 use of cryptography

SecurityInsight provides evidence and controls that **support** these requirements and **map to** these controls. It is not a certification, and SecurityInsight is not itself certified against DORA, NIS2 or ISO 27001.

Incident response and vulnerability handling

A VULNERABILITY IN SECURITYINSIGHT

- Report privately (security advisory or mail) - never a public issue.
- Acknowledged within 3 business days; initial assessment within 7 business days.
- A security defect is always P1: promotion stops, affected rings return to the last good version, a hotfix goes through the rings, affected customers are told directly.
- A leaked secret is rotated immediately on confirmation.
- The fix ships in the next patch release, with its severity in the release notes.

AN INCIDENT IN YOUR ESTATE

- Guards stop mass changes; a failed or partial run never closes, tickets or mails anything.
- Pro runs and guard trips open an Invardia support ticket automatically.
- You can switch off updates, the Support app or telemetry at any time; the install keeps running.
- Every action - yours, the product's and Invardia's - is in your audit trail and SQL audit for the investigation.
- Rollback: a vendor rollback is a new, signed, higher release; your configuration backups are kept 90 days.

WHAT CHANGED IN 3.0

Delivered in this release cycle

Change	What it does for you
Strict guards after an engine release	Every numbered damage brake is halved for 24 hours after a release that changes engine code; visible, auditable, endable by a SuperAdmin with a reason
Install names recorded	Each install records its own resource names, so renamed or customer-named components keep working
One mail-subject format	[Environment] Failed / Warning / Action needed: what [ticket] [SecurityInsight] - sortable by rules
Licence proof	A licence renewal carries the current licence's SHA-256 and the install key: nobody else can ask for your licence
Security baseline	Registry admin user off, Key Vault purge protection on, SQL auditing to your workspace for 90 days; rolled out through the audited cut-over
Per-report loading	A failed report keeps its last complete rows, marked "from <date>", for up to 7 days; the rest of the run loads normally
Fail-closed hand-off	A report whose lookup cannot be read is NotComplete - never "complete with gaps"
Report completeness proof	Every report compares expected, retrieved and stored rows; differences are named
Risk Analysis performance + progress lines	Faster scoring with identical output; a progress line every 30 seconds and a time limit on each Defender read
Least-privilege engine database role	The engine moves from broad database roles to its own role without DDL or security rights

INVARDIA SecurityInsight PRO

Questions

Fix what an attacker would reach for first - with a design your security team can verify.

invardia.com/products/securityinsight · info@invardia.com