CERTYX

AUTONOMOUS CERTIFICATE LIFECYCLE MANAGEMENT

User guide

Every console view: purpose, workflow, inputs, results, permissions and limitations

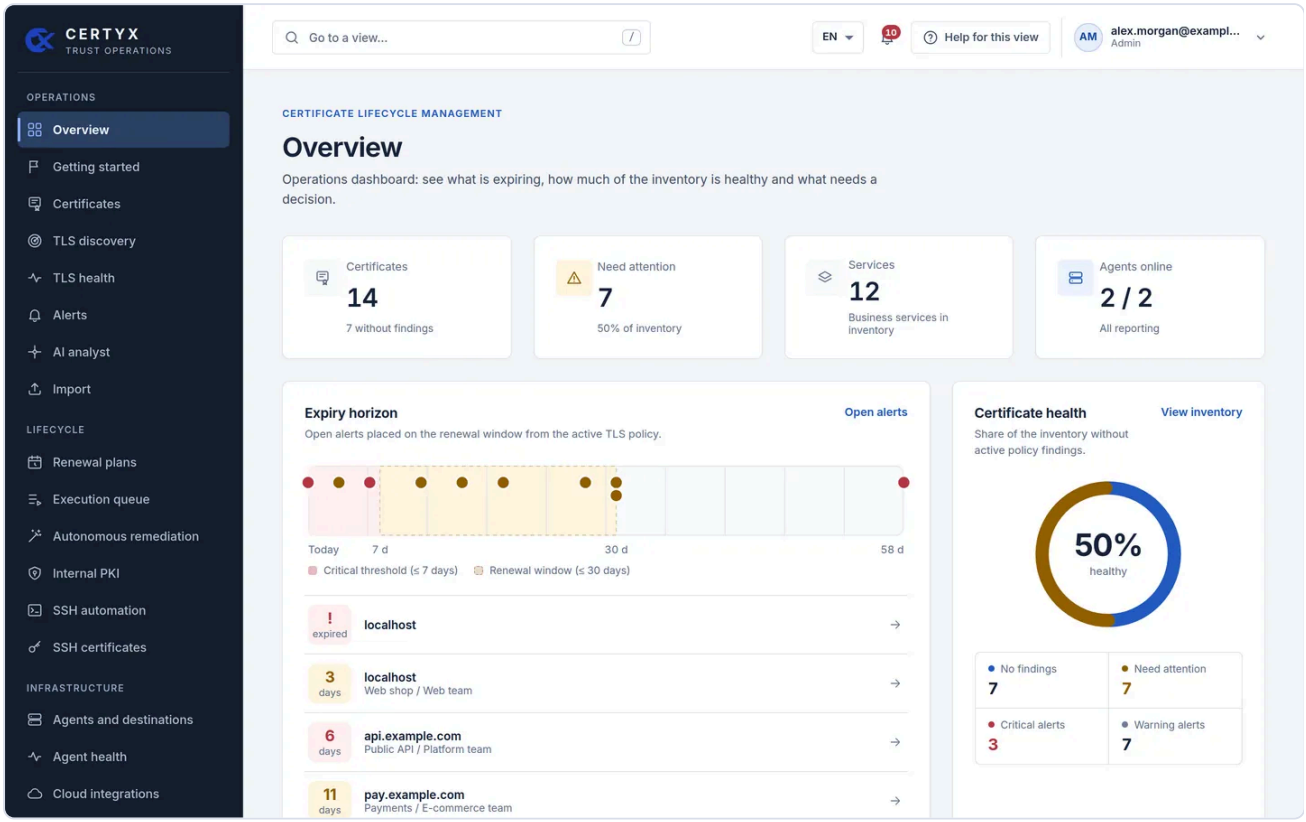
LANGUAGE	REVISION	DATE
English	8550060f	1 November 2026

Contents

1 Overview 3
2 Getting started 5
3 Certificate inventory 7
4 Certificate import 9
5 TLS discovery 11
6 TLS health 14
7 Renewal plans 17
8 Execution queue 19
9 Autonomous remediation 25
10 TLS policy 27
11 Access & security 29
12 Audit trail 36
13 Alerts 38
14 AI analyst 40
15 Platform administration 42
16 Platform operators 47
17 Sales 51
18 Internal PKI 55
19 Notification groups 57
20 SIEM export 60
21 Agents and destinations 68
22 Trust graph 75
23 Agent health 77
24 Platform operations 81
25 Mail suppression 84
26 AI analyst provider 87
27 Cloud integrations 89
28 Security event journal 93
29 SSH certificates 96
30 SSH automation 99

1 Overview

Operations dashboard: see what is expiring, how much of the inventory is healthy and what needs a decision.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Sign in to see your organization's actual data.
2. KPI cards show certificates, certificates needing attention, services and agents online (with current authenticated contact) out of all registered agents. Each card is a link: Certificates and Services open the inventory, Need attention opens the inventory filtered to those certificates, and Agents online opens Agent health — directly at the agent that needs attention when there is one; the card then names that agent, its cause (for example Not reporting, No first contact yet, Workload error) and since when.
3. The expiry horizon places open alerts on the renewal window from the active TLS policy; the red zone is the critical threshold. The list below shows the five nearest expiry dates; selecting one opens Alerts.
4. Certificate health shows the share of certificates without policy findings, plus critical and warning alert counts.
5. Operations activity counts audit trail events over the last 7 days and lists the three most recent.
6. The action center derives items from current data, most urgent first, each with its exact cause and a link to the record that explains it: expired certificates (filtered inventory), critical alerts, every agent that needs attention with its cause and explanation (its card in Agent health), failed or rolled-back renewal jobs with the agent's result (the certificate's renewal readiness), jobs waiting for a second approver, a failing SIEM export (administrators) and undeliverable notifications with the reason, blocked renewal

plans with their first blocker, discovery passes that reported errors or ended early (scan history), certificates in the renewal window, MFA disabled on your account and an empty inventory. It shows up to six items.

7. The Subscription and entitlements card shows the plan, use of the managed certificate family limit, available families, the number of certificates in the inventory (not billed) and license warnings. In the Certificate families list an administrator selects Manage family or Stop managing; only managed families consume the license.
8. To jump to any view, press "/" or use the "Go to a view..." field at the top of the screen.
9. Latest discovery shows the newest pass of each discovery agent and the newest server scan: state, start, live hosts, TLS endpoints and certificates found (new, replaced, unchanged) and the latest error; Scan history opens the full history in TLS discovery.

INPUTS & PARAMETERS

- Counts come from the inventory summary (whole organization, independent of any filter). Services are counted by distinct name.
- The horizon and action center use alerts, the active TLS policy, renewal plans and agent health; activity uses a server summary of the audit trail: events per local day of the browser (its current UTC offset) and the three latest events.
- Data loads when you open Overview and refreshes every 30 seconds while the view is open; the last update time is shown below the action center.
- When the TLS policy cannot be read, the horizon uses a fallback renewal window (30 days) and critical threshold (7 days).

OPERATION RESULT

The dashboard changes nothing, with one exception: an administrator switching management of a certificate family changes license usage. Links lead to the views where actions are performed.

PERMISSIONS

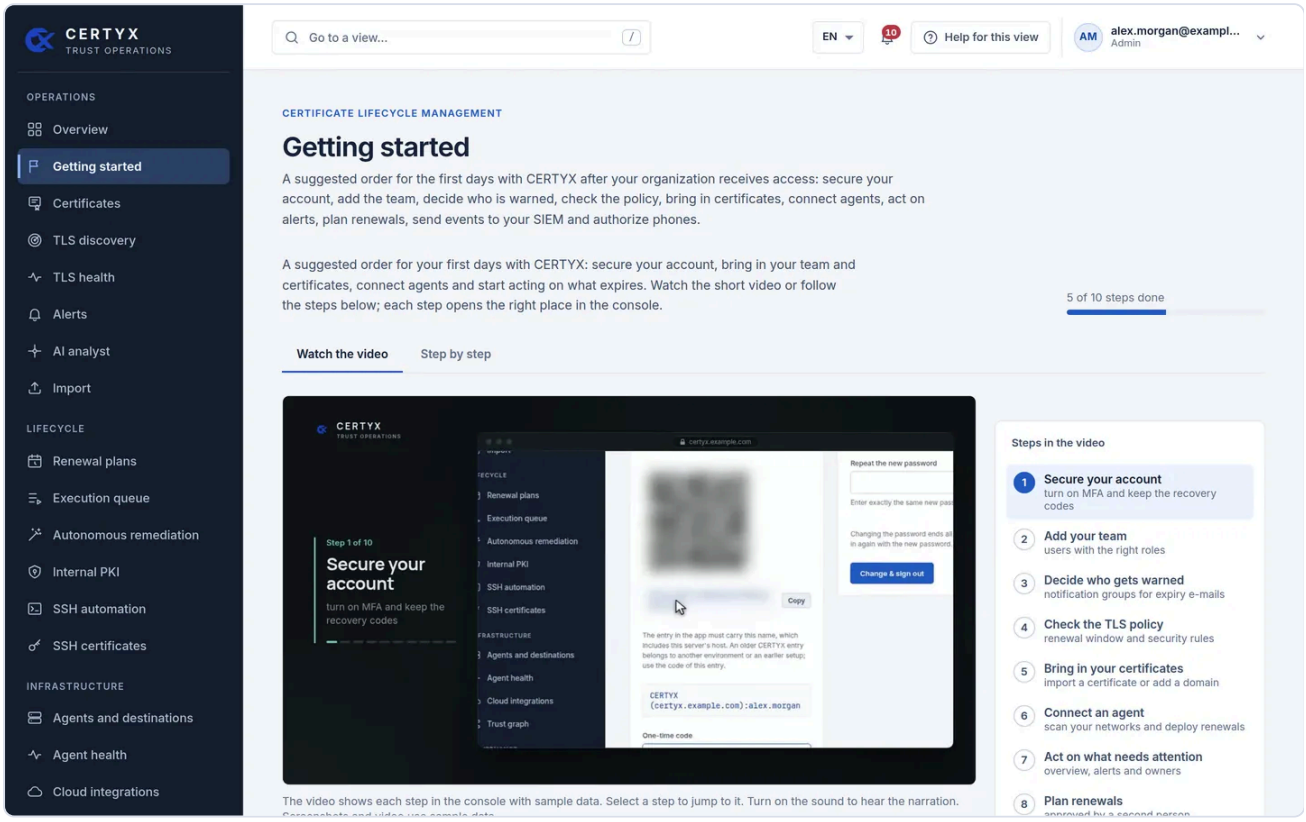
All signed-in users of the organization (Auditor, Operator, Admin). Only administrators switch management of certificate families. If a source cannot be read, its card shows "Data unavailable" instead of zero.

LIMITATIONS

The attention count reflects baseline policy, not verified deployments. The horizon shows only certificates with open alerts (inside the renewal window). Activity covers certificate audit trail events, not the platform security journal. The overview does not show renewal or deployment jobs; follow them in Execution queue. Managing a family is refused when the managed family limit is reached, the family changed meanwhile or the subscription does not allow new managed families.

2 Getting started

A suggested order for the first days with CERTYX after your organization receives access: secure your account, add the team, decide who is warned, check the policy, bring in certificates, connect agents, act on alerts, plan renewals, send events to your SIEM and authorize phones.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Watch the video: a recording of the ten steps in the console with sample data, in the console language, with captions. A narrator speaks each step: it starts muted and loops, and turning on the sound plays the narration once; it does not start on its own when your system asks for reduced motion. Select a step in the list beside it to jump to that part.
2. Or open Step by step: each step explains why it matters and how to do it, with a screenshot of the place in the console; steps only administrators can do (team, notification groups, policy, agents, SIEM export, phones) and steps that change data and need the Operator or Administrator role (certificates, acting on alerts, renewal plans) carry a note for roles that cannot do them.
3. Use Open ... on a step to go straight to the view where it happens; the button appears only when your role can open that view.
4. The progress bar counts completed steps. Steps the console can recognize (MFA or company sign-in, users, notification groups, certificates, agents that have contacted CERTYX — an agent still awaiting its first contact, revoked or expired does not count — renewal plans, a saved SIEM destination, active mobile devices) are marked done automatically. Mark the others with Mark as done and undo with Mark as not done.

5. After the steps, the console holds the operational journey together: the overview names what needs attention and why, Agent health explains each agent's cause with what to do, the scan history shows what discovery found, and a certificate's Renewal readiness shows each step from discovery to the updated inventory and what to configure.

INPUTS & PARAMETERS

- No input is required. Manual marks are stored in this browser for your account only.

OPERATION RESULT

A checklist that reflects your organization’s progress and links to every place you need. Nothing is changed by this view.

PERMISSIONS

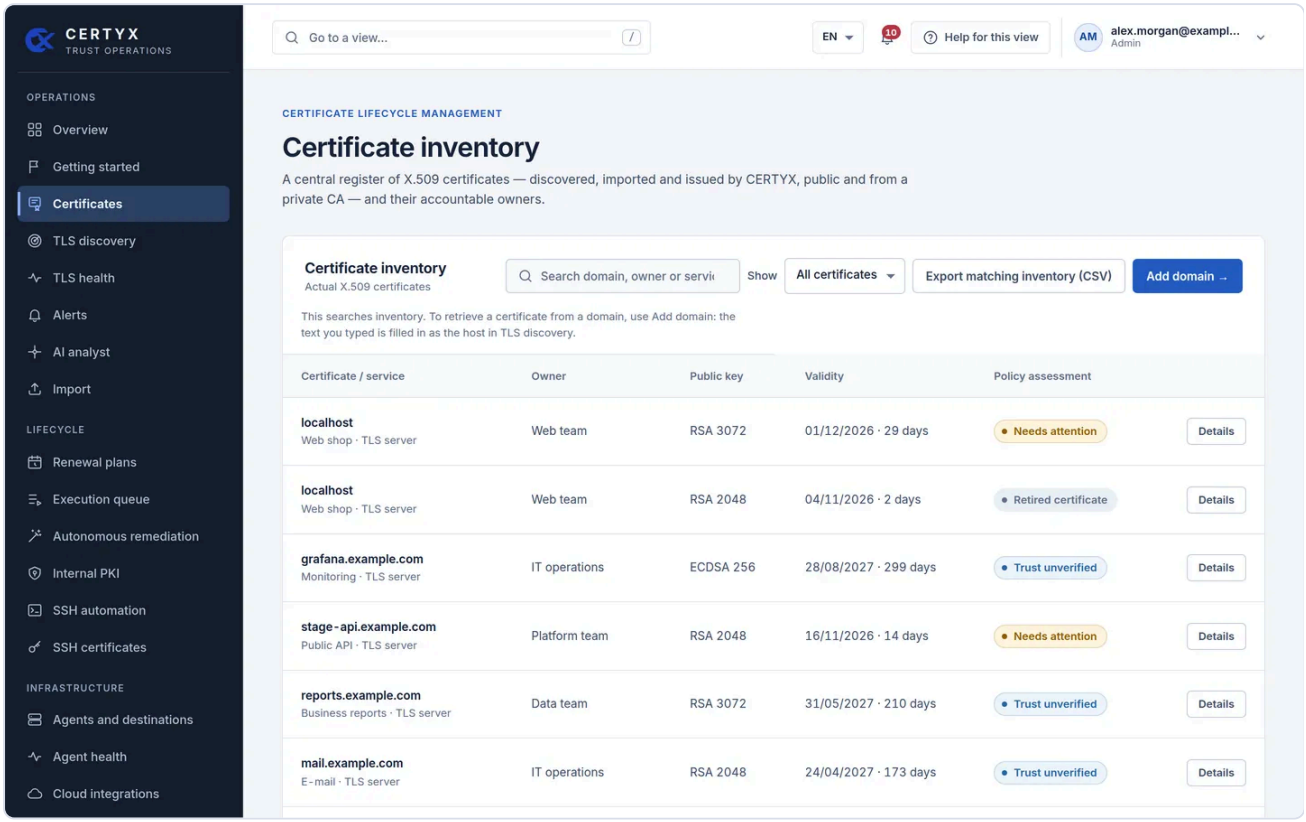
Every signed-in user of the organization. Steps that only administrators, or only operators and administrators, can do say so; detection that needs administrator access (users, SIEM destinations, mobile devices) is shown only to administrators.

LIMITATIONS

- The video and screenshots show sample data and may differ in detail from your configuration.
- Detection reads current data; a step done in another way (for example certificates brought in by cloud integrations) counts as done when its data exists.
- Check the TLS policy and Act on what needs attention cannot be detected; you always mark them yourself. Users, SIEM destinations and mobile devices are detected only for administrators, SIEM destinations only with the platform database that holds the security journal, and mobile devices only when the platform provides that feature.
- If the video cannot be loaded, the view shows a message and the steps below describe the same actions.
- Manual marks are not shared with colleagues or other browsers.

3 Certificate inventory

A central register of X.509 certificates — discovered, imported and issued by CERTYX, public and from a private CA — and their accountable owners.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Search a domain, owner or service.
2. To fetch the certificate of a domain that is not inventoried, choose Add domain →. It opens the inspection form of TLS discovery (inspecting requires the Admin role); the text in the search field is carried over as the host, and an empty search opens an empty form.
3. Open Details to inspect subject, issuer, purpose, serial, dates, public key, risk and SHA-256.
4. Review policy findings. Change Owner, Service, Production and the Responsibility groups (up to 10 non-archived groups; Manage recipient groups opens notification administration) and choose Save ownership, or choose Plan renewal — the plan is created and Renewal plans opens.
5. When the subscription includes the RISK_ENGINE feature, details also show the Business risk with its contributions and warnings about incomplete context.
6. When the AI analyst is enabled for the organization, Explain in the details opens it with a question about the certificate.
7. Each certificate shows its purpose from its extended key usages: TLS server, TLS client, TLS server and client, any usage (no EKU restriction) or other usage. Details list email, user principal name and URI identities. Server-only findings (localhost, loopback or IP identity) are not raised for TLS client certificates; a public certificate that permits client authentication gets an informational finding.

8. Show filters the list: all certificates, those that need attention (the dashboard count), expired ones, those inside the renewal window, or those valid beyond it. #inventory/validity/<filter> opens the list filtered.
9. Renewal readiness in the details shows every step of the renewal: inventory (where the certificate was last observed, with the service recognised at each endpoint, its TLS version and ALPN protocol, the suggested installation method with its confidence and evidence, and the issuer certificates the endpoint presented), policy and renewal window, the execution target that covers exactly its names, the installing agent and its diagnosis, where and how the agent installs (installation mode — the agent on the host or an agent as SSH relay with the remote host —, connector, deployment directory, the files it publishes such as fullchain.pem, privkey.pem and bundle.p12 or the configured paths, configuration check and reload commands, cloud resource, endpoint it checks and the service relay-check found — as the agent reports its destinations; a different installation mode or remote host blocks the step), the certificate authority, independent verification, what triggers the replacement (automatic profile, scheduled request or manual), the renewal plan, four-eyes approval, what the agent does step by step, the verification result and the updated inventory. A blocked step says exactly what to configure; when no target covers the certificate, administrators choose Create the target with these values, which opens Execution with the target the readiness suggests. The form shows under each field where its value came from and what the administrator must still enter.
10. Next renewal in the renewal readiness says how the certificate's next renewal starts — Automatic (no approval needed), Needs approval or Planned manually — and which automatic renewal setting decides it (installation default, organization, target or this certificate). Administrators change the certificate's Automatic renewal choice above the readiness: Organization setting, Renew automatically or Require approval; it also applies to the later renewals of the certificate.

INPUTS & PARAMETERS

- Search domain, subject, observed endpoint, owner, service, issuer, serial or SHA-256 fingerprint (up to 200 characters). PostgreSQL applies the filter to the full inventory of the organization; 50 rows are shown per page.
- Owner and Service: required, 1–200 characters each.
- CSV export streams the complete filtered result, without private keys. Spreadsheet formula cells are neutralized.

OPERATION RESULT

Metadata changes are saved and audited. A change of owner or service returns approved renewal plans for approval again (Ready for approval or Blocked).

PERMISSIONS

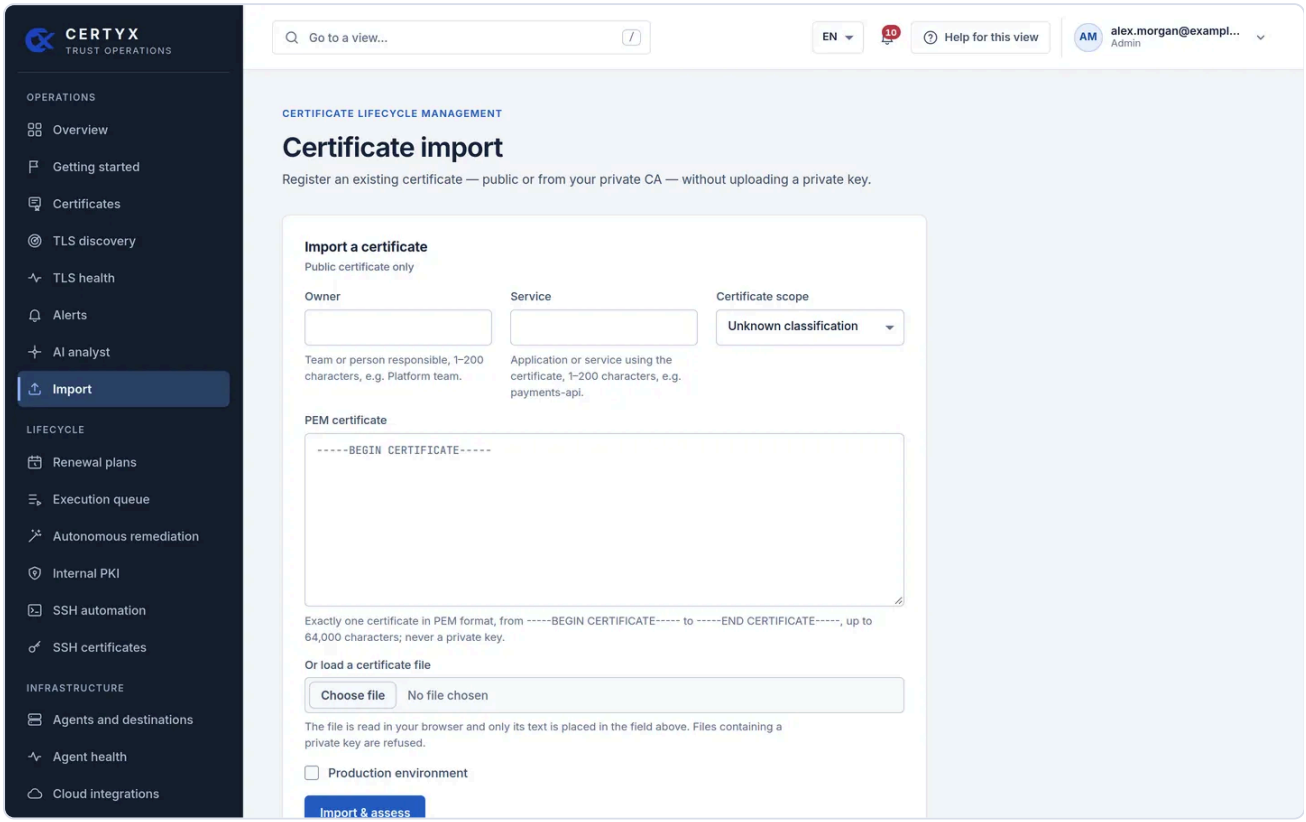
Every role of the organization: search, details and CSV export. Operator and Admin: save ownership and plan renewal. Business risk: only with the RISK_ENGINE feature.

LIMITATIONS

SHA-256 identifies a specific certificate. Risk 0–100 is not CA trust. Chain, revocation, SAN coverage and deployment correctness are not confirmed here.

4 Certificate import

Register an existing certificate — public or from your private CA — without uploading a private key.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Enter Owner and Service and choose the Certificate scope: Unknown classification (default), Public certificate or Internal certificate.
2. Paste one certificate, from -----BEGIN CERTIFICATE----- to -----END CERTIFICATE-----, into PEM certificate, or choose a file under Or load a certificate file (.pem, .crt or .cer in PEM text form). The file is read in your browser and only its text is placed in the field.
3. Tick Production environment if applicable and choose Import & assess.
4. The Import result panel shows the assessment, validity, key, risk and the number of policy findings. Open certificate inventory leads to the certificate in the inventory. The form is cleared, so you can import another certificate right away.

INPUTS & PARAMETERS

- PEM certificate: exactly one public X.509 certificate in PEM text form, up to 64,000 characters (a chosen file up to 64,000 bytes). Binary DER files and PKCS#12/PFX containers are not accepted — convert the certificate to PEM first, without the private key. A chain of several certificates is refused.
- Owner and Service: required, 1–200 characters each.

- Certificate scope: Internal certificate marks a certificate of your private CA — it then appears in the list of the Internal PKI view; Public certificate and Unknown classification remain in the inventory only.
- Content with a private key (any BEGIN … PRIVATE KEY block) is refused in the browser before it is sent, and again by the server. A certificate that is already inventoried is refused as a duplicate; an invalid X.509 PEM is refused with the reason.

OPERATION RESULT

The certificate is stored, X.509 metadata is extracted, current policy is assessed and an audit entry (certificate.import) is written.

PERMISSIONS

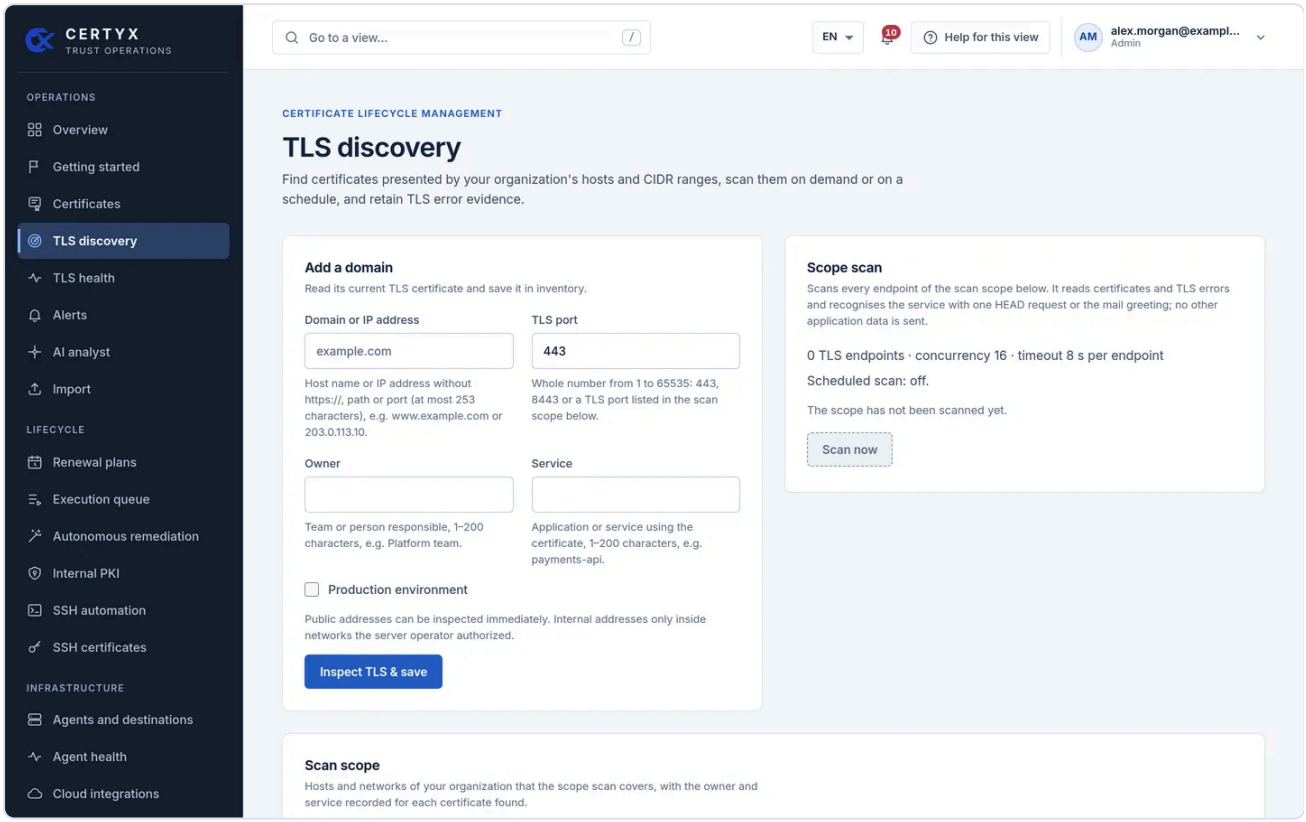
Operator or Admin. An Auditor sees the form disabled, with a note that the role can only view the inventory.

LIMITATIONS

Import does not issue or install a certificate and stores neither a private key nor a chain. A self-signed certificate can be inventoried, but its trust is not confirmed. The classification is the importer's declaration; CERTYX does not verify it.

5 TLS discovery

Find certificates presented by your organization's hosts and CIDR ranges, scan them on demand or on a schedule, and retain TLS error evidence.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. This view runs server-side discovery from the CERTYX server for your organization, also on dedicated instances. To find certificates inside networks the server cannot reach, use an enrolled agent: Agents and destinations → Automatic network discovery (reported networks, scope, mode, rate, windows); its results appear in the inventory and its alerts in Alerts.
2. For a single domain: enter its hostname without https://, choose a TLS port (443 by default), enter owner and service, then choose Inspect TLS & save. After success, choose Open certificate: the certificate is already stored in inventory.
3. Scan scope (Admin): under Hosts add a name or IP address with its TLS port, owner, service and production flag; under Networks add a CIDR range with explicit TLS ports and the same metadata; enter exclusions one per line.
4. To scan regularly, tick Scan automatically and enter an interval in minutes (the allowed bounds are shown below the field). Choose Save scope. Every save creates a new version and is recorded in the security journal; if another administrator saved the scope meanwhile, reload the view and apply your change again.
5. The Scope scan panel shows the number of TLS endpoints, the schedule with the next scan and the latest scan. Scan now starts a manual scan; progress updates live. Stop scan (Admin) stops the organization's running scan — yours, another administrator's or a scheduled one — on whichever

instance runs it; it ends as cancelled within seconds, and certificates already found remain in inventory. A scheduled scan or one started by another administrator appears when the status refreshes.

6. If the instance running a scan stopped, the scan is shown as interrupted. It continues from its saved checkpoint — endpoints already scanned are not probed again: the scheduler resumes it automatically at its next pass (unless the server operator switched this off), and Resume scan (Admin) resumes it at once. The note under the progress explains when a scan cannot be resumed (scope changed since it started, resume limit reached, stopped by an administrator or superseded by a newer scan); then start a new scan.
7. Review Last scan results (the certificate, or the reason there was no TLS response, for each endpoint), the observations and the certificates added to inventory. Each result names the detected service; each observation shows under Service and installation the service, TLS version and negotiated ALPN protocol, the suggested installation method for a renewed certificate with its confidence, and the evidence (for example Server: nginx/1.24.0 or port 6443). Observations of agents without service identification show Not identified.
8. The server operator sets the scope limits, the interval bounds and the authorized internal networks (Clm:Discovery). On an instance without dedicated organizations, a scope configured in CLM_DISCOVERY_TARGETS, CLM_DISCOVERY_CIDRS and CLM_DISCOVERY_EXCLUSIONS is taken over once as the scope of organization "default"; later changes are made in this view.
9. Agent discovery (Clm.Agent discovery scan on the customer host) retries transient identity-rotation and communication failures after Settings.RetrySeconds (30 seconds by default). Authentication, authorization and TLS trust failures stop the agent's discovery worker; with --once the command returns failure. On the agent host configure owner-only state and lock paths. A reporting outage can delay diagnostics; monitor missed agent contact as well.
10. Scan history lists every agent discovery pass and server scan of the organization, newest first: start, source (the agent, linked to its health, or the CERTYX server), scope (networks, mode and exclusions), state with phase and progress or when it ended (completed, interrupted because the scope changed, the profile was paused or the pass restarted), live hosts, TLS endpoints, certificates new to the inventory, replaced at an endpoint and unchanged, discovery alerts raised, reduced rate, the latest errors and whether it is the baseline pass. Results link to the inventory records of new or replaced certificates and to Alerts. #discovery/history opens it.

INPUTS & PARAMETERS

- Host: name or IP address without URL, path or port; TLS port 1–65535; owner and service (at most 200 characters by default); production flag.
- Network: IPv4/IPv6 CIDR naming the network address (for example 203.0.113.0/28) with 1–16 explicit TLS ports.
- Exclusions: IP addresses or CIDRs, also applied to the address a host name resolves to.
- The same endpoint (host or address and port) listed twice with a different owner, service or production flag is refused.
- Default server limits (shown below the form): 100 hosts, 64 networks, 4,096 addresses before exclusions, 16,384 TLS endpoints, 1,024 exclusions. Scan interval by default from 15 minutes to 7 days.
- Addresses: public ones always; private, local and reserved ones only inside networks the server operator authorized — a scope with other such addresses is refused, and host names resolving to them end as "no TLS response".
- By default 16 concurrent probes with an 8-second timeout per endpoint.

- Single domain: port 443 or 8443, or a TLS port of the organization's scan scope.

OPERATION RESULT

Public certificates and TLS errors are stored with the endpoint and observation time, together with the service fingerprint (server name sent, TLS version, cipher suite, ALPN protocol, the issuer certificates the endpoint presented, the HTTP status with the Server and X-Powered-By headers or the mail greeting) and the recognised service with the suggested installation method; per-endpoint results are kept for the organization's latest 20 scans, and a running scan saves the results gathered so far as its checkpoint. Scope changes, scan start, stop requests, resumption and the end of a scan (completed, cancelled, failed or interrupted) are recorded in the security journal and the inventory audit.

PERMISSIONS

Read the scope, status, results and observations: signed-in roles of the organization. Change the scope, inspect a domain, start, stop or resume a scan: Admin.

LIMITATIONS

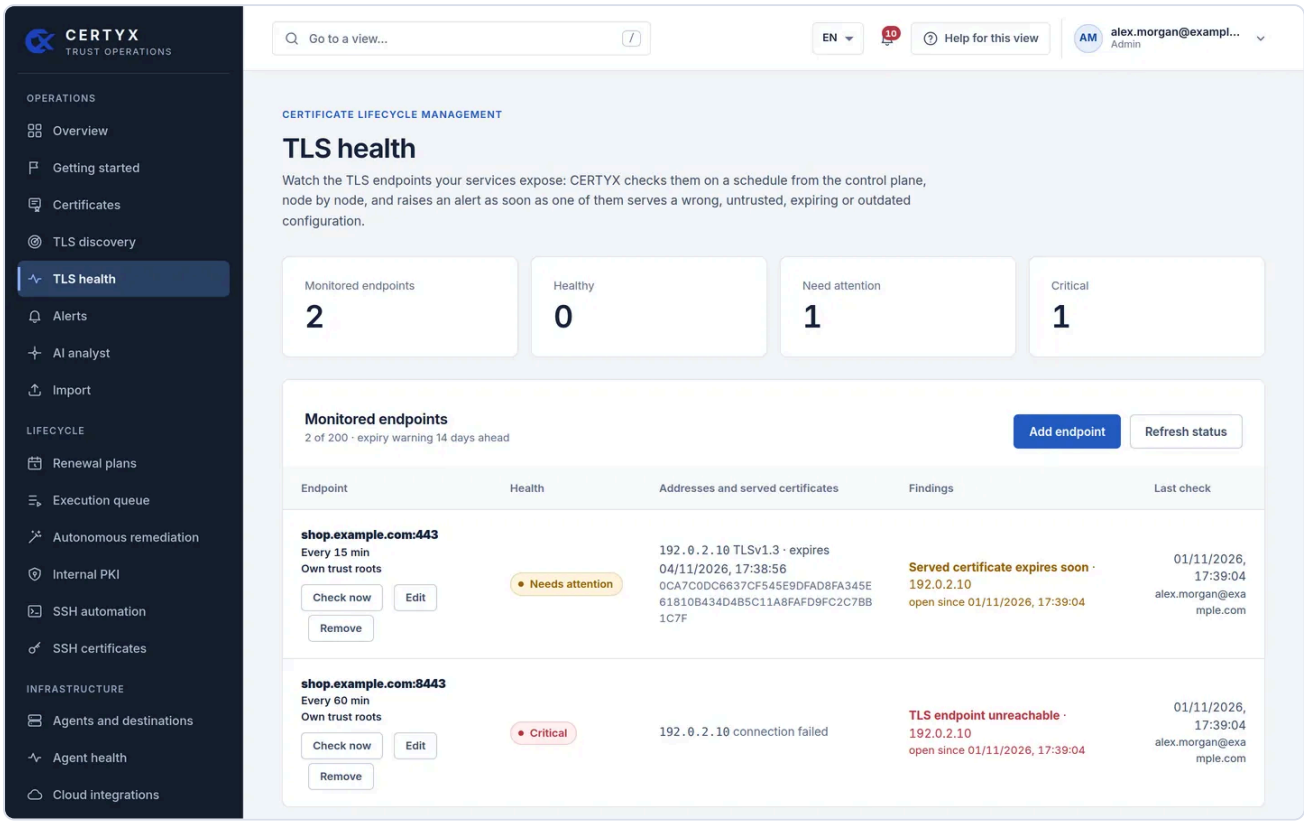
This is not an all-services scanner: only a TCP connection and TLS handshake on the listed ports and, unless the server operator switched it off (Clm:Discovery:ServiceIdentification:IdentifyServices), one HEAD / HTTP/1.1 request to an HTTP endpoint or the reading of the SMTPS, IMAPS or POP3S greeting; no other application data, no UDP or STARTTLS. The detected service is a suggestion from the configured rules: the Server header names the software that answered HTTP, which behind a proxy that passes headers through can be the backend; an endpoint that shows nothing distinctive is listed as an unidentified web server or TLS service with the generic PEM files installation and low confidence. One scan per organization at a time. Each instance scans up to 4 organizations at the same time by default (Clm:Discovery:SchedulerConcurrency); when more organizations are due, the most overdue start first and the rest on a later pass, so a scan may start later than scheduled. A stop requested while another instance runs the scan takes effect at its next progress save (every 2 seconds by default). A scan whose instance died without saving is shown as interrupted after 10 minutes without progress (default) and is resumed at most 5 times (default). The console shows only the latest finished scan: its first 100 endpoint results with the total number of endpoints; the full per-endpoint results remain stored for the kept scans. It also shows the latest 50 observations with their total number; the server sends only those 50 observations. Inspecting a single domain does not add it to the scope. An existing certificate keeps its current owner and service. The server-side limits do not apply to agent discovery, which has its own limits (Clm:Discovery:AgentLimits).

TROUBLESHOOTING

- No agent pass appears: the agent needs the discovery capability and an enabled, approved profile (Agents and destinations); check its diagnosis in Agent health.
- A pass ends as interrupted: the approved scope or ports changed, the profile was paused or the agent restarted the pass; the next pass starts by itself.
- Errors on a pass: the agent reported a fixed reason (for example a communication or local I/O failure, or a profile beyond its local allowlist); Agent health explains it with the remedies.
- No certificates found although hosts are alive: the common ports of the profile do not include the TLS ports of the services, or the hosts answer without TLS; add the ports or use full mode after agreeing it with the security team.

6 TLS health

Watch the TLS endpoints your services expose: CERTYX checks them on a schedule from the control plane, node by node, and raises an alert as soon as one of them serves a wrong, untrusted, expiring or outdated configuration.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Administrators select Add endpoint and enter the host name or IP address (no https://, path or port), the TLS port (443 by default) and the check interval in minutes (the limits are shown in the form). Optionally enter the expected SHA-256 fingerprint of the certificate every node must serve, and trust roots (PEM CA certificates) for endpoints of a private CA. Save: the endpoint is checked at the next scheduler pass, usually within a minute.
2. Each check resolves the name once and connects to every resolved address itself (each node behind a DNS name or load balancer, up to Clm:EndpointHealth:MaximumAddresses), so a changing DNS answer cannot redirect it. Per address it records the served certificate (fingerprint, subject, issuer, expiry), the negotiated TLS version and cipher, whether the certificate covers the host name, whether the chain is trusted with online revocation checking, whether the server omits intermediate certificates, and whether it still accepts TLS 1.0 or 1.1.
3. Findings: unreachable address, host not resolving, untrusted chain (unknown issuer, revoked, invalid), revocation status not checkable, missing intermediate, host name mismatch, TLS 1.0/1.1 accepted, expired or expiring certificate (warning Clm:EndpointHealth:ExpiringWarningDays ahead), a certificate other than the expected fingerprint, and node drift when addresses serve different certificates (for example after a rotation that missed one node).

4. Every finding is an alert in Alerts with the endpoint and address, open since its first sighting. The certificate's owner, service and notification groups come from the inventory when the served certificate is known. Open alerts reach the organization's notification channels (e-mail, Slack, Microsoft Teams, webhooks, PagerDuty) and critical ones the mobile app, like every other alert. An alert can be acknowledged; it stays acknowledged while the finding persists and closes by itself when a check no longer finds it. A recurrence is a new alert.
5. Check now (operators and administrators) checks one endpoint immediately, for example right after a deployment. Edit changes the monitor (versioned: when another administrator changed it meanwhile, reload and repeat); Remove stops monitoring and removes its results and open health alerts. Disabled monitors are not checked and raise no alerts.
6. The table shows per endpoint its health (Healthy, Needs attention, Critical, Awaiting first check, Disabled), each address with TLS version, expiry and fingerprint or the reason it failed, the findings with since when they are open, the last check and who changed the monitor. The view refreshes automatically; Refresh status reloads it.

INPUTS & PARAMETERS

- Host name or IP address, port 1–65535 and an interval within Clm:EndpointHealth:MinimumIntervalMinutes–MaximumIntervalMinutes (default DefaultIntervalMinutes, 15).
- Optional expected fingerprint: 64 hexadecimal characters, colons allowed. Optional trust roots: 1 to 10 CA certificates in PEM, never private keys (refused).
- Non-public addresses (private, loopback, link-local, shared, documentation and transition ranges) are checked only inside the networks the installation operator lists in Clm:Discovery:AuthorizedPrivateNetworks; otherwise they are reported as Non-public address not authorized and never contacted.

OPERATION RESULT

Per monitored endpoint the latest result of every address and its findings, alerts that open and close with the findings, and entries endpoint-health.monitor.created, .updated and .deleted in the security journal (exported to the SIEM as CONFIG-035).

PERMISSIONS

Every organization user reads the results. Administrators add, change and remove monitors; operators and Operator API tokens run Check now. API tokens cannot manage monitors.

LIMITATIONS

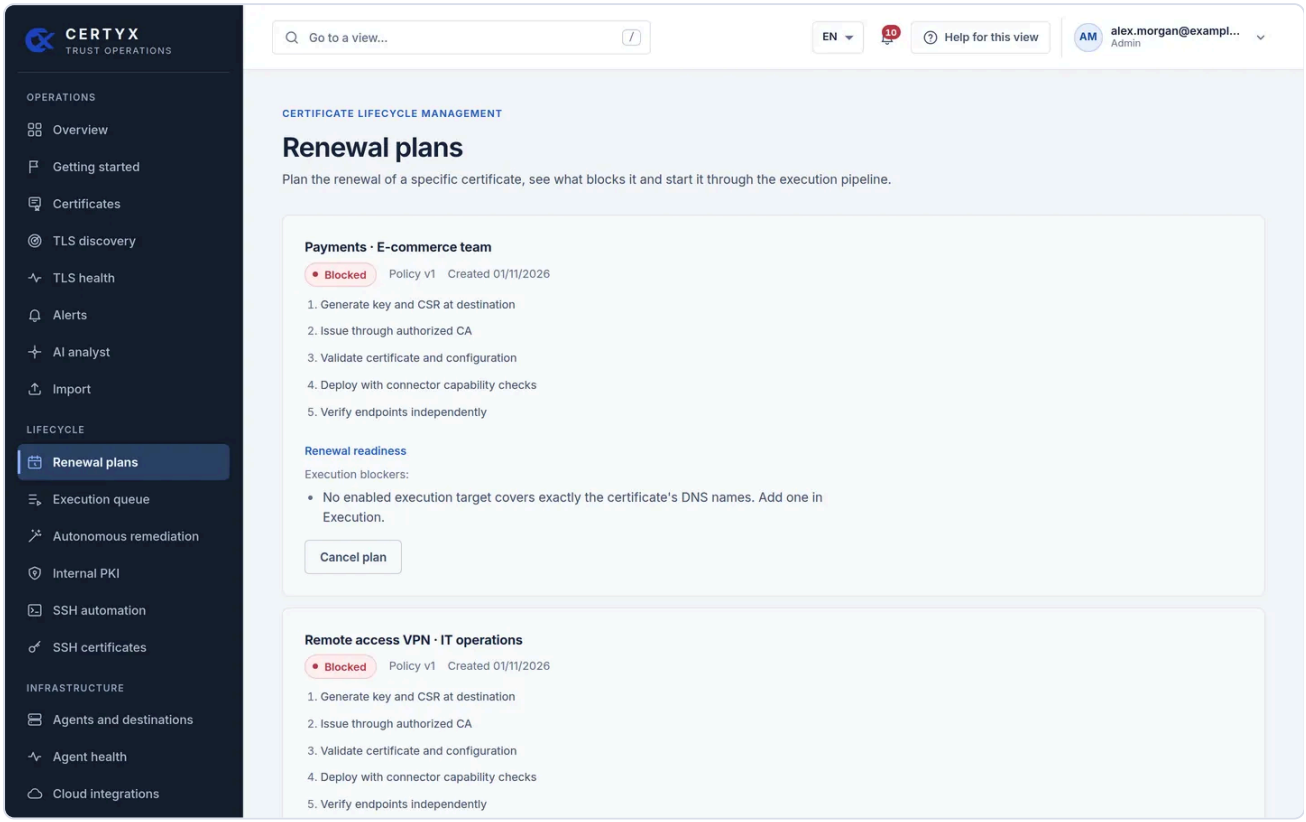
Checks run from the CERTYX control plane, so they see what the internet (or the authorized internal networks) sees; endpoints reachable only inside a customer network need an authorized network or remain not permitted. An organization keeps at most Clm:EndpointHealth:MaximumMonitors monitors. Revocation is checked online through the CRL or OCSP addresses in the certificates; without them the finding Revocation status not checkable appears. The TLS 1.0/1.1 test needs a TLS library on the CERTYX host that can still offer those versions; where it cannot, the result is left open instead of reported as safe. The checks do not send application requests after the TLS handshake.

TROUBLESHOOTING

- Non-public address not authorized: the name resolves to an internal address. Ask the installation operator to authorize that network (Clm:Discovery:AuthorizedPrivateNetworks) or monitor a public name.
- TLS endpoint unreachable: check the firewall, the port and that the service listens on every address the name resolves to; a single unreachable node behind a load balancer is reported for that address only.
- Untrusted certificate chain on a private CA: add the CA's root certificate as trust roots of the monitor. Missing intermediate: configure the server to send its full chain (leaf followed by the intermediates).
- Nodes serve different certificates: deploy the renewed certificate to the remaining nodes (Execution) and run Check now.

7 Renewal plans

Plan the renewal of a specific certificate, see what blocks it and start it through the execution pipeline.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. While automatic renewal is on for a certificate (Execution queue, policy or certificate details), it renews without a plan; plans are for certificates that renew with approval or need a renewal outside the scheduler.
2. Open certificate details and choose Plan renewal.
3. Review the five stages and the Execution blockers. They are computed each time from the current state: an enabled deployment target must hold exactly the certificate's identities (DNS names and, for client certificates, also email addresses, user principal names and URIs) and the certificate's extended key usages must permit the target's purpose; the target's agent (and verifier agent, if set) must have contacted CERTYX within Clm:Lifecycle:AgentContactWindow (10 minutes by default), and the target must name its certificate authority. Fix a blocker in Execution queue (deployment targets) or in Agents and destinations.
4. An administrator chooses Approve plan on a plan that is Ready for approval. Approval requests a renewal job for the matching target through the regular execution pipeline, under the current policy and subscription.
5. A different administrator approves the job (here with Approve job as second person, or in Execution queue); the person who requested the job sees a note that another administrator must approve it. The plan status then follows the job: Queued for the agent, Running on the agent, Outcome needs

reconciliation, Deployed and verified, Failed, Rolled back or Job cancelled.

6. After a failed, rolled-back or cancelled job, choose Approve again with a new job, or Cancel plan (after confirmation; a cancelled plan cannot be restored).
7. Each plan links to the Renewal readiness of its certificate (every step with what is missing) and to the health of the target's agent.

INPUTS & PARAMETERS

- Plans reference a certificate, owner, service and policy version.
- One active plan per certificate; repeated requests return the existing plan.
- The target is chosen automatically: among enabled targets whose identities equal the certificate's and whose purpose it permits, the one with the fewest blockers (then the lowest ID).
- Every decision carries the plan's version; if the plan changed meanwhile, the decision is refused and must be repeated after the list refreshes.

OPERATION RESULT

A versioned, audited plan linked to its execution job. A verified deployment completes the plan; a policy change, or a change of the certificate's owner or service, returns an approved plan for approval again (Ready for approval or Blocked).

PERMISSIONS

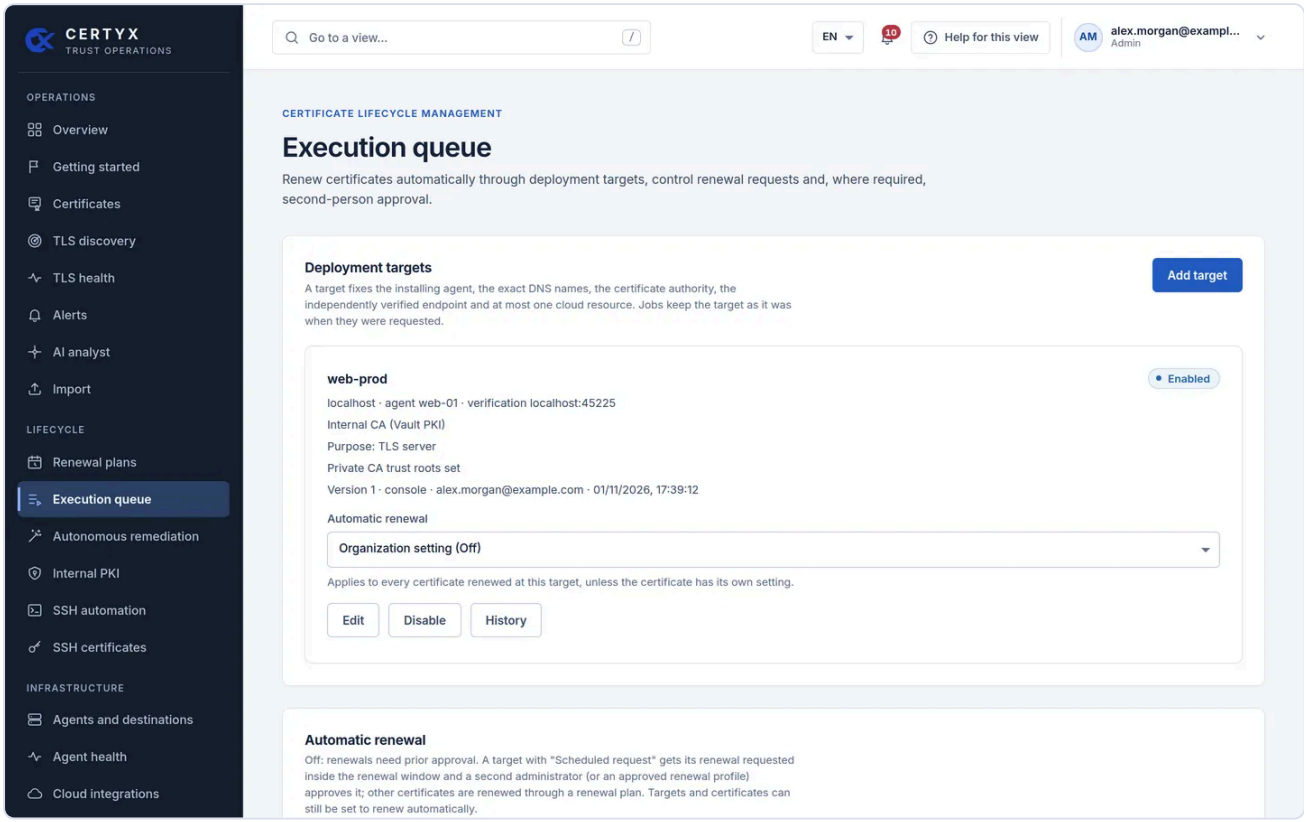
Operator or Admin: create plans (in certificate details). Admin: approve and cancel plans. A different Admin: approve the job. Auditor and Operator: read, with a note that only administrators approve or cancel plans.

LIMITATIONS

Approval never bypasses four eyes: the administrator who approved the plan cannot approve its job. Cancelling a plan cancels its job only while no agent has claimed it; a running job finishes or is reconciled in Execution queue. A pending job from an earlier policy version must be cancelled in Execution queue before the plan is approved again. Agent contact is checked from job polling and health reports; it does not prove the CA or deployment will succeed — the job result and independent endpoint verification do. On a service with subscriptions the job requires the AUTO_DEPLOYMENT feature, and the certificate becomes a managed family within the subscription's capacity; when the capacity is reached, approval is refused.

8 Execution queue

Renew certificates automatically through deployment targets, control renewal requests and, where required, second-person approval.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Under Deployment targets, an administrator adds a target or creates it from an agent draft in Agents and destinations (Add target opens the form; Certificate purpose, see below): installing agent, exact DNS names, verification host, port and health path, the certificate authority (internal Vault PKI with mount and role, or public ACME) with its HTTPS URL, optional private CA trust roots (PEM certificates only), an optional customer verifier agent and at most one cloud resource (AWS ACM certificate ARN and workload role; Azure Key Vault URL, certificate, tenant and managed identity; Google Certificate Manager project, location, certificate and service account). Every save is versioned and listed under History; Disable stops new jobs. A job keeps the complete target as it was when requested, so later edits never change pending jobs. Then choose an enabled target.
2. Installation: Agent on the host that uses the certificate (default) means the installing agent writes the files and reloads the service on its own machine. Agent as SSH relay to a host without an agent means the installing agent connects over SSH to the Remote host you enter, generates the key there (it never leaves that host), installs the certificate and reloads the service; the agent needs this destination with Remote in its configuration and listed in RelayDestinations, and it refuses jobs for any other host. SSH relay applies to TLS server certificates without a cloud resource, so the cloud choice is hidden. Without an agent on the host or a relay with SSH access, CERTYX cannot install the certificate automatically.

3. Under Request issuance or renewal, choose an enabled Deployment target. In Request type / certificate choose Issue a new certificate and enter Owner, Service and Production under New certificate ownership, or select a current (not superseded) certificate of the inventory to renew; renewal keeps its ownership.
4. Automatic renewal (panel above the request form) is on by default: every certificate that has a matching enabled deployment target renews without manual approval. Inside the policy's renewal window the scheduler requests the job, automatic renewal approves it (recorded as approved by system:auto-renewal), the agent issues and installs the certificate, CERTYX verifies the endpoint independently and the inventory is updated. An administrator chooses Switch automatic renewal off to require approval for the whole organization (Switch automatic renewal on asks for confirmation). Each target card has its own Automatic renewal choice — Organization setting, Renew automatically or Require approval — and the certificate details offer the same choice per certificate; the most specific choice wins (certificate, then target, then organization).
5. Choose Request issuance or renewal. A different administrator approves the job with Approve as second person under the current policy, or cancels it with Cancel while no agent has claimed it.
6. The agent generates the key and CSR where the certificate is used (on its own host or, as SSH relay, on the remote host), signs through Vault, ACME or DigiCert and installs with the destination's connector: pem-files, nginx, apache, haproxy, lighttpd, caddy, traefik, envoy, cribl, tomcat, java-keystore, postfix, dovecot, kubernetes-secret, windows-store, iis, or the target's cloud resource (AWS ACM, Azure Key Vault, Google Certificate Manager), or files for a client-only target. File-based connectors publish the configured files atomically with their owner and mode, run the configuration check and the reload and verify the endpoint over TLS; a failure publishes the previous generation again. CERTYX independently verifies the configured endpoint before accepting success.
7. For an expired lease (job in reconciliation), an administrator chooses Inspect the actual destination, reviews the observed evidence and then chooses Accept the verified certificate or Confirm the original installation. CERTYX rechecks the fingerprint before resolving; while a customer verifier agent has not observed the endpoint yet, the button shows that it is waiting. The requester cannot reconcile their own job.
8. Renewal automation profiles apply when automatic renewal is off: on a target with Scheduled request, the scheduler requests due renewals and they wait for approval. An administrator chooses Request 90-day automation, a different administrator chooses Approve as separate administrator, and either can choose Revoke automation; while the profile is valid the scheduled requests are approved by it. Requires the AUTO_RENEWAL subscription feature; a change of the target or of the policy invalidates the approval.
9. Unix lifecycle agents require owner-only state, secret files and execution journals without symbolic paths or symbolic parent directories. Unsafe or oversized journals stop execution for administrator reconciliation. Startup does not change permissions on an existing state directory. On Windows the agent applies the same rule to secret files and journals through ACLs: only LocalSystem, Administrators or the agent's own identity may own them and have access; anything else is refused. The Windows installer restricts the ACL of the agent directory accordingly.
10. Unix lifecycle agents can resolve approved Azure Key Vault, AWS Secrets Manager and Google Secret Manager text secrets through the customer-local provider CLI identity. Configure exact SecretBindings and run Clm.Agent secret-check with the protected lifecycle configuration as the service identity before enabling execution. Values stay local and are never printed. See CUSTOMER-VAULTS.en.md. Provider

CLI resolution runs on Unix agents only; on Windows only protected-file secrets are available. Private keys are generated and held in software on the agent host (no HSM), and the agent does not send e-mail from the customer host.

11. HashiCorp Vault KV is also supported through an exact HTTPS mount/path#field binding and a customer-local native token helper. TLS verification is required; namespaces are not supported. A read fails when access is denied, the identity is revoked or the Vault certificate is not trusted. The Vault token comes from the customer's token helper; CERTYX does not create or renew Vault identities.
12. Persisted lifecycle results are retransmitted after temporary connection failures, request timeouts and HTTP 408/429/500/502/503/504, using PollSeconds and the OperationTimeoutSeconds bound. The same job, lease and result are reused; issuance and installation are never replayed. Restart recovery renews the lease while delivering the saved result. Authentication, authorization, expired/reconciled leases and TLS trust failures stop for reconciliation. A lost lease cannot be renewed by this mechanism; an outage exceeding the lease or operation limit still requires administrator reconciliation.
13. All agent JSON configuration readers reject symbolic files or parent directories, Unix group/other permissions, files over 1 MiB and invalid UTF-8 before deserialization. Protect the configuration with chmod 600 on Unix. Relative command-line paths are resolved to an absolute path and receive the same checks. UTF-8 with or without BOM is supported. Settings and asynchronous readers use the same rules; no configuration values are included in failure output. On Windows the readers refuse a file that a principal other than LocalSystem, Administrators or the agent's own identity owns, can read or can change; provision restrictive ACLs before deployment.
14. Private endpoints can use a separate enrolled endpoint verifier. The operator binds its ID to the execution target; in Agents and destinations, under Registered agents, select Enable certificate lifecycle execution on the installer and Enable independent endpoint verification on a different agent identity. The verifier checks approved local TLS/HTTP endpoints before success is accepted. Setup takes one command on the verifier host: the target card in Execution → Deployment targets shows Set up the independent endpoint verifier with the grant, Download verifier scope (target ID, verification host, port, health path and trust roots of the target; public data only) and the command configure-verifier-linux.sh add-target (Linux, clm-endpoint-verifier.service under the agent account) or configure-verifier-windows.ps1 add-target (Windows, scheduled task CERTYX Endpoint Verifier as LOCAL SERVICE). The script writes the owner-only configuration from the agent's own identity, runs one verified cycle, starts the service and waits up to ten minutes for the grant while CERTYX answers HTTP 403; remove-target, disable and status manage it, and the uninstaller removes it. A new agent receives the role during installation (CLM_AGENT_ENDPOINT_VERIFIER on Linux, -EndpointVerifierTarget on Windows). Without trust roots in the target the verifier host must trust the issuing CA itself.
15. Enrolled installers report their own Lifecycle workload in Agent health (starting, idle or error). Missed reports remain visible even with healthy discovery. Process health does not replace the execution result and independent endpoint proof. Inspect protected local journals when a cycle stops for reconciliation.
16. Certificate purpose: each target is a TLS server (default), TLS client or TLS server-and-client target. A TLS client certificate authenticates a user, device or service to a server in mutual TLS (EKU clientAuth). Choose Internal CA (Vault PKI) with a role that sets client_flag (server_flag=false for client-only); public CAs are refused because ACME and DigiCert TLS products issue server certificates only. Enter the identities under Client identities: DNS names, email addresses, user principal names and URIs, plus an optional subject common name. A client-only target has no verification host, verifier agent or cloud resource: the agent generates the key locally, the CSR requests exactly these identities, the agent installs fullchain.pem and privkey.pem (plus bundle.p12 when its destination names a PKCS#12 password reference) and verifies the installed files locally (key, chain to its trust roots, clientAuth EKU)

before reporting. CERTYX then accepts the public leaf only when it carries exactly the job's identities and the purpose's EKUs. The agent destination must declare the same Purpose and Client identities (see AGENT-INSTALLATION).

17. From a certificate's renewal readiness, Create the target with these values opens the target form filled with the suggestion; under every filled field it says Filled from: … and under every field CERTYX could not determine it says what to enter and why. ID and purpose: those of the agent's local destination with exactly the certificate's names (the agent executes only jobs whose target ID, names and purpose equal a local destination), otherwise an ID from the first name and the purpose from the certificate; names and client identities come from the certificate. Installing agent: the agent of that destination, otherwise the only agent with the lifecycle workload, otherwise the only agent whose reported network map contains the address where the certificate was observed, when it has the lifecycle workload; otherwise empty with the reason. Verification host: it must be a name (or IP address) in the certificate, because CERTYX validates the certificate against it. Network discovery observes IP addresses, so CERTYX resolves the certificate's DNS names (wildcards skipped, bounded number and time) and fills the name that resolves to the observed address; if none does, the first name with a warning to check it; a certificate with only wildcard names leaves the host empty; an IP address only when the certificate contains it or has no DNS names. Port: where the certificate was observed (e.g. 9000); 443 only when it was never observed. Certificate authority: when an enabled target already issued a certificate from the same CA (same issuer and key identifier), its CA type, URL, Vault mount and role and trust roots; otherwise the type from the agent destination, the CA type recorded in the inventory, or the observation (an untrusted chain means an internal CA; a trusted chain means a public CA only when all names are public, not under suffixes such as .internal, .local or .home.arpa). For an internal CA without such a target, enter the Vault URL, mount and role your agent uses. Trust roots are filled only from that target or with the issuing CA's actual root certificate from the inventory; otherwise paste the private root (PEM) unless the CERTYX host or the verifier agent already trusts it. Nothing is stored before you save. Lifecycle agents report their local destinations (connector, deployment directory, files, configuration check and reload commands, cloud resource, endpoint) with their workload telemetry; secrets and secret references are never reported.
18. A failed or rolled-back job shows the agent's result and what to do next; every job links to the renewal readiness of its certificate and to the health of its agent.

INPUTS & PARAMETERS

- Deployment target: an enabled target of the organization. Targets in the server configuration CLM_EXECUTION_TARGETS (Clm:Execution:Targets) are copied into their organization once, on first use, and are then managed only here. Server file paths (trust roots file, offline revocation test aid, ACME renewal information directory) can only come from that configuration and are kept when such a target is edited. Invalid input is refused with the reason.
- Certificate: inventoried and not superseded, with exactly the target's identities (DNS names, and for client targets email addresses, user principal names and URIs) and extended key usages that permit the target's purpose; otherwise the request is refused with that reason.
- New issuance: Owner and Service required, 1–200 characters each; it reserves a managed certificate family within the subscription's capacity.
- Approval uses the job version and approver identity.
- Certificate authority Not set is accepted when saving, but a renewal plan for such a target stays blocked until the authority is chosen.

- Target formats (each field shows its own help and the browser names the expected format): IDs 1–100 letters, digits, hyphens and underscores; 1–100 distinct DNS names without wildcards; verification host as a host name or IP address without https://, path or port; health path starting with / (e.g. /healthz); CA URL starting with https://; Vault mount and role up to 120 letters, digits and . _ / -; trust roots as PEM certificate blocks up to 64 KiB; AWS arn:aws:acm:REGION:ACCOUNT:certificate/UUID with a role arn:aws:iam::SAME-ACCOUNT:role/NAME; Azure https://NAME.vault.azure.net/, certificate name of letters, digits and hyphens, tenant and client ID as GUIDs; Google project ID, location global or a region such as europe-west1, certificate ID and NAME@PROJECT.iam.gserviceaccount.com.
- Key/deployment roots must be absolute and link-free. Private keys and recovery journals require owner-only permissions; linked/shared/oversized state is refused. Only the managed current-generation UUID link is allowed.
- Client identity formats (each field shows its help; the server names the expected format when a value is refused): email jane.doe@example.com (ASCII, one @, DNS domain, at most 254 characters); user principal name jdoe@corp.example.com (sign-in name, @, directory DNS suffix, not CORP\jdoe); URI with a scheme such as spiffe://example.org/ns/prod/sa/payments or urn:uuid:..., no spaces or user:password; common name 1–64 characters, required when only URIs are entered. One value per line; at most 100 identities.
- Remote host (SSH relay): a DNS name or IP address, at most 253 characters, exactly as Remote.Host in the relay agent's configuration, e.g. web-01.internal.example.

OPERATION RESULT

Durable execution result, returned public certificate in inventory and audit evidence. The replaced certificate is retired.

PERMISSIONS

Admin: create, edit, disable and enable targets, cancel unclaimed jobs, reconcile jobs and request or revoke automation profiles. Operator or Admin: request jobs (on a service with subscriptions only with the AUTO_DEPLOYMENT feature). A different Admin than the requester: approve jobs and automation profiles; the requester of a job cannot reconcile it. Auditor (and Operator): read targets, profiles and jobs; for an Auditor the request form is read-only. Every role sees the History button of a target and its change history (version, change, account e-mail, time); the Edit and Disable/Enable buttons are shown to administrators only. Admin: switch automatic renewal on or off for the organization and choose it per target and certificate; every role sees the setting.

LIMITATIONS

The renewal scheduler checks the enabled targets of every organization every Clm:Lifecycle:SchedulerInterval (5 minutes by default). Automatic renewal applies to the newest certificate holding exactly a target's identities; it needs the AUTO_RENEWAL subscription feature (without it every renewal waits for approval) and starts short-lived certificates only in the last third of their lifetime. After a failed, rolled-back or cancelled job at a target, the next renewal waits for approval; after it succeeds, renewals run automatically again. At most Clm:Lifecycle:AutomaticRenewalsPerDay (200 by default) renewals per organization are approved automatically within 24 hours; further due renewals wait for approval. Before the agent receives an automatically approved job, CERTYX checks it again: a changed target cancels it, and a switched-off setting returns it to approval. Manually requested jobs and jobs from renewal plans always need a second administrator. The installation default is Clm:Lifecycle:AutomaticRenewalByDefault (true). On a shared multi-organization service set Clm:Execution:AllowPrivateVerificationEndpoints to false: loopback, private and link-local verification hosts

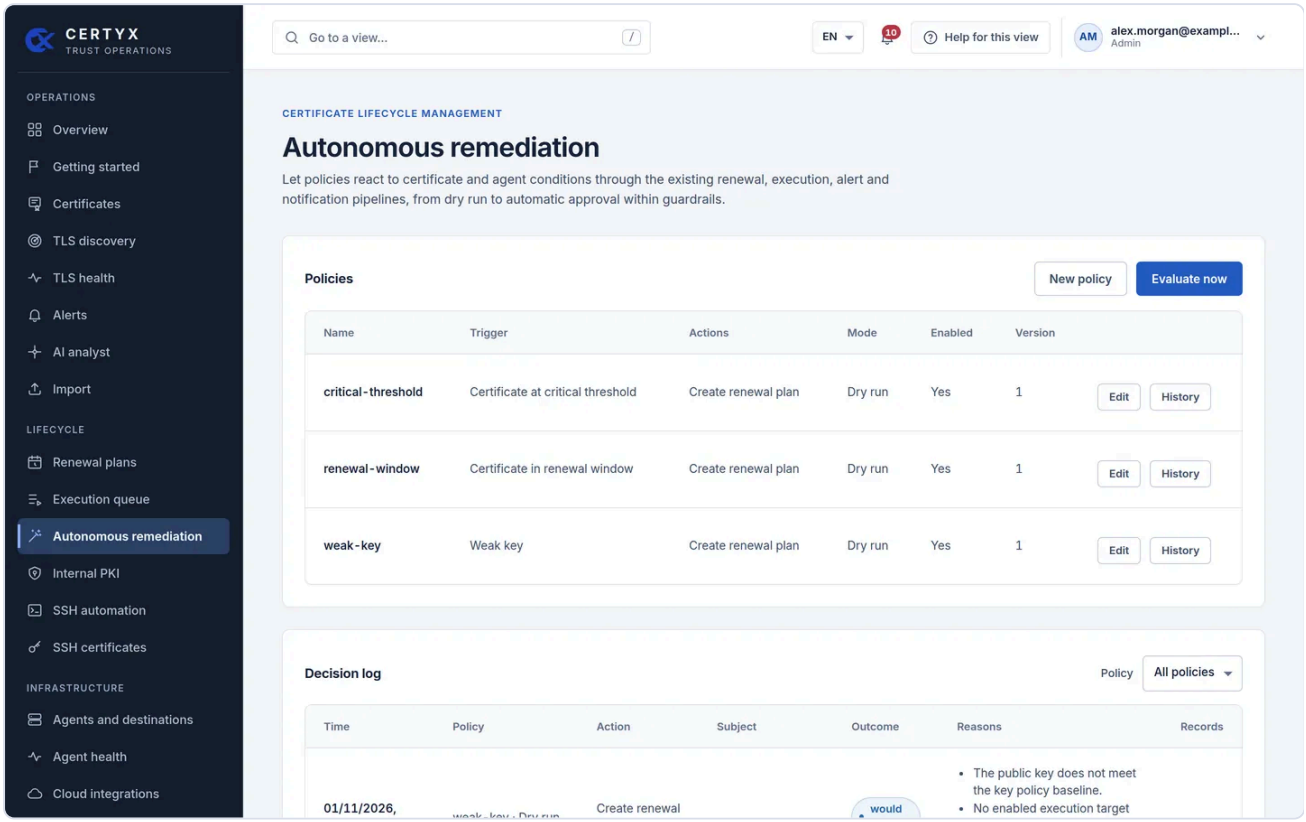
then need a customer verifier agent, and the control plane refuses to verify a host that resolves to such an address. Agent enrollment uses single-use codes in Agents; the installed agent's local configuration (CA credentials, deployment paths) stays on the customer host. Verification of loopback, private or link-local endpoints by CERTYX itself needs the platform operator's permission for your organization; otherwise assign a verifier agent. TLS client certificates are issued only through Vault PKI roles; ACME, DigiCert and cloud TLS stores are refused with a specific reason. Client certificates are delivered as PEM files and an optional PKCS#12 bundle on Unix agents; the Windows certificate store, browser enrolment and smart-card provisioning are not supported. Their verification is the installing agent's local check, not an independent endpoint observation, and a client-only job in reconciliation cannot be resolved by observation: the agent delivers a recorded result when it runs again, otherwise the operator checks the deployment directory on the agent host. The SSH relay installs TLS server certificates only (no client certificates, no Windows store) with an RSA-3072 key generated on the remote host; one relay should own a remote destination. Mail services are verified on implicit TLS ports (465, 993, 995); STARTTLS is not spoken.

TROUBLESHOOTING

- Readiness says the agent has no local destination with the target ID: add a destination with exactly this ID, the same DNS names and purpose to the agent configuration (agent.json, Destinations), restart the agent and wait for its next report.
- Verification fails with a name mismatch after creating a target from the readiness: the verification host must be a name (or IP address) in the certificate. Replace an IP address with the certificate's DNS name that reaches the endpoint; if the hint says the names did not resolve to the observed address, check DNS on the CERTYX host or enter the name the endpoint serves.
- The agent's destination lists other names, another purpose or another kind of CA: make the target and the destination identical; internal targets need a Vault destination, public targets ACME or DigiCert.
- The agent has not reported its destinations: it reports them with each lifecycle workload report once it runs with an enrolled identity; check its diagnosis in Agent health.
- Rolled back: the agent installed the new certificate, but the endpoint check failed and the previous generation was restored and verified; check the reload command, the served host and port and the trust roots, then approve the plan again.
- Reconcile: the lease expired before a result; inspect the actual destination here and accept the verified certificate or confirm the original installation.
- Readiness says the agent installs in another mode: the target says SSH relay but the agent destination has no Remote (or the other way round). Make both the same: choose the installation in the target, add or remove Remote and RelayDestinations in the agent configuration and wait for its next report.
- Readiness says the relay deploys to another host: set the target's remote host to exactly Remote.Host of the agent destination. Run Clm.Agent relay-check on the relay to test the host key pin, SSH login, helper, sudo rule and configuration check without changing a certificate.

9 Autonomous remediation

Let policies react to certificate and agent conditions through the existing renewal, execution, alert and notification pipelines, from dry run to automatic approval within guardrails.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Requires the AUTONOMOUS_REMEDIATION subscription feature. An Admin chooses New policy and enters an identifier (1–100 letters, digits, hyphens or underscores; it cannot be changed later), a name (1–100 characters), Enabled, the trigger and the actions it supports, the mode, optionally up to 50 comma-separated services the policy is limited to (each up to 200 characters; empty means every service), the owner to assign (1–200 characters, required for Assign owner) and the responsibility group to notify (required for Notify group).
2. Triggers: certificate in the renewal window, at the critical threshold, weak key, weak signature (each also for expired certificates where it applies), certificate replaced outside CERTYX (open discovery alert) and agent offline (offline, in error or identity expired).
3. Preview (dry run) evaluates the draft against the current data and shows what it would do, without changing or recording anything. Save policy stores a new version; Edit opens an existing policy; History lists every version with who changed it.
4. Modes: Dry run only records what would happen. Propose creates renewal plans and requests execution jobs that still need the normal four-eyes approval, sends group notifications and records acknowledgements and owner changes as recommendations. Automatic approval also approves plans and jobs within the guardrails, in the name of the Admin who saved the policy after acknowledging the

warning. Saving in automatic mode requires ticking I understand and enable automatic approval for this policy, at least one change window and, when the policy creates renewal plans or requests jobs, at least one eligible target.

5. The evaluator runs every Clm:Remediation:Interval (default 5 minutes) for every active, licensed organization, on one application instance at a time, unless the server operator set Clm:Remediation:WorkerEnabled to false; Evaluate now runs it immediately. The decision log lists each decision with its outcome (would, proposed, executed, skipped, refused, failed), translated reasons and links to the target, plan and job; Explain opens the AI analyst when it is enabled.

INPUTS & PARAMETERS

- Actions per trigger: certificate triggers support acknowledge alert, assign owner, notify group, create renewal plan and request execution job; replaced outside CERTYX supports acknowledge alert and notify group; agent offline supports notify group.
- Assign owner changes only certificates without an owner (empty or "unassigned"); a certificate that already has an owner is skipped with that reason.
- Guardrails of automatic approval: eligible targets (up to 200 execution targets of the organization), maximum executed actions in 24 hours (1 to Clm:Remediation:MaxActionsPerDayLimit, default 500), up to 20 change windows in UTC (at least one day each; start hour 0–23 before end hour 1–24) and whether production certificates are allowed. Renewals are approved only for an eligible target that has an independent verification endpoint.
- The production flag and the owner come from the certificate metadata in the inventory; notifications go to responsibility groups with members through the organization's SMTP relay.

OPERATION RESULT

Plans and jobs created as system:remediation:<policy>; automatically approved jobs carry the acknowledging Admin as approver and run through the normal execution pipeline, whose verification rolls a failed deployment back. Every decision is stored in the decision log (kept Clm:Remediation:DecisionRetentionDays, default 400 days) and in the security journal; an identical decision is recorded at most once per Clm:Remediation:DecisionRepeatWindow (default 24 hours). Policy changes are versioned and journaled.

PERMISSIONS

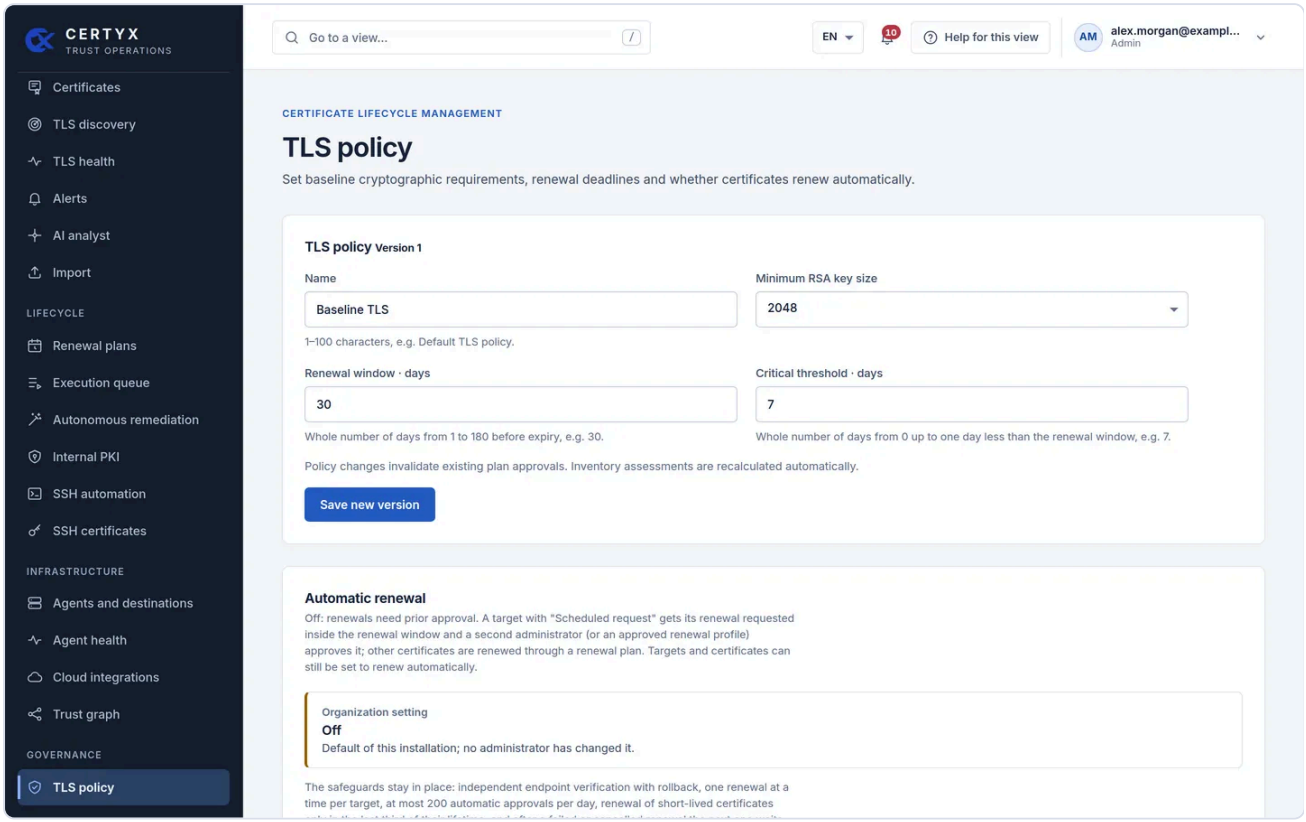
Read policies, history and the decision log: every role of the organization. Create, change, preview and evaluate: Admin.

LIMITATIONS

Revocation is not available: no CERTYX pipeline revokes TLS certificates, so a policy requesting it is refused. Propose mode does not change owners or alerts. A renewal only proceeds when the plan has no blockers (target holding exactly the certificate's identities with a matching purpose, reachable agent, configured certificate authority). Without the subscription feature the view states that the subscription does not include autonomous remediation and no policy is evaluated. Automatic approval does not re-check the acknowledging Admin's role at approval time; editing a policy in automatic mode requires a new acknowledgement by the editing Admin. At most Clm:Remediation:MaxPoliciesPerOrganization policies (default 50) and Clm:Remediation:MaxSubjectsPerPolicy subjects per policy and pass (default 200). The analyst's explanation needs the AI analyst to be enabled.

10 TLS policy

Set baseline cryptographic requirements, renewal deadlines and whether certificates renew automatically.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Review the current version shown next to the title.
2. Enter "Name", "Minimum RSA key size", "Renewal window · days" and "Critical threshold · days".
3. Choose "Save new version". Approved renewal plans return to planned and need approval again; review the reassessed certificates.
4. Automatic renewal shows the organization setting: On (the default) renews every certificate with a matching deployment target inside the renewal window without manual approval; Off requires a second administrator's approval for every renewal. Administrators choose Switch automatic renewal off or Switch automatic renewal on; targets and certificates can override it in Execution queue and in the certificate details.

INPUTS & PARAMETERS

- Minimum RSA key size: 2,048, 3,072 or 4,096 bits.
- Renewal window: 1–180 days. Critical threshold: 0–179 days, below the renewal window.
- Name: required, 1–100 characters.
- Default policy before the first save: "Baseline TLS", RSA 2,048 bits, renewal window 30 days, critical threshold 7 days.

OPERATION RESULT

A new policy version and audit entry. Assessments are recalculated on read; plan and job approvals are protected by version checks.

PERMISSIONS

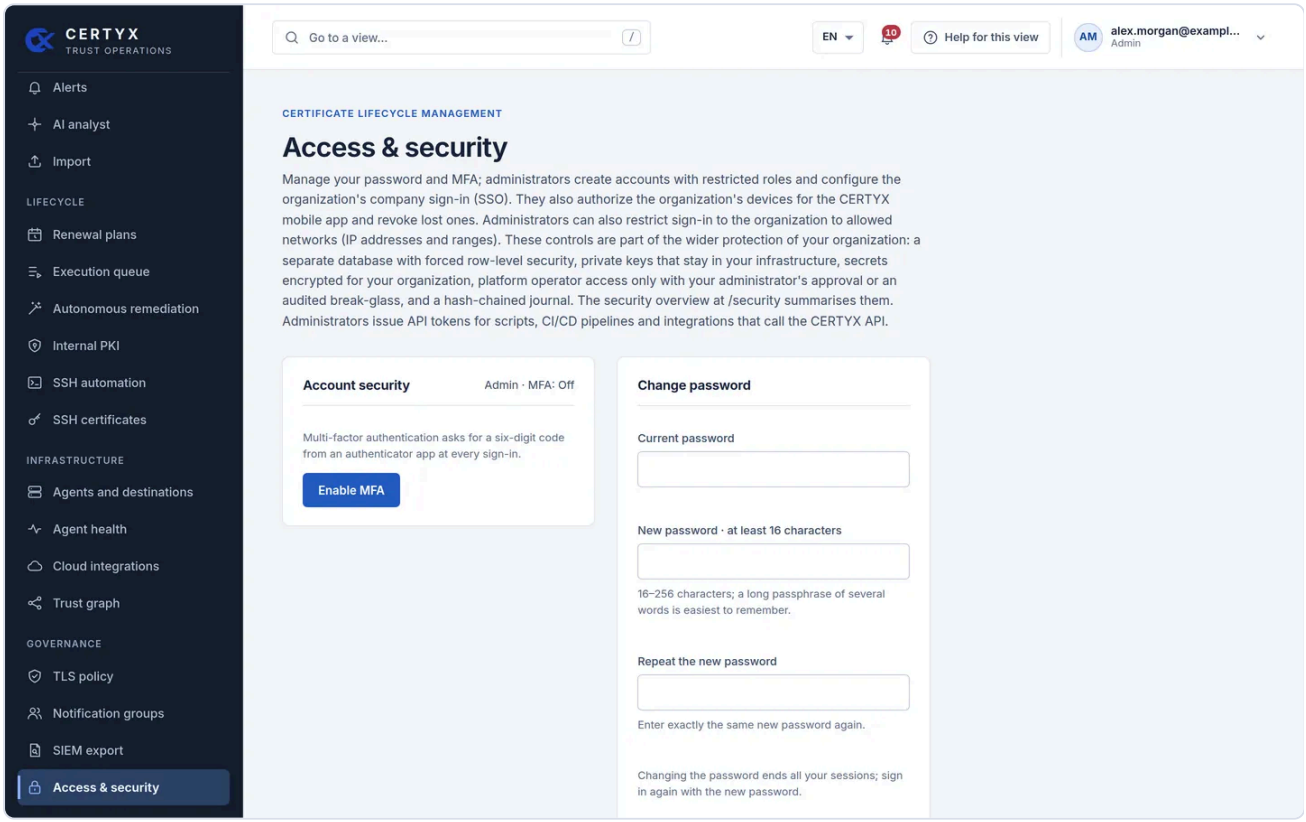
Read: all roles. Write (policy and automatic renewal): Admin.

LIMITATIONS

This is a baseline validity and public-key policy, not a complete organizational rule engine. Fixed baseline rules are not editable: ECDSA keys below 256 bits, algorithms other than RSA and ECDSA, and SHA-1 or MD5 signatures are always critical. If someone saved another version in the meantime, the save is refused and asks you to reload. Changing it does not modify issued certificates.

11 Access & security

Manage your password and MFA; administrators create accounts with restricted roles and configure the organization's company sign-in (SSO). They also authorize the organization's devices for the CERTYX mobile app and revoke lost ones. Administrators can also restrict sign-in to the organization to allowed networks (IP addresses and ranges). These controls are part of the wider protection of your organization: a separate database with forced row-level security, private keys that stay in your infrastructure, secrets encrypted for your organization, platform operator access only with your administrator's approval or an audited break-glass, and a hash-chained journal. The security overview at /security summarises them. Administrators issue API tokens for scripts, CI/CD pipelines and integrations that call the CERTYX API.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Account menu: select your avatar and name at the top right. Profile shows your login, e-mail address, role, organization, sign-in method and MFA state; it changes your password (enter the new one twice) and your own e-mail address, which is also your sign-in name – confirm with your current password and, with MFA, an MFA or recovery code. Both changes end all your sessions. Sign out ends the current session.
2. Enable MFA, then scan the QR code with an authenticator app or enter the key by hand. Check the entry name: it is CERTYX followed by this server's host in brackets, for example CERTYX (certyx.example.com). An older entry named CLM, CERTYX alone or another host belongs to another

environment or an earlier setup, and its codes are refused here.

3. Confirm with the current six-digit code. Ten recovery codes are shown once: copy or download them and store them offline, then sign in again.
4. If you lose the authenticator, enter one recovery code instead of the MFA code at sign-in. Each code works once; Recovery codes shows how many remain. Generate new recovery codes (password and current MFA code) replaces all earlier codes.
5. An administrator who signed in with MFA can use Reset MFA for another user of the organization after confirming that person's identity outside the system. The user's sessions end; they sign in with the password and enable MFA again.
6. The Change password panel of this view changes your password as well: enter the current password and the new one twice, then Change & sign out ends all your sessions. Repeated failed sign-ins lock the account for 15 minutes, and each further failure after a lock has ended doubles it (up to 24 hours by default; the operator sets the values). A session ends after 30 minutes, or after 15 minutes without activity (defaults). In production, administrator functions require a session signed in with MFA: until an administrator enables MFA and signs in again with a code, administrator views answer that MFA is required, while the profile and MFA enrollment stay available.
7. Users & roles (administrator): Create user takes the e-mail address, username, initial password and role. The user list shows each user's e-mail address, role and MFA state; Change e-mail address sets the address the user signs in with (unique in the installation) and ends that user's sessions.
8. Company sign-in (administrator signed in with MFA): choose the provider – Microsoft Entra ID, Okta, another OpenID Connect provider or SAML 2.0. For OpenID Connect, register the redirect address shown at the provider and enter the issuer address, client ID and secret. For SAML, import the service provider metadata at the provider (or enter the entity ID and assertion consumer address), then enter the provider's SAML sign-in URL, entity ID and signing certificate in CLM.
9. SAML options: in Service provider key (SAML) generate a key; its SHA-256 fingerprint and validity are shown, the private key never leaves the server. Re-import the metadata at the provider (it now carries the signing and encryption certificate and the single logout service), then turn on Sign authentication requests and Require encrypted assertions as the provider supports them, and enter the provider's single logout URL for single logout. To rotate, choose Prepare next key, let the provider import the metadata again, then Activate next key; the former key keeps decrypting during the overlap period (30 days by default) and Discard next key drops a prepared key that was not activated. Signing out of the console then also signs out at the provider, and a signed logout request from the provider ends the user's CERTYX sessions.
10. Map provider groups to Admin, Operator or Auditor and save. Add an e-mail domain, publish the _certyx-verification TXT record shown and choose Verify. From then on users of that domain see the company sign-in button on the sign-in page. The user name the provider asserts (preferred_username, email or upn) must belong to a verified domain or one of its subdomains once the organization has verified a domain (before that, names are accepted and journaled as unverified): a name of another domain is refused (the sign-in page says the identity provider named a user whose e-mail domain is not verified; journal reason sso-unverified-domain), unless the administrator selects Accept user names outside the verified domains, in which case the sign-in is journaled with the user marked unverified. Names without a domain (a login name) are accepted.
11. Optionally generate a SCIM token and enter it with the SCIM address in the provider's provisioning settings: then only users provisioned and active in the directory can sign in, and deactivating a user ends their sessions. The provider can also provision groups (SCIM Groups); Provisioned groups lists

them with their members and mapped role. Map a group by its name or external ID like a token group. Company sign-in only blocks the organization's local passwords. A SCIM group change that raises a user's role is journaled as its own entry, identity.scim.role-raised, with the group, its mapped role and each user's role before and after.

12. Company sign-in (Microsoft Entra ID, Okta, other OpenID Connect, SAML 2.0) is configured by an administrator in the Company sign-in (SSO) panel of this view. Users then choose Sign in with a company account on the sign-in page; it starts as soon as their e-mail belongs to a verified domain.
13. Mobile devices (administrator): choose a user (or any user of the organization) and an optional label, then Generate activation code. The code and its QR code are shown once and work once within the configured lifetime (Clm:Mobile:EnrollmentCodeLifetime, 15 minutes by default). When the server's certificate comes from the organization's own (private) certificate authority, the operator configures its SHA-256 fingerprint (Clm:Mobile:TrustedServerCertificates or Clm:Mobile:TrustedServerCertificateFile); the QR code then carries it, the panel shows it under the code, and the app trusts that certificate authority for this server only, so scan the QR code instead of typing the code. On the device, install CERTYX from the organization's app store (Custom App through Apple Business Manager or a private app in managed Google Play), scan the QR code or enter the code, then sign in with the user's account (e-mail, password and, when the account has MFA, its code) or with company sign-in; the app offers company sign-in only when the organization has it set up with a verified domain. In the app the user sees the overview, certificates, alerts, approvals, agent health and SSH certificates; alerts are acknowledged and approvals granted with the same roles as in the console. Authorized devices shows the label given with the code together with the name and model the phone reported. Codes waiting for a device can be cancelled. Revoke access blocks a device immediately: the app is refused on its next request, before sign-in, and must be activated again with a new code. Remove this device in the app's settings revokes the device on the server too (the list then shows Removed on the device); when the phone has no connection, the app deletes its data anyway and tells the user that an administrator must revoke the device here.
14. API tokens (administrator signed in with MFA): enter a name, choose the role – Auditor reads, Operator also requests renewals, deployments and imports – and the lifetime in days, then Create token. The token (cx_...) is shown once: copy it into your secret manager and send it as the header Authorization: Bearer <token> to the CERTYX API (/api/...); no cookie or CSRF token is needed. The list shows the role, who created the token and when, its expiry, last use and state (active, expired, revoked); Revoke blocks it immediately.
15. Authenticator setup e-mail (administrator): Create user also asks for the e-mail language (English, Deutsch or Polski; the console language is preselected). The new user then receives an e-mail through the organization's SMTP relay (Notification groups → Delivery) with a single-use link to set up an authenticator app. The e-mail contains the link as text and as a QR code that opens it on a phone, never the authenticator secret or the password; give the initial password separately. Without a switched-on relay, or when sending fails, the panel shows why and the link to copy: pass it on through a trusted channel. The user list shows "Setup link valid until ..." while a link is pending; Send setup e-mail issues a new link (the previous one stops working) for an enabled user with an e-mail address and without MFA.
16. Setup page: the person opens the link, enters the account password, scans the QR code (or types the key; the entry is named CERTYX with this server's host in brackets), confirms with the current six-digit code and keeps the ten recovery codes shown once; their sessions end and every later sign-in asks for the code.

17. Allowed sign-in networks (administrator signed in with MFA): enter the addresses and ranges your users connect from, one per line – a single address (203.0.113.7, 2001:db8::7) or a CIDR range starting at its first address (203.0.113.0/24, 2001:db8::/48) – and choose Save sign-in networks. The panel shows the address you are connected from and whether the list includes it. If it does not, saving is refused with a warning: make sure you can connect from an allowed network, tick the confirmation, enter your current authenticator code and save again. From then on every password and company sign-in, every device activation and sign-in in the mobile app and every request of a browser session is checked against the list; a session from an address outside it ends on its next request. Empty the list and save to remove the restriction.
18. Mobile app and the sign-in networks: by default (Also check every request of signed-in mobile apps off) activating a device and every sign-in in the app must come from an allowed network, and an app that is already signed in keeps working from any network until its session ends (Clm:Mobile:SessionLifetime, 30 minutes by default) – for example on mobile data. Turn the option on to check every app request too; the app then works only on allowed networks, for example through your VPN.
19. Requests from the platform operator (Admin signed in with MFA): CERTYX platform operators cannot create accounts, raise roles, re-enable accounts or reset MFA in your organization on their own while it has an enabled Admin. Such a change is listed here with the requested change, the time and the state until an Admin approves it (it is carried out at once; when the platform relay could not send a new account's setup e-mail, its one-time link is shown to you) or rejects it; it expires after 72 hours by default. A break-glass by two different operators is shown with its reason; every Admin receives an e-mail and granted access ends after 24 hours by default unless you change the account. Platform operator activity lists the operators' latest changes to your organization from its security journal (also sent to your SIEM destinations).
20. Passkeys (local accounts with MFA enabled): enter a name and your current authenticator code, choose Add passkey and follow your browser's prompt (device, phone or security key). Afterwards Sign in with a passkey on the sign-in page takes your password and the passkey instead of the code, and every code field and the control room's confirmation dialog offer Confirm with a passkey. The policy chooses what the second factor may be: authenticator code or passkey (default), passkey only (codes are refused; needs a passkey) or authenticator code only. Recovery codes work with every policy. Remove passkey takes the passkey and a code; removing the last one returns the policy to code or passkey.
21. Your sessions: every browser and mobile app session of your account with client, start, last activity, address and state, the current one marked. End session signs that session out on its next request; End all other sessions keeps only this one. Organization administrators also see Sessions in the organization (every user, with the device label of a mobile session) and can end any of them; each ending is recorded in the security journal.

INPUTS & PARAMETERS

- TOTP: 6 digits, SHA-1, 30-second period; confirm the enrollment within 10 minutes, otherwise start Enable MFA again.
- Recovery code: 16 characters in four groups (letters and digits); case, spaces and dashes are ignored.
- Password: 16–256 characters. Username: up to 120 characters.
- Auditor reads; Operator manages inventory and plans; Admin manages policies and approvals.
- Entra: issuer https://login.microsoftonline.com/<directory ID>/v2.0 (one directory). OpenID Connect: an HTTPS address without a query. SAML: the HTTPS sign-in address (HTTP-Redirect binding), the provider's entity ID and its signing certificate in PEM or base64 (not expired, RSA or ECDSA key).

- SAML single logout URL: the provider's HTTPS SingleLogoutService address (HTTP-Redirect). SCIM groups: display name unique per organization (up to 256 characters), members must be users already provisioned over SCIM; filters displayName, externalId or id eq "value".
- Group mapping: one line "group = role", up to 100 lines. Domains: up to 20 per organization; a domain verified by another organization cannot be claimed.
- Mobile activation code: 16 characters in four groups; one use; a user binding limits sign-in on that device to the chosen user.
- Setup link: valid for Clm:Identity:MfaSetupLinkLifetime (default 24 hours), works once, needs the account password; a newer link, a password change or enabled MFA invalidates it. A new link for the same user at most once per Clm:Identity:MfaSetupResendInterval (default 1 minute).
- Allowed sign-in networks: IPv4 in dotted decimal (four parts, no leading zeros) or IPv6, optionally with a prefix length (/0–/32, /0–/128); the address must be the first of its range (203.0.113.0/24, not 203.0.113.5/24); an IPv4-mapped IPv6 entry (::ffff:203.0.113.0/120) is stored as its IPv4 range. Entries may also be separated by commas or spaces; duplicates are dropped. At most Clm:Identity:MaximumSignInNetworks entries (100 by default).
- Company sign-in rules: Accept user names outside the verified domains is off by default (refuse); turning it on accepts such names and journals them as unverified.
- Passkey: a name (up to 64 characters) and the current authenticator code; the policy: authenticator code or passkey, passkey only, or authenticator code only.
- API token: name up to 80 characters; role Auditor or Operator; lifetime 1 day up to Clm:Identity:ApiTokens:MaximumLifetimeDays (365 by default, 90 offered); at most Clm:Identity:ApiTokens:MaximumPerOrganization active tokens (25 by default).

OPERATION RESULT

Enabling MFA, a password change and an MFA reset end the account's sessions. Recovery codes are stored only as hashes; using, regenerating and resetting are recorded in the security journal. Roles are enforced by the API, independently of button visibility. Company sign-in creates a session with the role of the most privileged mapped group; it is refused without a mapped group, without MFA when required, from another Entra directory, with too many groups for the token, and with SCIM for a user who is not provisioned or is deactivated. Every settings change ends existing company sessions. A SAML response is accepted only when signed by the pinned certificate, for the right audience and issuer, within its validity and as the answer to a request started in the same browser; reusing the same response is refused. With SCIM groups, the role is the most privileged one mapped from the token groups and the SCIM groups together; a membership change, rename or deletion that changes a user's effective role ends that user's sessions at the next request, and every change is journaled with the users whose role changed. With Require encrypted assertions an unencrypted assertion is refused; encrypted assertions must use AES-GCM or AES-CBC with RSA-OAEP key transport (RSA PKCS#1 v1.5 is refused). With Sign authentication requests every AuthnRequest is signed with RSA-SHA256. Single logout: console sign-out ends the session on every instance at once, then sends a signed logout request to the provider; a signed, fresh logout request from the provider (checked against the pinned certificate, issuer and destination, refused when replayed) ends the subject's sessions on every instance and is answered with a signed logout response. Mobile devices: every app request, including sign-in, is signed with a key that stays on the device and is checked against the authorized device, so revocation applies at once; codes are stored only as hashes, and code creation, device authorization and revocation are recorded in the security journal. Sending, using and an expired link are recorded in the security journal (mfa-setup.mail.sent, .used, .expired) without the link and reach the organization's SIEM export. Company sign-in (SSO and SCIM) users get no setup e-mail;

with Company sign-in only none is sent, because MFA is enforced by the identity provider. Allowed sign-in networks: every change (before and after, actor) and every refused change is recorded in the security journal (identity.signin-networks.changed, .refused); every refused sign-in, device activation or app request from outside the list is a sign-in event with the reason network-not-allowed, the source address and, where known, the user, in the organization's journal and SIEM export, and a session ended that way is a sign-out with that reason. The sign-in page and the app tell the user that their network is not allowed, without revealing the list. API tokens are stored only as SHA-256 hashes; creation, revocation and refused uses (for example from outside the sign-in networks) are recorded in the security journal and exported to the SIEM like other identity events, and the last use is shown in the list.

PERMISSIONS

Own account: signed-in local user. Account creation and MFA reset: Admin; the reset also requires the administrator's own MFA sign-in and cannot target their own account. Platform administrators reset MFA of customer accounts in Platform administration. SSO accounts manage password/MFA at the identity provider. Company sign-in, domains and the SCIM token: Admin signed in with MFA; they apply only to that administrator's organization. Service provider keys (generate, activate, discard) also require an Admin signed in with MFA; the list of SCIM groups is read-only for Admins. Changing your own password and e-mail: every signed-in local user for their own account; a failed confirmation counts toward account lockout. Mobile devices: organization Admin; generating codes and revoking devices require the administrator's own MFA sign-in. The app is for customer users only; platform operators cannot sign in from it. Allowed sign-in networks: organization Admin; saving requires the administrator's own MFA sign-in, and a list excluding the administrator's current address also a current authenticator code of a local account (administrators signed in through company sign-in must keep their own address in the list). Requests from the platform operator: Admins of this organization only; approving and rejecting require the Admin's own MFA sign-in. Passkeys and the own session list: every signed-in local user with MFA for their own account (sessions also for company sign-in users). Sessions in the organization: organization Admin. API tokens: creating and revoking them requires an Admin signed in with MFA in the browser; a token itself acts only as Auditor or Operator in its organization and is refused by every administrator endpoint (approvals, settings, users, devices, tokens), so four-eyes approvals stay with people.

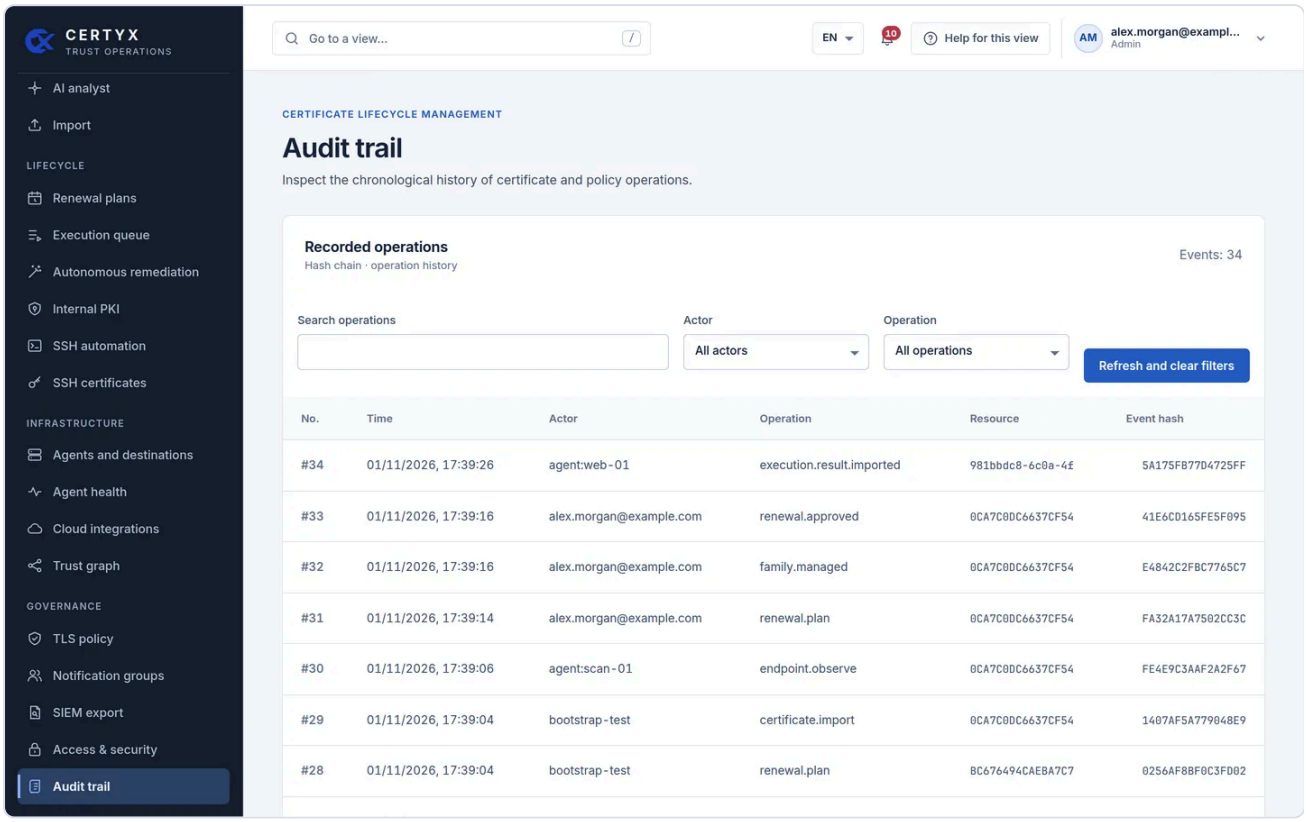
LIMITATIONS

No self-registration. A console session ends after Clm:Identity:SessionLifetime (default 30 minutes) regardless of activity; a mobile app session likewise after Clm:Mobile:SessionLifetime (default 30 minutes); then sign in again. Sessions also end through sign-out, password, e-mail or MFA changes, an MFA reset or revocation of a mobile device. The session lists need the platform database; without it they say sessions are not recorded. Passkeys need the installation's relying party (Clm:Identity:Passkeys) and a browser with WebAuthn on the console's https address; a passkey works only for the address it was registered on. Mobile sessions started before this version are not listed. An administrator who lost both the authenticator and all recovery codes needs another administrator of the organization; a platform administrator in that situation is recovered offline by two people with tools/Clm.OperatorBootstrap reset-mfa while the service is stopped. Accounts created before recovery codes existed get them through Generate new recovery codes. With the platform database, accounts, MFA state and data-protection keys are shared by all application instances; without it they stay in the local data directory of a single instance. Never share your TOTP secret, recovery codes or password. Customer credentials and private keys remain customer-controlled. Agents read customer secrets from protected files, HashiCorp Vault KV, Azure Key Vault, AWS Secrets Manager and Google Secret Manager; other vault products are not supported, and agents do not send e-mail from the customer host. On Windows, protected files are accepted only when LocalSystem, Administrators or the agent's own identity own and access them. Unix Vault token permissions are

rechecked on each use. Company sign-in: the platform operators' organization always keeps local break-glass sign-in and cannot enable Company sign-in only. SAML: paste the provider's new signing certificate before it rotates its key. Signed requests, required encryption and single logout need a service provider key, and single logout also the provider's logout URL; a prepared next key must be imported at the provider before activation, otherwise it refuses signed requests. If the provider does not confirm a console sign-out, the CERTYX session is still ended and the sign-in page says so. Nested SCIM groups are not supported (members must be users), and SCIM bulk operations are not offered. SAML requires HTTPS because the request cookie is marked Secure. Entra group overage is refused – limit the token to groups assigned to the application. Mobile devices need the platform database; without it the panel says so. Push notifications are sent only when the service operator has configured Firebase Cloud Messaging; they never contain certificate names. Allowed sign-in networks need the platform database; without it the panel says so and sign-in is not restricted. The client address is the connection's address, or with Clm:Ingress:TrustedProxies the address the trusted proxy forwards (X-Forwarded-For from any other peer is ignored); users behind a shared NAT or proxy share its public address. Agents (mutual TLS) and SCIM provisioning tokens are not user sign-in and are not checked. If every administrator is locked out, a platform operator can remove the restriction (customer card → Access); that removal appears in this organization's security journal and SIEM export. A request can only be decided while it is pending and at the version shown; an approved account change is not carried out when the account changed after the request, the name is taken or the active-account limit of the contract is reached (the request then shows Not carried out). Platform operator activity shows the newest 50 entries without before/after details; reads by operators are not listed. MFA is enforced by our provider policy makes every company sign-in count as MFA without the provider reporting it: the panel shows a warning while it is selected, and every save with it is journaled as identity.sso.mfa-delegated. Creating an account or assigning an e-mail address that is already used, in this or any other organization, is refused with the same message; only the platform journal records the real reason (account.create-refused, account.email-refused), so administrators cannot find out whether an address exists elsewhere. API tokens need the platform database; without it the panel says they are unavailable. Every token expires and cannot be extended – create a new one and revoke the old one. A token is subject to the organization's sign-in network list (403 network-not-allowed outside it). Requests with an unknown, malformed, revoked or expired token answer 401 invalid-token and never fall back to a browser session.

12 Audit trail

Inspect the chronological history of certificate and policy operations.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Open the workspace after sign-in.
2. Search operations, filter by actor or operation and page through the list, 50 operations per page, newest first; paging stays on the same snapshot until Refresh and clear filters loads the latest operations. Review sequence, timestamp, actor, operation and resource identifier.
3. Compare events with plans, policy changes or endpoint observations.
4. Export CSV or Export JSON downloads every operation that matches the current search and filters, newest first, with the hash chain links; each export is recorded in the security journal. Export and retention shows the retention that applies, the number and time range of recorded operations and the export limit.
5. The operator can additionally export signed inventory-audit checkpoints using tools/Clm.Backup anchor --tenant <organization-id> with a protected signing key and an explicit non-bypass-RLS database identity. verify-anchors --tenant verifies the tenant-bound signature, segment and chain; --db-env also compares the live database. Keep the signing key and immutable copies independent of the database host.

INPUTS & PARAMETERS

- The server creates events during changes; this workspace cannot edit them.
- Sequence and hashes link consecutive events. Times use the browser’s local time zone.

- The server verifies the whole hash chain whenever the newest event changes and at least every 5 minutes, and refuses inventory reads when the chain is inconsistent.

OPERATION RESULT

History helps reconstruct who changed data and when. Hash-chain validation detects inconsistent stored history.

PERMISSIONS

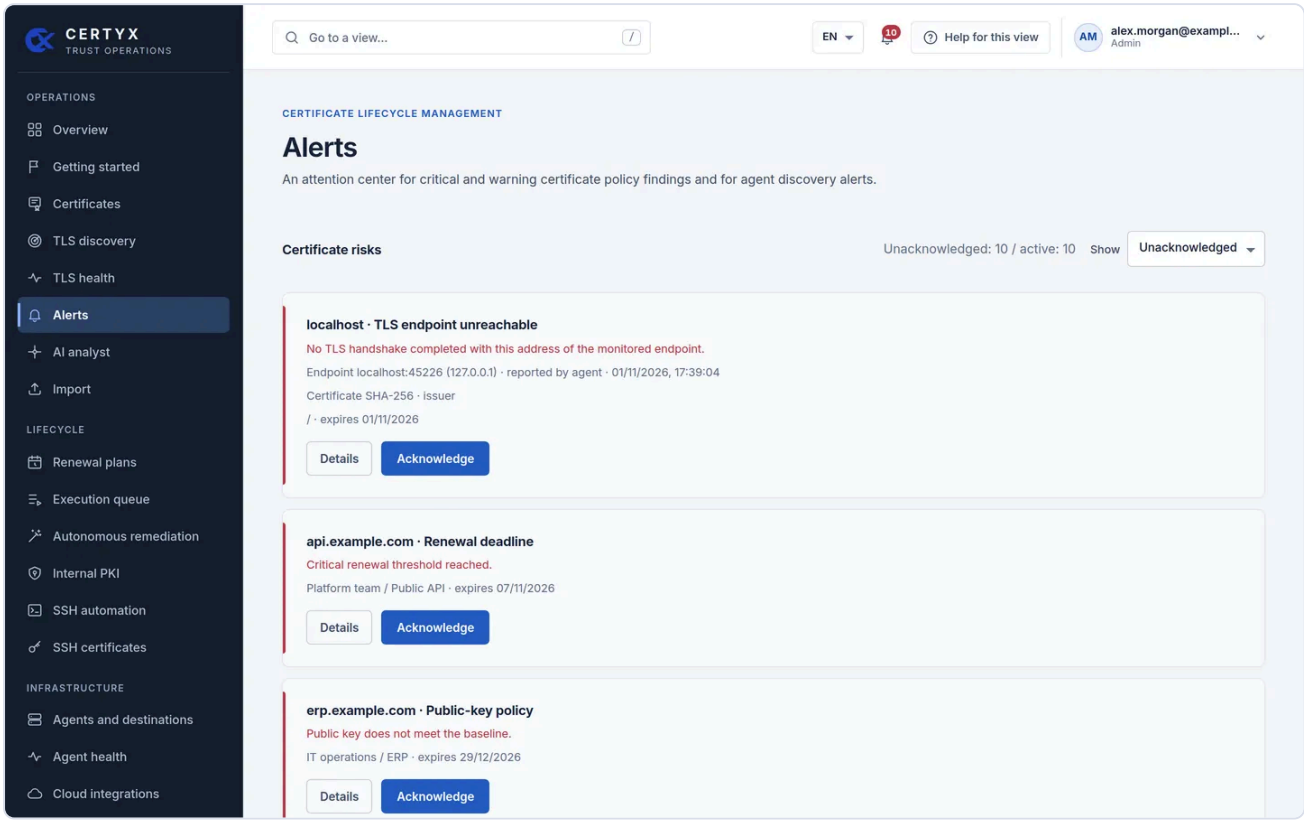
All signed-in roles view and export.

LIMITATIONS

The live hash chain alone cannot prove non-repudiation against a database administrator. A change to an earlier stored event made directly in the database is detected at the next full verification (at the latest after 5 minutes). Independent signed checkpoints detect rewriting of already anchored history; protection against deletion requires external immutable retention and independent verification. Retention: operations are kept for the life of the organization without automatic deletion, and the application role can only add and read them. A configurable retention period is not offered, because deleting entries would break the hash chain. One export holds at most Clm:AuditExport:MaxRows operations (default 100,000); narrow the filters for larger views.

13 Alerts

An attention center for critical and warning certificate policy findings and for agent discovery alerts.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Review unacknowledged alerts, ordered by severity and expiry.
2. Details opens the certificate in the inventory, where its facts, findings, ownership and renewal show the evidence behind the alert. Each card already shows the finding, its explanation, owner and service, expiry and, once acknowledged, who acknowledged it and when.
3. Choose Acknowledge. Select All active risks to include acknowledged risks.
4. Notification outbox & history below lists the organization's latest 100 notifications (alerts for outbox channels, group email and mobile push) with channel, state, attempts, delivery time and the last failure reason; Preview email opens a queued group email as it is sent.
5. Each entry names its event type (certificate alert, escalated alert, job and renewal events). An administrator chooses Send again on a delivered, given-up or cancelled channel entry to deliver it once more (attempts reset, recorded in the security journal); group e-mail and mobile push entries cannot be sent again, and sending a PagerDuty trigger again reopens the incident.
6. When the AI analyst is enabled for the organization, Explain opens it with a question about the alert.

INPUTS & PARAMETERS

- Alerts come from current policy assessments: validity, renewal thresholds, public keys or signatures.

- Identity includes certificate, policy version, finding code and severity. A policy or severity change requires a new acknowledgment.
- Agent discovery raises alerts after the profile's baseline (its first complete pass): Unknown certificate found (warning; critical when self-signed or from an issuer never seen in the organization), Certificate replaced at endpoint (warning; not raised when a CERTYX execution job deployed exactly that certificate to that endpoint, or in place of the previous one, within Clm:Discovery:ReplacementSuppressionWindow, default 7 days) and New TLS endpoint (info). These cards show the endpoint, the reporting agent and the SHA-256 fingerprint and issuer of the new and the previous certificate. They are raised once per endpoint and certificate, journaled and sent through the configured notification channels like other open alerts.

OPERATION RESULT

Acknowledgment is persistent, idempotent and audited with identity and timestamp. It does not alter the certificate or resolve the risk.

PERMISSIONS

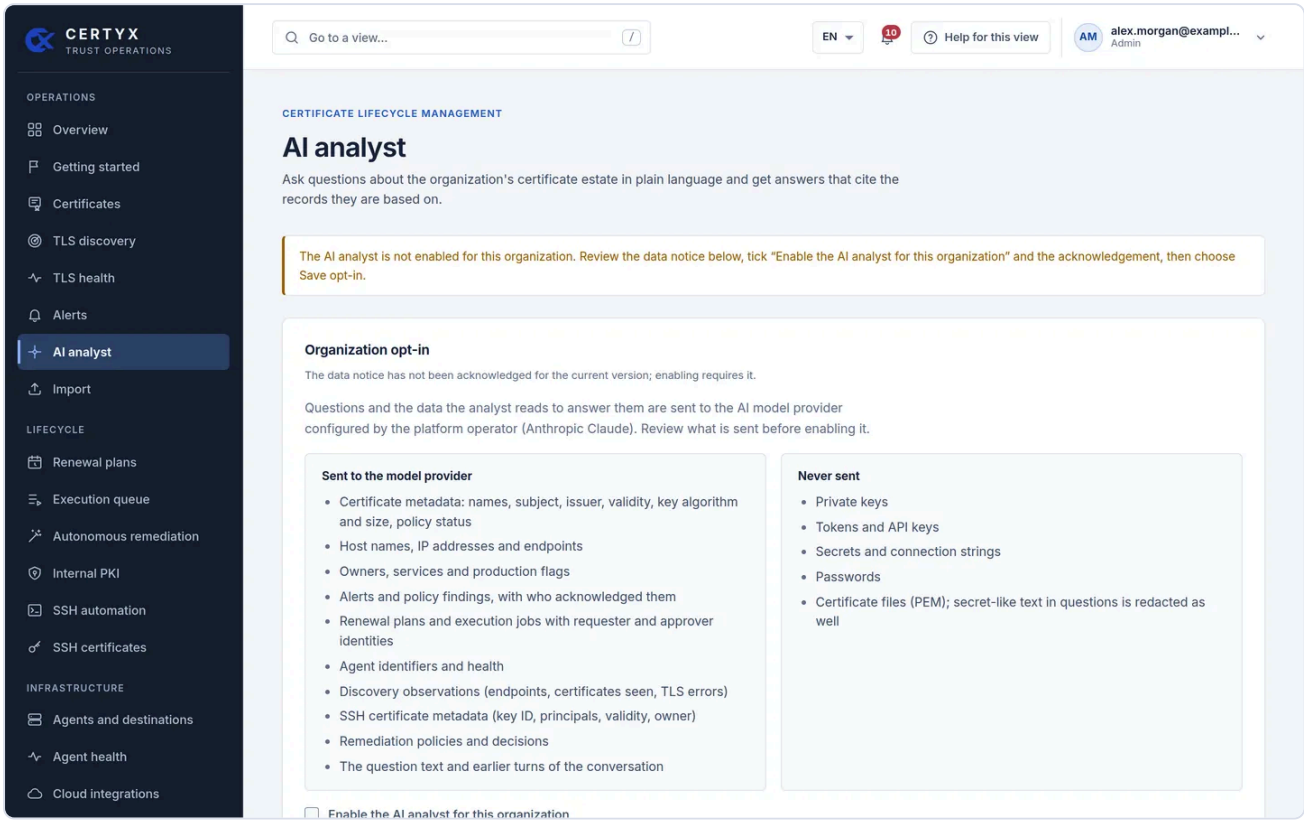
Read: all signed-in roles. Acknowledge: Operator or Admin.

LIMITATIONS

Acknowledgment does not remediate risk. Open alerts are queued for every enabled outbox channel of the organization (webhook, Microsoft Teams, Slack, PagerDuty) and delivered at least once; escalation channels receive only critical alerts that are still unacknowledged after their delay. Channels and the SMTP relay are set up in Notification groups → Delivery, where each shows its waiting and failed messages. When the service operator has configured push notifications, open critical and expiry alerts also reach the organization's authorized mobile devices with notifications turned on, as a generic message without certificate names. The list shows the latest 100 notifications. A discovery alert stays listed after acknowledgement; assign an owner to the certificate or investigate the endpoint to act on it. A replacement by another tool, or a CERTYX deployment whose verification endpoint differs from the observed address, is reported.

14 AI analyst

Ask questions about the organization's certificate estate in plain language and get answers that cite the records they are based on.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Prerequisites: the subscription includes AI_ANALYST; the platform operator connects the analyst in the control room view AI analyst provider (Anthropic API key, model, Test connection, “Offer the AI analyst to organizations”), or, in a standalone installation without a platform database, whoever runs it sets Clm:Analyst in the configuration; and an organization Admin enables it here after reading and acknowledging the data notice. Save opt-in records the acknowledgement of the data notice even when the analyst stays disabled; the card then shows the box ticked (it cannot be unticked) with the administrator and time of the acknowledgement, and a confirmation appears next to the button. The acknowledgement stays valid until the data notice changes; a new notice version must be acknowledged again.
2. Until everything is in place, the view names the missing step and who must act: not in the subscription (sales or the platform operator), not offered by the operator, no API key or an unreadable key (platform operator in AI analyst provider, or the installation configuration), not enabled for the organization (an Admin, in the Organization opt-in card of this view). Administrators see the opt-in card and the data notice even before the provider is configured.
3. Type a question, for example which production certificates expire within 30 days and who owns them, and choose Ask. The analyst reads data only through read-only tools of this organization: certificate search and details (findings and, when licensed, business risk), alerts, renewal plans, execution jobs,

agent health, discovery observations, the SSH inventory and remediation policies and decisions.

4. The answer shows numbered references; each links to the cited record (certificate details, Alerts, Renewal plans, Execution, Agent health, Discovery, SSH or Autonomous remediation). Only records a tool actually returned can be cited. Below the answer you see which data was read and the token counts.
5. Explain on certificate details, on an alert and on a remediation decision opens this view with a prefilled question about that record; review it and choose Ask.
6. Conversations are kept for the organization's retention and are visible only to you: open one to continue, delete it, or start a New conversation. Administrators see the request journal of the organization.

INPUTS & PARAMETERS

- Question of up to Clm:Analyst:MaxQuestionLength characters (default 4000). A follow-up sends at most Clm:Analyst:MaxHistoryTurns earlier turns (default 10).
- Sent to the model provider (Anthropic Claude, chosen by the operator): certificate metadata, host names and endpoints, owners and services, alerts and findings, plans and jobs with their requester and approver, agent health, discovery observations, SSH certificate metadata, remediation decisions and the question text.
- Never sent: private keys, tokens, secrets, passwords and certificate files (PEM). Tools project records without them, and a redaction pass removes secret-like text (PEM blocks, API and access keys, bearer tokens, JWTs, password= assignments, long encoded data) from tool results, questions and earlier turns.

OPERATION RESULT

An answer with numbered references to records of this organization. Every question is recorded in the request journal (user e-mail, time, tools used, token counts, outcome) and in the security journal, both without question or answer text. Opt-in changes are journaled with the acknowledging administrator, whose e-mail address the Organization opt-in card shows.

PERMISSIONS

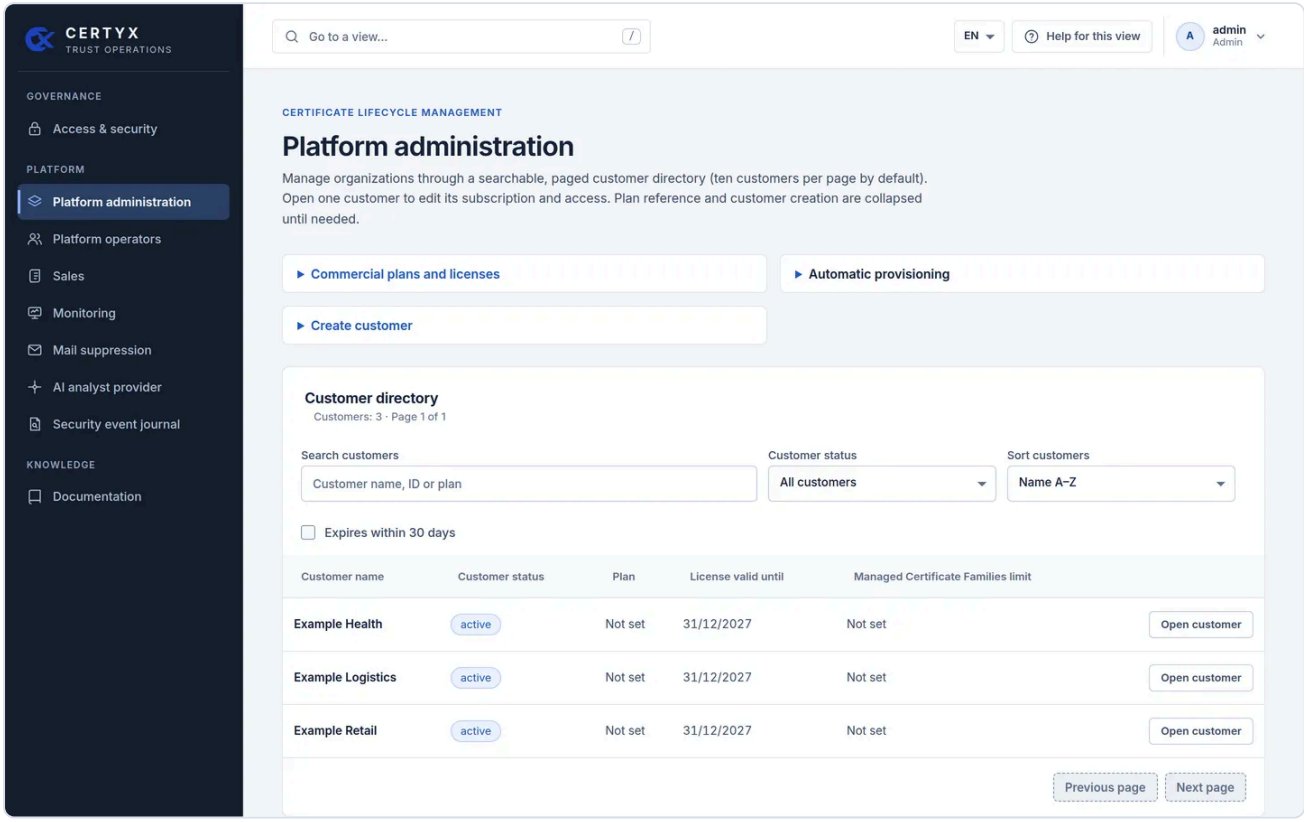
Every signed-in user of the organization sees the status. When the analyst is enabled, Auditors, Operators and Admins can ask. Enabling it, setting the retention and reading the request journal: Admin.

LIMITATIONS

Answers are generated by an AI model and can be wrong or incomplete; check the cited records before acting. The analyst is read-only and cannot change anything. Quotas set by the operator (defaults: 30 questions per user and hour, 300 per organization and 2,000,000 tokens in 24 hours) refuse further questions with a message. One answer uses at most Clm:Analyst:MaxToolRounds model calls (default 6) and Clm:Analyst:Timeout (default 2 minutes); otherwise it is marked incomplete or a timeout is shown. Each tool returns at most 25 records and 40,000 characters. Answers appear when complete (no streaming). A provider failure or refusal is shown as a message and journaled. With retention 0 nothing is kept; expired conversations are deleted by an hourly sweep. Your conversation list shows your newest Clm:Analyst:ConversationListLimit conversations (default 50); the request journal shows the newest 50 requests of the organization and keeps entries for Clm:Analyst:RequestJournalDays (default 400 days). Every number stated here is a configurable default: the quotas are set by the platform operator in AI analyst provider, the other values in the installation configuration (Clm:Analyst).

15 Platform administration

Manage organizations through a searchable, paged customer directory (ten customers per page by default). Open one customer to edit its subscription and access. Plan reference and customer creation are collapsed until needed.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Search by name, ID or plan; filter status or licenses expiring within the configured window (30 days by default; the checkbox shows the number of days), sort and change pages. Open customer shows only the selected organization; close its details to return to the directory.
2. Create a customer with a license period.
3. Provision its dedicated instance: with automatic provisioning on (the default of the container deployment), creating the customer starts it on its own and the customer becomes active when the Instance & provisioning tab shows the job completed; when automatic start is off, choose Provision automatically. Without automatic provisioning, prepare the instance with the server-side procedure and choose Verify and activate instance.
4. Create customer accounts with customer roles: in the Access tab, Create customer account takes the sign-in e-mail address, an initial password and the role (Admin, Operator or Auditor). Each listed account has its own role, Account enabled and Save access.
5. Manage license periods and suspend access when required: in Overview → Customer details change the name, the licence end or the Customer state (active, Suspended or Archived) and choose Save customer. The details editor is not shown while the customer is provisioning (activation happens through the instance tab); an archived customer shows only its Overview.

6. Set the subscription plan, status, dates, managed-family limit, grace allowance, billing period, contract amount, currency (PLN, EUR or USD) and feature/add-on entitlements, then choose Save subscription. Choosing a plan applies its feature preset and keeps optional modules; Discovery is always included. Operations and discovery do not increase usage-based fees.
7. Contract and billing on the customer card shows the signed contract from Sales, overdue and next payments and a suspension for non-payment; Open in Sales opens the contract. While a contract is in force, plan, certificate limit, validity, billing period, amount, currency and licence end follow it: a different value is saved only with a reason for the exception, recorded in the security journal.
8. In the Access tab, Reset MFA (shown for accounts with MFA switched on) removes MFA from a customer account that lost its authenticator and recovery codes, after identity confirmation outside the system. The account's sessions end and the action is written to the platform audit and the security journal.
9. Automatic provisioning: the Automatic provisioning panel above the directory shows whether it is switched on and starts on its own for new customers, the configured infrastructure and migration commands, whether they are fixed executables others cannot modify, the state directory and the retry limits. A customer created in the panel or by a signed contract, and a customer that waited while provisioning was off, is provisioned automatically: the platform creates the customer's own database and roles, migrates the database and records the customer identity, stores the owner-only connection secret, verifies the instance and activates the customer. Status, step and any stop reason refresh on the Instance & provisioning tab; each stop reason names its cause and remedy, and after fixing it choose Retry provisioning. Verify and activate instance names the exact check that failed and how to fix it; for an instance that was never prepared it starts automatic provisioning instead. In the container deployment the jobs run in the separate provisioning job (the provisioner container), the only part of the deployment that holds the database provisioning credential, so the application itself cannot create databases or roles; the panel then shows that job's last report under Runs in and warns when the report is missing or older than five minutes (check that the provisioner container runs).
10. The customer card has the tabs Overview, Subscription and entitlements, Access, Cloud sources and Instance & provisioning (the last only while provisioning; an archived customer has only Overview); the chosen tab stays open while the card reloads after a change, but not when the page is reloaded or the view is opened again. Overview shows the customer facts, Contract and billing, the subscription usage of an active or suspended customer and the customer details editor. In the subscription each feature has one choice: not included, included in the plan or optional module. Close the customer details with the button in the card header.
11. Cloud sources: assign the customer's collector inbox with a source ID, the provider (AWS, Azure, Google Cloud or DigiCert CertCentral), the scope exactly as the collector reports it (Azure: the vault name; AWS: ACCOUNT:REGION, e.g. 123456789012:eu-central-1; Google Cloud: PROJECT:LOCATION, e.g. my-project:global; DigiCert: account:<destination ID>), the inbox (absolute path of the owner-only file on the platform host, or an inbox identifier when the deployment has an inbox directory) and the maximum snapshot age (5–1440 minutes). The next synchronization pass imports it into the customer's inventory and trust graph. The table shows the state (waiting for the first synchronization, working, partial result, failed with the reason, paused), certificates, last success and last attempt; edit, pause, resume or remove a source. Each change is versioned and recorded in the security journal. The organization sees its sources and their state in Cloud integrations, never the inbox location.
12. Access tab → Private endpoint verification: allow CERTYX itself to verify the organization's deployments on loopback, private or link-local addresses (confirmation and security-journal entry). Without it, the organization verifies such endpoints with its own verifier agent.

13. Customer account setup e-mail: Create customer account sends the new account a single-use authenticator setup link through the platform's technical mail relay (Platform operations → E-mail relay for alerts; never the sales relay), in the chosen e-mail language. The e-mail contains the link as text and as a QR code of the link, never the authenticator secret or the password. While that relay has no host or sender, or fails, the Access tab shows the reason and, only for an organization without an enabled Admin, the link to copy; pass it to the customer through a trusted channel, separately from the initial password. Send setup e-mail in an account row issues a new link (the previous one stops working) and is written to the platform audit.
14. Sign-in networks of a customer (customer card → Access): shows the addresses the organization's administrators allow for sign-in. When they locked themselves out (for example a wrong range or a changed office address), enter your authenticator code and choose Remove the restriction: the list is emptied (the mobile app setting is kept), the change is recorded in the platform audit (customer.signin-networks.cleared with before and after) and appears as identity.signin-networks.cleared-by-operator in the organization's own security journal; as a platform record it is not sent to the organization's SIEM destinations. The organization's administrators then set a correct list in Access & security.
15. Customer approval and break-glass: in an active organization that already has an enabled Admin, Create customer account (any role), Save access that raises the role or re-enables an account, and Reset MFA do not change the account. They create an access request (the console says so) that the Access tab lists under Access requests with the requested change, time and state; a customer Admin approves or rejects it in Access & security → Requests from the platform operator, and an approved request is carried out at once. A request expires after Clm:Platform:AccessRequests:RequestLifetime (default 72 hours). An organization without an enabled Admin (fresh provisioning) or that is not active keeps the direct flow. Lowering a role and disabling an account always run directly.
16. Break-glass, only in an emergency (for example the organization's only Admin lost the authenticator): in the request row enter the reason (10–500 characters, shown to the customer) and your current authenticator code and choose Start break-glass. A second, different operator enters their own current code and chooses Confirm break-glass within Clm:Platform:AccessRequests:BreakGlassConfirmationWindow (default 1 hour). The action then runs; a created account, a raised role or a re-enabled account is reverted after BreakGlassAccessDuration (default 24 hours) unless a customer Admin changed the account meanwhile; every enabled customer Admin (and, for an access change or MFA reset, the account itself) receives an e-mail in their language through the platform relay, retried every minute (NoticeAttempts, default 30) until delivered.

INPUTS & PARAMETERS

- Customer name, license end date, account e-mail address, initial password and customer role (Admin, Operator or Auditor).
- Formats: customer name 1–200 characters; license end a date from today up to 5 years ahead, stored as 23:59:59 UTC; subscription limit a whole number from 1 to 10,000,000, grace 0–100 %, billing period 1–36 months, contract amount 0–1,000,000,000 with up to 2 decimal places, a trial at most 30 days; account e-mail up to 120 characters and initial password 16–256 characters; cloud source ID 1–100 letters, digits, - or _, scope up to 300 characters, inbox up to 1024 characters, maximum age 5–1440 minutes. The help under each field states its format, and a rejected value names the expected format.
- Plan and subscription status labels follow the selected language; their stored API identifiers remain stable.
- Reason for departing from the contract (10–500 characters), required only when a contract-governed value differs from the signed contract.

- Break-glass: reason 10–500 characters; the 6-digit code of your own authenticator (step-up, a code is accepted once; repeated wrong codes lock your account).
- Reserved organization names: *, platform, platform-control, unattributed and default (any case) are refused when a customer is created or renamed, because the journal and the SIEM export use them for the platform. A customer that already has such a name shows a warning on its card until it is renamed.

OPERATION RESULT

A versioned customer registry and platform audit. Every state-changing action on a customer (accounts, requests, break-glass, subscription and state, activation, provisioning, cloud sources, private verification) is also written to that customer's security journal with actor platform:<operator account ID>, shown to the customer in Access & security → Platform operator activity and exported only through the platform's own SIEM scopes, never to the customer's destinations; account actions record the target account ID in the platform audit.

PERMISSIONS

Explicitly allowlisted platform administrator with an MFA-authenticated session. Customer Admin is insufficient. Break-glass needs two different operators, each confirming with a current authenticator code; operators who sign in through company sign-in cannot confirm (no local authenticator). Approving a request is reserved to the customer's own Admins. Step-up: every change in this view (except tests, connectivity checks and acknowledging an incident) asks for the current 6-digit code of your own authenticator; a confirmed code also covers your further control-room changes for Clm:Identity:OperatorStepUpWindow (default 5 minutes). A wrong or replayed code is refused, counts toward your account lockout and is journaled; Cancel abandons the change. Operators who sign in through company sign-in have no local authenticator and cannot make these changes. Operator roles: every role reads customers; creating customers, the details editor, subscriptions and provisioning need commercial or administrator, customer accounts, access requests, break-glass, sign-in networks, cloud sources and private verification need security administrator or administrator, contract and billing need commercial or administrator; the card shows only what the role may use.

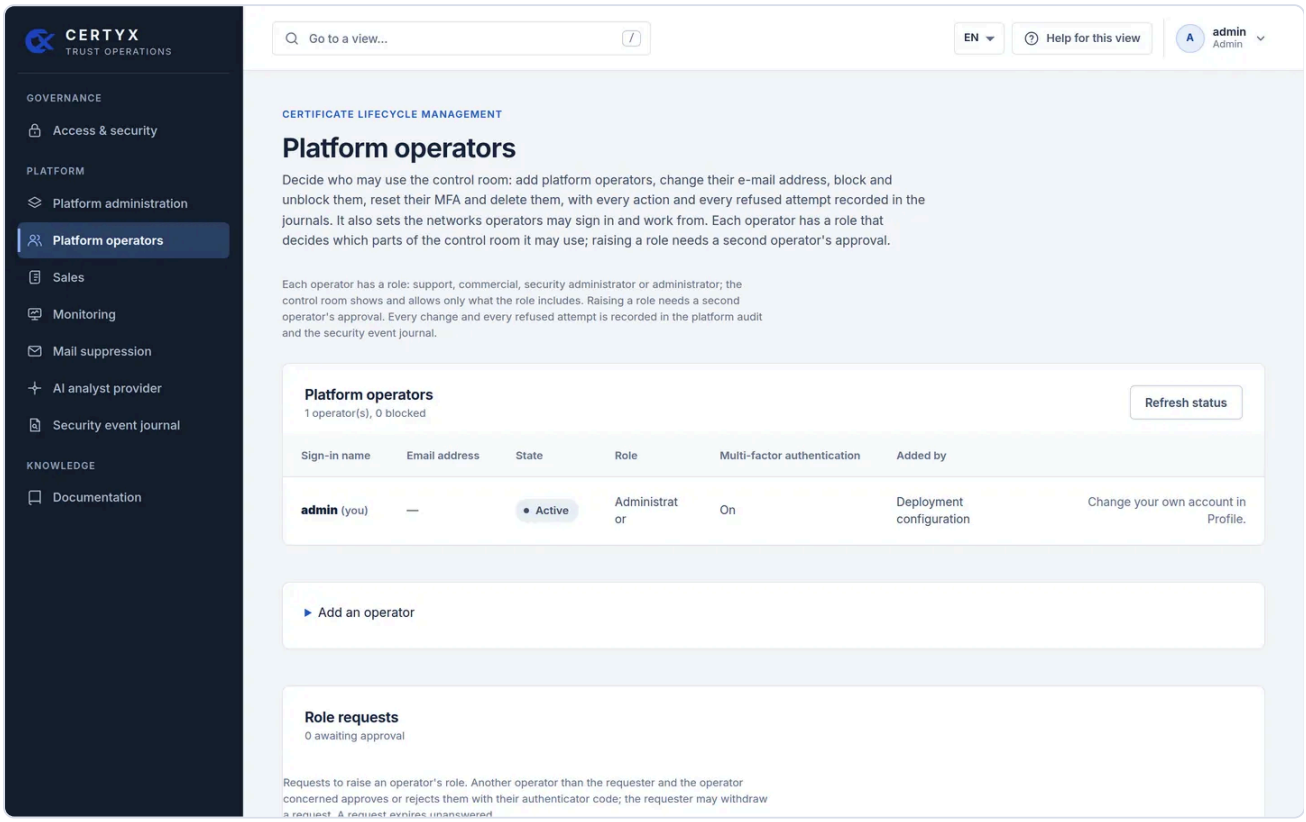
LIMITATIONS

- Platform administration needs the platform database (Clm:Database:Platform). Creating a customer additionally needs an absolute customer secret directory (Clm:Storage:CustomerSecretDirectory), where the customer's connection secret is expected; without it creation is refused as unavailable.
- Provisioning commands, the state directory, the automatic start and the retry limits are deployment settings (container deployment: CERTYX_PROVISIONING settings in certyx.env) and are shown read-only in the Automatic provisioning panel. While provisioning is switched off, prepare the instance manually; activation verifies it. A customer whose connection secret an operator already placed is not provisioned automatically; activate it with Verify and activate instance.
- With the platform database, customer accounts are kept in its shared account table and are the same on every application instance; without it they stay in the local data directory of a single instance. Each organization connects its own company sign-in (Microsoft Entra ID, Okta, OpenID Connect or SAML 2.0) and SCIM provisioning in Access & security.
- Contract amount is a recorded commercial term, not an invoice, automated charge or fixed price list. Search, filters, sorting and paging of the directory run on the server: the browser loads only the current page and the open customer. Search text matches literally (% and _ are not wildcards). The operator sets the page size and the expiry window in Clm:Platform:Directory.

- Automatic provisioning requires a configured infrastructure command and runs only on Unix hosts. The container deployment creates each customer's database with its own owner and runtime roles on the deployment's tenant database server, over TLS with full certificate verification; one customer's credentials cannot open another customer's database. Locally it can create a Docker PostgreSQL per customer, in the cloud use the Terraform modules for AWS RDS, Azure Flexible Server and Cloud SQL. A database or role of the same name that provisioning did not create, a database belonging to another customer and an existing different secret are refused without changes. Deleting a customer's infrastructure is not automated.
- Cloud source inboxes are files on the platform host written by the customer's collector; the deployment can restrict them to an inbox directory, and an inbox outside it is shown as not read. A synchronization registry file used before is taken over once into the customer cards. A source of an unlicensed customer (trust graph) fails with that reason. The catalog features HSM integration and On-premise are marked Not implemented in the plan reference and the entitlement table; entitling them enables no function.
- Setup links are valid for Clm:Identity:MfaSetupLinkLifetime (default 24 hours), work once and still require the account password; set Clm:Identity:PublicConsoleUrl so links use the public console address.
- Operator-created and re-enabled customer accounts count toward the active-account limit of the customer's contract (the package user limit, taken from the plan when a signed contract is applied and 5 for a customer without an applied contract); accounts created by the customer's own Admins and by SCIM are not counted against it yet. For an organization with an Admin, the setup link of a customer account is never shown in the control room: without a working platform relay the console says so and the customer's Admin sends or receives the link. A break-glass e-mail that cannot be delivered after the configured attempts stays recorded in the request and the customer's journal.

16 Platform operators

Decide who may use the control room: add platform operators, change their e-mail address, block and unblock them, reset their MFA and delete them, with every action and every refused attempt recorded in the journals. It also sets the networks operators may sign in and work from. Each operator has a role that decides which parts of the control room it may use; raising a role needs a second operator's approval.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Sign in at the operator entry (/operator-access) as an active platform operator with MFA and open Platform operators in the control room. The list shows each operator's sign-in name, e-mail address, state (Active or Blocked), whether MFA is set up, and whether the operator comes from the deployment configuration (the bootstrap operator, Clm:Platform:AdminIds) or was added in the control room. Your own row has no actions.
2. Every change (add, e-mail change, block, unblock, MFA reset, setup e-mail, delete) is confirmed with the current 6-digit code of your own authenticator (step-up): enter it in the form, or in the confirmation dialog of a row action, which also states the consequence. A missing, wrong or already used code refuses the change with the reason; the refusal is journaled.
3. Add an operator: enter the sign-in name, the e-mail address, an initial password and the e-mail language, then choose Add operator. CERTYX creates the account in the operators' organization (platform-control) and issues a one-time authenticator setup link. With the platform mail relay configured (Monitoring settings) the link is e-mailed; otherwise the view shows it with the reason, to

pass on through a trusted channel. Pass the initial password on separately. The new operator opens the link, enters the password, adds the authenticator and confirms a code, then signs in at the operator entry with password and code.

4. Roles: support (customers read-only, monitoring and incidents), commercial (sales, subscriptions and the customer lifecycle), security administrator (operators, customer accounts and access requests, integrations, SIEM, journals, alerting, sessions) and administrator (everything; the bootstrap operator and the configured ones). The control room shows only the views and actions of your role, and the server refuses everything else with the reason. Add operator offers a role: a new operator always starts as support, and any other role becomes a request in Role requests. Change role in a row lowers a role at once; raising it creates a request.
5. Role requests: another operator than the requester and the operator concerned approves or rejects a pending request with their own code or passkey; their own role must include the requested one (only an administrator approves an administrator). The requester sees Withdraw instead. A request expires unanswered (3 days by default). The last active administrator cannot be blocked, deleted or given another role, and you cannot change an operator whose role is above your own.
6. Sessions (security administrator and administrator): Show sessions lists the operators' browser and mobile app sessions, or those of an organization when you enter its identifier, with client, start, last activity, address and state. End session (confirmed with your code or passkey) signs it out on its next request and records it in that organization's security journal.
7. Change e-mail: choose Change e-mail, enter the new address and save. The operator's sessions end; the sign-in name stays the same.
8. Block: choose Block and confirm. The operator's sessions end at once (on their next request), sign-in is refused, and the block stays in force even when the operator's ID is still on the configured list. Unblock and confirm to give access back; the operator signs in again with password and authenticator.
9. Reset MFA (an operator who lost the authenticator): confirm the reset; the authenticator and recovery codes are removed and the sessions end. Then choose Send setup e-mail so the operator can set up a new authenticator. Send setup e-mail is also offered for an operator who never finished the setup.
10. Delete: confirm the deletion. The operator leaves the list, the sessions end and the account is disabled for good. The membership is kept as a tombstone, so adding the ID to the configured list again does not bring the operator back, and the sign-in name stays reserved so the journals keep naming it.
11. Every action and every refused attempt is recorded in the platform audit (Security event journal → Platform audit: action platform.operator.<action>, the target account and the outcome, refused: <reason> for a refusal; before and after are stored with the entry) and in the hash-chained security event journal with the acting operator, the target, before, after and the outcome.
12. Control room sign-in networks: enter the addresses and ranges operators connect from, one per line (203.0.113.7, 203.0.113.0/24, 2001:db8::/48), enter your authenticator code and save. Until a list is saved here, the deployment setting Clm:Platform:AllowedNetworks applies (shown as ranges from the deployment configuration); the saved list takes precedence afterwards. The panel shows your own address and whether the list includes it; a list without it is refused until you tick the confirmation and enter a fresh code. Every operator sign-in at the operator entry and every request of an operator session is then checked; a session from outside the list ends on its next request. Empty the list and save to remove the restriction.

INPUTS & PARAMETERS

- Sign-in name: up to 120 characters without spaces, unique across all accounts; it cannot be changed later.
- E-mail address: a valid address, unique across all accounts; it receives the setup link.
- Initial password: 16 to 256 characters.
- E-mail language: English, Deutsch or Polski, for the setup e-mail.
- Your authenticator code: the current 6-digit code of your own authenticator, asked for with every change; recovery codes are not accepted.
- Deployment settings (not in the console): Clm:Platform:AdminIds (CLM_PLATFORM_ADMIN_IDS), the account IDs that are operators until a stored membership exists (written by the deploy-time bootstrap); the platform mail relay (Monitoring settings) for setup e-mails.
- Control room sign-in networks: IPv4 or IPv6 addresses or CIDR ranges starting at their first address, at most Clm:Identity:MaximumSignInNetworks entries (100 by default); Clm:Platform:AllowedNetworks in the deployment configuration seeds the list (invalid entries stop the start-up).
- Role: support, commercial, security administrator or administrator; any role above support is a request until a second operator approves it.

OPERATION RESULT

A membership per operator in the platform database (active, blocked or deleted, with a version), an account in platform-control for operators added here, ended sessions after a block, deletion, e-mail change or MFA reset, and an entry in the platform audit and the security event journal for every action and refused attempt. Each change of the control room's sign-in networks is recorded in the platform audit (platform.signin-networks.changed, target platform-control, before and after) and in the security journal; refused operator sign-ins and requests from outside the list are sign-in events with the reason network-not-allowed and the source address.

PERMISSIONS

Only active platform operators signed in with MFA open this view and use its API; customer administrators and blocked or deleted operators are refused (403) and cannot reach it. This view needs the security administrator or administrator role; other roles do not see it. An operator changes only operators whose role is covered by their own, and never approves their own role request. An operator never changes their own account here (no self-block, self-deletion or own MFA reset; use Profile for your own e-mail and authenticator), so the last active operator always keeps access. The control room's sign-in networks: any active operator signed in with MFA, with a current authenticator code of a local operator account for every change.

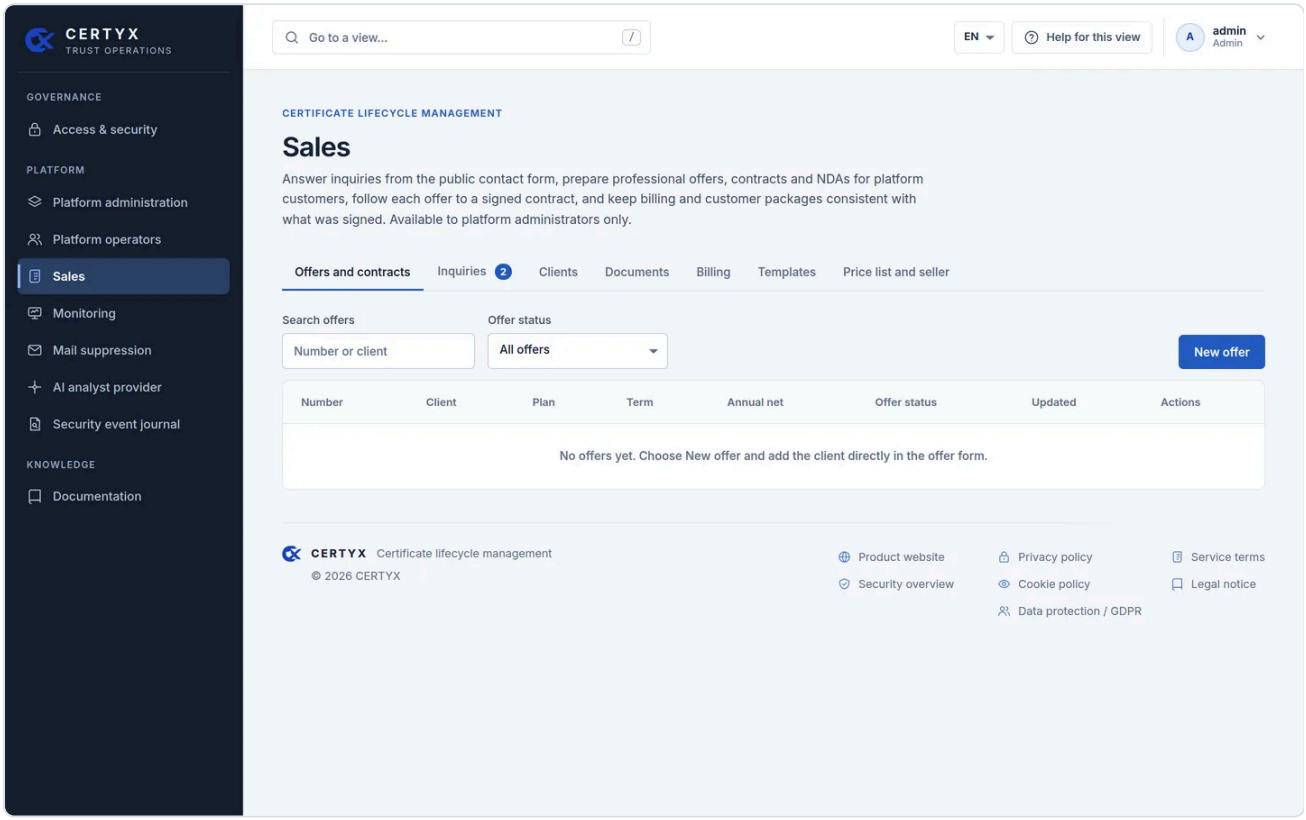
LIMITATIONS

Needs the platform database; without it the view's requests answer 503. A change made from an older list is refused as a conflict: refresh and repeat. The sign-in name cannot be renamed. Deleted operators cannot be restored; add a new operator instead. Each code confirms one change: a code of a time step that already confirmed a change (or the sign-in) is refused, so wait for the next code; wrong codes count toward your account lockout like failed sign-ins. Operators signed in through company sign-in have no local authenticator and cannot confirm changes: a local operator account with an authenticator must make them. The last sign-in time is not shown; find sign-ins in the Security event journal. Without a mail relay the setup link must be passed on by hand. Recovery when no operator can sign in any more: the offline bootstrap tool (reset-mfa) described in DEPLOYMENT.md. Operators who use the platform's company sign-in (an

sso:platform-control:<subject> ID on the configured list) have no local account: they are listed without actions, and their access ends when the ID is removed from Clm:Platform:AdminIds. Sign-in networks need the platform database to be saved here; without it only Clm:Platform:AllowedNetworks applies. If every operator is locked out of the control room by the saved list, the service owner removes the stored list in the platform database (DELETE FROM platform_settings WHERE key='signin-networks:platform-control', as the database owner, see PLATFORM-OPERATIONS.md); the configured list applies again. Roles of operators that existed before roles were introduced are administrator. When no administrator can sign in any more, the owner sets a role in the platform database (offline recovery).

17 Sales

Answer inquiries from the public contact form, prepare professional offers, contracts and NDAs for platform customers, follow each offer to a signed contract, and keep billing and customer packages consistent with what was signed. Available to platform administrators only.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. In Price list and seller, check the seller details shown on every document and the price list: plan fees, certificate bands, SSH, add-ons, term discounts, approval threshold and minimum margin. In Sales e-mail delivery, switch e-mail on and enter the SMTP relay (host, STARTTLS port, sender, optional user name and a write-only password or secret:<name>), the optional operator copy (billing notices) and up to 10 inquiry notification recipients; Send test e-mail sends one message at once and names the reason when it fails.
2. Inquiries lists requests from the contact form on the public site (general question, presentation request, offer request, pilot request), newest first; the tab badge counts new ones. Details shows the contact, message and questionnaire answers and lets you set the status (New, Contacted, Converted, Closed) with an internal note. For presentation, offer and pilot requests, Create offer adds the client from the inquiry (once; later clicks reuse it), marks the inquiry Converted and opens a new offer prefilled with the plan, managed certificates, SSH hosts (at least the SSH minimum), extra organisations, onboarding, premium support, term, monthly billing where the plan allows it, currency, start month, pilot flag and the inquiry reference and message in the notes. Review the quote and save the offer.

3. Each new inquiry sends two e-mails through the sales relay, usually within seconds: every inquiry notification recipient gets a notification in the language set by Clm:Sales:Inquiries:NotifyLanguage (English by default) with a headline, colour and priority per type (general inquiry: standard, reply within 1 business day; presentation request: high, presentation dates within 1 business day; quote request: high, priced offer within 1 business day; pilot request: top, contact within 4 business hours), all form details (contact, job title, company, e-mail, phone, country, tax ID, language, sizing answers including deployment preference and preferred start, pilot goals, message, consent, reference and receipt time in UTC), recommended next steps (for a pilot also a prerequisites checklist), Reply to <name> (Reply-To is the prospect) and Open in console, which opens this inquiry under Sales › Inquiries. The prospect gets a confirmation in their reply language with the reference, what happens next and the sales sender address. Both are HTML with a plain-text alternative and an embedded logo; they load nothing from the internet.
4. Clients are added with New client: in the Clients tab, or directly next to the client field of an offer or NDA, where the new client is selected as soon as it is saved and the values already entered stay. Record legal details, contact and invoice e-mail, document language and the public-sector flag. Optionally link an existing platform customer; otherwise the first signed contract creates one.
5. Optional pilot: in Documents generate an NDA for the client before the pilot starts.
6. In Offers and contracts choose New offer: the form starts with the Professional plan, a 24-month term and onboarding. Pick or add the client, then adjust package, term, billing and discount. The quote on the right recalculates as you type and warns about the minimum term, discount approval, maximum discount and margin. Save to assign the number OF/year/sequence.
7. Choose Generate offer PDF. Every PDF is encrypted with AES-256 and its own password, shown once after generation and available later through Show password. Send the PDF and the password by different channels.
8. Move the offer through its stages: Mark as sent, Client accepted, Prepare contract (assigns UM/year/sequence), then Generate contract PDF in the client's language (available from Client accepted on). Final edits are possible until the offer is signed. Other transitions: Back to draft, Client rejected, Mark as expired and Cancel offer; a rejected or expired offer can be reopened as a draft, while cancelling, rejecting and signing ask for confirmation and cancellation is final.
9. After signature, open Signed contract and configuration: record the terms as written in the contract (possible while the offer is accepted or the contract is prepared), upload the scan, use Check terms to compare offer, contract and customer configuration, then Mark as signed, which stays disabled until the signed terms are recorded. This locks the offer, creates the payment schedule and creates or configures the platform customer from the terms: plan, features, certificate limit with tolerance, validity, billing period, contract amount and user limit.
10. A newly created platform customer starts in provisioning: provision and verify its instance in Platform administration (Open platform customer). If configuration failed at signing, Apply to customer subscription retries it; it also re-applies the signed terms later.
11. In Billing, record payments with their date and reference. Recording the payment that cleared an overdue item automatically reinstates a customer that billing suspended. The Needs attention panel above the tabs (shown only when something is due) lists payments due, overdue items, suspended customers and contracts nearing their end; Prepare renewal offer creates a draft starting the day after the contract ends.
12. In Templates, edit the offer, contract and NDA text for each language, insert data fields and generated blocks, preview the PDF and save a new version.

13. Record a payment in a dialog with the payment date and a reference (up to 120 characters). The price list shows each plan as its own card with a summary (fee, limits, margin, PL/EN/DE text completeness) and expandable editing; a card warns about missing or mismatched PL/EN/DE lists and support texts, a margin below the minimum and a margin that falls below the minimum at the maximum sales discount. Save or discard price-list changes with the bar that stays visible while scrolling.

INPUTS & PARAMETERS

- Inquiry (from the public form): type, name, company, work e-mail, optional phone, country, job title and tax ID, reply and document language (EN, DE or PL), message (required for a general question), consent, and for offer and pilot requests the optional questionnaire: plan, managed certificates, SSH hosts, organisations, deployment preference (hosted by CERTYX, own environment or not decided), start month, term in years, annual or monthly billing, currency, onboarding, premium support, public sector, certificate sources and pilot goals.
- Client: legal name, address, tax ID, register number, representative, contact and invoice e-mail, language, public sector, linked platform customer.
- Offer: plan, managed certificates, SSH hosts, additional organisations, onboarding, extra days, premium support, pilot, term (minimum 12 months by default, 24 recommended, at most 60), annual or monthly invoicing, sales discount and approver, start date, validity, notes, PLN, EUR or USD (EUR and USD at the price list rates). A discount above the approval threshold of the price list needs an approver, and no discount may exceed the maximum sales discount.
- Formats: managed certificates and SSH hosts are whole numbers from 0 to 10,000,000, additional organisations 0–50, extra days 0–365; sales discount 0–100 % with up to 2 decimal places, approver up to 300 characters, notes up to 4,000 characters; client text fields up to 300 characters; payment term 1–120 days; amounts with up to 2 decimal places; SMTP host a host name or IP address without scheme or port, port 1–65535, sender and operator copy single e-mail addresses, inquiry notification recipients up to 10 addresses separated by commas; seller profile fields up to 300 characters; templates up to 200,000 characters. Inquiry questionnaire: managed certificates and SSH hosts whole numbers of 0 or more (up to Clm:Sales:Inquiries:MaxQuantity), organisations 1 or more (up to MaxOrganizations, 100 by default), term 1–3 years; message and pilot goals up to Clm:Sales:Inquiries:MaxMessageLength (4,000 characters by default); name, company, phone, country, job title and tax ID up to 300 characters. The help under each field states its format, and a rejected value names the field and the expected format.
- Signed terms: plan, quantities, term start and end (the end after the start), billing period (monthly or annual), annual net amount, currency (PLN, EUR or USD), payment term (1–120 days) and agreed changes. Scan: PDF, JPEG or PNG, recognised by its content, within the configured size limit (Clm:Sales:MaxSignedScanBytes, 20 MB by default).
- Templates: headings (#), bullets (-), numbered items (1.), **bold**, {{data fields}}, {{#if condition}} passages and blocks such as [[pricing-table]], [[payment-schedule]] and [[signatures]].

OPERATION RESULT

Numbered offers and contracts with a recorded lifecycle; password-protected PDFs with logo, headers and page numbers, stored with template version and SHA-256; a payment schedule with reminders; and a platform customer created or configured from the signed terms, whose contract then governs its subscription in Platform administration. Document generation, password display, terms, scans, status changes, customer configuration, suspensions and reinstatements are recorded in the security journal.

PERMISSIONS

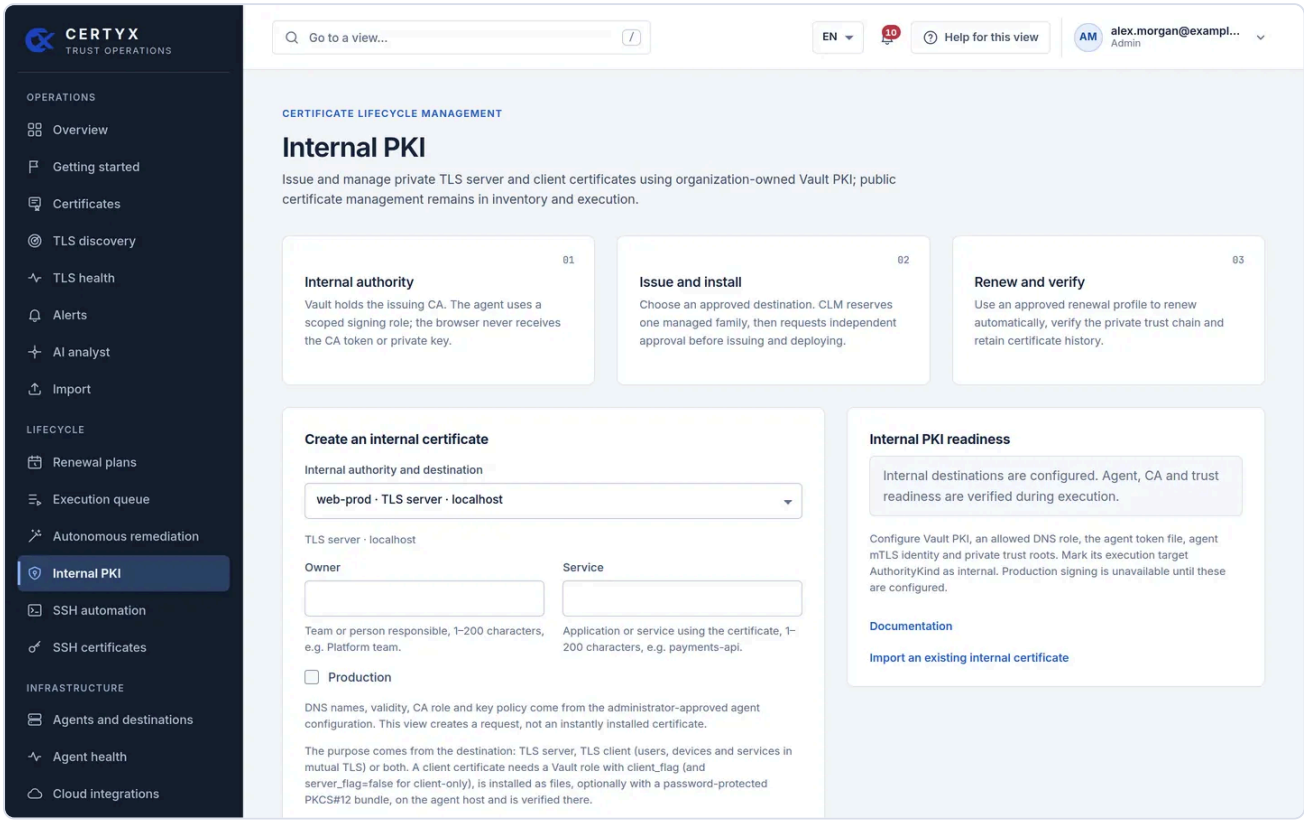
Platform administrator with an MFA-authenticated session. Customer accounts, including customer Admin, cannot open this view or its API. Step-up: signing an offer (which registers or configures the customer), applying a contract to the subscription, recording a payment and changing the sales mail relay ask for the current 6-digit code of your authenticator (covered for a few minutes by a confirmed code); quotes, offers, parties, documents and templates do not.

LIMITATIONS

- Requires the platform database (Clm:Database:Platform) with the current platform/schema.sql; without it the view reports that sales is unavailable.
- The public contact form needs the platform database; without it the form shows that it is unavailable. It accepts 10 requests per minute per address (Clm:Http:InquiriesPerWindow), requires a CSRF token and silently drops submissions that fill the hidden honeypot field. The notification and the confirmation are sent only while Sales e-mail delivery is on; until then they wait under Billing and are sent once it is switched on (the queue is checked every Clm:Sales:NoticeInterval, 30 seconds by default, and at once after a new inquiry). Without inquiry notification recipients the notification goes to the seller contact e-mail; without either, only the confirmation is queued. Open in console uses Clm:Sales:Inquiries:ConsoleUrl, or the address the form was sent from when it is empty, and needs a platform administrator sign-in. Every visitor sees the reference on screen; a filtered honeypot submission sends no e-mail.
- Default contract, DPA and NDA templates are drafts and must be reviewed by a lawyer before use.
- Monthly amounts in offers are an illustration only; contracts are concluded for the full term.
- The platform does not issue VAT invoices or connect to a bank: payments are recorded manually and invoices come from your accounting system.
- E-mail reminders and notices are sent only while Sales e-mail delivery is on; otherwise they remain queued and visible under Billing and are sent once it is switched on. Values of the configuration file are taken over once into the console; the relay timeout stays a deployment setting. TLS is always required.
- An unpaid item suspends the linked platform customer after the internal grace period configured in Clm:Sales:SuspensionGraceDays; the grace period is not stated in reminders. When the payment is recorded and nothing else is overdue past the grace period, the customer is reinstated automatically. A suspension set by an operator in Platform administration is never lifted by billing, and an operator who reactivates a customer with a payment still overdue past the grace period will see it suspended again at the next billing check.
- Signed scans are stored for comparison; the platform does not read their text. Terms are compared with the values you record.
- Creating the platform customer at signing requires Clm:Storage:CustomerSecretDirectory; the dedicated instance is still provisioned and verified in Platform administration.
- If the running server is older than the console files, the view says so instead of showing empty lists; restart the application and reload the page.

18 Internal PKI

Issue and manage private TLS server and client certificates using organization-owned Vault PKI; public certificate management remains in inventory and execution.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Configure Vault PKI with an issuing CA and DNS-scoped signing role. Store its token in a private agent-side file.
2. Configure the outbound mTLS agent and private trust roots. In Execution queue → Deployment targets an administrator creates the target with Certificate authority Internal CA (Vault PKI), its CA URL, mount and role (AuthorityKind internal). The agent destination uses AuthorityKind vault.
3. Internal PKI readiness states whether internal destinations exist and how many are disabled; without an enabled one, the form is disabled. Disabled targets stay in the list, marked disabled and not selectable, because the server refuses new jobs for them. Under Create an internal certificate choose the destination in Internal authority and destination (every target with an internal CA is listed with its purpose and identities), enter Owner, Service and Production, and choose Request internal issuance. CERTYX reserves one managed family before creating the issuance request; the confirmation shows the job ID.
4. A different administrator approves the job in Execution queue (Open approval and execution queue). The agent generates its private key/CSR, signs through Vault, deploys and verifies TLS.
5. Review the resulting certificate under Internal certificate inventory (50 per page). It lists only certificates classified internal: issued through an internal destination, or imported with Internal certificate (Import an existing internal certificate). Configure a separately approved renewal profile in

Execution queue for automation.

6. For client certificates, create the target in Execution with purpose TLS client (or server and client) and its identities. Use a dedicated Vault role: client_flag=true, server_flag=false for client-only, allowed_domains covering email/DNS domains, allowed_other_sans for user principal names (1.3.6.1.4.1.311.20.2.3;UTF8:*@corp.example.com) and allowed_uri_sans for URIs; cn_validations=disabled only when a free-form common name such as a person's name is used. A role without client_flag is refused with that setting named, and a client-only request is refused when the role would also grant server authentication.

INPUTS & PARAMETERS

- DNS names, client identities, purpose, CA URL and Vault mount/role come from the execution target; validity and key policy come from the Vault role and the agent configuration. Imported certificates may be explicitly classified internal.
- Owner and Service: required, 1–200 characters each. The destination must be enabled; a request for a disabled destination is refused (enable it in Execution queue).

OPERATION RESULT

An auditable issuance job followed by an inventoried internal certificate after verified successful execution. Renewal inherits its family.

PERMISSIONS

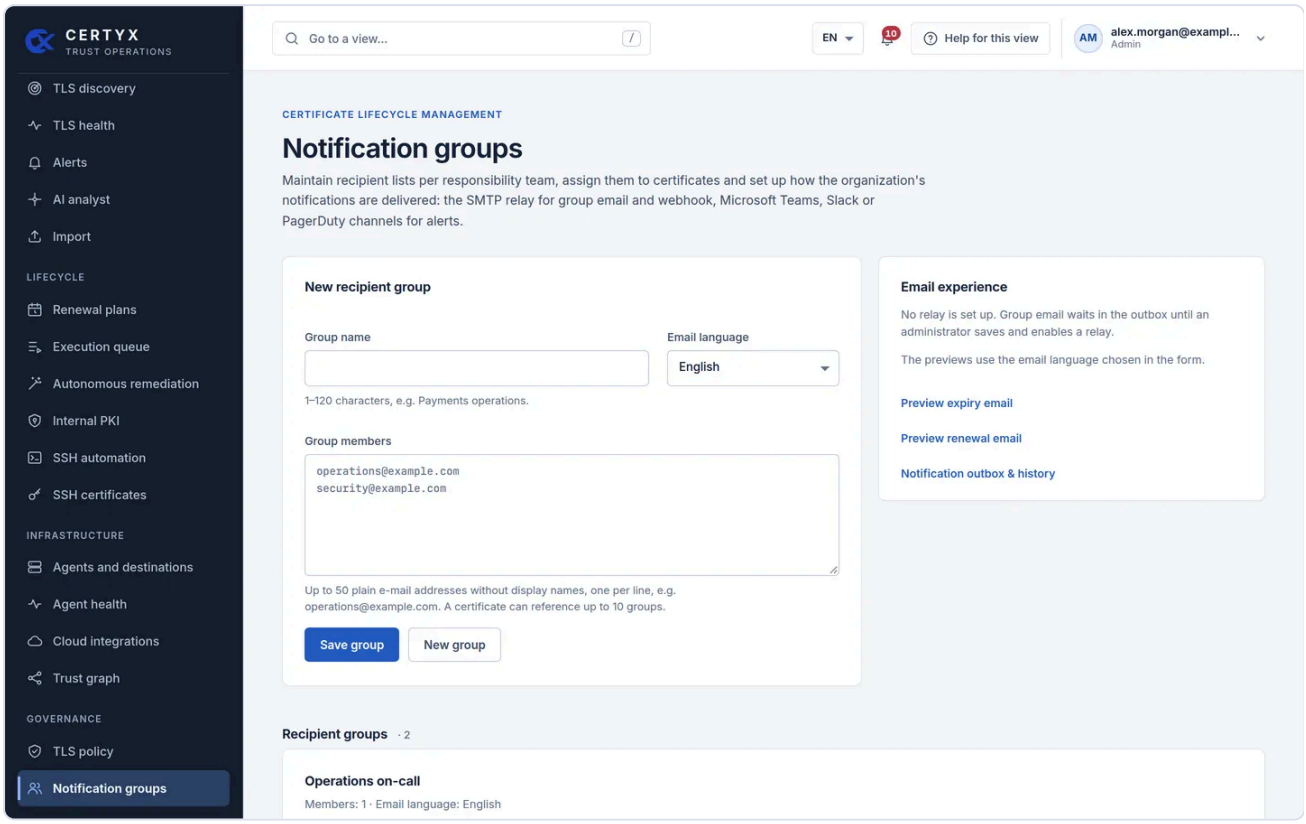
Operator or Admin requests issuance (on a service with subscriptions only with the AUTO_DEPLOYMENT feature and free managed-family capacity); a different Admin approves the job; Auditor reads only (the form is disabled). Tenant-specific targets and agent identities are enforced.

LIMITATIONS

Requires a configured Vault instance and agent. This view does not initialize a root CA, distribute trust roots to every client or issue code-signing certificates. Client certificates (users, devices, services in mutual TLS) are issued when the destination's purpose is TLS client or server and client and the Vault role sets client_flag; they are delivered as files on Unix agents only (no Windows certificate store, browser or smart-card enrolment). Public imports are never automatically trusted.

19 Notification groups

Maintain recipient lists per responsibility team, assign them to certificates and set up how the organization's notifications are delivered: the SMTP relay for group email and webhook, Microsoft Teams, Slack or PagerDuty channels for alerts.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Create a recipient group with a name, an email language (English, German or Polish) and up to 50 plain addresses; assign groups in certificate details. Archive group takes a group out of use: it receives no further email and can no longer be assigned to certificates, but it stays listed (marked Archived) with its members; Restore group makes it active again. Editing a group keeps its archived state. A refused save names the reason (identifier or name too long, unsupported language, invalid members, changed meanwhile).
2. A group receives email when a certificate assigned to it reaches an expiry threshold or has expired, after a verified renewal of that certificate, for SSH certificate events of SSH profiles assigned to it and when a remediation policy notifies it. Email experience → Preview expiry email and Preview renewal email show the message layout with sample data in the email language chosen in the form (the console language until you choose one, or the group's language while you edit it); group email waits in the outbox until an enabled SMTP relay exists.
3. Delivery → SMTP relay: enter the host, port (587), sender address and, when the relay requires it, user name and password. STARTTLS is always used and the relay certificate is validated. Enable the relay and save.

4. Send test e-mail sends a short message through the saved relay to the address you enter and shows whether the relay accepted it; the result is kept with the relay.
5. Delivery → Outbox channels: add a webhook (HTTPS JSON, optionally signed with HMAC-SHA256 in X-Certyx-Signature), a Microsoft Teams incoming webhook or Workflows URL, a Slack incoming webhook (https://hooks.slack.com/…; alert content arrives as plain text, without mentions or links), or a PagerDuty integration (routing) key. Escalation delay 0 sends every open alert at once; a delay makes it an escalation channel that receives only critical alerts still unacknowledged after that time. Send test checks the channel (a PagerDuty test is resolved right away).
6. Events of a channel: certificate alerts are chosen by default; tick job requested, job approved, certificate deployed and verified, job failed, deployment rolled back, job cancelled and certificate renewed to let pipelines, chat rooms or ITSM tools follow every renewal. Each event is sent once per channel as it happens (within Clm:Notifications:LifecycleEventWindow, one day); a webhook receives { eventId, type, version 1, event { id, at, actor, jobId, targetId, certificateId, domains, state } }, Teams and Slack a short text, PagerDuty only alerts and failed or rolled-back jobs.
7. Each relay and channel shows waiting and dead-lettered messages, the last delivery and the last failure reason; Alerts → Notification outbox & history lists the individual entries.
8. Before sending group email the system rechecks active group membership: removed recipients do not receive queued messages, new recipients do not inherit older ones, and a message without authorized recipients is cancelled. Recipients are delivered using Bcc; the sender address is not added.
9. Recipients that no longer receive e-mail lists the members of your groups that are on the platform's mail suppression list (a permanent bounce or a complaint reported by the mail provider), with the reason, the date and their groups; the group cards show how many members are suppressed. CERTYX sends them nothing, through any relay, until a platform operator removes them. Correct or replace the address in the group, or have the mailbox fixed and ask your platform operator (support) to remove it. Only platform-wide entries and entries caused by your organization's own mail apply; a bounce of another organization's mail never suppresses your recipients.

INPUTS & PARAMETERS

- Group name (1–120 characters), email language and members (plain addresses without display names, at most 50 per group). Channel names are also 1–120 characters.
- Relay host, port, sender address, optional user name and password; transport security STARTTLS (a plain-text loopback relay is offered only in the Development environment).
- Channel name, identifier (1–64 lower-case letters, digits and hyphens, starting with a letter or digit; group-email and mobile-push are reserved), kind, escalation delay (0–1440 minutes), HTTPS address without user name or password or PagerDuty routing key (20–64 letters and digits), optional webhook signing secret (up to 256 characters).

OPERATION RESULT

Audited, versioned delivery configuration of the organization and actual delivery of group email and alerts. Passwords, channel addresses, routing keys and signing secrets are stored encrypted, are never shown again and are not written to the journal or logs; leave a secret field empty to keep the stored value, or choose Remove the stored password or Remove the stored signing secret to delete it. The SMTP relay also sends the authenticator setup e-mail of users created in Access & security, at once and only to that user. Without an enabled relay the console shows the setup link to copy instead.

PERMISSIONS

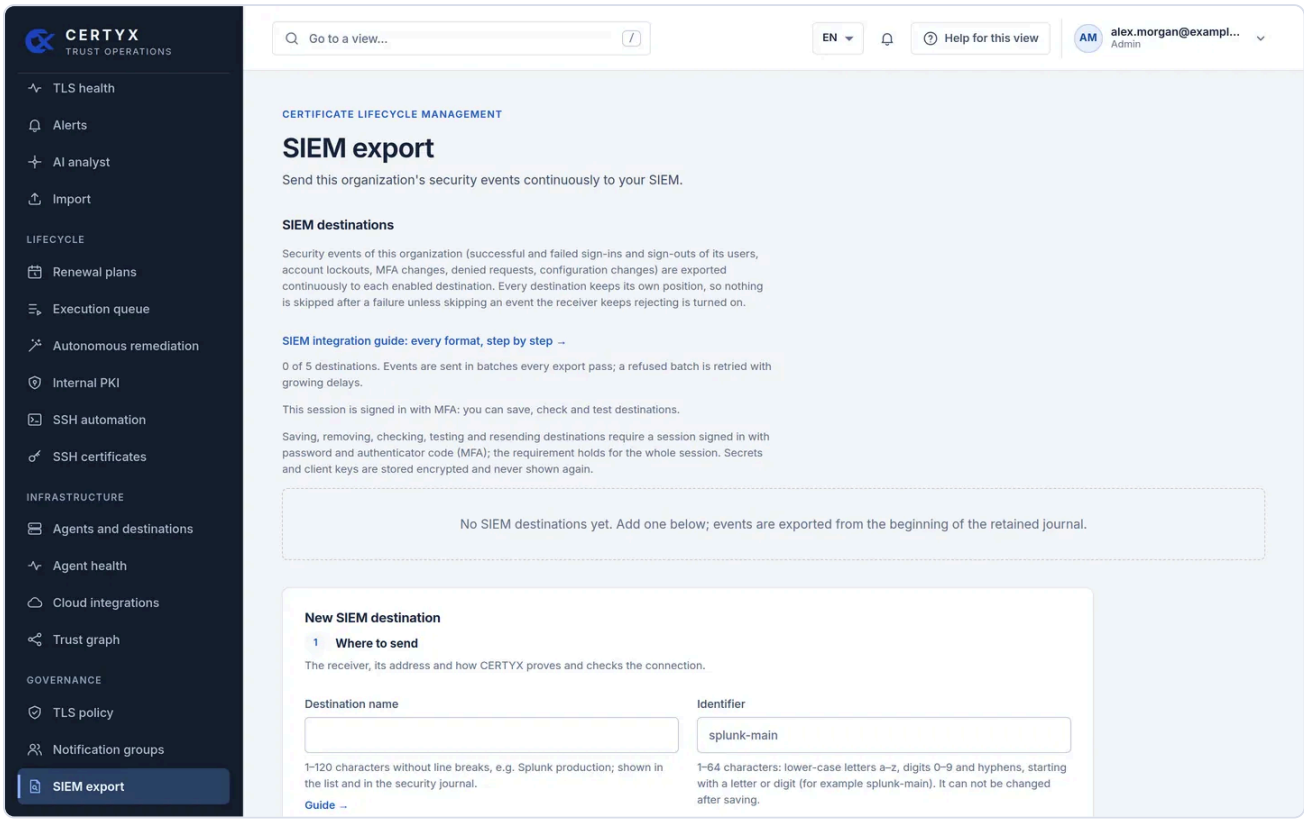
Admin creates, changes, archives and restores recipient groups, changes and tests the SMTP relay (disabling it instead of removing it) and adds, changes, tests and removes outbox channels; other signed-in roles read them. Recipient groups are archived rather than removed, so certificates and the journal keep referring to them.

LIMITATIONS

Delivery is at least once: a relay or channel that accepted a message just before a failure can receive it again (X-CLM-Event-Id and X-Certyx-Event-Id identify it). Failed messages are retried with growing pauses (at most one hour) and dead-lettered after 10 attempts. An organization can define at most Clm:Notifications:MaxChannelsPerOrganization outbox channels (default 20). Implicit TLS (SMTPS, port 465), OAuth2 SMTP authentication and SMTP client certificates are not supported. Settings copied once from the server configuration are marked as such; afterwards the values saved here apply. Queued messages keep the recipients from the time they were queued, minus anyone removed from the group since. Removing a channel cancels its undelivered messages. Delivery settings, the outbox and its history need the organization database; without it they are unavailable. Recipients on the platform's mail suppression list are skipped: a group email whose recipients are all suppressed is cancelled with that reason, and a test message or setup e-mail to a suppressed address is not sent.

20 SIEM export

Send this organization's security events continuously to your SIEM.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Sign in as an organization Admin with your password and authenticator code (MFA). The top of the view states whether this session meets the MFA requirement; without it the form and the Check connection, Send test event, Resend from..., New secret and Remove buttons are disabled and the notice says exactly what to do (for example: open Profile, finish Enable MFA with the current 6-digit code, then sign in again with the code). The requirement holds for the whole session; there is no time window.
2. Add a destination with a name and an identifier (1–64 lower-case letters, digits and hyphens, fixed after saving), then choose the Transport (HTTPS, Splunk HTTP Event Collector, Syslog over TLS, Azure Monitor Logs Ingestion for Microsoft Sentinel, OpenTelemetry Logs (OTLP/HTTP), or File archive on the server), a Format and a Schema profile. Only the formats and profiles the server allows for the chosen transport are offered: HTTPS sends the JSON record (CERTYX journal fields) or NDJSON (canonical event); Splunk HEC the JSON record or NDJSON; Syslog over TLS RFC 5424 syslog with structured data, CEF, LEEF 1.0, LEEF 2.0, key=value (Generic KV), the JSON record, or the CEF and LEEF layouts of the first release; Azure Monitor the Microsoft Sentinel table rows; the file archive CSV (canonical or the layout of the first release) or NDJSON. NDJSON and RFC 5424 offer the profiles Native CERTYX event, OCSF 1.1.0 (validated class) and ECS 8.11.0, Splunk HEC NDJSON additionally Splunk CIM fields; every other format uses the native event. The box What the receiving side needs lists the receiver settings for the chosen transport and format; every field states its expected format, and refused input names the

field. Clearing Export to this destination pauses the destination without losing its position. Destinations saved before this release keep what they sent (plus the event ID): editing one pre-selects the transport, format and profile its former kind stands for.

3. Step-by-step setup for every format, on the receiving side and in CERTYX, is in the SIEM integration guide: Splunk HTTP Event Collector (token, index, sourcetype, certificate), Cribl Stream (syslog TLS source, route and index), a Splunk TCP-SSL input with newline framing, IBM QRadar (Universal LEEF, TLS Syslog), ArcSight (Syslog NG Daemon SmartConnector with TLS), Microsoft Sentinel (data collection endpoint and rule, table columns, Entra application with the Monitoring Metrics Publisher role) and webhook receivers with signature verification code. Each field and the box What the receiving side needs link to the matching section, a failed check stage links to its remedy, and the guide's page offers the same content as a PDF in English, German and Polish (also under Documentation).
4. Sign-in events: check Sign-in events (sign-in, sign-out, lockout, MFA changes) in Outcomes to export, or leave every box unchecked. The organization's destinations then receive every successful and failed sign-in of its users (password, password with MFA or a recovery code, company sign-in with OIDC or SAML, the mobile app), sign-outs (including sessions ended by expiry, a password change or a logout at the identity provider), account lockouts and MFA enrolment, confirmation, administrator reset and recovery-code regeneration. Each event names the user's e-mail address, the source IP, the user agent, the method, the result (success or failure) and a fixed reason code; the claimed password or code is never sent.
5. Splunk: choose the transport Splunk HTTP Event Collector. In Splunk enable HEC (Global Settings: All Tokens Enabled, Enable SSL, port 8088), create a token whose Allowed indexes include the target index, then enter https://<splunk-host>:8088/services/collector/event, the token value, the index (for example clm) and optionally a sourcetype. The format JSON record uses the default sourcetype certyx:security; NDJSON (canonical event, profile native, OCSF, ECS or Splunk CIM fields) uses certyx:event and carries event_id as an indexed field; give that sourcetype KV_MODE = json in props.conf. The HEC certificate must be issued for the host name (Splunk's default SplunkServerDefaultCert is refused); paste a private CA in Trusted root certificates.
6. Syslog over TLS: enter the host name only and the port, and choose the framing: octet counting (RFC 5425; Cribl Syslog source, syslog-ng, rsyslog, QRadar, ArcSight) or newline (Splunk [tcp-ssl:<port>] input). The receiving input must use TLS. For LEEF 2.0 enter the LEEF delimiter (one character such as ^ or its hexadecimal code such as x5E; empty is TAB) and use the same delimiter in the QRadar log source; LEEF 1.0 always uses TAB. The line under the framing shows the syslog header the server operator set (facility, APP-NAME and IANA enterprise number; without a number no certyx@ structured data element is sent). Through Cribl, the index (for example clm) is set in the Cribl route or pipeline, not by CERTYX.
7. HTTPS (POST of a JSON array or NDJSON, HMAC-SHA256 of the body in X-Certyx-Signature) and Azure Monitor Logs Ingestion for Microsoft Sentinel (data collection endpoint, immutable ID and stream of the data collection rule, Microsoft Entra tenant, application ID and client secret) use the same form. For an HTTPS destination added without a secret, CERTYX generates the signing secret and shows it once to copy into the receiver; New secret on an HTTPS destination replaces it immediately (the previous secret stops working) and shows the new one once.
8. OpenTelemetry Logs (OTLP/HTTP): choose the transport OpenTelemetry Logs and the format OTLP protobuf (application/x-protobuf, the default of OTLP/HTTP) or OTLP/JSON, enter the full logs URL of the receiver (https only, for example https://collector.example.com:4318/v1/logs for the OpenTelemetry Collector), the Log record body (the message, or the whole canonical event as a key-value list), optionally an Authentication header name (empty is Authorization) and its value as the secret (for

example Bearer <token>; stored encrypted, never shown again), and Compress requests with gzip. Every batch is one ExportLogsServiceRequest: resource service.name certyx, service.version, deployment.environment.name, host.name, host.ip and certyx.organization.id/name/kind; one log record per event with Timestamp, ObservedTimestamp, SeverityNumber/SeverityText (low INFO 9, medium WARN 13, high ERROR 17, very high FATAL 21), Body, TraceId and SpanId of the request and the attributes (semantic conventions event.name, user.id, user.name, user.roles, session.id, client.address/port, server.address/port, http.request.method, http.route, http.response.status_code, error.type; certyx.event.id and the other certyx.* fields). TLS, trusted roots, client certificate (mTLS), revocation, delivery policy, filters, check and test event work as for the other HTTPS transports; a partial success with rejected log records is not resent (OTLP specification) but journaled as siem.event.rejected with the request's event IDs and the receiver's message, with the siem-rejected incident; Resend from sends the records again. A Retry-After on 429 or 503 is honoured up to the longest retry delay.

9. Field mapping (network destinations with a canonical format: NDJSON, RFC 5424, CEF, LEEF, Generic KV, OTLP): Add rule adds an ordered rule on the canonical event before it is written: Rename (a canonical field under a new name), Drop (deny-list), Keep (allow-list: with at least one Keep rule every other optional field is removed) or Add constant (a new field with a fixed value of at most 256 characters). event_id, schema_version, the three timestamps, event_code, category, action, outcome, severity and message can not be dropped or renamed; a new name has 1–64 letters, digits, underscores and dots (CEF letters and digits only, ECS and OCSF without dots), must not start with certyx and must not equal a canonical field or a field the format writes itself (for example dst in CEF or server.address in OTLP). New fields follow the format's own fields: inline in JSON, under labels in ECS, under unmapped in OCSF, as extensions in CEF and LEEF, as keys in KV and as attributes in OTLP. At most 32 rules; they are saved, versioned and journaled with the destination (before and after). Preview test event shows what the form as filled in would send for the test event in the chosen format and encoding (OTLP protobuf in its JSON encoding, with the size of the binary body); nothing is sent or saved, and invalid rules are refused with the reason.
10. Save, then choose Check connection (DNS, TCP and TLS without sending anything; for a file archive, whether its directory can be written) and Send test event. Each stage is listed with its result: resolved addresses, connected address and port, TLS version and certificate subject, issuer and expiry, then the HTTP status or the Splunk HEC code with its meaning and what to change. The test event is in category Integration and has a random UUID as event_id, which the result names; search the receiver for that event ID (actor siem-test). A file archive has no test event.
11. The list shows each destination's transport, format and profile, its position in the journal, events still waiting, delivered events, the last success, the last failure reason, the consecutive failures and the time of the next attempt, an event the receiver keeps rejecting (its sequence and attempts) and the last check. Edit loads a destination into the form: a secret or client key field left empty keeps the stored value, but changing the transport requires entering the secret again. Remove deletes the destination and its position; adding it again starts from the beginning of the journal. Destinations the platform operator configured for the organization are listed read-only. Each card also lists the saved configuration: transport, format and profile, target, framing and delimiter, the events sent, the delivery policy (server defaults named as such), Splunk and Sentinel settings, trust roots, client certificate, revocation check and whether a secret is stored. When the view opens, the form shows the first destination with its stored values; a stored secret or client key is never shown and appears as "Stored – leave empty to keep". Edit opens another destination, New starts an empty form, and after Save the form keeps the saved destination.

12. Send only reduces what a network destination receives, to lower the SIEM's licence usage: the checked outcomes are combined with or (for example Access denied, Server errors, Sign-in events and Configuration and business changes, which adds every event that carries a configuration or business change). The categories (Authentication, Authorization, User management, Configuration change, Data access, Administrative, Security, Application error, System, Integration) and the selected codes of the event-code catalogue are combined with or among each other and narrow the outcome choice further. Minimum severity sends only events at or above the chosen severity (received 2, success 3, rejected 5, denied and failed sign-in 7, server error and lockout 8). Every request has two records in the journal, a received record before it runs (not a failed attempt) and an outcome after it; a destination receives only the outcome unless Received requests is checked. The received record stays in the journal and in a file archive. Fields sent (under Shape and size) lists every field of the chosen format with a check box, so a destination receives exactly the fields checked and the SIEM's licence counts nothing else; only the fields the format requires stay on, and the first release's CEF and LEEF compatibility layouts are fixed, so choose the CEF or LEEF format there. Health and readiness probes are not sent. Events left out are not lost: they stay in the journal and in a file archive.
13. The destination form has four numbered sections: 1 Where to send (name, transport, format, address, secret, TLS), 2 What to send (outcomes, categories, event codes, severity and the folded More filters), 3 Shape and size of each event (preset cards, folded Detailed settings and Field mapping) and 4 Delivery (folded delivery policy, Export to this destination). Folded parts open when you click their title.
14. More filters: Send only these routes and Never send these routes (one per line; a route covers every route that starts with it, for example /agent/v1/discovery/telemetry for the agents' regular reports), Never send events of these actors (for example pending, requests before authentication), Never send these categories and Never send these event codes. An exclusion always wins.
15. Event shaping decides how large each event is, because many SIEMs bill by indexed volume: Full sends every field and the complete before and after of each change (as before); Balanced sends only the changed values, no empty fields and the standard fields; Economical sends a change's action and identifiers, the minimal fields and at most 2 KB per event; Custom uses the Detailed settings. A preset fills the Detailed settings; changing one selects Custom. Detailed settings: Changes (complete, only changed values, action and identifiers, none), Fields (all, standard, minimal; the required fields are always sent), Leave out empty fields, Largest event (0 or 512–1048576 bytes; a longer event loses change details, then its longest texts are shortened, and is marked truncated), Field name style and Time format (JSON documents of the native or Splunk CIM profile only), Syslog MSGID (event category by default, event code, your own text such as security-journal, or none) and Syslog APP-NAME (empty is clm). Preview test event shows the size with and without shaping. Measured on an agent observation of three certificates (json): Full 4578 bytes, Balanced 591, Economical 378.
16. Delivery policy (network destinations): Attempts before a critical alert, First retry delay and Longest retry delay (seconds), Batch size and Backlog alert above; an empty field uses the server default shown below the fields. A failed batch is retried with exponential backoff and jitter; after the attempts the siem-delivery incident becomes critical and retrying continues. A batch the receiver refuses for its content is split down to the single refused event; everything before it is delivered and that event is retried and alerted. Skip an event the receiver keeps rejecting is off by default; when on, the event is skipped after the attempts and journaled as siem.event.skipped with its event_id. Nothing is dropped on a backlog: the journal is the queue, and above the backlog limit the siem-backlog incident is raised.
17. TLS of network destinations: the receiver's certificate is always validated against the system trust store or the Trusted root certificates entered here. For mutual TLS enter a Client certificate (PEM) with its unencrypted private key (RSA or EC); the key is stored encrypted, never shown again, and an empty

key field keeps the stored key while the certificate is unchanged. Revocation check: Soft-fail (default) refuses a certificate reported revoked and accepts an unknown status (no CRL or OCSP address, responder unreachable); Hard-fail also refuses an unknown status.

18. Resend from... on a network destination's card opens a dialog: choose a journal sequence (1 or higher) or a date and time, read that every event from there is sent again, and confirm with Resend. The position moves back to just before that point, the backoff and the rejected-event state are cleared, and the notice names the new position; the receiver must deduplicate by event_id. The position never moves forward (a point the destination has not reached yet is refused), a file archive can not resend, and every resend is journaled (siem.destination.replayed) with the previous and the new position.
19. Syslog framing must be the same on both sides. Octet counting sends each record as <length> <message> (the UTF-8 byte length, no line feed) in one write; a Cribl syslog source then needs Octet count framing set to Yes (its default is No) and TLS enabled, one source per CERTYX port, with index and sourcetype set by an Eval function. A receiver that splits at line feeds shows fragments such as 412 <84>1 ...; a receiver without TLS shows binary events that start with \x16\x03 and contain its own host name. A successful syslog test states what the receiving input must do with the chosen framing.
20. File archive: choose the transport File archive on the server (daily files) and a format: CSV (layout of the first release, certyx-security-YYYY-MM-DD.csv), CSV (canonical event, certyx-events-YYYY-MM-DD.csv with a versioned header) or NDJSON (one document of the chosen profile per line, certyx-events-YYYY-MM-DD.ndjson). Enter the Retention (days, 1–3650), optionally Compress closed days with gzip, save and choose Check connection, which checks that the archive directory can be written and shows the files, used and free space. The archive writes every event of the organization to one file per UTC day in the organization's own area under the archive root the platform operator configured; filters, delivery policy, client certificate and revocation do not apply. A closed day is sealed (optionally .gz) with its SHA-256 in the SHA256SUMS manifest, and days older than the retention are deleted and journaled (siem.archive.expired).
21. Archive files on the archive's card lists every file with its size, sealed state and SHA-256; Download fetches one file and Download all (zip) every file with the manifest. Check a download with sha256sum -c SHA256SUMS. Downloads require the MFA session and are journaled (siem.archive.downloaded).

INPUTS & PARAMETERS

- Exported fields: sequence, time, event category and event code, organization, actor, source IP, agent fingerprint, method, route, HTTP status, outcome, trace ID, error type, business changes and the event hash, which identifies the entry in the hash-chained CERTYX journal.
- CEF mapping: rt, suser, src, requestMethod, request, outcome, externalId (sequence), dvchost (instance), cn1 httpStatus, cs1 organization, cs2 traceId, cs3 eventHash, cs4 agentFingerprint, cs5 errorType, cs6 changes; severity 2–8 by outcome. LEEF 2.0 carries the same values as devTime, usrName, src, cat, sev, method, resource, httpStatus, organization, traceId, sequence, eventHash, agentFingerprint, errorType, instance and changes with ^ as delimiter.
- Sign-in events (outcome authentication) are separate journal entries named authentication.sign-in.succeeded, authentication.sign-in.failed, authentication.sign-out, authentication.locked, authentication.mfa.enrolled, authentication.mfa.confirmed, authentication.mfa.reset and authentication.mfa.recovery-codes-regenerated. Methods: password, password+mfa, password+recovery-code, oidc, saml, mobile-password (with +mfa or +recovery-code), mobile-oidc, mobile-saml, mobile-sso and totp. Reason codes include unknown-account, invalid-credentials, disabled, locked, too-many-failures, mfa-required, mfa-invalid, mfa-enrollment-required, enrollment-expired, sso-required, not-operator, operator-entry, organization-unavailable, organization-mismatch,

account-not-allowed, ticket-invalid, device-refused, invalid-request, sso-error, sso-disabled, sso-no-subject, sso-group-overage, sso-not-provisioned, sso-no-role, sso-expired, sso-not-configured, sso-unverified-domain, and for sign-outs user, credentials-changed, idp-logout, session-ended and administrator-reset.

- Sign-in fields per format: the JSON record (webhook, syslog JSON, Splunk HEC event) adds SignIn {Action, User, UserVerified, Organization, Method, Result, Reason, UserAgent}; CEF uses the event name as class ID, suser = user e-mail, suid = account, src, outcome = success or failure, cat = authentication, act = event name, reason, requestClientApplication = user agent, flexString1 authMethod and flexString2 userVerified, with severity 3 for success, 7 for failure and 8 for a lockout; LEEF uses usrName = user e-mail, src, actor, action, authMethod, result, reason, userAgent and userVerified; Sentinel adds the columns User, UserVerified (boolean), UserOrganization, AuthAction, AuthMethod, AuthResult, Reason and UserAgent.
- Microsoft Sentinel: the stream columns are TimeGenerated, Sequence, Organization, Actor, SourceIp, AgentFingerprint, Method, Route, Status, Outcome, TraceId, ErrorType, Instance, EventHash and Changes, plus the sign-in columns listed above (a rule that does not declare them stores the other columns only); the application needs the Monitoring Metrics Publisher role on the data collection rule.
- Splunk HEC request: POST with Authorization: Splunk <token>, X-Splunk-Request-Channel (a stable GUID per destination) and concatenated envelopes {time, host, source certyx:security-journal, sourcetype, index, event}. With indexer acknowledgement on the token, CERTYX polls /services/collector/ack until Splunk confirms indexing (Clm:Siem:HecAckTimeout, default 30 seconds); an unconfirmed batch is sent again.
- Organization identification in every format: JSON, Splunk HEC and syslog JSON carry Tenant (organization ID), OrganizationName, OrganizationKind (organization, platform or unattributed) and Instance; CEF adds certyxOrganizationName and certyxOrganizationKind next to cs1 organization and dvchost; LEEF adds organizationName and organizationKind; Sentinel adds the columns OrganizationName and OrganizationKind. The name and kind are added at export; the hash-chained journal record is unchanged. The RFC 5424 HOSTNAME of syslog messages is the instance ID of the CERTYX instance that recorded the event.
- CSV archive columns: Sequence, At, OrganizationId, OrganizationName, OrganizationKind, Instance, Actor, SourceIp, AgentFingerprint, Method, Route, Status, Outcome, Severity, TraceId, ErrorType, SignInAction, SignInUser, SignInMethod, SignInResult, SignInReason, Changes, PreviousHash and Hash; UTF-8, header row, quoted cells, CRLF line ends; a cell starting with = + - @, a tab or a carriage return is prefixed with an apostrophe so spreadsheets show it as text.
- Canonical event (NDJSON, CEF, LEEF 1.0/2.0, KV, RFC 5424, canonical CSV): event_id (<installation id>:<sequence>; a UUID for test events), event_code from the event-code catalogue, category, action, outcome, severity 0–10, time, organization, actor, source IP and request details; the profiles OCSF 1.1.0 (Authentication 3002, Account Change 3001, API Activity 6003, otherwise Base Event 0 when a class's required attributes are missing), ECS 8.11.0 and Splunk CIM fields (Authentication and Change data models as HEC indexed fields) map the same event. The JSON record, the Sentinel rows and the layouts of the first release keep their fields and add the event ID.
- Enrichment fields trace_id, span_id, src_port, host_ip, dst_ip, dst_port, user_role and session_id are filled for journal entries recorded since this release (span_id and dst_port since the OTLP release); older entries send them empty.
- Destination address dst_ip and dst_port: for a request the server address and port that accepted the connection (the listener of the CERTYX server; behind a reverse proxy or a container network that is the local interface address and port, not the public name, because a forwarded host header can be set by

the client), for an agent or endpoint event the observed target the event names (for example the endpoint of a discovery alert); unknown otherwise. Mapped as CEF dst/dpt, LEEF dst/dstPort, ECS destination.ip/port, OCSF dst_endpoint, Splunk CIM dest/dest_port, OTLP server.address/port and dst_ip/dst_port in KV, CSV and NDJSON.

OPERATION RESULT

Continuous, at-least-once export of the organization's events with a durable position per destination, and a stage-by-stage diagnosis of each destination. Secrets (HMAC key, HEC token, client secret) and client keys are stored encrypted, never shown again and not journaled; every change, check, test and resend is recorded in the security journal (siem.destination.created, updated, deleted, checked, tested, secret-rotated, replayed), and a skipped rejected event as siem.event.skipped with its event_id. A file archive keeps the complete record on the server for its retention, with a SHA-256 manifest of every closed day, so the SIEM can receive less (Send only).

PERMISSIONS

Organization Admin whose session was signed in with MFA: a local account with a confirmed authenticator, or a company sign-in whose identity provider reports MFA (or whose organization declares that the provider enforces it). Other roles do not see this view. Changes, checks, tests, resends and new secrets are refused for sessions of the CERTYX mobile app.

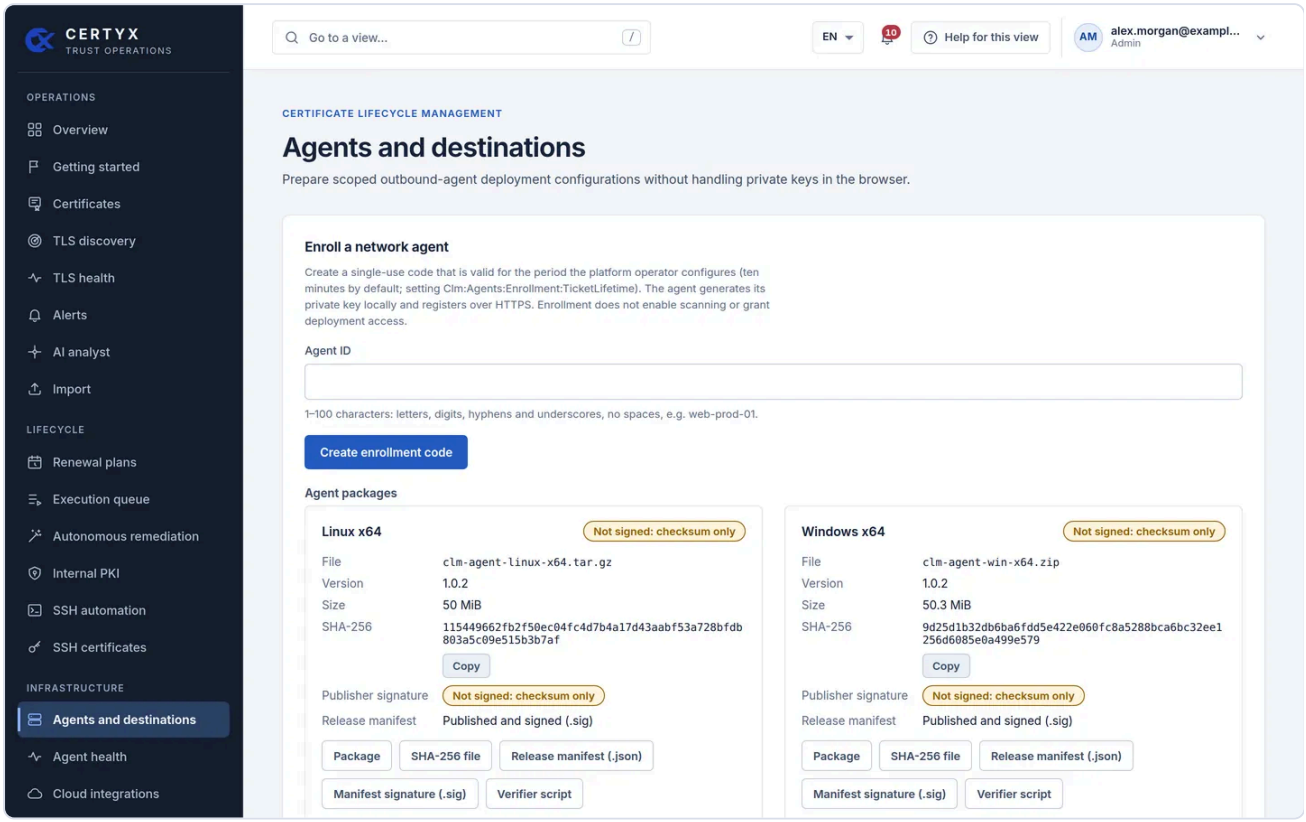
LIMITATIONS

Requires the platform database that holds the security journal; otherwise the view states that export is unavailable. At most Clm:Siem:MaxDestinationsPerOrganization destinations (default 5). Delivery is at least once: a refused batch is retried with exponential backoff and jitter (export passes every Clm:Siem:Interval, default 30 seconds), and a resend sends events again, so the receiver must deduplicate by event_id (<installation id>:<sequence>; a UUID for test events). TLS certificates are always validated; a client certificate is presented only when one is entered, and the revocation check in soft-fail mode accepts an unknown status while hard-fail refuses it. Sentinel uses the Microsoft Entra public cloud authority unless the operator sets Clm:Siem:SentinelAuthorityHost. A new destination starts at the beginning of the retained journal of the organization. Syslog has no delivery acknowledgement: a successful syslog test proves only DNS, TCP, TLS and the write, so routing and index must be checked at the receiver. Plain TCP and UDP syslog are not supported. A failing stage (name not resolvable on the CERTYX server, connection refused or dropped, TLS handshake, certificate name or trust, HTTP status or HEC code) is shown with its remedy and kept as the last check. Destinations at private, loopback or link-local addresses (for example an internal Splunk or Cribl receiver) are allowed by default (Clm:Siem:AllowPrivateDestinations, default true). The server operator of a shared installation can set it to false to restrict organization destinations to public unicast addresses: every address the host resolves to must then be public (private, loopback, link-local, shared 100.64.0.0/10, documentation, benchmarking, multicast and non-global IPv6 ranges are refused), the connection goes to the checked address and redirects are not followed. The check then stops at the DNS stage with the reason that the address is private, and export to that destination fails with the same reason (the siem-delivery incident, position unchanged) until a public address is used. The setting never restricts destinations of the platform operator or of the server configuration. Sign-in events belong to the organization of the account or company sign-in; a failed sign-in whose e-mail address matches no account is attributed to the organization whose verified company sign-in domain it belongs to, otherwise to the platform, so this organization does not receive it. Sign-ins blocked before the application (TLS, firewall, rate limit at the ingress) are not observed, and behind a proxy the source IP is the proxy's address. A CSV archive is written at the export pace (Clm:Siem:BatchSize events per Clm:Siem:Interval) with the same durable position: after a crash it continues after the last complete row, so a file never contains a row twice.

Plan the disk space (roughly 0.5–1 KB per event uncompressed, about a tenth with gzip); a full disk, a missing volume or an archive path with a symbolic link fails the archive with the reason shown on its card and the siem-delivery incident, and the position stays until the cause is fixed. The organization cannot choose the archive location. Removing an archive stops writing; its files stay until the operator deletes them. CSV archives cannot be defined in the server configuration. OTLP export uses OTLP/HTTP only (no gRPC) and one authentication header; a partial success is not resent (OTLP specification): it is journaled as siem.event.rejected with the event IDs of the request and raises siem-rejected, and Resend from sends the records again after the receiver is fixed. The field mapping applies to the canonical formats of network destinations only; file archives keep the complete event and the formats of the first release keep their layout. The enrichment fields trace_id, span_id, src_port, host_ip, dst_ip, dst_port, user_role and session_id are present only for entries recorded since this release (span_id and dst_port since the OTLP release). An organization's destinations and archive receive only its own events. They never receive platform records, even those concerning the organization: operator actions and requests, billing suspension and reinstatement, provisioning and re-encryption of stored secrets. The organization reads these in its console, and they leave only through the platform's own scopes, which the platform operator configures.

21 Agents and destinations

Prepare scoped outbound-agent deployment configurations without handling private keys in the browser.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Enter an agent ID and select Create enrollment code. The console shows the code's deadline (single use; ten minutes by default; the platform operator changes it with Clm:Agents:Enrollment:TicketLifetime), the address agents must use (the configured agent host and port, otherwise this console's HTTPS address), your organization ID and numbered installation steps with ready-to-copy commands for Linux (systemd, x64) and Windows (x64): when the agent address uses the operator's own CA, first download its root and verify the SHA-256 checksum shown next to the command (on Linux the following downloads of the same terminal trust it and the installer adds it to the system trust store itself through CLM_AGENT_TRUST_ROOT, removing it again when its connection check fails; on Windows the step imports it into the machine store before the download); copy the downloaded clm-enrollment.token into an empty working directory on the server; there download the package (the archive is served only with a valid enrollment code, which the command reads from the file, or to a signed-in console session through its own download links; a missing, expired or used code or a wrong organization ID ends in HTTP 401 and needs a new code), check its checksum, verify its publisher signature with verify-agent-package (replace <PUBLISHER-SHA256> in the commands with the publisher certificate fingerprint from your CERTYX provider; the console never shows it), extract it, in the clm-agent directory run the installer with ../clm-enrollment.token (Windows ..\clm-

enrollment.token) and run the status check (sudo clm-agent status on Linux). Download the token file from the first step; it is never displayed. Below the steps the console follows the new agent: waiting for enrollment, enrolled and waiting for the first contact, then "first contact received" with the time.

2. SSH workload: the same agent runs SSH automation. Grant "SSH operations" (and "SSH identity deployment" for installation) here or in SSH automation → Install and connect the SSH agent, which also shows the downloads, the host commands (configure-ssh-linux.sh / configure-ssh-windows.ps1 init, add-profile, trust-user-ca) and each agent's SSH workload report. sudo clm-agent status prints the granted capabilities.
3. Under Agent packages the console shows one card per platform (Linux x64, Windows x64) with the archive's file name, release version, size, SHA-256 (with a Copy button), the publisher signature state ("Signed (.p7s published)" or "Not signed: checksum only") and the release manifest state, plus a download button for every file the server publishes: the package, its .sha256 file, the .p7s signature, a .tsr time stamp, the release manifest and its .sig signature, and the separate verifier (verify-agent-package.sh or verify-agent-package.ps1). Every button saves the file under its own name. The card reports which signature files are published; it does not verify them, the verifier on the host does. A package the operator has not published is shown as "Not published on this server" with the operator's publishing step instead of a link. Compare the SHA-256 on the card with sha256sum (Linux) or Get-FileHash (Windows) of the downloaded file. The archive itself requires a valid enrollment code (the console's commands) or a signed-in console session, while checksums, signatures and verifiers are public. Verify it against the publisher certificate fingerprint obtained from your CERTYX provider through an independent channel; do not install a refused package. Follow the linked installation guide and pass the same fingerprint to the installer (CLM_AGENT_PUBLISHER_SHA256 or -PublisherSha256); use a trusted HTTPS control-plane address.
4. Run the commands on the target server before the deadline shown: the validity covers download and installation together; checking the code during the download does not use it up, enrollment does. The installer checks every argument before changing anything and names a wrong one with the next step (missing token file, placeholder or example address, non-HTTPS address, existing installation). It then checks the connection without using the code (name resolution, TCP, TLS trust including online revocation, clock difference, whether the address serves agents) and stops with the cause of the first failed step, explains enrollment failures (expired or used code), and ends by verifying that CERTYX accepts the new identity and has recorded its first contact. Run the status check again at any time: every line must start with [OK] and the last one must read "Result: connected"; a [FAIL] line names the cause and the next step, and the installation guide lists every message under Troubleshooting. Only when your operator publishes a release signing key, add ./release-signing.pub as a fourth Linux argument (daily clm-agent-update.timer) or -ReleaseKeyFile .\release-signing.pub on Windows (daily scheduled task CERTYX Agent Update, LocalSystem); it enables automatic signed updates.
5. Prerequisites for the agent address: HTTPS reachable from the agent host, a certificate for exactly that host name or address that the host trusts and whose revocation list it can download, and clocks within a few minutes. When the console is opened over plain HTTP without a configured agent host (for example the local ./start.sh) or through a proxy without one, no agent address is offered and the view explains why. For a local test, start CERTYX with CLM_AGENT_HOST=<address of this computer that the agent host reaches> ./start.sh: it adds an HTTPS agent listener with a local CA and revocation list, and the console then shows the address and the trust step. The agent creates a local private key and registers its public identity; discovery is allowed by default. Registered agents show the first and last contact, or that no contact has arrived yet. The agent reports a network map at start and then every hour (NetworkMapIntervalMinutes): the private IPv4 networks of its interfaces and routing table and private neighbour addresses. Public ranges are never reported or suggested.

6. In Automatic network discovery select the agent. Discovered networks lists the reported networks; use Add to scope per network or Add all (private). Networks wider than the allowed prefix (default /16) cannot be added; type narrower ranges. Add exclusions, and type public ranges manually only when they belong to you. The renewal readiness also uses the reported networks: when exactly one agent's network contains the address where a certificate was observed and that agent has the lifecycle workload, it is suggested as the installing agent of a new target.
7. Choose the mode: Standard checks which hosts are alive on the liveness ports and then tries the common TLS ports; Full also tries every remaining TCP port 1–65535 on live hosts. Set the connection attempts per second (1–2000), the simultaneous connections per host (1–4), the batch limits and, optionally, weekly scan windows (days, from, to, IANA time zone; an end before the start continues on the next day, 24:00 ends at midnight; no windows means any time).
8. Before saving, check What one pass checks below the ports. It counts the addresses one pass checks exactly as the agent does (ranges with 4 or more addresses skip their network and broadcast address, overlaps count once, exclusions are subtracted) and estimates the duration at the approved rate: the liveness check (each address is tried on the liveness ports until one answers; an address that does not answer waits the agents' connection time-out per port, 500 ms by default, Clm:Discovery:AgentDefaults:ConnectTimeoutMilliseconds), the common TLS ports per live host and, in Full mode, every other TCP port per live host. Example at 50 attempts per second: a /24 (254 addresses) needs 5–25 s for liveness and well under a second per live host for 16 common ports, while Full mode needs at least about 22 minutes per live host and up to about 9 hours for a host whose firewall drops connections instead of refusing them; if all 254 addresses answered, one full pass would take days. Start with Standard mode and run Full mode in a scan window outside working hours. A scope with no address to check is refused; a single address ending in .0 or .255 (for example 10.10.10.0/32, the network address of 10.10.10.0/24) and saving a Full mode profile ask for confirmation with the estimate.
9. Full mode, a range shorter than /20 or a public range opens a warning that must be confirmed: full discovery connects to every address and port, so intrusion detection systems will see it and may block the agent or alert your security team. Agree the scan with your security team and tell them the agent addresses, the windows and the rate so they can register it as an approved scanner. The server records who confirmed it and when; the profile shows it. Saving such a profile without confirmation is refused. An earlier confirmation stays valid while you add nothing that would need a new one.
10. Enable the profile and save. Discovery progress shows the phase (liveness, common ports, all ports), progress, live hosts, certificates found, waiting for a scan window (with the next window) and a reduced rate after many connection time-outs. The first complete pass of a scope is the baseline and raises no alerts; afterwards unknown certificates, replaced certificates and new TLS endpoints appear in Alerts. Changing networks, exclusions, mode or ports starts a new baseline; changing rate, windows or batch limits does not. When a pass completes, the progress and the agent log (journalctl -u clm-discovery.service) show the outcome of that pass: addresses checked, addresses that answered and TLS certificates found; Scan history in TLS discovery keeps every pass.
11. Unix lifecycle agents require owner-only state, secret files and execution journals without symbolic paths or symbolic parent directories. Unsafe or oversized journals stop execution for administrator reconciliation. Startup does not change permissions on an existing state directory. On Windows the agent applies the same rule to secret files and journals through ACLs: only LocalSystem, Administrators or the agent's own identity may own them and have access; anything else is refused. The Windows installer restricts the ACL of the agent directory accordingly.

12. Unix lifecycle agents can resolve approved Azure Key Vault, AWS Secrets Manager and Google Secret Manager text secrets through the customer-local provider CLI identity. Configure exact SecretBindings and run Clm.Agent secret-check with the protected lifecycle configuration as the service identity before enabling execution. Values stay local and are never printed. See CUSTOMER-VAULTS.en.md. Provider CLI resolution runs on Unix agents only; on Windows only protected-file secrets are available. Private keys are generated and held in software on the agent host (no HSM), and the agent does not send e-mail from the customer host.
13. HashiCorp Vault KV is also supported through an exact HTTPS mount/path#field binding and a customer-local native token helper. TLS verification is required; namespaces are not supported. A read fails when access is denied, the identity is revoked or the Vault certificate is not trusted. The Vault token comes from the customer's token helper; CERTYX does not create or renew Vault identities.
14. For customer-controlled remote KRL transfer, run ssh publish-krl with protected publisher configuration and serve only its public envelope through a separate HTTPS relay identity. Run ssh sync-krl with a protected consumer subscription. Signed publications expire after one hour and are refreshed automatically; redirects, bad trust, oversized data, tampering and expired publications are refused. Existing revocations remain after failures. OpenSSH does not enforce envelope expiry itself. Central health is reported only by consumers enrolled for KRL reporting; CERTYX does not determine which hosts are expected to consume the KRL. Old envelopes without signed time bounds must be re-exported. A fresh CA can initialize an empty KRL before issuing; loss of KRL on a used CA requires recovery.
15. For central visibility, enroll a separate consumer identity and explicitly enable KRL consumer reporting; it grants no SSH issuance or host-installation capability. Run ssh monitor-krl with the protected identity and subscription configuration. Agent health shows the reported revision, signed publication expiry, synchronization errors and missed reporting windows; healthy discovery cannot hide a KRL error. Platform monitoring includes those agent errors. These are authenticated agent observations, not independent proof that every host rejects the revoked identity. Use one registered identity per consumer.
16. Persisted lifecycle results are retransmitted after temporary connection failures, request timeouts and HTTP 408/429/500/502/503/504, using PollSeconds and the OperationTimeoutSeconds bound. The same job, lease and result are reused; issuance and installation are never replayed. Restart recovery renews the lease while delivering the saved result. Authentication, authorization, expired/reconciled leases and TLS trust failures stop for reconciliation. A lost lease cannot be renewed by this mechanism; an outage exceeding the lease or operation limit still requires administrator reconciliation.
17. All agent JSON configuration readers reject symbolic files or parent directories, Unix group/other permissions, files over 1 MiB and invalid UTF-8 before deserialization. Protect the configuration with chmod 600 on Unix. Relative command-line paths are resolved to an absolute path and receive the same checks. UTF-8 with or without BOM is supported. Settings and asynchronous readers use the same rules; no configuration values are included in failure output. On Windows the readers refuse a file that a principal other than LocalSystem, Administrators or the agent's own identity owns, can read or can change; provision restrictive ACLs before deployment.
18. Private endpoints can use a separate enrolled endpoint verifier. The operator binds its ID to the execution target; under Registered agents select Enable certificate lifecycle execution on the installer and Enable independent endpoint verification on a different agent identity. The verifier checks approved local TLS/HTTP endpoints before success is accepted. Setup takes one command on the verifier host: the target card in Execution → Deployment targets shows Set up the independent endpoint verifier with the grant, Download verifier scope (target ID, verification host, port, health path and trust

roots of the target; public data only) and the command configure-verifier-linux.sh add-target (Linux, clm-endpoint-verifier.service under the agent account) or configure-verifier-windows.ps1 add-target (Windows, scheduled task CERTYX Endpoint Verifier as LOCAL SERVICE). The script writes the owner-only configuration from the agent's own identity, runs one verified cycle, starts the service and waits up to ten minutes for the grant while CERTYX answers HTTP 403; remove-target, disable and status manage it, and the uninstaller removes it. A new agent receives the role during installation (CLM_AGENT_ENDPOINT_VERIFIER on Linux, -EndpointVerifierTarget on Windows). Without trust roots in the target the verifier host must trust the issuing CA itself.

19. Enrolled installers report their own Lifecycle workload (starting, idle or error). Missed reports remain visible even with healthy discovery. Process health does not replace the execution result and independent endpoint proof. Inspect protected local journals when a cycle stops for reconciliation.
20. Discovery retries transient identity-rotation and communication failures after Settings.RetrySeconds. Authentication, authorization and TLS trust failures stop the worker; --once returns failure. Configure owner-only state and lock paths. A reporting outage can delay diagnostics; monitor missed contact as well.
21. Destination drafts: save a draft (agent, target ID, DNS names, CA type and URL, Vault mount and role, verification endpoint, lifetime, Deployment connector and Installation; for an SSH relay also the remote host, SSH account and port), download its configuration bundle for the agent host, and choose Create target from this agent to add the organization's deployment target with the same scope and installation mode under Execution. The bundle contains the connector's documented default files and commands and, for a relay, the Remote object and RelayDestinations; replace its placeholders (SET-HOST-KEY-FINGERPRINT, SET-ACCOUNT-EMAIL, YOUR-CLM-HOST) before installation. Creation is refused when a target with that ID exists or the draft fails target validation; edit the target in Execution afterwards (for example to add private CA trust roots).
22. SSH relay to hosts without an agent: grant Enable certificate lifecycle execution to the relay agent. On the remote host run configure-relay-host-linux.sh from the agent package with the relay's public key and the helper printed by Clm.Agent relay-helper (account with key login only, authorized_keys with restrict and from=, helper, allowed directories, minimal sudoers rule); pin the printed host key fingerprint in HostKeyFingerprints and the known_hosts file on the relay; then run Clm.Agent relay-check. The relay's SSH identity and known_hosts file stay owner-only on the relay and are never sent to CERTYX; reports carry only the remote host name. Step-by-step: agent installation guide, "SSH relay".
23. Each registered agent shows its diagnosis (cause and explanation, with Why, and what to do leading to Agent health) and its latest discovery pass (start, state, live hosts, endpoints and certificates found, with Scan history).

INPUTS & PARAMETERS

- Organization-scoped settings; no private keys or CA credentials are accepted.
- Key/deployment roots must be absolute and link-free. Private keys and recovery journals require owner-only permissions; linked/shared/oversized state is refused. Only the managed current-generation UUID link is allowed.
- Discovery limits come from the server configuration Clm:Discovery:AgentLimits (defaults: prefix at least /16, at most 65,536 addresses per range and 262,144 per profile, 64 ranges, 256 exclusions); default ports, rate and time zone come from Clm:Discovery. Loopback, link-local, multicast and reserved ranges are refused. Only IPv4 is scanned; IPv6 neighbours are liveness hints only.

- Formats (each field shows its own help): agent and destination IDs 1–100 letters, digits, hyphens and underscores (web-prod-01); networks and exclusions one IPv4 CIDR per line written with its network address (10.20.0.0/16, a single address as 10.20.0.1/32); ports as a list such as 443, 8443 (1–64 common, 1–16 liveness, each 1–65535); scan window times HH:mm (24:00 only as an end); IANA time zone such as Europe/Warsaw; DNS names without wildcards, URLs, ports or IP addresses; CA API URL starting with https://. Network profile: Connection attempts per second 1–2000, Simultaneous connections per host 1–4; under Batch limits, Parallel connections 1–64, Attempts per batch 1–10000 and Pause between passes 10–86400 seconds. Destination drafts: Verification port 1–65535 and Certificate lifetime in hours 1–8760.

OPERATION RESULT

Audited, durable administration configuration.

PERMISSIONS

Administrators create enrollment codes, see registered agents and discovered networks, grant or withdraw capabilities, revoke agents, edit network profiles and save destination drafts or create targets from them. Other signed-in users of the organization see only the destination configuration drafts and a note that administrators manage the rest; Agent health remains available to them.

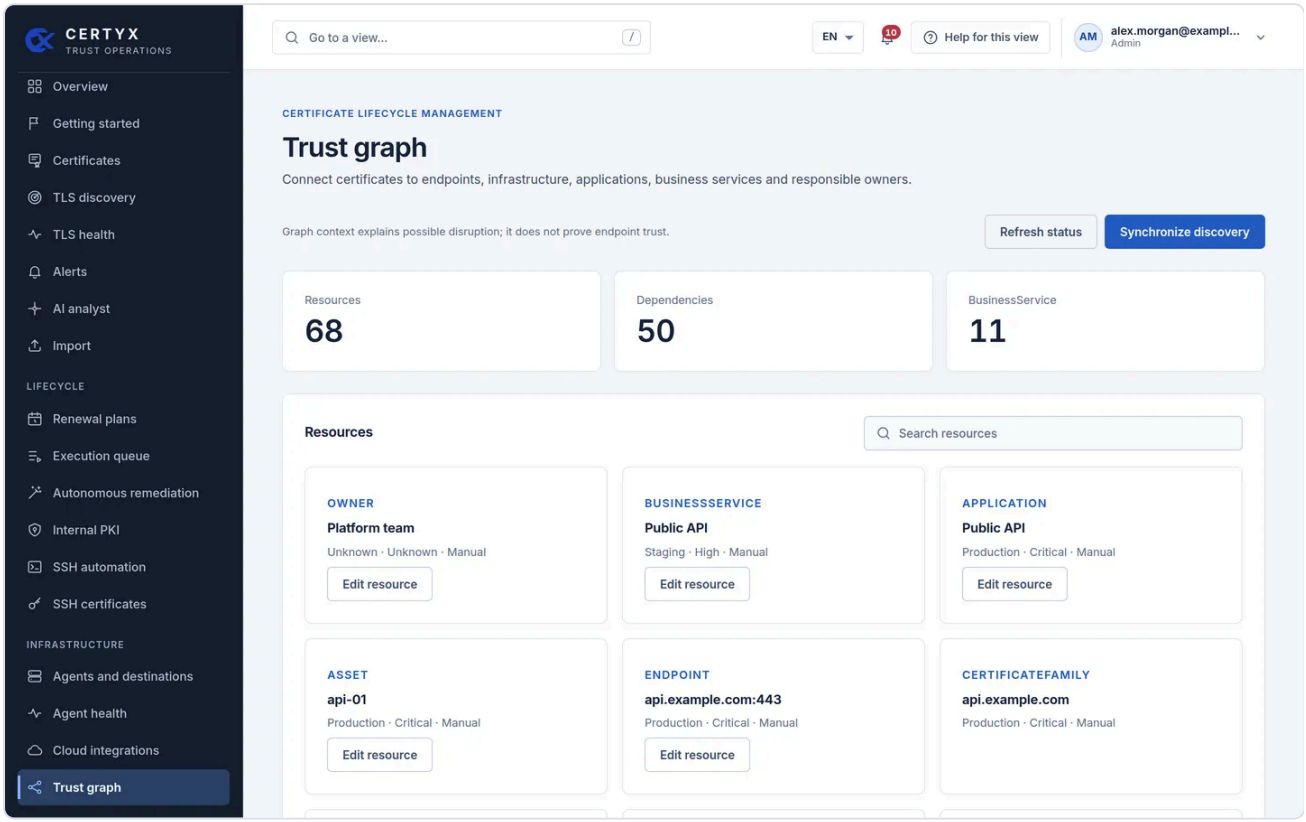
LIMITATIONS

Direct TLS discovery supports IPv4 TCP. The agent scans in a fixed, documented order at a steady rate; pacing, windows and the reduced rate after time-outs limit the load on your network and are not meant to hide the scan from security tools. The server re-checks every observation against the approved profile. To recognise the service, after the TLS handshake the agent sends one HEAD / HTTP/1.1 request to an endpoint that negotiated an HTTP protocol (or none on an HTTP port; h2 endpoints get it over one more HTTP/1.1 connection) or reads the SMTPS, IMAPS or POP3S greeting; Settings.ServiceIdentification.IdentifyServices false in the scan configuration switches this off. The agent does not probe UDP or STARTTLS services and does not enumerate SNI virtual hosts. An agent ID can be enrolled only once: a code for an already registered ID is refused, so a replacement agent needs a new ID; remove the old installation first with uninstall-linux.sh (in /opt/clm-agent/current) or uninstall-windows.ps1, which delete every unit or task, file, trust anchor and account the CERTYX scripts created. When the operator publishes a release signing key, releases also carry an offline-signed manifest: verify it with verify-release.sh (served at /downloads/ on the console address and on a separate agent address, together with the packages, their signature files and verify-agent-package.sh/.ps1) before installing; Linux and Windows agents installed with the release key update themselves daily (Linux clm-agent-update.timer, Windows scheduled task CERTYX Agent Update), only to a newer signed version, and roll back when the restarted agent is not healthy. Packages are signed by the publisher: a CMS signature over the archive hash and over SHA256SUMS of every packaged file, plus Authenticode on Windows executables, CLM assemblies and PowerShell scripts. The Linux and Windows installers verify the package against the pinned publisher fingerprint and refuse to install an unsigned, modified or foreign package; the development override (CLM_AGENT_ALLOW_UNSIGNED=1 or -AllowUnsignedDevelopment) prints a warning and must not be used in production. Agents installed without the release key are upgraded in place with the upgrade scripts described below, because the installers refuse a host that already has an enrolled agent. Customer credentials and private keys remain customer-controlled. Secrets resolve from protected files, HashiCorp Vault KV, Azure Key Vault, AWS Secrets Manager and Google Secret Manager; other vault products are not supported, provider CLIs run on Unix agents only, and the agent does not send e-mail from the customer host. Unix Vault token permissions are rechecked on each use. Discovery identities rotate automatically 30 days before expiry by default (IdentityRenewalDays in the agent configuration Settings), with keys kept

locally and recovery of lost acknowledgments. Deployment identities in the file registry require controlled manual rotation. Customer-local connector locks must be regular owner-only files on Unix, with no symbolic parents. Unsafe existing locks are refused; an administrator must correct storage permissions before retrying. These connectors run on Unix agents only. For an explicitly configured Unix AWS CLI transport, set AwsCliExecutable in the customer-local AWS destination to a trusted absolute executable path. CLI history must be disabled. Certificate imports use private standard input; credentials stay local. SDK remains the default transport. GCP lifecycle uses customer-local resource/identity configuration and gcp-adopt baseline preparation; see GCP-LIFECYCLE.md. Credentials are not job fields. The Google Certificate Manager target itself is registered in Execution under Deployment targets. When this server publishes a package without a publisher signature (an unsigned development build), the installation commands say so, skip the signature file and check the checksum only, with the development overrides of the verifier and installer; use such packages only for development. Installed agents are upgraded in place with upgrade-linux.sh or upgrade-windows.ps1 from a newer package (installation guide, "In-place upgrade"), without a new enrollment code; download the newer archive through the signed-in console and copy it to the host. upgrade-linux.sh also adds AgentConfigurationFile to an update.json written by an older installer, so its signed updates can download archives again. An SSH relay installs TLS server certificates only and needs SSH access, OpenSSL 1.1.1 or later and a POSIX shell on the remote host; Windows store and IIS install on a Windows agent host only.

22 Trust graph

Connect certificates to endpoints, infrastructure, applications, business services and responsible owners.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Select Synchronize discovery (administrators) to take over observed TLS endpoints: for every endpoint observation of an inventoried certificate the graph gets an infrastructure asset (server), its endpoint (host and port) and the certificate family, linked as asset → endpoint → family; a newer observation moves the endpoint to the newly observed family. The server also runs this synchronization automatically for licensed organizations (Clm:Assets:SynchronizationInterval, default every minute). Existing resources keep their manual data.
2. Under Manage business context select Add resource: choose the resource kind (Asset, Endpoint, Application, BusinessService or Owner), name, environment, criticality, status and owner; endpoints need a host name and port, owners an e-mail and owner type. Save resource; Edit resource on a card in the Resources list loads it into the form (certificate families come from discovery and cannot be edited).
3. Under Dependencies add a dependency from the Dependent resource to the Dependency target it depends on (for example application → endpoint, business service → application).
4. Under CMDB enrichment enter the CMDB source and choose a CMDB JSON file (up to 100 KB, format as in the JSON schema example), then Import CMDB JSON. Imports are incremental: the same source and external IDs update the same resources, and manual criticality overrides are kept.
5. Under Business impact select a certificate family and Analyze impact to see the affected resources, dependency paths, owners, the resulting criticality and warnings about incomplete context.

INPUTS & PARAMETERS

- Versioned typed resources; incremental CMDB JSON enrichment with source identity.
- Prerequisite: the organization's subscription includes the trust graph (TRUST_GRAPH). Without it the view shows only a note and the server refuses the trust-graph requests.

OPERATION RESULT

Explainable business impact with criticality, owners and incomplete-context warnings.

PERMISSIONS

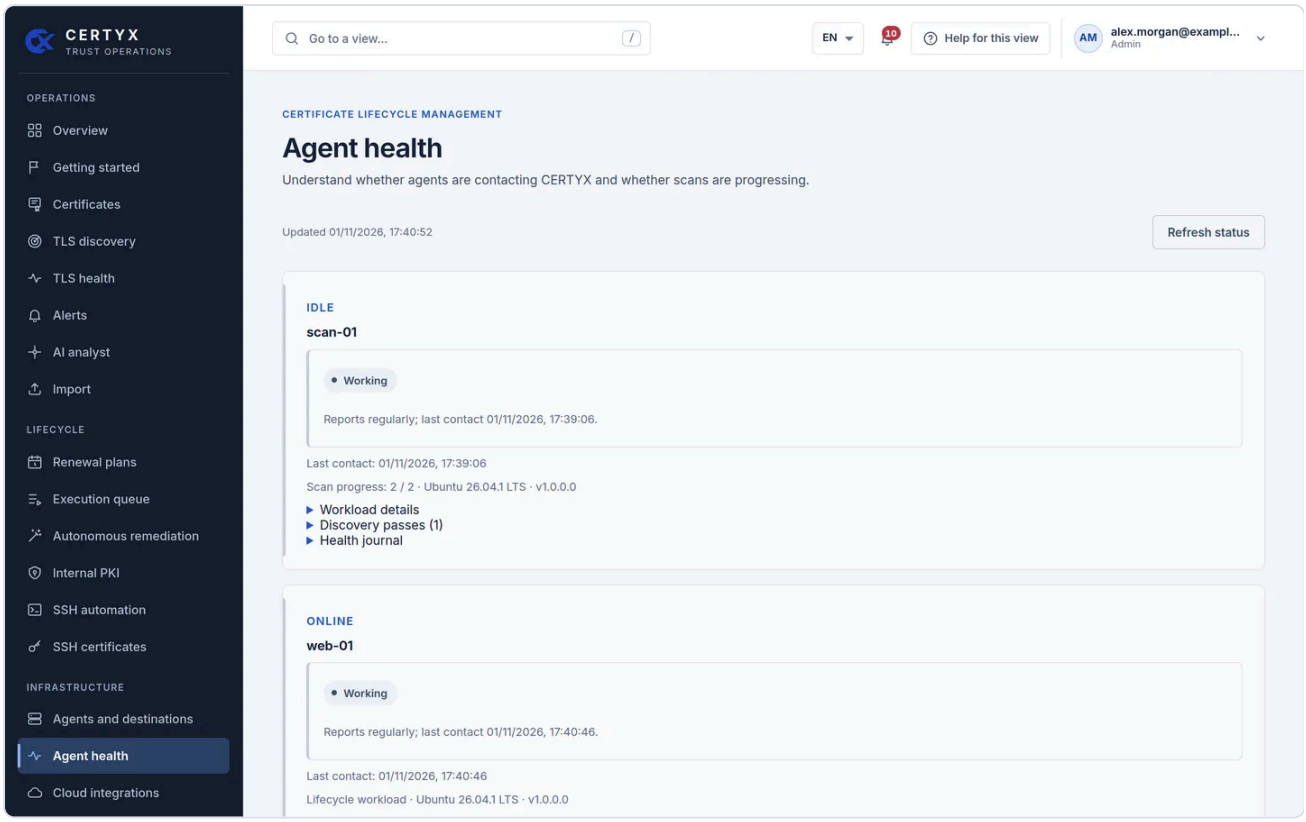
Requires the trust-graph license. Administrators synchronize discovery, edit resources and dependencies and import CMDB data; other signed-in users of the organization search resources and analyze impact read-only.

LIMITATIONS

CMDB import is a normalized push connector. Vendor-specific pull adapters are not implemented. Unknown criticality is not low risk; graph context does not verify TLS trust.

23 Agent health

Understand whether agents are contacting CERTYX and whether scans are progressing.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Register and install a network agent.
2. Assign an approved profile.
3. Review authenticated last contact, progress and error reasons.
4. Expand Workload details for a discovery agent to see its phase (liveness, common ports, all ports), live hosts, a reduced rate after connection time-outs, the next scan window while it waits outside the approved windows (state Waiting for a scan window) and when the baseline pass completed.
5. SSH-only agents do not require a discovery profile. Online means authenticated recent contact, not successful installation. Check SSH jobs and revocation results separately; missed contact becomes offline.
6. Managed SSH reports its own safe cycle status. A healthy discovery process cannot hide a failed or silent SSH worker. Review the error in Agent health and protected local journals; fixed summaries contain no secrets. Loss of connectivity can prevent immediate reporting, so missed-contact monitoring is also required.
7. Expand Workload details to see SSH cycle state and last report; SSH metadata is not labeled as scan progress.

8. For central visibility, enroll a separate consumer identity and explicitly enable KRL consumer reporting; it grants no SSH issuance or host-installation capability. Run ssh monitor-krl with the protected identity and subscription configuration. Agent health shows the reported revision, signed publication expiry, synchronization errors and missed reporting windows; healthy discovery cannot hide a KRL error. Platform monitoring includes those agent errors. These are authenticated agent observations, not independent proof that every host rejects the revoked identity. Use one registered identity per consumer.
9. A newly authorized KRL consumer remains unconfigured until its first report; another healthy workload cannot imply synchronization. State and reason must agree. Reports are tenant-isolated and a removed capability rejects later updates.
10. Private endpoints can use a separate enrolled endpoint verifier. The operator binds its ID to the execution target; in Agents and destinations, under Registered agents, select Enable certificate lifecycle execution on the installer and Enable independent endpoint verification on a different agent identity. The verifier checks approved local TLS/HTTP endpoints before success is accepted; its workload appears in Workload details. Setup takes one command on the verifier host: the target card in Execution → Deployment targets shows Set up the independent endpoint verifier with the grant, Download verifier scope (target ID, verification host, port, health path and trust roots of the target; public data only) and the command configure-verifier-linux.sh add-target (Linux, clm-endpoint-verifier.service under the agent account) or configure-verifier-windows.ps1 add-target (Windows, scheduled task CERTYX Endpoint Verifier as LOCAL SERVICE). The script writes the owner-only configuration from the agent's own identity, runs one verified cycle, starts the service and waits up to ten minutes for the grant while CERTYX answers HTTP 403; remove-target, disable and status manage it, and the uninstaller removes it. A new agent receives the role during installation (CLM_AGENT_ENDPOINT_VERIFIER on Linux, -EndpointVerifierTarget on Windows). Without trust roots in the target the verifier host must trust the issuing CA itself.
11. Enrolled installers report their own Lifecycle workload (starting, idle or error). Missed reports remain visible even with healthy discovery. Process health does not replace the execution result and independent endpoint proof. Inspect protected local journals when a cycle stops for reconciliation.
12. Discovery retries transient identity-rotation and communication failures after Settings.RetrySeconds. Authentication, authorization and TLS trust failures stop the worker; --once returns failure. Configure owner-only state and lock paths. A reporting outage can delay diagnostics; monitor missed contact as well.
13. Each card starts with the diagnosis: the exact cause (Working, Not reporting, No first contact yet, Identity expired or expires soon, Clock skew, Workload error, Workload not reporting, Configuration missing, Discovery paused, Waiting for a scan window, Reduced scan rate, Observation refused), a plain-language explanation with its facts (last contact and the reporting window it missed, the workload and the fixed reason it reported, the expiry date, the measured clock difference) and What to do: the remedies in order, each with a link to where it is done, plus the troubleshooting guide. #agent-health/<agent ID> opens one agent directly (from the overview, Agents and destinations, Renewal plans and the renewal readiness).
14. Health journal lists every change of the diagnosis, newest first, with the time, the new and the previous cause and the explanation. A health watch evaluates every agent every Clm:Agents:Health:WatchInterval (1 minute by default) and records each change; it is also written to the security journal as agent.health.changed with the explanation and remediation steps, so the platform operator and the organization's SIEM export receive it. Discovery passes shows the agent's latest passes (see TLS discovery, Scan history).

INPUTS & PARAMETERS

- Server receipt time, certificate expiry and bounded authenticated agent telemetry.
- An agent becomes Offline, and a workload report is marked as an error, when nothing arrives within its reporting window: the profile or workload interval plus Clm:Agents:Health:ReportingGrace (default 1.5 minutes), at least Clm:Agents:Health:MinimumReportingWindow (default 2 minutes).

OPERATION RESULT

One card per registered agent with its state (Online, Awaiting first contact, Awaiting configuration, Paused, Waiting for a scan window, Offline, Error, Identity expired, Revoked), the reason, the last contact, the primary workload with operating system and agent version, and Workload details per reported workload. The list refreshes automatically while the view is open; Refresh status reloads it at once.

PERMISSIONS

Authenticated organization users.

LIMITATIONS

Offline means contact is missing; the platform cannot know the precise cause of an unreachable machine. Agent-reported operating system is not hardware attestation. Discovery identities rotate automatically 30 days before expiry by default (IdentityRenewalDays in the agent configuration Settings), with keys kept locally and recovery of lost acknowledgments. Deployment identities in the file registry require controlled manual rotation. Linux agents installed with the operator's release signing key update themselves daily from signed releases; other Linux agents and all Windows agents have no in-place update: the installers refuse a host that already has an enrolled agent, so a newer version is installed by replacing the agent (revoke it, remove the installation as described in the installation guide and enroll the verified new package under a new agent ID).

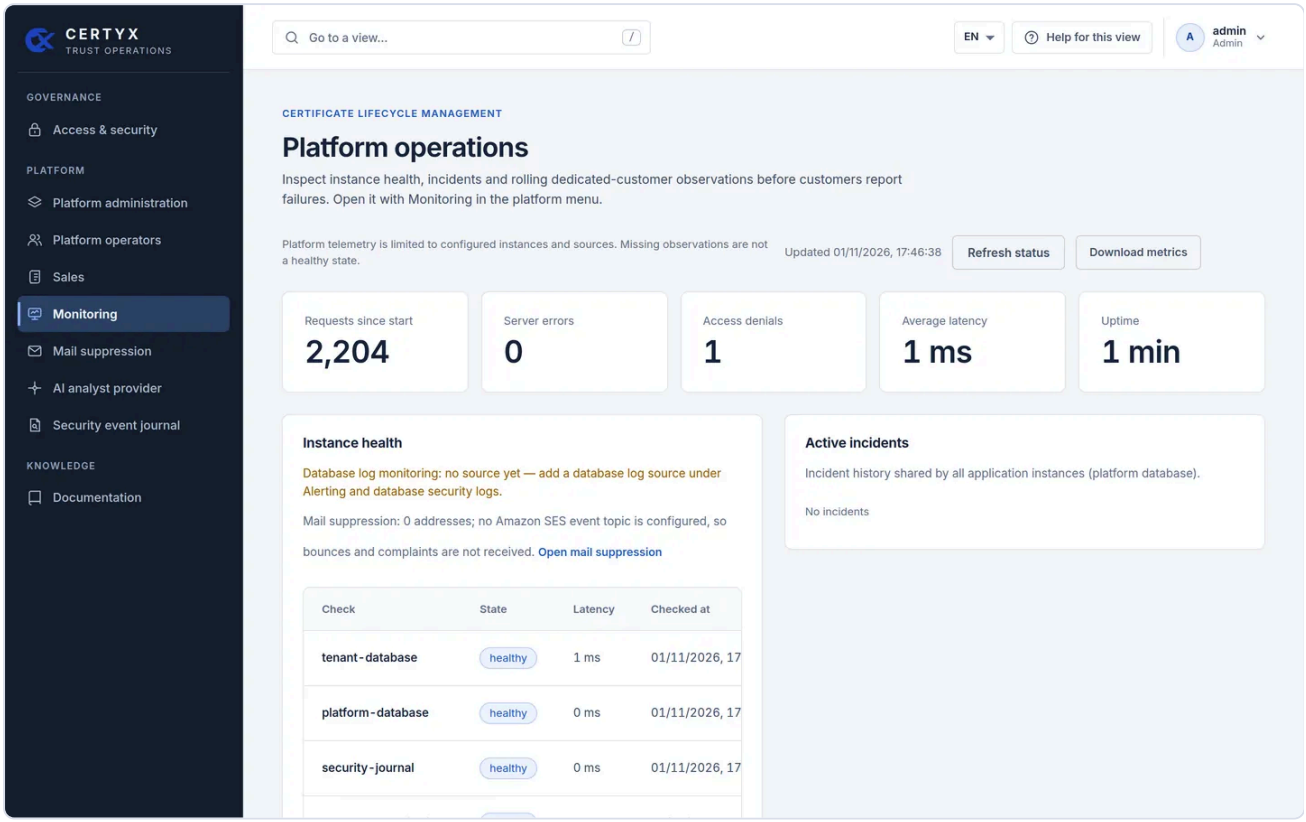
TROUBLESHOOTING

- Not reporting: start the service (sudo systemctl status clm-discovery.service; clm-ssh.service for SSH; the scheduled task on Windows) and read sudo clm-agent log; run sudo clm-agent status, which names the failing step (service, address, TLS trust with revocation, clock, identity); check firewall, proxy and DNS towards the agent address; install the private root of the agent address when TLS trust fails; synchronize the clock; check that the identity is not expired or revoked. On Windows read the task logs C:\ProgramData\CLM-Agent\state\discovery.log and C:\ProgramData\CLM-SSH\state\ssh-workload.log (Get-Content -Tail 50) and run Clm.Agent.exe status agent.json in an administrator PowerShell.
- No first contact yet: run the install command shown after the enrollment before the code expires (Agents and destinations); create a new code when it expired; then sudo clm-agent status.
- Identity expired or expires soon: automatic rotation runs IdentityRenewalDays (30 by default) before the expiry; when it did not, enroll the agent again under a new agent ID and revoke the old identity.
- Clock skew: the agent's network map carries its clock; synchronize the host with NTP. TLS and identity checks fail when clocks differ by minutes (tolerance Clm:Agents:Health:ClockSkewTolerance).
- Workload error: the lifecycle workload failed — open the failed job in Execution queue and the protected local journal; discovery failed — check communication with sudo clm-agent status; the approved profile exceeds the local allowlist — widen AllowedCidrs or narrow the profile; KRL synchronization failed — check the customer relay and pinned trust; SSH cycle failed — check the SSH journal and SSH automation.

- Configuration missing: assign and approve a discovery profile, or configure and start the workload the capability was granted for (lifecycle, endpoint verifier, KRL consumer).
- Reduced scan rate: connection time-outs halved the rate; check whether a firewall or intrusion prevention blocks the agent, or lower the rate and connections per host.

24 Platform operations

Inspect instance health, incidents and rolling dedicated-customer observations before customers report failures. Open it with Monitoring in the platform menu.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Sign in as an allowlisted platform administrator with MFA.
2. Review the metric tiles of this instance (requests since start, server errors, access denials, average latency, uptime), Instance health (checks of the tenant and platform databases, customer monitoring and each database log source, with state, latency and check time; the line above the table shows the database log monitoring) and Active incidents. Refresh status reloads everything; Download metrics saves the Prometheus metrics of this instance.
3. Under Alerting and database security logs, add the managed PostgreSQL log of each database: AWS CloudWatch Logs (full RDS log group ARN), Azure Log Analytics (workspace GUID|Flexible Server resource ID) or Google Cloud Logging (project:Cloud SQL instance), with the customer ID for a dedicated customer database. Each source is read in the next monitoring cycle with the machine identity of the platform host (CloudWatch through the AWS SDK, so the host needs only read access to the log group; Azure and Google Cloud through their provider CLI), and authentication and authorization failures become incidents. With the database containers of the standard deployment nothing has to be added: both databases write a JSON security log that the application reads; the instance health line then says “local PostgreSQL log”. On AWS with RDS, add the CloudWatch source once with the log

group ARN that the infrastructure output database_log_group_arn names. An unreadable source appears as its own failed health check and in the source list; the instance health line shows how many sources are watched and whether the local PostgreSQL JSON log of the host is read.

4. Customer health lists dedicated customers with database state, customer status, offline agents, failed lifecycle jobs and the last observation (hover a row for the reason). Choose Acknowledge incident after investigation; acknowledgement records who owns the investigation and does not resolve the fault.
5. Alert channels: add Microsoft Teams (incoming webhook or workflow URL), Slack (Incoming Webhook URL; message layout Blocks or Plain text), PagerDuty (Events API v2 routing key), HTTPS webhook or e-mail channels with a minimum severity, then choose Send test alert; a failed test names the reason (relay missing, invalid destination, unreadable secret, refused, unreachable, timeout). E-mail channels use the alert e-mail relay (host, STARTTLS port, sender, optional user name and password). URLs, routing keys and passwords are write-only: enter a new value to replace one, or secret:<name> to read it from an approved platform secret binding; an empty field keeps the stored value. Incident openings and resolutions go to every channel whose minimum severity they reach; a channel that keeps failing raises the alert-delivery incident. One instance at a time sends alerts and each change reaches a channel once, also with several instances.
6. For central visibility, enroll a separate consumer identity and explicitly enable KRL consumer reporting; it grants no SSH issuance or host-installation capability. Run ssh monitor-krl with the protected identity and subscription configuration. Agent health shows the reported revision, signed publication expiry, synchronization errors and missed reporting windows; healthy discovery cannot hide a KRL error. Platform monitoring includes those agent errors. These are authenticated agent observations, not independent proof that every host rejects the revoked identity. Use one registered identity per consumer.
7. The AI analyst provider (Anthropic API key, model, connection test, quotas, offer to organizations) has its own control room view: AI analyst provider.
8. Under the offline count, each customer row lists the agents that need attention with their cause and since when (the same diagnosis the organization sees in Agent health); the customer's security journal entries agent.health.changed carry the explanation and the remediation steps.
9. Technical and sales mail are separate: the e-mail relay for alerts is the platform's technical sender (for example support@...) for alert e-mails and the authenticator setup e-mails of customer accounts; sales mail uses its own relay under Sales (for example sales@...). Send sample setup e-mail sends a recipient a preview of the setup e-mail in a chosen language through this relay; its link and QR code open no setup, and the sending is recorded in the security journal (mfa-setup.mail.sample-sent). The relay uses STARTTLS (port 587); implicit TLS on port 465 is not supported.
10. Under Instance health, the mail suppression line shows how many addresses are suppressed and the last Amazon SES event received (none yet is normal, events are sparse) or that no event topic is configured; Open mail suppression leads to the list. An alert e-mail channel whose address is suppressed is not retried and raises an alert-delivery incident naming the cause.

INPUTS & PARAMETERS

- Instance counters since process start; authenticated database probes; bounded incident history and customer observations.
- Alert channel: name (1–64 letters, digits, . _ -, starting with a letter or digit), kind, minimum severity, recipient e-mail or URL/routing key (a URL must start with https://; a PagerDuty routing key has up to 200 characters without spaces; either can instead be a secret:<name> reference), Slack message

layout (Slack only). Relay: host, port, sender, optional user name and password. Database log source: name, log service, location in the format shown for the service, optional customer ID (GUID). Minimum severity is Critical only or Warnings and critical; at most 20 alert channels and 50 database log sources.

OPERATION RESULT

Persistent incidents (shared by all instances when the platform database is configured, so an incident raised on one instance can be acknowledged or closed on another), protected Prometheus metrics and operational status. The incident panel shows whether the history is shared, local or unavailable. Versioned alert channel, relay and database log source settings in the platform database.

PERMISSIONS

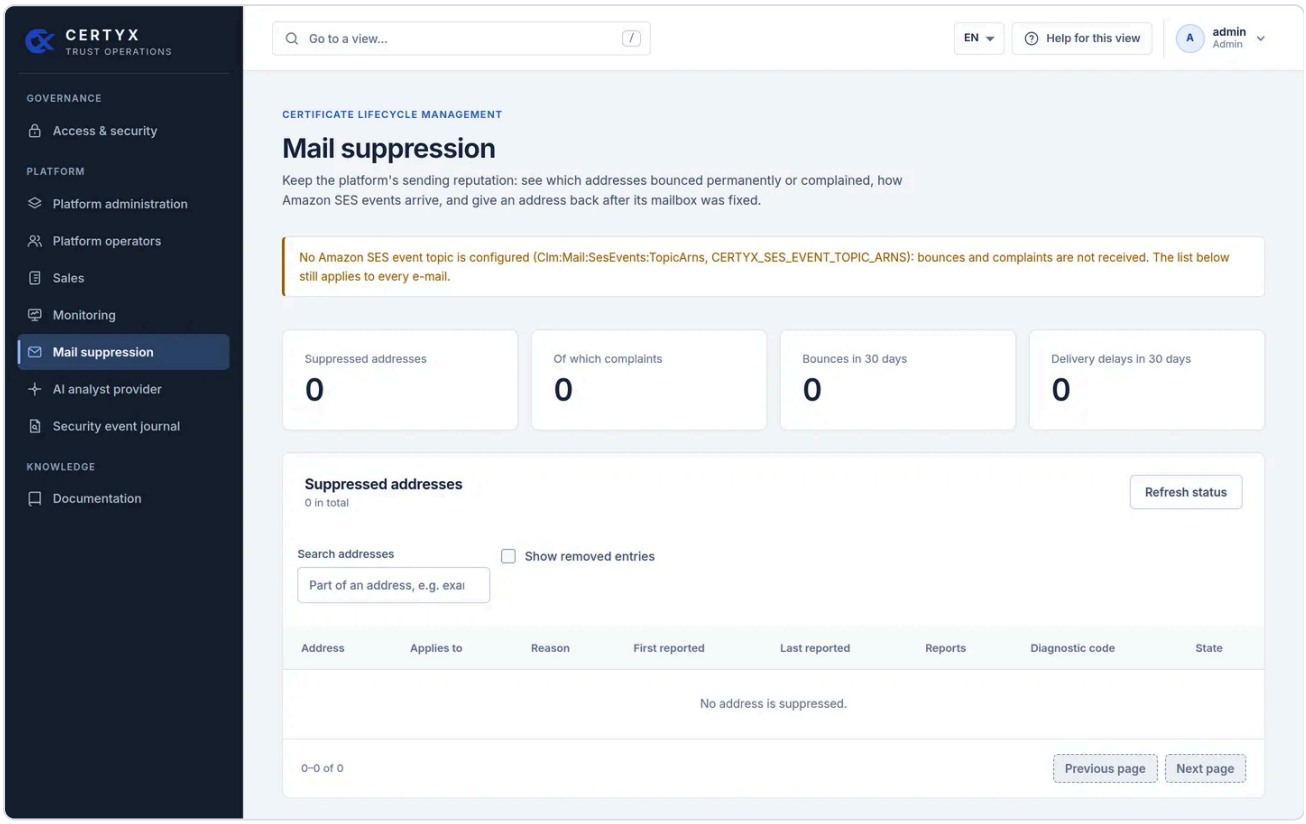
Allowlisted platform administrator with an MFA-authenticated session only. Every settings change and test alert is recorded in the security journal without secret values. Step-up: every change in this view (except tests, connectivity checks and acknowledging an incident) asks for the current 6-digit code of your own authenticator; a confirmed code also covers your further control-room changes for Clm:Identity:OperatorStepUpWindow (default 5 minutes). A wrong or replayed code is refused, counts toward your account lockout and is journaled; Cancel abandons the change. Operators who sign in through company sign-in have no local authenticator and cannot make these changes. Operator roles: support, security administrator and administrator see monitoring and handle incidents; the alerting settings (channels, relay, log sources) need security administrator or administrator.

LIMITATIONS

Counters reset on restart. This is detection of selected symptoms, not complete intrusion prevention. Each instance runs its own probes and metrics; without the platform database it also keeps its own incident history and sends its own alerts, so run a single instance then. While the platform database is unavailable, new incidents are not recorded until it returns. Incidents from before the upgrade or imported from the file are not alerted again, and a new channel receives only later changes. Managed log cursors restart after a restart, so an old failure can be reported again. The security journal is exported to SIEM destinations configured in Security event journal → SIEM export (platform scopes) and by each organization (SIEM export); full database audit collection is not part of this view. Dedicated customers are probed in rolling batches (Clm:Monitoring:CustomerProbeBatchSize, default 10 per cycle of Clm:Monitoring:ProbeInterval, default 30 seconds); timestamps expose stale observations. The incident history keeps at most Clm:Monitoring:IncidentRetention incidents (default 5,000), evicting resolved ones first. Alerting and log source settings need the platform database; the values of the configuration file are taken over once at the first start and the console is authoritative afterwards. The provider CLI paths, the local PostgreSQL JSON log and the alert delivery timeout are deployment settings. A successful test proves that the destination accepted the message, not that someone was paged.

25 Mail suppression

Keep the platform's sending reputation: see which addresses bounced permanently or complained, how Amazon SES events arrive, and give an address back after its mailbox was fixed.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Sign in as an allowlisted platform administrator with MFA and open Mail suppression in the control room. The status line says whether Amazon SES events are received (the number of allowed SNS topics and the last event), and the tiles show suppressed addresses, complaints among them, and bounces and delivery delays of the last 30 days.
2. Amazon SES reports every bounce, complaint, reject and delivery delay of the platform's mail to the SNS topic of the deployment, which posts it to https://<console>/public/api/mail-events/ses. CERTYX checks the SNS signature (certificate only from sns.<region>.amazonaws.com, chain to a public root, subject sns.amazonaws.com), the topic against Clm:Mail:SesEvents:TopicArns, the message age and the MessageId (a replay changes nothing), and confirms the subscription itself.
3. A permanent bounce or a complaint puts each named recipient on the list (reason, first and last report, number of reports, the provider's diagnostic code; never message content) and is journaled as mail.suppression.added. Temporary bounces, delivery delays and rejects are only counted. Each entry has a scope (column Applies to): mail the platform sends for an organization (its authenticator setup e-mails) carries the organization in the Amazon SES message tag certyx-scope, so its bounce or complaint suppresses the address for that organization only; platform mail (operators, alerts, sales) and

events without the tag (for example SES feedback notifications, or entries from before per-organization suppression) suppress it for all mail. An organization's mail checks the platform-wide entries and its own, never another organization's.

4. Every e-mail of the platform checks the list before sending: organization notification e-mail (suppressed recipients are skipped; the outbox entry says so, or is cancelled when nobody is left), test messages, authenticator setup e-mails (the console shows the link with the reason instead), alert e-mails (the channel raises an incident instead of retrying) and sales e-mails (the notice is marked Not sent: recipient suppressed). The operator copy of billing notices is left out when it is suppressed.
5. Search by part of an address; Show removed entries also lists removed addresses with who removed them, when and why.
6. Once the mailbox works again, choose Remove from suppression, enter the reason and confirm. The removal is journaled as mail.suppression.removed and CERTYX sends to the address again at once; a new bounce or complaint suppresses it again. Remove the address from the Amazon SES account-level suppression list as well (SES console → Suppression list, or aws sesv2 delete-suppressed-destination --email-address <address>), otherwise SES still drops messages to it. Removal applies to the entry of the row you chose (its scope); an address suppressed in several scopes keeps its other entries.

INPUTS & PARAMETERS

- Search: part of an address (up to 254 characters, case-insensitive).
- Reason for the removal: 3 to 500 characters without line breaks, for example: mailbox recreated by the customer on 14 October.
- Deployment settings (not in the console): Clm:Mail:SesEvents:TopicArns (on the host CERTYX_SES_EVENT_TOPIC_ARNS), the allowed SNS topic ARNs; MaxMessageAge (default 1 hour), MaxClockSkew (5 minutes), MaxBodyBytes (256 KiB), EventRetention (90 days) and SigningCertificateSubjects (sns.amazonaws.com).

OPERATION RESULT

One suppression list in the platform database, shared by every e-mail path and every organization, with a journal entry for each address added (actor system:amazon-ses) and removed (the operator). Organization administrators see their suppressed group members in Notification groups. Entries are scoped: platform-wide, or to the organization whose mail caused them.

PERMISSIONS

Only platform administrators with MFA see the list and remove addresses. Organization users see only the suppressed members of their own recipient groups (Notification groups) and cannot remove them: the list protects the shared sending reputation of the platform. Step-up: every change in this view (except tests, connectivity checks and acknowledging an incident) asks for the current 6-digit code of your own authenticator; a confirmed code also covers your further control-room changes for Clm:Identity:OperatorStepUpWindow (default 5 minutes). A wrong or replayed code is refused, counts toward your account lockout and is journaled; Cancel abandons the change. Operators who sign in through company sign-in have no local authenticator and cannot make these changes.

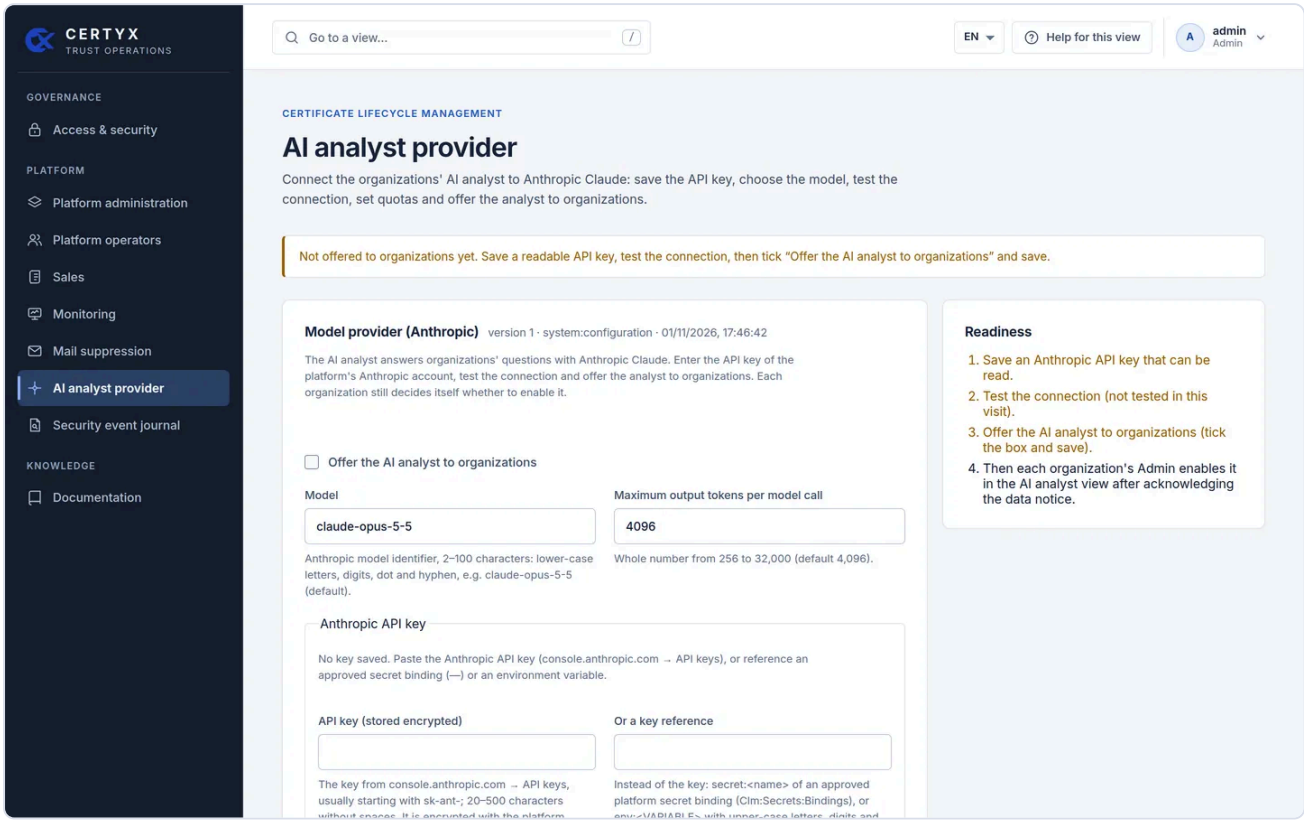
LIMITATIONS

Needs the platform database; without it nothing is suppressed by CERTYX (the list is shown as unavailable) and SES events cannot be received. Without an allowed topic the endpoint answers 404 and the status line says that no events are received; the list still applies. Events are sparse: no event for a long time is normal. While the platform database cannot be read, mail is sent without the check (Amazon SES's own account-

level suppression still applies). Only Amazon SES events through SNS are understood; other providers' bounces (for an organization's own relay) are not received. Removing an address here does not remove it from the SES account-level list. Suppression is per address for all organizations.

26 AI analyst provider

Connect the organizations' AI analyst to Anthropic Claude: save the API key, choose the model, test the connection, set quotas and offer the analyst to organizations.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Sign in as an allowlisted platform administrator with MFA and open AI analyst provider in the control room. The status line and the Readiness list show what is still missing: a readable key, a successful connection test and the offer to organizations.
2. Under Anthropic API key, paste the key from console.anthropic.com → API keys into API key (stored encrypted), or enter a reference instead: secret:<binding> for an approved platform secret binding or env:<VARIABLE> for an environment variable of the application. Fill in only one of the two; empty fields keep the saved key; Remove the saved key deletes it. A saved key is never shown again: the empty key field then reads “saved key — type a new one to replace it”, and the line above the fields names where the key comes from.
3. Choose Test connection. It sends one short request (a few output tokens, no tools, no organization data) with the values in the form — also an unsaved key or model — or the saved settings, and reports the outcome: Connection works, Key rejected (401), Not permitted (403), Model not found (404), Rate limit or quota reached (429), Billing problem (credit balance), Request rejected (400), Provider overloaded (529), Provider error, Provider not reachable (DNS, firewall, proxy, TLS) or No answer in time, with the HTTP status, the provider's error type and its message, the model and the duration. A missing or unreadable key is reported without sending anything.

4. Check the model (default claude-opus-5-5) and the output tokens, tick Offer the AI analyst to organizations and choose Save settings. The status line then says Ready and names the provider address.
5. Each organization's Admin still enables the analyst in its own AI analyst view after acknowledging the data notice; only then can its users ask questions.

INPUTS & PARAMETERS

- API key: 20–500 characters without spaces (Anthropic keys usually start with sk-ant-). Encrypted with the platform data protection keyring (shared by all instances through the platform database) and stored as protected:<ciphertext>; never shown again, never returned by the API, never journaled or logged.
- Key reference: secret:<name> of a binding listed in Clm:Secrets:Bindings (read with the machine identity when a question is sent) or env:<VARIABLE> (upper-case letters, digits, underscores).
- Model: 2–100 lower-case letters, digits, dots or hyphens, e.g. claude-opus-5-5. Maximum output tokens per model call 256–32,000 (default 4,096).
- Quotas: questions per user and hour 1–10,000 (default 30), questions per organization and day 1–100,000 (default 300), tokens per organization and day 1,000–1,000,000,000 (default 2,000,000).

OPERATION RESULT

A versioned settings document in the platform database (key analyst); each save is journaled as analyst.settings.updated with a stored key shown only as "stored", and each connection test as analyst.connection.tested with its outcome, HTTP status, model and key source. Organizations see the analyst as available once it is offered and the key resolves.

PERMISSIONS

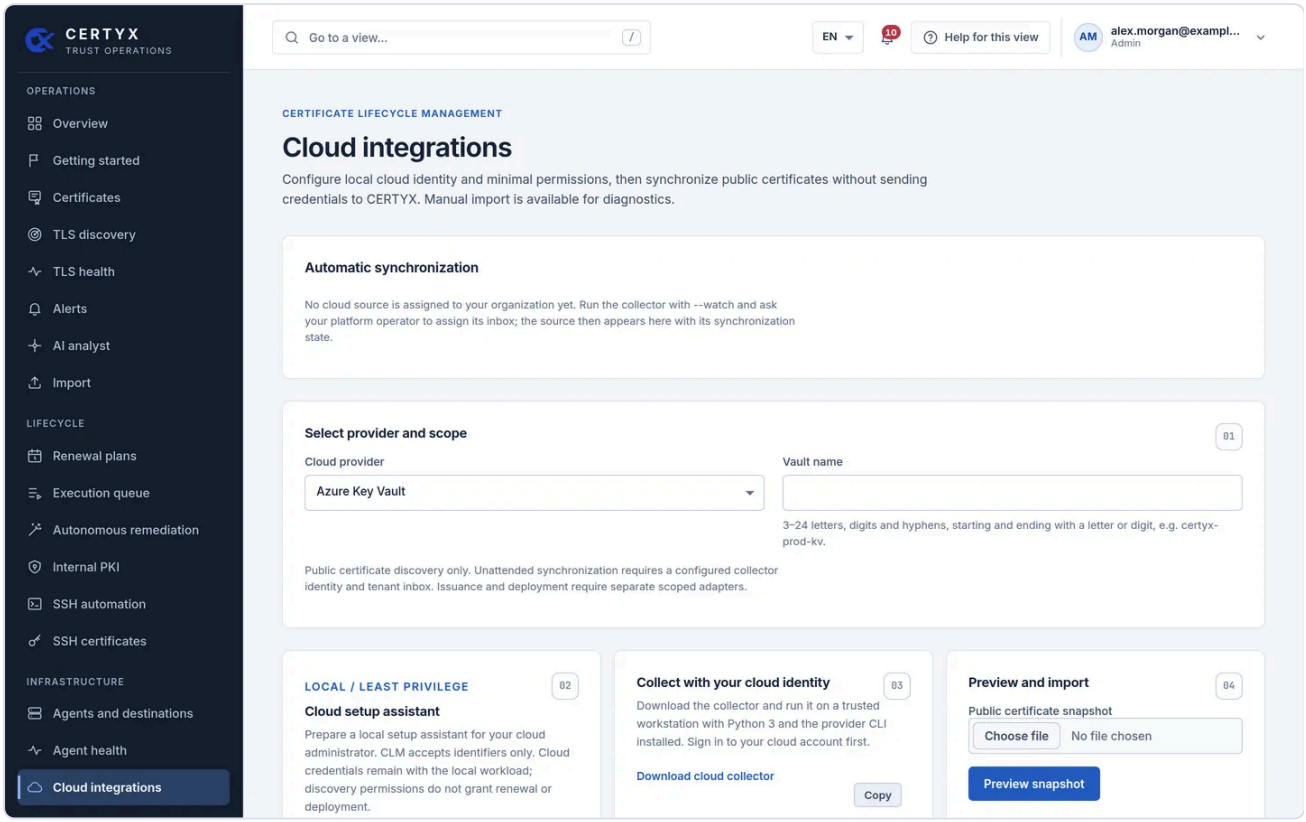
Platform administrators with MFA only. Organization users never see the key or this view. Step-up: every change in this view (except tests, connectivity checks and acknowledging an incident) asks for the current 6-digit code of your own authenticator; a confirmed code also covers your further control-room changes for Clm:Identity:OperatorStepUpWindow (default 5 minutes). A wrong or replayed code is refused, counts toward your account lockout and is journaled; Cancel abandons the change. Operators who sign in through company sign-in have no local authenticator and cannot make these changes.

LIMITATIONS

Without a platform database (standalone installation) the values from Clm:Analyst in the configuration are shown read-only and a key cannot be entered here; Test connection still checks them. A key encrypted with another keyring cannot be read and must be entered again. The provider address (Clm:Analyst:BaseUrl, default https://api.anthropic.com) is set in the configuration only. The connection test waits at most 30 seconds and does not retry. A successful test proves the key, model and network path at that moment; it does not evaluate answer quality. Data is processed by Anthropic under the operator's agreement with Anthropic.

27 Cloud integrations

Configure local cloud identity and minimal permissions, then synchronize public certificates without sending credentials to CERTYX. Manual import is available for diagnostics.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. Select AWS, Azure or GCP and the scope. Fill the local IAM setup assistant with public identifiers only. GCP requires an existing service account in the selected project; its one-permission role grants public listing, without creating account keys or federation.
2. Choose "Download setup configuration" (clm-cloud-setup.json; a wrong identifier is refused with its expected format) and "Download local assistant" (cloud-setup.py). Run plan locally to prepare minimal policies without changing your cloud.
3. Review policies. Run apply with an authorized cloud administrator identity and explicitly confirm the scope. This creates discovery permissions only.
4. Configure the collector workload identity, then run doctor under that identity. Failed checks return a nonzero exit code; empty scopes are reported as not fully tested.
5. In Collect with your cloud identity, download cloud-collect.py and run the displayed command after signing in with the provider CLI; it writes a new clm-cloud.json and refuses to overwrite an existing file.
6. For unattended synchronization, run the collector with --watch SECONDS (at least 300) and ask your platform operator to assign its owner-only inbox to this organization. Automatic synchronization then lists each assigned source with its state: waiting for the first synchronization, working, partial result,

failed with the reason, or paused by the operator, together with the certificate count, last success and last attempt. The panel refreshes every 30 seconds while the view is open. When no source is assigned yet, the panel says so. Manual snapshots remain available for troubleshooting.

7. Manual import (Preview and import): choose the snapshot file and "Preview snapshot". The server checks it without storing anything and shows provider, scope, certificate count, warnings and the certificates. "Import into this organization" then adds new certificates and reports added, already existing and partial results. Without the trust graph in the subscription only the preview is available.
8. Unix lifecycle agents can resolve approved Azure Key Vault, AWS Secrets Manager and Google Secret Manager text secrets through the customer-local provider CLI identity. Configure exact SecretBindings and run Clm.Agent secret-check with the protected lifecycle configuration as the service identity before enabling execution. Values stay local and are never printed. See CUSTOMER-VAULTS.en.md. Provider CLI resolution runs on Unix agents only; on Windows only protected-file secrets are available. Private keys are generated and held in software on the agent host (no HSM), and the agent does not send e-mail from the customer host.
9. HashiCorp Vault KV is also supported through an exact HTTPS mount/path#field binding and a customer-local native token helper. TLS verification is required; namespaces are not supported. A read fails when access is denied, the identity is revoked or the Vault certificate is not trusted. The Vault token comes from the customer's token helper; CERTYX does not create or renew Vault identities.
10. The AWS lifecycle manages certificates imported into ACM. On the customer Unix host, aws-adopt <protected-agent-config> <target-id> <protected-key-reference-json> prepares a protected rollback baseline after checking the matching key, real TLS endpoint and unchanged ACM fingerprint. This command only reads AWS; it does not replace the cloud certificate. Existing managed state requires recovery instead of adoption.
11. The AWS package includes agent.aws-vault.json, aws-baseline-key.json and AWS-LIFECYCLE.md. Replace placeholders locally; the Vault example issues private-CA certificates. A renewal job succeeds only after ACM holds the new certificate and independent TLS verification of the served endpoint passes. No credentials are entered into this view.
12. For an approved AWS imported certificate, aws-plan <protected-agent-config> <target-id> generates a minimal resource permission policy without contacting AWS. It does not grant role trust or create infrastructure. See AWS-LIFECYCLE.md before applying the plan.
13. The AWS ACM and Azure Key Vault store operations replace the certificate, read it back, restore the protected baseline when a change has to be rolled back and refuse a certificate whose key does not match. AWS renewal derives key algorithm/size and usage constraints from the protected baseline; incompatible migrations require a new resource.
14. DigiCert CertCentral: on the customer host, Clm.Agent digicert-collect <config> <destination-id> <output.json> reads purchased certificates with the locally held API key and writes a snapshot (provider digicert) to import here; warnings list domain and organization validations that expire within 60 days. Agent destinations of kind digicert order or reissue certificates within the configured purchase policy.
15. For Azure agent execution, configure the exact AzureKeyVault scope (VaultUrl, CertificateName, TenantId, ManagedIdentityClientId) on the server and customer agent, plus an independent VerifyHost. Use one user-assigned managed identity. On the customer host, azure-adopt <protected-agent-config> <target-id> <protected-key-reference-json> <protected-public-chain-reference-json> reads and verifies the existing certificate and prepares a protected rollback baseline. Both references resolve locally. Private keys are not exported from Azure or uploaded to CERTYX. See AZURE-LIFECYCLE.md.

Importing a vault version alone is not a successful deployment: the job succeeds only after independent TLS verification, and CERTYX does not switch application bindings of consuming Azure services to the new version. This cloud-generation workflow runs on Unix agents only.

16. GCP discovery: select Google Certificate Manager and enter the exact project ID. Run the displayed customer-local cloud-collect.py gcp PROJECT --location global command using an approved read identity. Import the public snapshot here or use --watch with an owner-only synchronization inbox. Use --location for a separately approved regional location. Missing public PEM, including certificates still provisioning, is reported as a warning. The IAM assistant creates a project-scoped public listing role for the existing GCP service account. Run doctor with explicit impersonation of that account; this requires approved federation or token minting authorization already configured on the customer side. No token-creator grants or account keys are created.
17. GCP setup records an owner-only gcp-apply-receipt.json before changing IAM. An existing receipt blocks repeated writes. Run cloud-setup.py reconcile --config CONFIG.json --output PLAN-DIRECTORY for read-only checks of exact permissions and binding. Customer IAM administrators resolve partial changes using provider audit logs; reconciliation never broadens permissions.
18. GCP baseline adoption (gcp-adopt <protected-agent-config> <target-id> <protected-key-reference-json> [protected-public-chain-reference-json]) accepts an optional protected public issuer-chain reference as its final argument. Use it if Google omits issuers; the agent validates the chain against approved trust roots before saving rollback state. Keys remain customer-local.
19. On Compute Engine, an agent can explicitly select gcpIdentityKind=metadata for its approved attached service account, with no credential reference or account key. IAM and VM OAuth scope must be configured by the customer. Federation and customer-vault alternatives remain available.

INPUTS & PARAMETERS

- Public PEM certificates and cloud resource references; no credentials, secrets or private keys.
- Snapshot file: JSON of at most 5 MB with at most 250 certificates; snapshots older than 30 days are rejected. Providers accepted in snapshots: azure, aws, gcp and digicert.
- Scope and identity formats (shown under each field; a wrong value is refused with the expected format): Azure vault name 3–24 letters, digits and hyphens; AWS region such as eu-central-1, account ID of 12 digits and role arn:aws:iam::ACCOUNT:role/NAME; GCP project ID of 6–30 lowercase letters, digits and hyphens; Azure vault resource ID /subscriptions/.../providers/Microsoft.KeyVault/vaults/VAULT-NAME and object ID as GUID; GCP service account NAME@PROJECT-ID.iam.gserviceaccount.com of the selected project.
- Permissions the assistant prepares (plan) and grants (apply): AWS — a customer role trusted only by the given workload role with a unique external ID (sts:AssumeRole), allowed acm:ListCertificates in the selected region and acm:DescribeCertificate and acm:GetCertificate on certificates of that account and region; Azure — a custom role with the single data action Microsoft.KeyVault/vaults/certificates/read, assigned to the managed identity on the one vault; Google Cloud — a project role with the single permission certificatemanager.certs.list, bound to the existing service account. No secret, private-key, issuance or deployment permission is granted. Snapshot scopes are reported by the collector as the Azure vault name, AWS ACCOUNT:REGION and GCP PROJECT:LOCATION.

OPERATION RESULT

Certificates appear in inventory and cloud resource dependencies in the trust graph. Repeated imports do not duplicate certificates.

PERMISSIONS

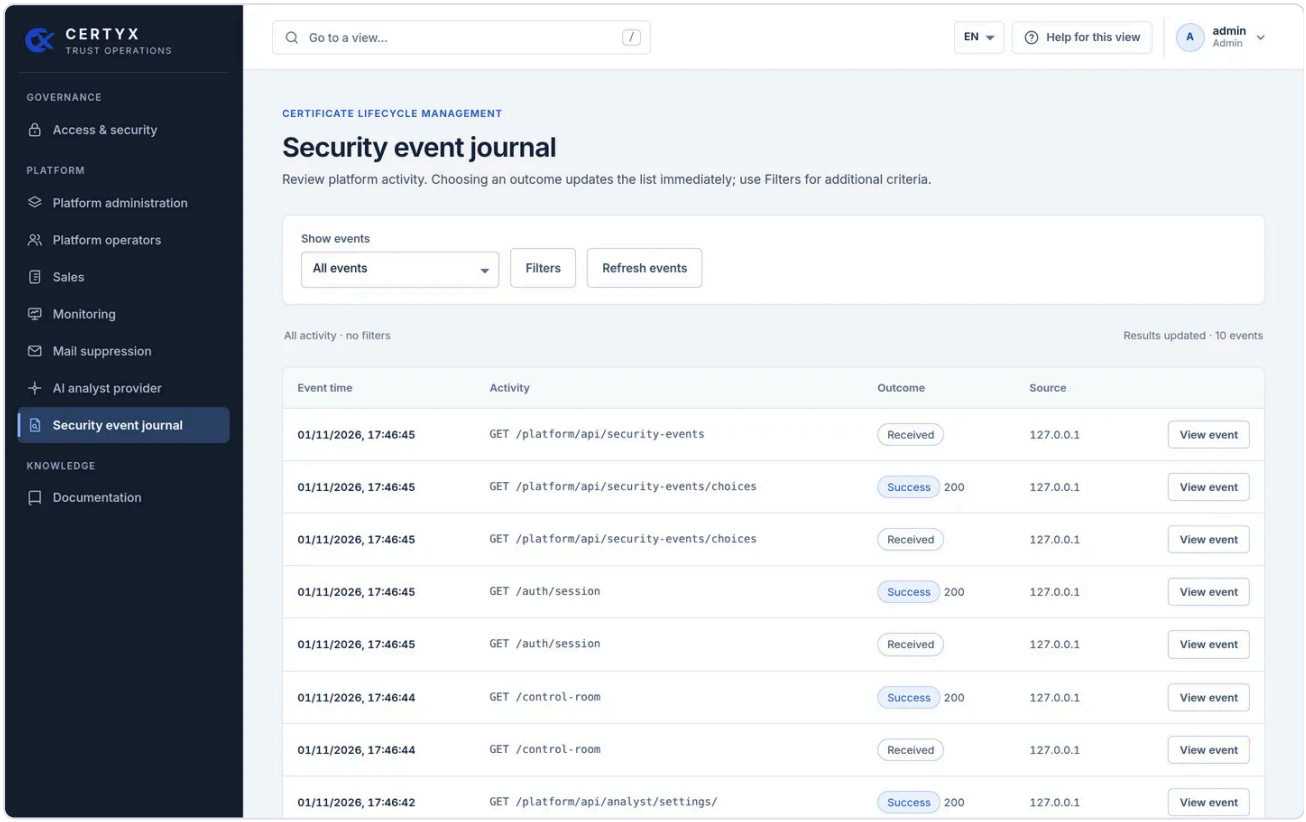
Every organization role can use the setup assistant and the collector command. Synchronization status, snapshot preview and import: Admin. Import additionally requires the trust graph in the subscription.

LIMITATIONS

The assistant requires Python 3 and the provider CLI. It creates roles for an existing workload identity; it does not create or attach VM identities or federation. Azure requires RBAC-enabled vaults. Cloud provisioning needs local administrator permissions, while the collector needs only certificate read access. Credential values are not accepted. Successful read tests do not prove absence of broader privileges. Renewal and deployment run through the lifecycle agent on an execution target with one cloud resource: AWS ACM (re-import into the same certificate ARN), Azure Key Vault (new version of the same certificate) and Google Certificate Manager (update of the same self-managed certificate); a job succeeds only after independent TLS verification of the served endpoint. Importing into a store does not switch consuming Azure or Google services to the new version. Customer credentials and private keys remain customer-controlled. Secrets resolve from protected files, HashiCorp Vault KV, Azure Key Vault, AWS Secrets Manager and Google Secret Manager; other vault products are not supported, provider CLIs run on Unix agents only, and the agent does not send e-mail from the customer host. Unix Vault token permissions are rechecked on each use. The Azure Key Vault store operation preserves the existing certificate policy, tags, enabled state and PEM/PFX storage format. Software import refuses missing policy, disabled certificates and HSM-backed keys. It does not deploy the certificate to Azure endpoints. The workspace does not provision load balancers (ALB). Customer cloud is optional: customer-hosted Vault KV/PKI remains an alternative under customer control; protected files are a compatibility fallback, not an encrypted vault. DigiCert is a provider choice, not an account prerequisite. Reading Google Secret Manager secrets is separate from the Google Certificate Manager lifecycle. Public certificate import fields must never contain private keys. AWS verification is bounded by the agent's convergence window (six minutes for AWS ACM targets unless the agent's Settings.Connectors.CloudVerification sets another ConvergenceWindow, because ALB can stall TLS for more than two minutes while it propagates a re-import): aws-store-unread means no ACM read completed within the bound; aws-store-not-converged means the latest ACM read did not match the approved fingerprint; aws-endpoint-not-verified means ACM matched but not every resolved load-balancer address served it with a valid chain and health. Throttling and timed-out reads are retried within the bound; access, identity and validation errors stop immediately. Both require investigation; neither proves successful deployment. Customer cancellation remains cancellation. An unfinished AWS restoration journal forces a guarded import of its previous protected generation even when one ACM read already matches. Completion still requires independent TLS verification; on failure the journal is retained for reconciliation. GCP apply and reconcile run on Unix only, because their receipts require owner-only Unix paths.

28 Security event journal

Review platform activity. Choosing an outcome updates the list immediately; use Filters for additional criteria.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. In Show events above the table (ten events per page, newest first) choose All events, Denied, Server error, Rejected request, Success, Received or Sign-in; the list updates immediately. Refresh events reloads the newest events with the current filters.
2. Sign-in events (Sign-in) are recorded for every successful and failed sign-in, sign-out, account lockout and MFA change, attributed to the user's organization; platform operators' sign-ins and sign-ins that match no account are recorded under platform. View event shows the event name, the user's e-mail address (marked as entered and not verified when the sign-in failed), the sign-in method, the result, the reason code and the user agent; the claimed password or code is never recorded.
3. Open Filters for account, organization, IP, UTC range or technical criteria. Apply filters returns to results.
4. Loading, result count and active-filter count show what changed. Clear filters restores all activity.
5. View event opens focused details. Filter this request follows a request without copying its identifier. Every request has two entries: Received, written when the request arrives and before the server checks who sent it (user and organization are not identified yet, status 0), and the outcome entry with the user, organization and HTTP status. A Received entry is not a failed attempt or an attack; it keeps a trace of a request even when it is interrupted before the response. Filter this request shows both entries.

6. Older events replaces the page. Journal tools contain the integrity verification, the export of every event matching the filters (CSV or JSON, not only the visible page) and the retention that applies.
7. After moving to older pages choose Newest events to return to the start. In Journal tools, Verify journal integrity checks the hash chain of the whole retained journal, and Platform audit lists the latest 100 business changes made by platform administrators (customers, subscriptions, instances and customer accounts).
8. SIEM export manages the platform-wide destinations: scope Platform operators (the operators' own events) or All events (every organization), then the transport (HTTPS, Splunk HTTP Event Collector, Syslog over TLS, Azure Monitor Logs Ingestion for Microsoft Sentinel, OpenTelemetry Logs (OTLP/HTTP) or File archive), a format and a schema profile among those the server allows for it, as in the organization's SIEM export view. Each destination shows its route, position, backlog, last success and failure, consecutive failures with the next attempt and an event the receiver keeps rejecting; Check connection tests DNS, TCP and TLS without sending anything and Send test event also delivers one test event (category Integration, a UUID as event_id), each with its stages and remedy. Network destinations have filters (outcomes, categories, event codes, minimum severity), a delivery policy, optional client certificate (mutual TLS) and a revocation check mode, and Resend from... rewinds the position to a sequence or time (never forward; receivers deduplicate by event_id). Changes require the MFA session. For an HTTPS destination added without a secret, CERTYX generates the signing secret and shows it once to copy into the receiver; New secret replaces it immediately (the previous secret stops working) and shows the new one once. Each field links to the matching section of the SIEM integration guide (also as PDF under Documentation).
9. The Platform operators scope receives the operators' sign-in events and sign-ins that match no organization; All events receives every organization's sign-in events as well. Check Sign-in events in Outcomes to export to send only those, or leave every box unchecked; the SIEM export guide of the organization view lists the sign-in fields of every format.
10. File archives for the platform: in SIEM export choose the scope Platform operators or All events, the transport File archive on the server and the format CSV (layout of the first release or canonical) or NDJSON, with retention and optional gzip. The files are written under Clm:Siem:PlatformArchiveDirectory (default the platform folder of Clm:Siem:ArchiveRoot), listed with size and SHA-256 under Archive files and downloadable one by one or as a zip; downloads require the MFA session and are journaled. Use Send only on the network destinations (outcomes, categories, event codes, minimum severity) to send the SIEM less while the archive keeps every event; health probes are not sent unless Clm:Siem:ExportHealthProbes is on. Exported events carry the organization ID, name and kind and the instance, so the All events scope can be split by organization in one SIEM.

INPUTS & PARAMETERS

- Select a recorded account or system actor. Empty selection includes everyone.
- Select an organization by name; system entries may use technical scope names.
- Select an instance recorded by the journal.
- Select an observed HTTP status. Zero marks a Received entry, written before the response.
- Use Filter this request in an expanded event to populate this automatically, or enter a request ID.
- Optional IP, route fragment and UTC time range.
- SIEM destination: transport, format and schema profile; HTTPS address or syslog host, port, framing and LEEF 2.0 delimiter; transport-specific fields (Splunk sourcetype and index; Sentinel data collection rule, stream, Microsoft Entra tenant and application); filters, delivery policy, optional trusted root

certificates, client certificate and key (PEM) and revocation mode; OTLP body, authentication header name and gzip; the field mapping rules; the event shaping (preset Full, Balanced, Economical or Custom with change detail, field set, empty fields, largest event, name style, time format, syslog MSGID and APP-NAME) and the additional filters (routes, actors, excluded categories and event codes), described in the SIEM export help; the secret (HMAC key, HEC token, client secret or OTLP header value) and the client key are stored encrypted and never shown again.

OPERATION RESULT

Filtered, bounded access to the retained security journal, downloadable exports of the filtered journal and continuous export to SIEM destinations. Authenticator setup links appear as mfa-setup.mail.sent (delivery by e-mail or as a link shown to the administrator, with the failure reason), mfa-setup.mail.used and mfa-setup.mail.expired, attributed to the account's organization; the link itself is never recorded.

PERMISSIONS

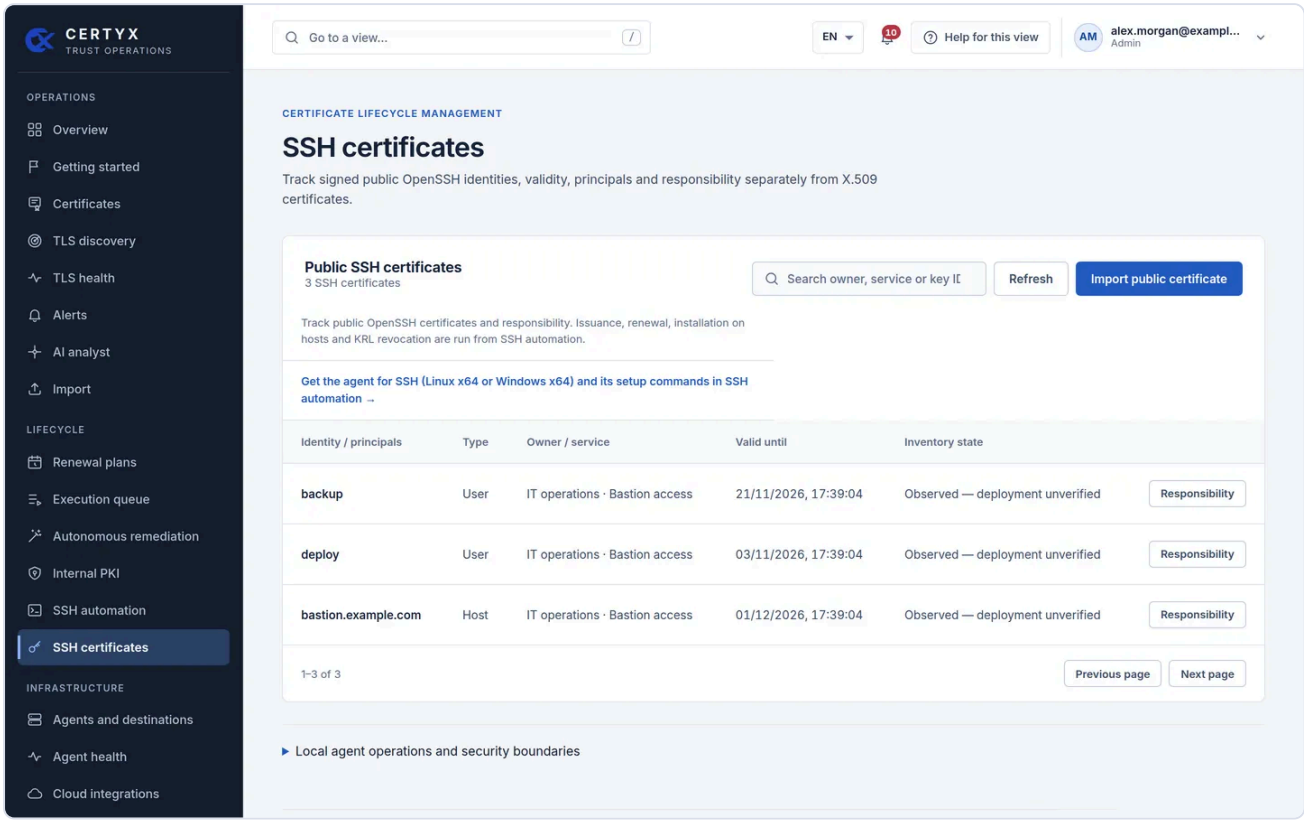
Platform administrator on the operator allowlist whose session was signed in with MFA; other identities cannot open the view or its API. SIEM changes, checks, tests, resends and new secrets are refused for sessions of the CERTYX mobile app. Step-up: every change in this view (except tests, connectivity checks and acknowledging an incident) asks for the current 6-digit code of your own authenticator; a confirmed code also covers your further control-room changes for Clm:Identity:OperatorStepUpWindow (default 5 minutes). A wrong or replayed code is refused, counts toward your account lockout and is journaled; Cancel abandons the change. Operators who sign in through company sign-in have no local authenticator and cannot make these changes.

LIMITATIONS

Source IP is not proof of person or location. Pagination does not delete events. One export holds at most Clm:AuditExport:MaxRows events (default 100,000) and is itself journaled. Choice lists contain up to 500 distinct values; without the platform database (local file journal) the latest 5,000 events are examined and SIEM export is unavailable. Events are kept without automatic deletion; the application role can only add and read them. SIEM delivery is at least once: a refused batch is retried with backoff and jitter and raises the siem-delivery incident, which becomes critical after the attempts; a backlog above its limit raises siem-backlog and nothing is dropped; an event the receiver keeps rejecting is retried until it is accepted unless skipping is turned on (siem.event.skipped). Receivers deduplicate by event_id. Resend from... never moves forward and is refused for file archives; on platform destinations it asks for the step-up code like other changes. Destinations defined in Clm:Siem are listed read-only. The receiver's certificate is always validated; a client certificate (mutual TLS) is presented when one is entered, and the revocation check accepts an unknown status in soft-fail mode (default) and refuses it in hard-fail mode. OpenTelemetry Logs use OTLP/HTTP only; a partial success is not resent but journaled (siem.event.rejected, incident siem-rejected) and can be resent with Resend from. Platform-wide destinations may use private, loopback or link-local addresses; Clm:Siem:AllowPrivateDestinations restricts only the destinations organizations manage themselves. A platform CSV archive keeps every event of its scope (All events: every organization) for its retention; plan the disk space of the archive volume and mount the same directory on every instance. Requests of platform operators are recorded under the tenant platform (like their sign-ins), so the Platform operators SIEM scope exports them; entries from before 2026-10-18 keep the tenant platform-control. Every change an operator makes to a customer is additionally recorded under that customer (actor platform:<operator account ID>).

29 SSH certificates

Track signed public OpenSSH identities, validity, principals and responsibility separately from X.509 certificates.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. To get the agent for SSH (Linux x64 or Windows x64) with checksums, signatures and setup commands, open SSH automation → Install and connect the SSH agent.
2. On an approved Unix customer host, configure the local SSH CA, protected state directory and allowed user/host principals.
3. Run Clm.Agent ssh issue <config> <operation-UUID> user|host <principal> <minutes>; retries with the same operation ID recover the same certificate.
4. Choose "Import public certificate" and paste only the public identity-cert.pub line. Assign its owner, service and environment. Importing the same certificate again returns the existing entry without a duplicate. Search matches owner, service or key ID; use the page buttons for longer lists.
5. Use Responsibility to update owner, service, environment or archive an inventory entry. The dialog shows the CA fingerprint; signature verification does not establish trust in that CA. Archiving does not revoke host access. If someone changed the entry in the meantime, the save is refused; refresh and retry.
6. To revoke locally, run Clm.Agent ssh revoke <config> <operation-UUID>. Publish the resulting revoked.krl to each SSH server using RevokedKeys and verify access rejection.

7. Further local commands: ssh list <config> prints the issued certificates of the local CA; ssh deploy-host or ssh deploy-user <config> <target-config> <operation-UUID> installs an issued host certificate or a workload user certificate on the locally configured target with verification and rollback; ssh export-krl <config> <tenant-ID> <CA-fingerprint> <new-file> writes the signed public KRL envelope, and ssh apply-krl <consumer-config> <envelope> applies one envelope on a consumer.
8. Standalone local renewal without reporting to CERTYX: configure-ssh-linux.sh enable-renewal <downloaded profile> writes renewal.json from the local CA policy and the profile, runs one Clm.Agent ssh renew cycle through systemd and enables clm-ssh-renewal.timer (one cycle a minute, no network access, journal identifier clm-ssh-renewal); on Windows configure-ssh-windows.ps1 enable-renewal registers the task CERTYX SSH Renewal. A failed, blocked or conflicting profile fails the cycle visibly (failed unit or non-zero task result); renewal-status shows it and disable-renewal removes it. A profile is renewed either by add-profile or by enable-renewal. ready-local confirms only a local bundle; no remote installation is implied.
9. For platform-managed renewal and verified host deployment, open SSH automation. Assign active notification groups and inspect the separate SSH workload health. Local KRL consumers can be configured in the managed agent; remote fleet distribution is still separate.
10. For customer-controlled remote KRL transfer, run ssh publish-krl with protected publisher configuration and serve only its public envelope through a separate HTTPS relay identity. Run ssh sync-krl with a protected consumer subscription. Signed publications expire after one hour and are refreshed automatically; redirects, bad trust, oversized data, tampering and expired publications are refused. Existing revocations remain after failures. OpenSSH does not enforce envelope expiry itself. Central health is reported only by consumers enrolled for KRL reporting; CERTYX does not determine which hosts are expected to consume the KRL. Old envelopes without signed time bounds must be re-exported. A fresh CA can initialize an empty KRL before issuing; loss of KRL on a used CA requires recovery.
11. For central visibility, enroll a separate consumer identity and explicitly enable KRL consumer reporting; it grants no SSH issuance or host-installation capability. Run ssh monitor-krl with the protected identity and subscription configuration. Agent health shows the reported revision, signed publication expiry, synchronization errors and missed reporting windows; healthy discovery cannot hide a KRL error. Platform monitoring includes those agent errors. These are authenticated agent observations, not independent proof that every host rejects the revoked identity. Use one registered identity per consumer.

INPUTS & PARAMETERS

- One Ed25519 OpenSSH certificate signed by an Ed25519 CA, with a finite validity interval. Private keys are rejected.
- Local CA/state paths must be absolute, without symbolic links, and owner-only. No CA key or endpoint private key is sent to SaaS.
- Import format: one line ssh-ed25519-cert-v01@openssh.com AAAA... with an optional comment, as in a -cert.pub file, at most 64,000 characters.

OPERATION RESULT

Paginated tenant-isolated SSH inventory, cryptographically verified signatures, versioned ownership and audited changes. The agent issues certificates and generates KRLs locally. Local renewal uses persistent pending operations, atomic bundle pointers and revocation/policy-conflict blocks.

PERMISSIONS

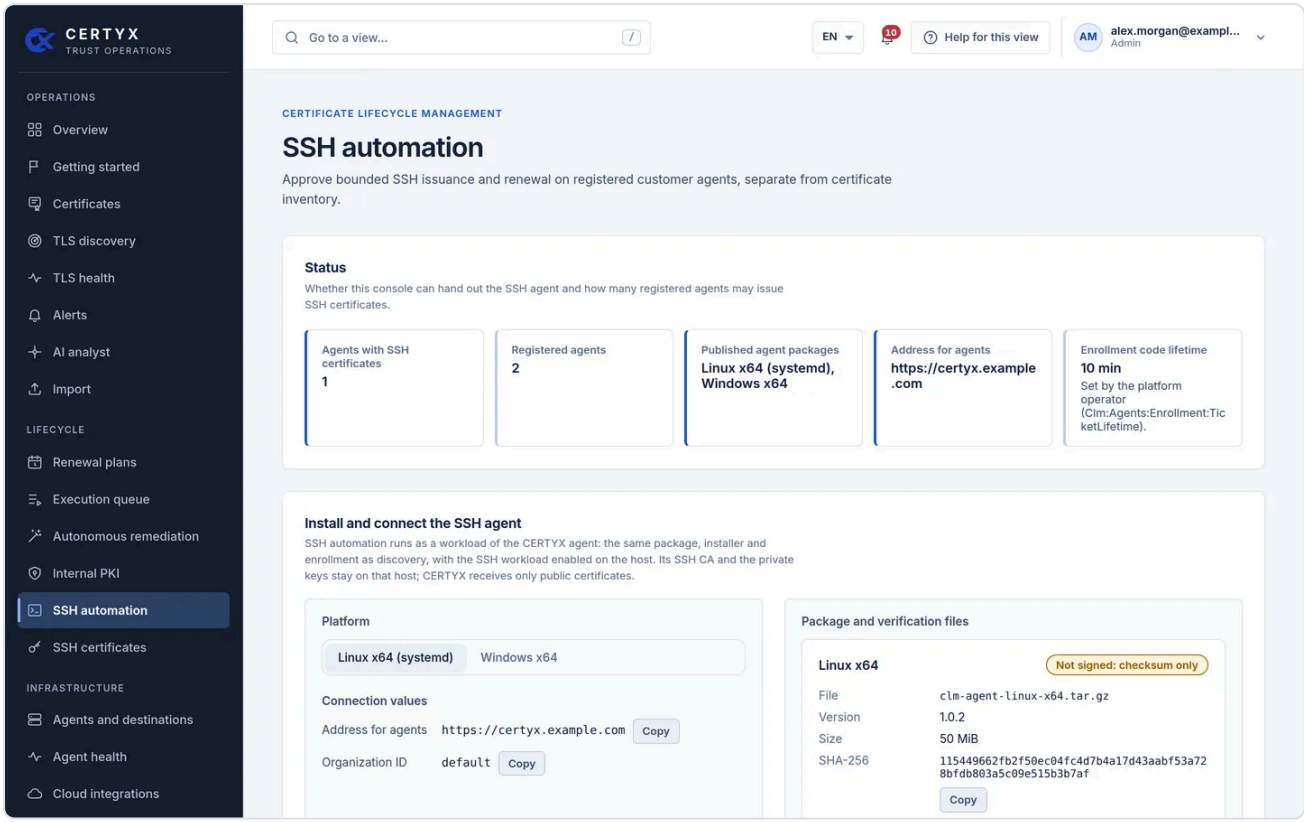
All organization roles read; Admin and Operator import and edit, Auditor reads only. Local CA use requires explicit principal and validity policy.

LIMITATIONS

SSH automation covers platform-approved jobs, unattended bounded renewal, verified Unix host installation/rollback, local signed KRL consumers and SSH notification events. CERTYX does not configure CA trust or revocation lists in the OpenSSH configuration of remote hosts and does not track which hosts are expected to consume the KRL. User certificates are installed only for customer-local Unix workload identities, not for human accounts. SSH operations run on Linux and Windows agents (OpenSSH for Windows), and the SSH CA uses Ed25519 keys only. SSH notifications are e-mailed to the profile's recipient groups through the organization's SMTP relay. A valid inventory signature does not prove serving state or fleet revocation. Local CA key files are not an HSM. The local SSH authority flushes serial-counter contents and its directory entry before signing. Storage synchronization failure blocks issuance; inspect protected local storage and retry the same operation ID after repair. Signing runs on one host; distributed signing is not supported. Customer-local connector locks must be regular owner-only files on Unix, with no symbolic parents. Unsafe existing locks are refused; an administrator must correct storage permissions before retrying. These connectors run on Unix agents only.

30 SSH automation

Approve bounded SSH issuance and renewal on registered customer agents, separate from certificate inventory.



Sample data: the organizations, people, hosts and addresses are fictional.

HOW IT WORKS

1. The view is divided into cards: Status (agents with SSH certificates, registered agents, published packages, the address for agents and the configured enrollment code lifetime), Install and connect the SSH agent, Agents with SSH permissions, and Profiles, job history and revocations. In Install and connect the SSH agent choose Linux x64 (systemd) or Windows x64: the panel shows the package card of that platform (the same card as in Agents and destinations: file name, version, size, SHA-256 with a Copy button, publisher signature and release manifest state, and download buttons for the package, its .sha256 file, the .p7s signature and .tsr time stamp when published, the release manifest with its signature and the verifier, each saved under its own file name), or states that the package is not published on this server together with the operator's publishing step. Administrators create the single-use enrollment code here and download the token file; the numbered steps give ready-to-copy commands built from the address for agents and your organization ID: trust step for a private agent CA, copying the token file into an empty working directory on the host, download (the archive is served only with a valid enrollment code, which the command reads from the file; the panel's download link works in your signed-in console session) and verification (replace <PUBLISHER-SHA256> with the publisher fingerprint obtained independently), the installer in the clm-agent directory with ../clm-enrollment.token, and the status check. There is one agent package; SSH is a workload of that agent.

2. The numbered installation steps (each its own block with a title and the command to copy) open by themselves while no agent has the SSH capability. In Agents with SSH permissions grant "SSH certificates" (claims approved profiles) and, for installation, "SSH installation" to the registered agent; the table shows last contact and the last SSH workload report. On the host run configure-ssh-linux.sh init (Linux, as root) or configure-ssh-windows.ps1 init (Windows, administrator PowerShell): it creates the host CA (/etc/clm-ssh or C:\ProgramData\CLM-SSH, owner-only) and prints the CA fingerprint for the profile. After the profile is approved and downloaded, add-profile ./ssh-profile-<ID>.json runs one verified ssh manage --once cycle and starts clm-ssh.service (Linux) or the scheduled task "CERTYX SSH Workload" (Windows, LocalSystem); a failed cycle names its cause (for example HTTP 403 without the SSH capability) and nothing is started. Optional trust-user-ca makes the local sshd trust user certificates of this CA (TrustedUserCAKeys, checked with sshd -T, backup restored if sshd does not reload). Verify with sudo clm-agent status (Capabilities line lists ssh) and the script status. On Windows Server 2025 sshd admits only members of administrators and OpenSSH Users by default (AllowGroups): add the accounts that log in with user certificates to OpenSSH Users, and start the OpenSSH server once before trust-user-ca so that sshd_config exists.
3. Select New profile and fill in the registered SSH-capable agent, principal and type (User or Host), lifetime, renewal window, approval validity, independently approved CA fingerprint, environment, responsible owner or team, service and optional responsibility groups, then Request approval. Without an enabled agent holding SSH capability the form says so and cannot be submitted.
4. A different administrator selects Approve scope on the profile in the Profiles tab (the requester cannot approve their own request). Download public profile and add it to the customer-local managed configuration; private CA and identity paths stay on that host.
5. Standalone local renewal without reporting to CERTYX: configure-ssh-linux.sh enable-renewal <downloaded profile> writes renewal.json from the local CA policy and the profile, runs one Clm.Agent ssh renew cycle through systemd and enables clm-ssh-renewal.timer (one cycle a minute, no network access, journal identifier clm-ssh-renewal); on Windows configure-ssh-windows.ps1 enable-renewal registers the task CERTYX SSH Renewal. A failed, blocked or conflicting profile fails the cycle visibly (failed unit or non-zero task result); renewal-status shows it and disable-renewal removes it. A profile is renewed either by add-profile or by enable-renewal. ready-local confirms only a local bundle; no remote installation is implied.
6. Run Clm.Agent ssh manage <protected-managed-config> [--once]. The agent connects outward over mTLS, retrieves durable jobs and retains all private keys locally.
7. Use Profiles and Job history tabs to inspect approval, expiry and failures. Stop automation to reject further claims/publication.
8. For host installation, select Enable SSH identity deployment for the agent under Registered agents and tick "Install and verify this SSH identity" in the profile. The customer administrator provisions a protected local target, existing baseline, fixed validation/reload tools and a CA-pinned SSH probe; on Linux, configure-ssh-linux.sh enable-host-deploy grants the agent account exactly sshd -t and the sshd reload through sudo (visudo-checked rule and service drop-in; disable-host-deploy removes them). The agent installs a complete key/certificate pair, validates, reloads and verifies; failure restores and verifies the previous pair.
9. Revoking SSH host deployment capability stops pending host jobs and further claims. The agent checks job authorization again immediately before installation. If an interrupted activation journal exists, recover the prior validated state before new work.

10. In Job history, choose Revoke certificate and confirm. The platform stops its profile and creates one durable request. The assigned agent verifies the exact local certificate, updates its local KRL and reports its hash. Inspect pending, failed or revoked-local in Revocations. Offline agents retain pending work. Existing sessions and other hosts are not automatically revoked.
11. If local revocation fails, inspect the safe error code and protected agent state, correct the cause, then use Retry revocation. A fresh request ID rejects delayed acknowledgements from the old attempt; issuance stays stopped.
12. Assign existing notification groups when creating the SSH profile. The local outbox creates bilingual HTML messages for approaching expiry, issuance, deployment, rollback, failures and local revocation. Repeated processing does not duplicate an event. Delivery requires an enabled tenant SMTP channel; a queued message is not proof of delivery. Local revocation does not distribute KRLs to other hosts or terminate sessions.
13. Customer-side SSH KRL transfer: Clm.Agent ssh export-krl signs a public list with the local CA; Clm.Agent ssh apply-krl <consumer-config> <envelope> verifies tenant, pinned CA, signature, digest and monotonic revision. Consumers must separately configure RevokedKeys or RevokedHostKeys. Distribute the list to other hosts with ssh publish-krl and ssh sync-krl; CERTYX does not test whether each host actually rejects a revoked key.
14. Optional customer-local KrlConsumers in managed agent configuration automatically receive signed KRL updates after revocation. Versions persist across restarts. A failed publication is visible as failed revocation and can be retried. This does not configure OpenSSH consumers or cover remote fleet hosts.
15. The managed agent reconciles configured local KRL consumers each polling cycle, including after restart or an authorized local revocation. Consumer failures stop the cycle safely. The OpenSSH consuming identity must have permission to read its provisioned destination; the agent does not grant filesystem access to another account.
16. Managed SSH reports its own safe cycle status. A healthy discovery process cannot hide a failed or silent SSH worker. Review the error in Agent health and protected local journals; fixed summaries contain no secrets. Loss of connectivity can prevent immediate reporting, so missed-contact monitoring is also required.
17. SSH email events persist atomically with the operation. Messages identify the exact signed certificate, expiry in UTC, CA fingerprint and safe error code. Delivery projection processes bounded batches and recovers interrupted acknowledgements without duplicate emails.
18. New SSH issuance requires the automatic-renewal entitlement. Host installation additionally requires deployment, verification and rollback entitlements. Existing in-flight results/recovery are not blocked solely by a feature change; tenant access and security revocation gates still apply.
19. For customer-controlled remote KRL transfer, run ssh publish-krl with protected publisher configuration and serve only its public envelope through a separate HTTPS relay identity. Run ssh sync-krl with a protected consumer subscription. Signed publications expire after one hour and are refreshed automatically; redirects, bad trust, oversized data, tampering and expired publications are refused. Existing revocations remain after failures. OpenSSH does not enforce envelope expiry itself. Central health is reported only by consumers enrolled for KRL reporting; CERTYX does not determine which hosts are expected to consume the KRL. Old envelopes without signed time bounds must be re-exported. A fresh CA can initialize an empty KRL before issuing; loss of KRL on a used CA requires recovery.

20. For central visibility, enroll a separate consumer identity and explicitly enable KRL consumer reporting; it grants no SSH issuance or host-installation capability. Run ssh monitor-krl with the protected identity and subscription configuration. Agent health shows the reported revision, signed publication expiry, synchronization errors and missed reporting windows; healthy discovery cannot hide a KRL error. Platform monitoring includes those agent errors. These are authenticated agent observations, not independent proof that every host rejects the revoked identity. Use one registered identity per consumer.
21. Issuance acknowledgements require an active registered identity with SSH capability. Deployment acknowledgements also require SSH deployment capability. Removing permission rejects pending results even if issuance previously started; retained local journals require reconciliation before restoring permissions.
22. User identity installation: select User and enable installation when requesting the profile. Independent approval and SSH deployment capability are required. Add a matching UserTargets entry to the current Unix agent configuration: fixed remote account/principal, endpoint and separately pinned host CA. The agent switches the local complete key/certificate generation, proves certificate authentication and restores the previous generation if rejected. Raw-key fallback is not accepted as certificate verification.
23. UserTargets must include HostRevocationFile: the protected customer-local KRL used for RevokedHostKeys. The client refuses missing/unsafe lists and revoked hosts even when their CA pin matches. Keep the list synchronized through the customer KRL publisher/subscriber; CERTYX does not check which hosts are expected to hold it.
24. All agent JSON configuration readers reject symbolic files or parent directories, Unix group/other permissions, files over 1 MiB and invalid UTF-8 before deserialization. Protect the configuration with chmod 600 on Unix. Relative command-line paths are resolved to an absolute path and receive the same checks. UTF-8 with or without BOM is supported. Settings and asynchronous readers use the same rules; no configuration values are included in failure output. On Windows the readers refuse a file that a principal other than LocalSystem, Administrators or the agent's own identity owns, can read or can change; provision restrictive ACLs before deployment.

INPUTS & PARAMETERS

- Registered enabled agent with explicit SSH capability; HTTPS trusted by the host.
- Owner-only managed config contains enrollment identity, local authority and approved profile array.
- Approval requires a different administrator and expires after 1–90 days.
- Profile formats: principal 1–128 letters, digits and . _ @ -, starting with a letter or digit (bastion.example.com or deploy); CA fingerprint SHA256: plus 43 Base64 characters as printed by ssh-keygen -lf; lifetime 2–10,080 minutes; renew before expiry at least 1 minute and shorter than the lifetime; approval 1–90 days.

OPERATION RESULT

Actual signed SSH public artifacts are validated against job UUID, principal, CA and lifetime; inventory is projected idempotently. Lost result acknowledgements recover without new keys.

PERMISSIONS

Organization users read; administrators request/revoke, a different administrator approves. Customer host administrators authorize local CA scope.

LIMITATIONS

Host deployment supports OpenSSH on Linux and OpenSSH for Windows with a preconfigured baseline and local target (examples ssh-host-target.linux.json and ssh-host-target.windows.json). The Windows SSH workload (ACL-protected files, copied current pair instead of a symbolic link, Restart-Service sshd) runs as LocalSystem with OpenSSH for Windows. issued-local is issuance only; deployed-verified records observed SSH acceptance. rolled-back stops automatic retries; intervention requires host administrator investigation. CERTYX does not configure CA trust on remote hosts and does not track which hosts are expected to consume the KRL. SSH notifications are e-mailed through the organization's SMTP relay. Local signed KRL consumers and durable HTML notification events are available; local revocation does not establish remote consumer coverage. Stopping a profile does not revoke certificates or existing sessions. Registered monitored consumers report synchronization health; this does not establish expected fleet coverage or independent revoked-access denial. User installation is currently a customer-local Unix workload identity under the same OS account with an existing verified baseline. It does not provision human home directories, Windows ACLs, server CA trust or live sessions. Workloads must use the deployment current/identity and current/identity-cert.pub paths for new connections. Upgrade agents before assigning user-installation profiles. When this server publishes a package without a publisher signature (an unsigned development build), the installation commands say so, skip the signature file and check the checksum only, with the development overrides of the verifier and installer; use such packages only for development.