

# Your incidents never leave. Nothing changes without a human.

Letting software near production should worry you. So here is everything your security team will ask, in the order they'll ask it.

Document
Security and architecture overview

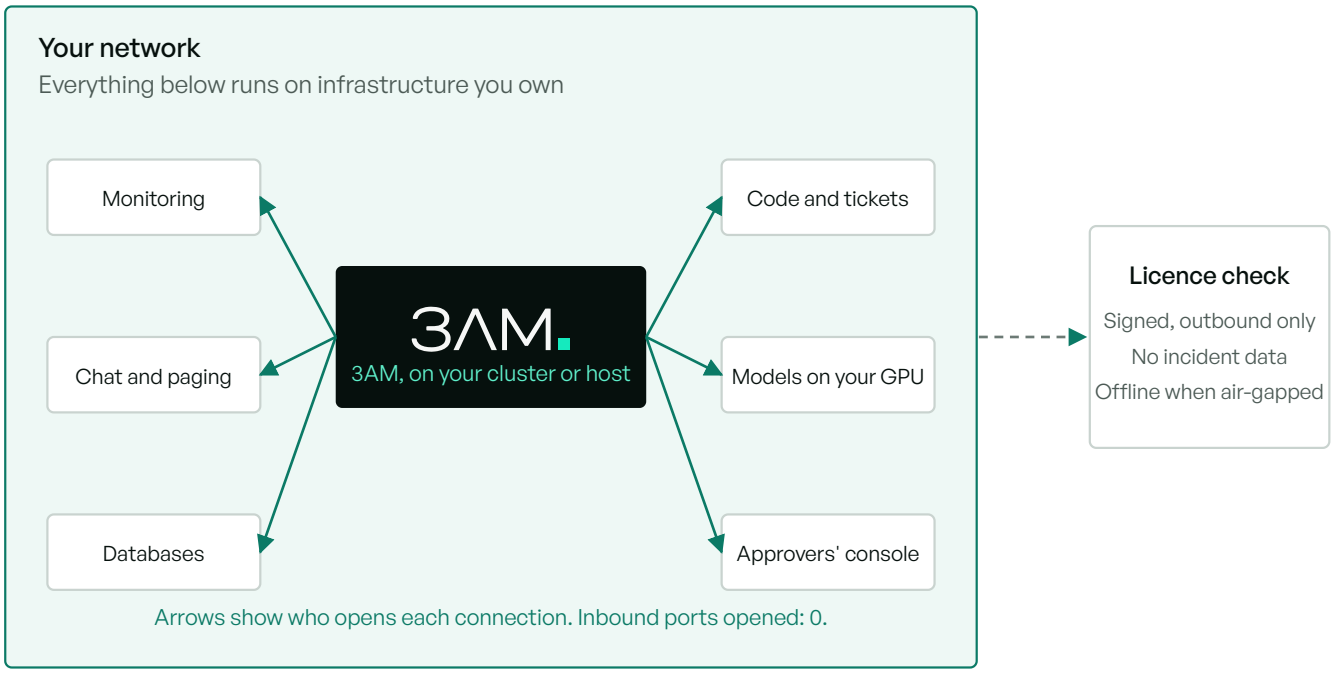
For
Security and risk

Published
November 2026

# At a glance.

Built for environments where every action needs an owner and every decision needs a record.

| 0 | 0 | 2 | 100% |
|---|---|---|---|
| inbound ports to open | bytes of incident data sent to us | approvers to raise autonomy | of actions on the ledger |




A Helm chart or a single host, inside your network. Models run on your GPU, or behind an endpoint you choose. Closed networks get an air-gap bundle with the weights included.

# First, can anyone get in? No.

3AM runs inside your network and opens no inbound ports. Probe every one of them from outside, and none answer. It only ever calls out, and you decide how far.

| | Connected | Air-gapped |
|---|---|---|
| **What it calls out to** | Calls out only to tools you connect, and a signed licence check. | Calls out to nothing. The licence verifies offline. |
| **The licence** | A signed check, outbound | Verifies offline |
| **The models** | On your GPU, or an endpoint you choose | Shipped in the bundle |
| **Approvals** | In the console, or in your chat tool | In the console |
| **Inbound ports** | None | None |

- ✓ Helm chart or single-host bundle
- ✓ Air-gap bundle, weights included
- ✓ Outbound only, to tools you connect

| | |
|---|---|
| **Do we need to open inbound firewall ports?** | No. Slack uses Socket Mode, Symphony its datafeed, and PagerDuty and other channels are polled. Every connection is opened from inside your network. |
| **Can it run in an air-gapped environment?** | Yes. The air-gap bundle ships the images and model weights together, so nothing needs to be fetched at install time. |
| **How does it handle disaster recovery and several installs?** | A licence can name your DR install, or carry grace hours for failover. On Enterprise and Strategic, Keychain sets up every install once, and that setup moves only between your own installs. |

# Next, what does it see? Never your secrets.

Logs are full of things that shouldn't travel. So passwords, tokens, keys and card numbers are redacted on the way in, before anything is written. Prompts are kept redacted too, with a hash for proof.

```
db.connect user=svc_3am [REDACTED] password=hunter2-prod! [REDACTED]
slack.post token=xoxb-2291-8812-Kq3vZ [REDACTED] channel=#approvals
payment.retry card=4111 1111 1111 1111 [REDACTED-PAN] amount=420.00
git.fetch Authorization: Bearer ghp_9fK2mQx7Lw0 [REDACTED]
ssh.exec key=-----BEGIN OPENSSH PRIVATE KEY----- [REDACTED]
```

Illustrative log lines. Each secret is removed before anything is written.

- ✓ No incident data sent to us
- ✓ Models on your GPU, or an endpoint you choose
- ✓ Card numbers caught by checksum

| | |
|---|---|
| **Does any of our data leave the network?** | No. It never copies your data anywhere: each question runs where the data already lives, and only a small answer comes back. It reaches out only to the tools you connect, to a model endpoint if you use one instead of a GPU, and for a signed licence check, which you can turn off when air-gapped. |
| **Which AI models does it use?** | Open-weight models. With a GPU they run inside your install, and nothing calls out. Without one, you point 3AM at a model endpoint: your own, inside your network, or a hosted one. It won't run them on a CPU, where every investigation would wait. |
| **What can your company actually see?** | The licence service sees fingerprints, public keys and licences. It never sees a connection, a secret or a person. Air-gapped, it sees nothing at all. |

# Then, who can change anything? Two people, always.

Seeing is one thing. Changing is another. Every write takes two people, each with their own key. Code review can't grant that, and neither can an admin. Only their signatures can.

- ✓ Shadow by default, per-service autonomy
- ✓ Quorums, allowed hours, action budgets
- ✓ Mandatory dry run, one global halt

## So locking takes a second. Unlocking takes days.

An attacker's first move is to loosen the rules, so that's the move that waits. Tightening them never does: any one person can do it, the moment something looks wrong.

Making it safer

### Now

Any one person, immediately.

- ✓ Revoke a key
- ✓ Halt everything
- ✓ Lower what it may do

Making it riskier

### Days

Several signatures, then a wait. Never the person who benefits.

- ✓ Let it do more
- ✓ Add an approver
- ✓ Upgrade the safety core

While it waits, everyone is alerted, and any one of them can cancel it. Your clock times it, not ours.

Suggested defaults. You set your own, above fixed floors.

| | |
|---|---|
| **Who holds the root of trust?** | You do. At install, named people from your bank each enrol their own key and sign the first policy: who may approve, which tables are the books, and how many signatures each change takes. 3AM staff are never among them. |

# Not the books doesn't mean harmless

A failover on a payments database can stop a bank. So every action is rated by what it could do if hijacked, and the rating decides who must approve it. You set the ratings for your own estate.

| The action | For example | Who must approve | Can it ever run alone? |
|---|---|---|---|
| **Contained** | Release a lock, clear a stuck job | Two people | Only once it has earned it |
| **Data, not the books** | Set-based fixes to tables you opt in | Two people | No |
| **Disruptive** | Failover, rollback, restarting a payment service | Two people and the business owner | Never |
| **The books** | Reversals and corrections | 3AM can't run it. Your DBA does, with their own key | Never |

Disruptive actions also stay out of your declared business-critical windows, and one person can stop everything at any moment.

| | |
|---|---|
| **Can it ever touch our books?** | No, at any setting. 3AM holds no key to your ledgers, journals or balances. For those it hands your DBA a signed package, already rehearsed on a copy, and a person runs it with their own key. 3AM checks the result afterwards. |

## Documents can suggest. Only evidence can act.

A log line, a ticket or a row can hide an instruction, and it might change what the AI suggests. That's survivable, because no suggestion can pass a gate. Only a live check can authorise a fix.

| Source | Tier | Can suggest | Can authorise |
|---|---|---|---|
| **Code and schema** | T1 | ✓ | ✓ |
| **Alert rules and live metrics** | T2 | ✓ | ✓ |
| **Runbooks and wiki pages** | T3 | ✓ | – |
| **Past postmortems** | T4 | ✓ | – |

# And whatever happened, you can prove it.

Every signal, check, approval and action is hash-chained, and your own key co-signs each seal, so nobody can quietly rewrite it. Not even us. Change one byte and the verifier shows where.

- ✓ Append-only, hash-chained, Ed25519-signed
- ✓ Independent verifier for edits, gaps, reorders
- ✓ Syslog to Splunk, QRadar or any SIEM

| | |
|---|---|
| **How do our auditors verify what happened?** | Every event is written to an append-only, hash-chained ledger with signed segments. An independent verifier detects any edit, deletion, insertion or reordering, and the ledger exports to your SIEM. |
| **Can we audit exactly what the AI was asked?** | Yes. Every model call goes on the ledger: the model, the redacted prompt with its SHA-256, the output, tokens and latency. Whoever holds the original prompt can prove which one it was. |

## Now assume all of that fails. Including us.

Keys get stolen and vendors get hacked, and we're a vendor. So take each place an attacker could land, and ask what they'd get. None of them is enough to change your data or loosen the rules.

| If an attacker lands in | They try to | What happens |
|---|---|---|
| **Your Git or CI pipeline** | Merge a malicious change | It's only a proposal until your approvers sign it. |
| **The cluster 3AM runs in** | Forge an approval | The signature fails, and opens an incident against 3AM itself. |
| **Your identity provider** | Add a new key | It means nothing until your existing approvers enrol it. |
| **One of your sites** | Reach beyond that site | It can't go beyond a plan your people have already approved. |
| **3AM, the company** | Push an upgrade | Upgrades need your signatures, and your people hold the root keys. Never our staff. |

Your data and your rules stay as they were.

# Built to be checked.

- ✓ Builds you can verify yourself
- ✓ SBOM with every release
- ✓ Source-free compiled build
- ✓ No shell access for the agent
- ✓ Signed, distroless images
- ✓ Signed licence, verified offline
- ✓ Read-only diagnostic credentials

| | |
|---|---|
| **Does it work with our single sign-on?** | Yes, on every licence tier. It signs in through OIDC, including Entra ID, ADFS, Okta, Ping and Keycloak, or directly against LDAP or Active Directory over TLS. Your provider's MFA and access rules apply, and directory groups decide each person's role. |
| **Who in our organisation can see what?** | There are four roles: owner, admin, responder and viewer. A viewer can read incidents and the audit log, and can see connections but never their secrets. Someone in no mapped directory group gets no access. |
| **How is the install itself locked down?** | It runs as non-root with a read-only root filesystem, all Linux capabilities dropped, seccomp RuntimeDefault and no service-account token. An in-cluster model is reachable only from the core, by NetworkPolicy. |
| **Could a log line or a ticket trick it into acting?** | Text inside anything it reads is treated as data, never as instructions. A fix still needs a cause confirmed by a live check, and your approval policy still applies. |

## Have a vendor security questionnaire? Bring it to the demo. We'll walk your team through it line by line.

See it on your estate
www.3am.si/demo

Write to us
hello@3am.si

Company
3AM Super Intelligence Inc.