VEILNET · SOVEREIGN POST-QUANTUM NETWORKING

Post-quantum, without the rebuild

A staged path from discovery to a post-quantum cloud: observe, protect one system, protect the network, take control, then run the cloud on it.

conflux · guardian stack · anchor protocol

ML-KEM-1024 · ML-DSA-87 · AES-256-GCM

THE PATH

Five stages, each useful on its own

STAGE 01

Discovery

Inventory every key exchange the host makes, serves or passes on.

`conflux observe`

STAGE 02

Scoped protection

Protect one application or system, with no host changes.

`conflux proxy`

STAGE 03

Network-wide

Protect every IP protocol on the machine, apps unchanged.

`conflux up`

STAGE 04

Guardian suite

Own the identities, renewal, revocation and telemetry.

`guardian stack`

STAGE 05 · ROADMAP

PQC Kubernetes

Run RKE2 clusters on the overlay, fully integrated.

`Roadmap: RKE2`

Every stage keeps the machine's identity and settings from the one before. Stop at any stage and keep what it delivered.

STAGE 01 · DISCOVERY

Find what a quantum computer could read

The observer reads the handshake at the start of every flow and records whether its key exchange was post-quantum, hybrid, classical or none, and which side held it back.

Carries nothing. It reads a copy, so it cannot slow, break or reroute a connection.

Joins nothing. No enrolment, no credential, no realm.

Least privilege. CAP_NET_RAW on Linux, BPF read on macOS, ETW on Windows.

```
$ sudo conflux observe
$ conflux inventory
```

SERVICE	PORT	KEY EXCH.	LAGGING
a.example	443/tcp	hybrid	-
[db.internal]	5432/tcp	classical	server
[idp.corp]	636/tcp	classical	client
[plc-gw]	502/tcp	none	-

Illustrative output. Bracketed rows are placeholders.

STAGE 01 · DISCOVERY

Three outputs that feed the PQC roadmap

`conflux inventory`

Inventory

One entry per service: protocol, key exchange, TLS version and suite, and the owning process.

`conflux inventory -cbom`

CBOM

A CycloneDX 1.6 cryptography bill of materials, ready for audit and the transition plan.

`--export HOST:4317`

Live records

Each flow sent as an anchor.observed_flow event over OTLP/gRPC, kept for good by the collector.

The result: which systems to protect first, and whether the client or the server is holding each one back.

STAGE 02 · SCOPED PROTECTION

Protect one application at a time with proxy mode

The overlay lives inside the daemon, in a userspace network stack. Nothing on the host sees it, and you publish only the ports you choose.

```
$ sudo conflux proxy \
  8080=127.0.0.1:3000 \
  53/udp=127.0.0.1:53
```

A peer on overlay port 8080 gets a fresh connection to 127.0.0.1:3000.

No privilege. No interface, no TUN device, no host network. anchord runs as its own account.

Fits locked-down hosts. Containers, CI runners and hosts where CAP_NET_ADMIN is not available.

Smallest exposure. Peers reach only the published TCP or UDP ports.

Change it live. anchorctl proxy -add adds a port without a restart.

Userspace router. --subnet and --serve-exit work too, with nothing set up on the host.

STAGE 03 · NETWORK-WIDE

Protect every protocol on the machine with TUN mode

The daemon creates a real network interface and the kernel owns the overlay addresses. ping, ssh, browsers and every application use the overlay without knowing it exists.

```
$ sudo conflux up \
  --taint [network-taint] \
  --ipv4 10.128.0.2/24
```

Every IP protocol. TCP, UDP, QUIC, ICMP and all built on them, protected with no fallback.

Apps unchanged. No proxy settings, certificates or library swaps. Publishing is an ordinary bind.

Subnet router. Devices that cannot run VeilNet keep their addresses behind a gateway node.

Compartments. Different taints have no address for each other at all.

Any link. --uplink runs over a serial device when there is no IP network.

STAGES 02 AND 03

Choosing a mode per machine

	conflux proxy	conflux up
Host interface	None	A real network interface
Anchor privilege	None	CAP_NET_ADMIN on Linux
Traffic protected	Published TCP and UDP ports	Any IP protocol
Typical host	Container, CI runner, locked-down host	Server, VM, gateway, laptop
Publishing a service	OVERLAYPORT=BACKEND spec	An ordinary bind
Subnets and exits	Yes, routed in userspace	Yes, forwarded by the host

One daemon holds one anchor, so a machine runs one mode at a time. Switching keeps its identity, address and taints.

STAGE 04 · GUARDIAN SUITE

Post-quantum identity and control, on your own host

Admits every machine

Operators commission each machine and issue its manifest. No machine can enrol itself.

Renews every credential

90 days by default. If the guardian is down, machines keep exchanging data.

Keeps keys protected

Realm keys stay protected in the state store, released only to make one signature.

Acts on running machines

Block, unblock, move compartments and set routed networks. Each power is lent for ten minutes at most.

Decides who operates it

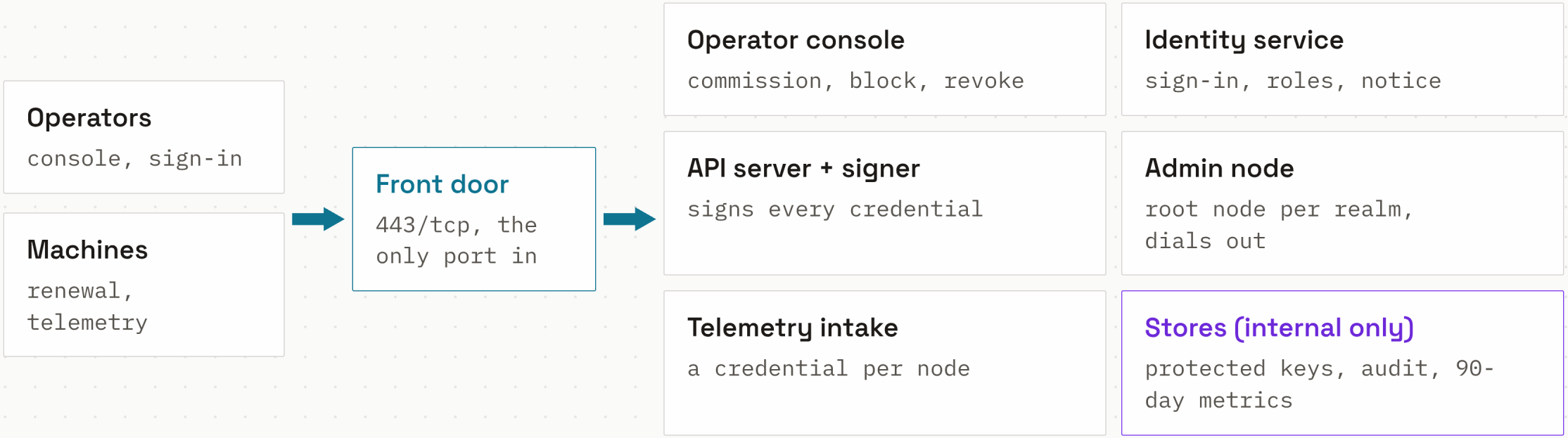
Sign-in through your own identity provider, four roles and a compliance profile.

Runs anywhere you need

Online, offline or air-gapped, with a FIPS option and a full audit of every action.

STAGE 04 · GUARDIAN SUITE

One stack, one port in, never on the data path



Machines exchange data directly or through your bootstrap nodes, protected end to end. No traffic between machines passes through the guardian.

STAGE 04 · INTEGRATION

OpenTelemetry out of every node



Four signals

Metrics, traces, logs and flows. Flow records name the peer, overlay addresses, protocol, port, bytes and packets. Payloads are never reported.

Separate credential

Each node exports with its own collector credential, apart from its renewal secret. Revoking a node cuts its reporting at once.

Standard schema

Service name, resource attributes and trace sampling are set per node, so records land beside existing telemetry.

STAGE 05 · PQC KUBERNETES

The whole cloud environment on the overlay

AVAILABLE NOW

Clusters across clouds, no VPN gateways

- A conflux proxy container in each cluster publishes selected services.
- AWS, GCP and Azure clusters join one overlay, direct or relayed through NAT.
- Pod ranges never appear on the overlay, so they may overlap.
- Unprivileged: no TUN device, no host network.

TARGET STATE · ON THE ROADMAP

Full integration with RKE2

- Planned: every RKE2 node on the overlay, east-west traffic protected.
- Planned: native deployment as part of the RKE2 cluster.
- Planned: guardian-commissioned identity for cluster nodes.
- Planned: cluster flows and metrics in the same OTLP pipeline.

Replaces IPsec, TLS 1.2, ECDHE and RSA between clouds with ML-KEM-1024, ML-DSA-87 and AES-256-GCM on every link.

SUMMARY

What each stage needs and delivers

Stage	Run	Needs	Delivers
1 Discovery	conflux observe	Packet capture only	Crypto inventory and CBOM
2 Scoped	conflux proxy	No privilege for the anchor	Chosen services protected
3 Network-wide	conflux up	CAP_NET_ADMIN	All IP traffic protected, apps unchanged
4 Guardian	guardian stack	One host, 443/tcp in	Own identity, renewal, revocation, OTel
5 PQC Kubernetes (roadmap)	RKE2, planned	An RKE2 cluster	RKE2 clusters on the overlay

NEXT STEP

Move from PQC strategy to implementation

01

Assess

Run the observer and map cryptographic dependencies.

02

Design

Choose proxy or TUN per system and plan the realm tree.

03

Validate

Prove it in a representative environment, air-gapped if needed.

sovereign@veilnet.net