



MintID

MintID.net



ZadQ.info

MintID's kill switch: switching off an agent's authority

Security & Privacy Working Group, AAIF · 27 October 2026

What the chain is and who issues credentials

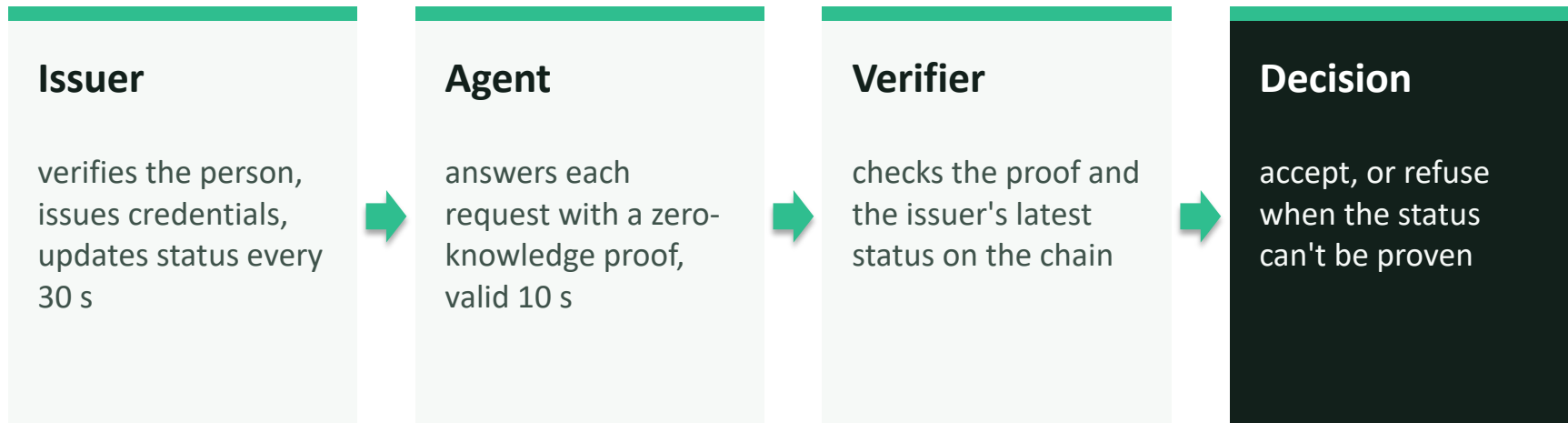


KYC-backed identity and accountability for AI agents

- 1 An approved issuer verifies a person once, with KYC.
- 2 The person's agents carry their own credentials.
- 3 Each credential proves that a verified, accountable person stands behind the agent.

The person's identity stays with the issuer. No personal data goes on the chain.

How a decision works



Every decision checks current status. That is where the kill switch acts.

How to revoke an agent's identity

Path	Who pulls it	What goes
Kill switch	the agent's own runtime, or the user with the agent's revocation kit	that agent's credential
Cascade	the person, by revoking their own credential	all their agents
Issuer	the issuer; an emergency update when urgent	anything it issued

A compromised agent can block only its own runtime. Verifiers refuse the killed credential from then on; nobody learns who pulled it or which one.

How long it takes

Up to **245 s** until every verifier on the network refuses all copies

60 s

for the issuer to remove it

+

180 s

oldest status a verifier accepts

+

5 s

clock tolerance

The clock starts when the chain records the kill.

If the issuer misses the 60 s, the chain refuses its updates.

Measured by Bitgrit for the Identity & Trust WG, Round 1: last acceptance +27.5 s, first refusal +33.8 s.

Your Kill Switch pattern, and one question

MintID covers

- Action 3 of the draft for MintID credentials
- Outside the agent's reach
- Never auto-closes
- Publishes its latency as a bound

Stays with the deployer

- API keys and other tokens
- Halting the process
- Cutting network and tools
- Detection
- Rollback of actions already carried out

For the group: should the standard ask for a maximum time per way of revoking?

Backup: the kill switch on the public testnet

Recording of 23 October, captioned. Our published harness, public access only.

1

Accepted

the agent presents itself

2

Kill switch

pulled through the privacy relay

3

Window

a stolen copy is still accepted

4

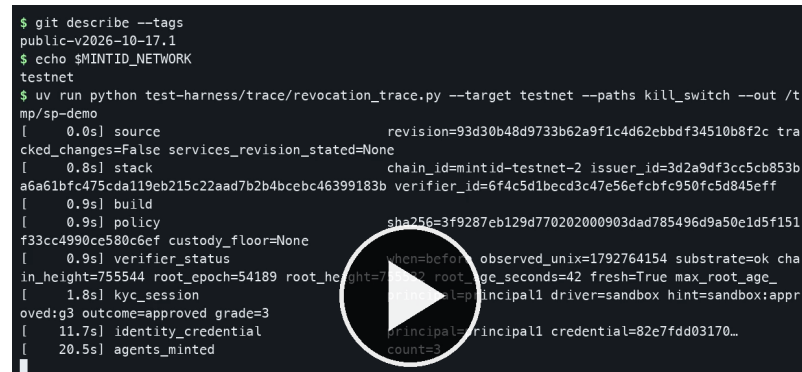
Refresh refused

this proves the revocation

5

Refused

the verifier refuses the copy

A terminal window showing a series of commands and their outputs. A large white play button icon is overlaid on the right side of the terminal output.

```
$ git describe --tags
public-v2026-10-17.1
$ echo $MINTID_NETWORK
testnet
$ uv run python test-harness/trace/revocation_trace.py --target testnet --paths kill_switch --out /t
mp/sp-demo
[ 0.0s] source revision=93d30b48d9733b62a9f1c4d62ebddf34510b8f2c tra
cked_changes=False services_revision_stated=None
[ 0.8s] stack chain_id=mintid-testnet-2 issuer_id=3d2a9df3cc5cb853b
a6a61bfc475cdal19eb215c22aad7b2b4bcebc46399183b verifier_id=6f4c5d1becd3c47e56efcbfc950fc5d845eff
[ 0.9s] build sha256=3f9287eb129d77020200903dad785496d9a50c1d5f151
f33cc4990cc580c6ef custody_floor=None
[ 0.9s] verifier_status when=before observed_unix=1792764154 substrate=ok cha
in_height=755544 root_epoch=54189 root_height=755544 root_age_seconds=42 fresh=True max_root_age_
[ 1.8s] kyc_session principal=principal1 driver=sandbox hint=sandbox:appr
oved:g3 outcome=approved grade=3
[ 11.7s] identity_credential principal=principal1 credential=82e7fdd03170...
[ 20.5s] agents_minted count=3
```

Recording of 23 October, on the public testnet

KYC sandbox on the public testnet.