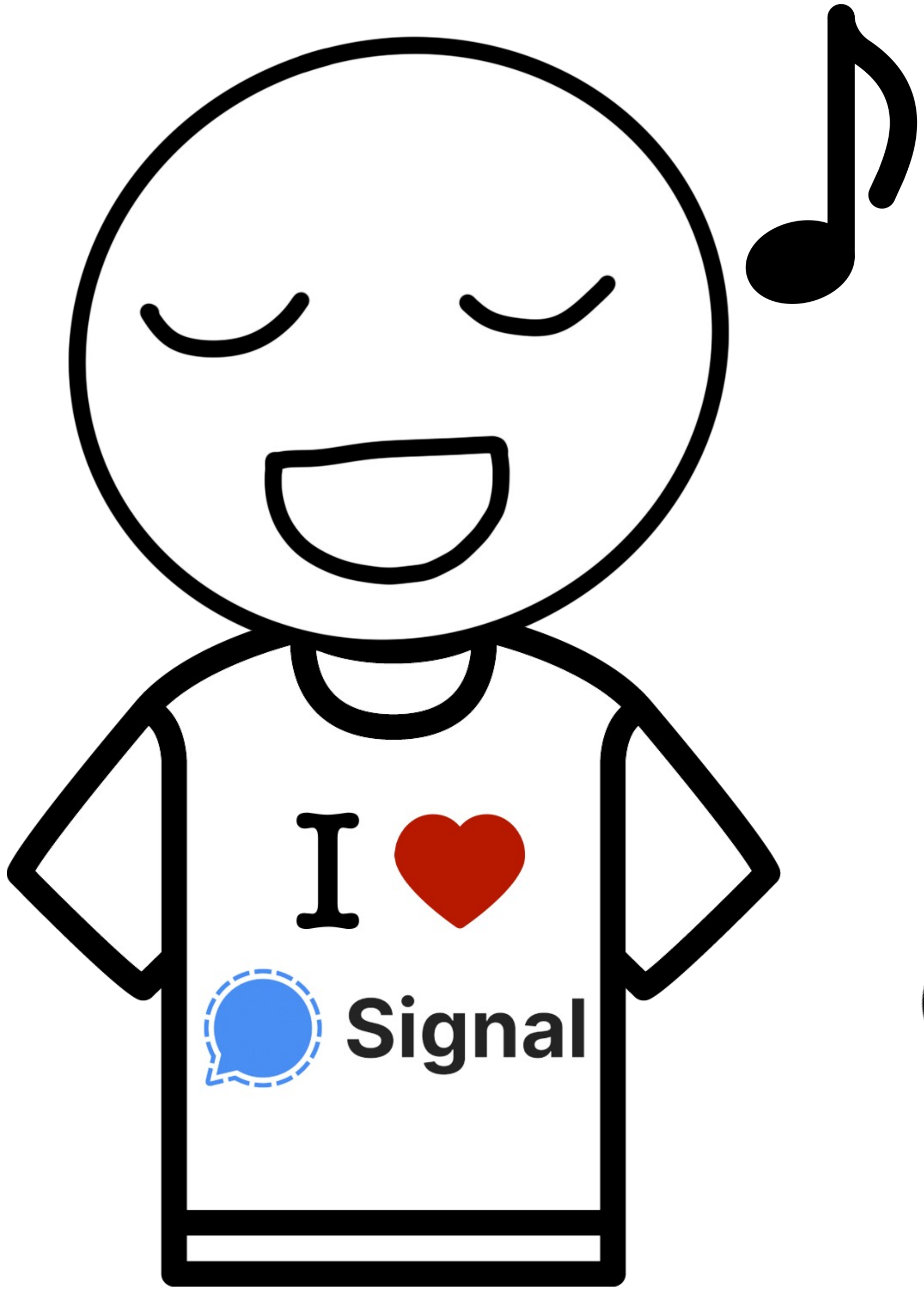
Secret Key Recovery in a Global-Scale End-to-End Encryption System

Graeme Connell*[a] **Vivian Fang***[b] Rolfe Schmidt*[a]
Emma Dauterman[b] Raluca Ada Popa[b]

[a]*Signal Messenger* [b]*UC Berkeley*

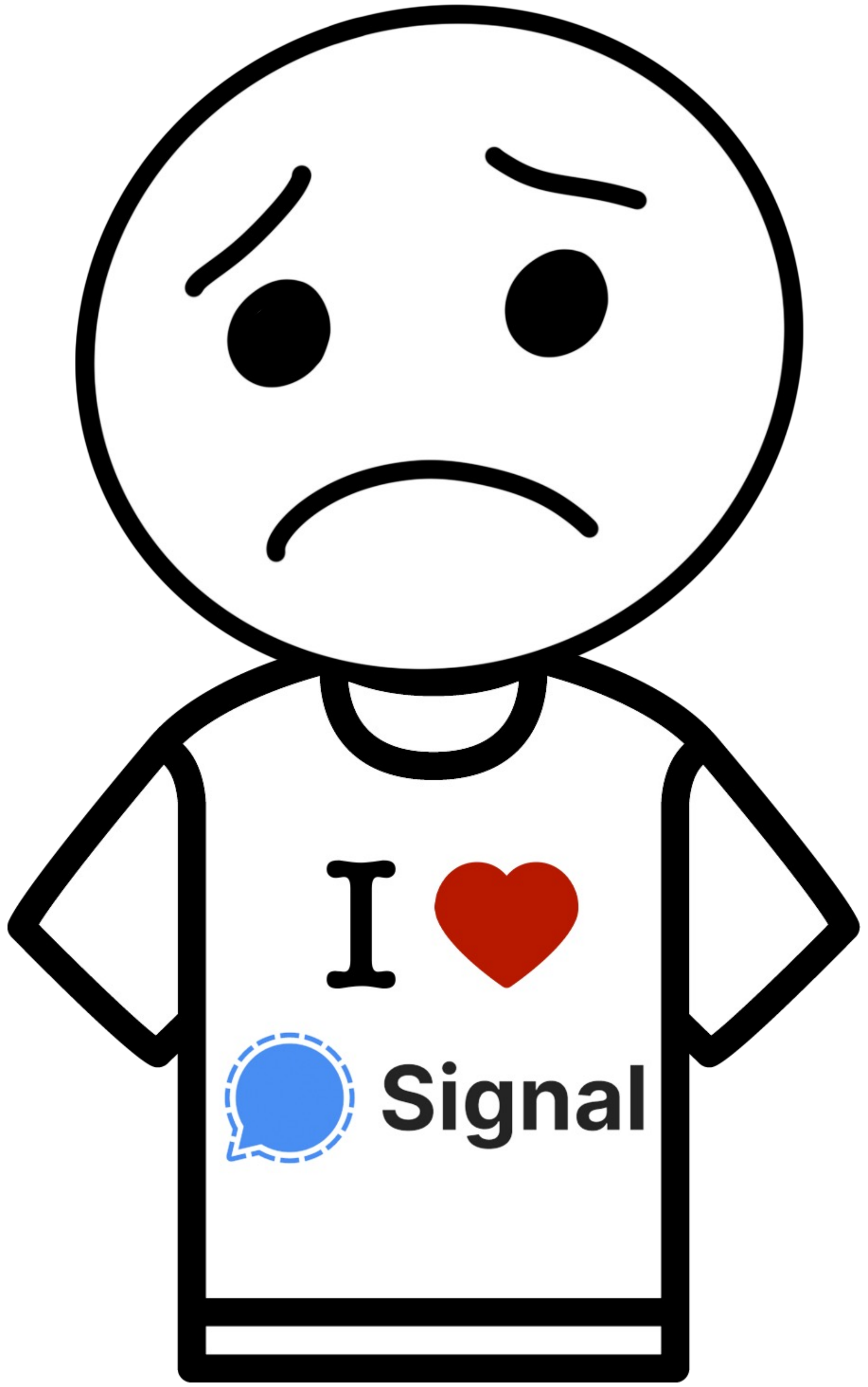
ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE
ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL
ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

OSDI 2024



Bob is using end-to-end encrypted messaging.

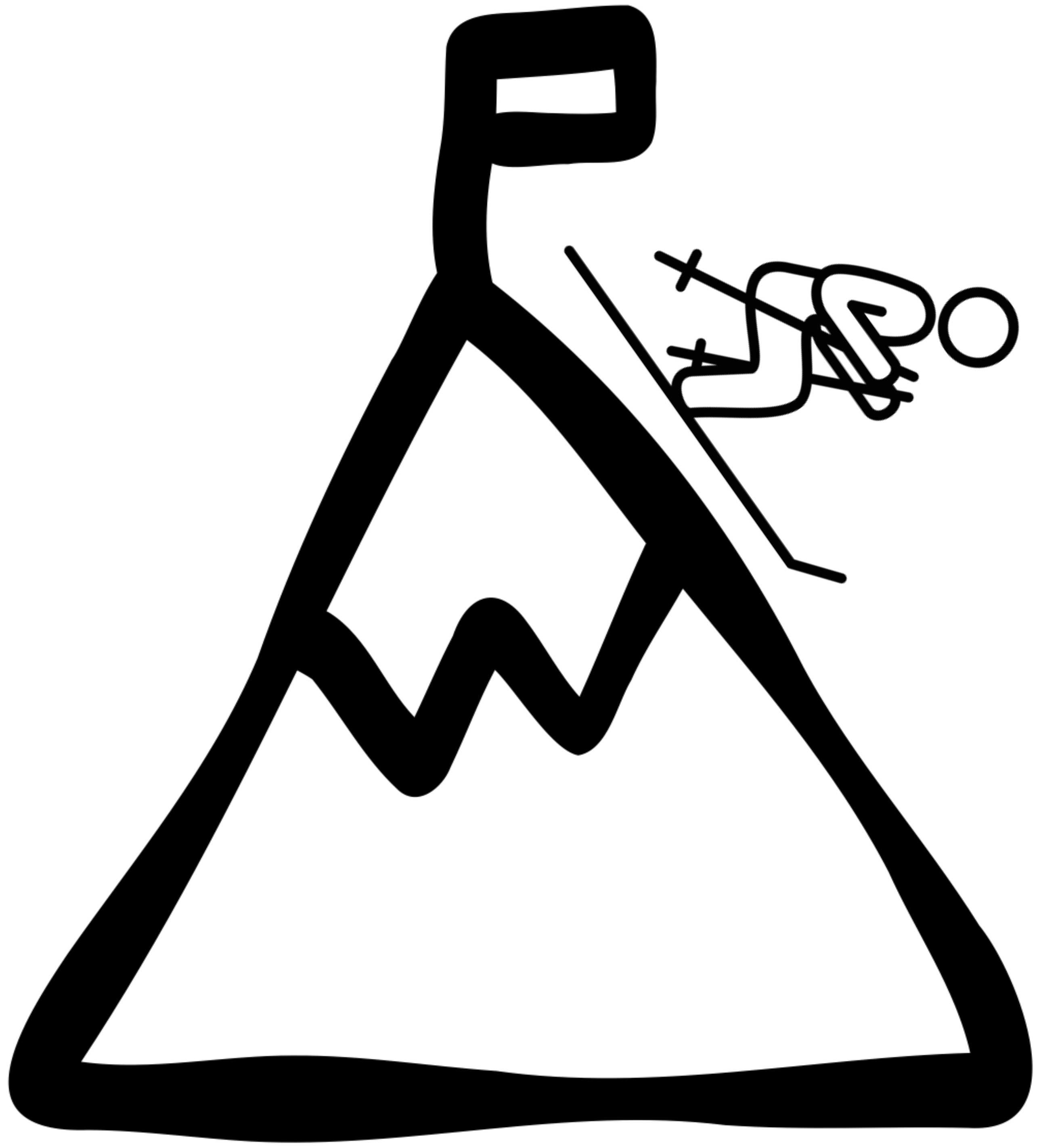
Bob is pleased!

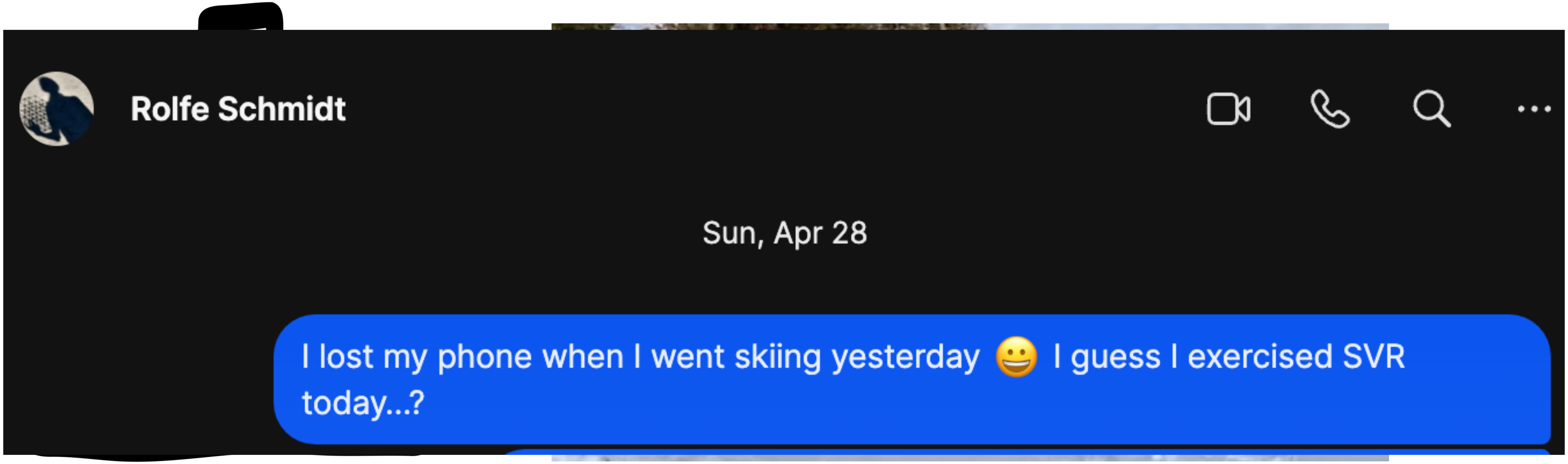


Bob broke his phone!

His secret key is gone.

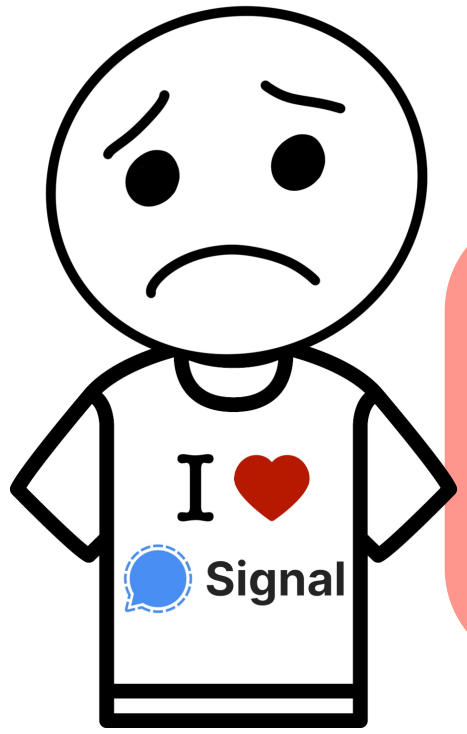
Bob is displeased.




Rolfe Schmidt
Sun, Apr 28
I lost my phone when I went skiing yesterday 😀 I guess I exercised SVR today...?

How to back up secret keys?

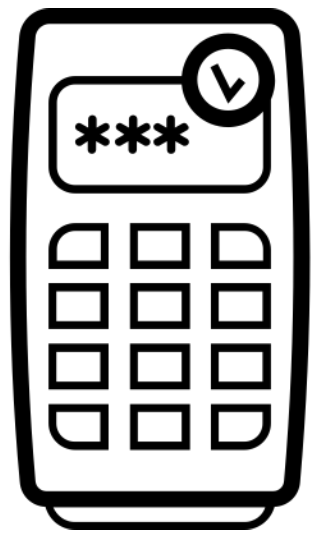
Signal



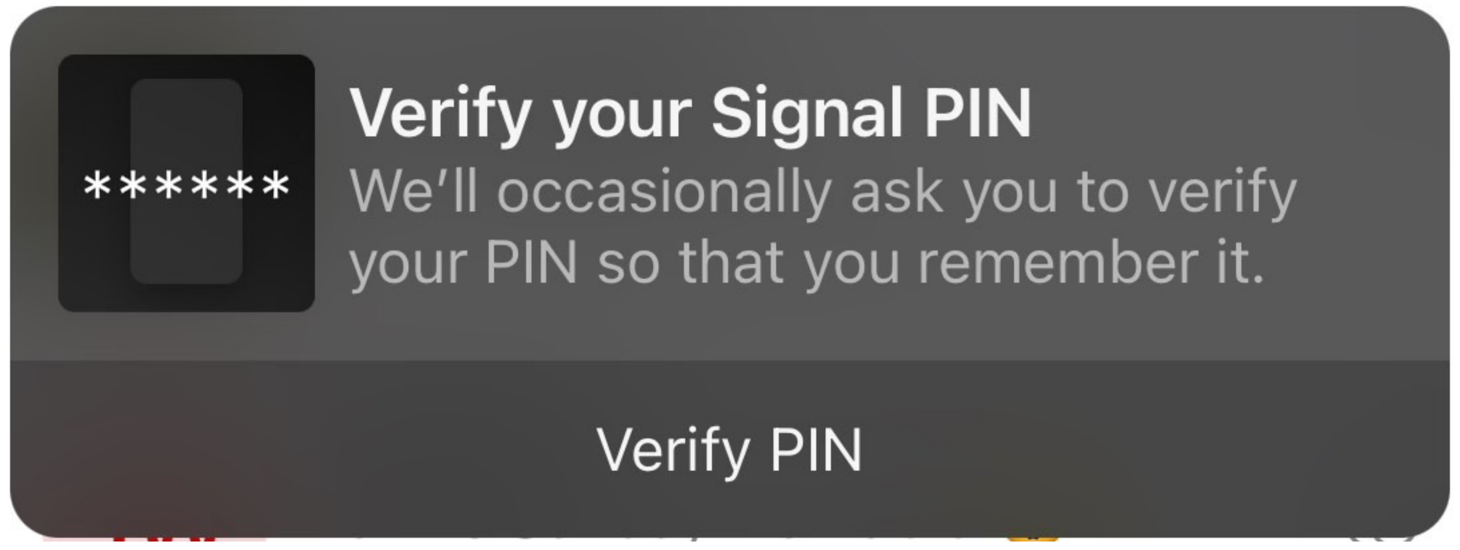
Problem: Signal has Bob’s secret key and can decrypt Bob’s messages

How to back up secret keys?

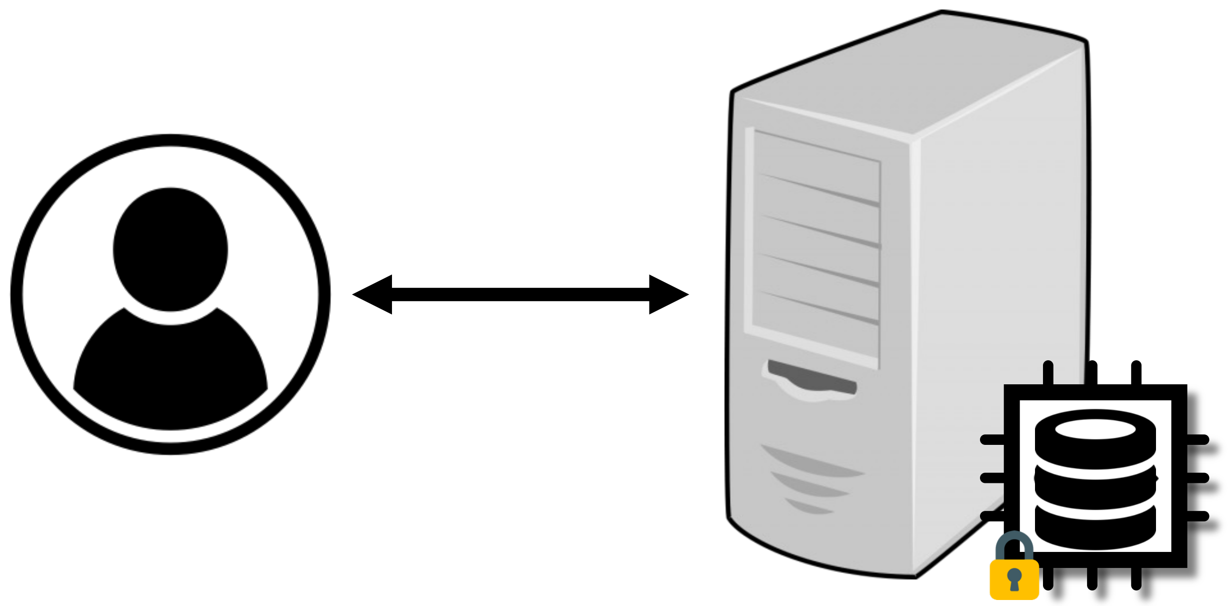
Printing out keys is not user-friendly.



Use low-entropy PIN to derive secret key 



Limit PIN guesses with secure hardware

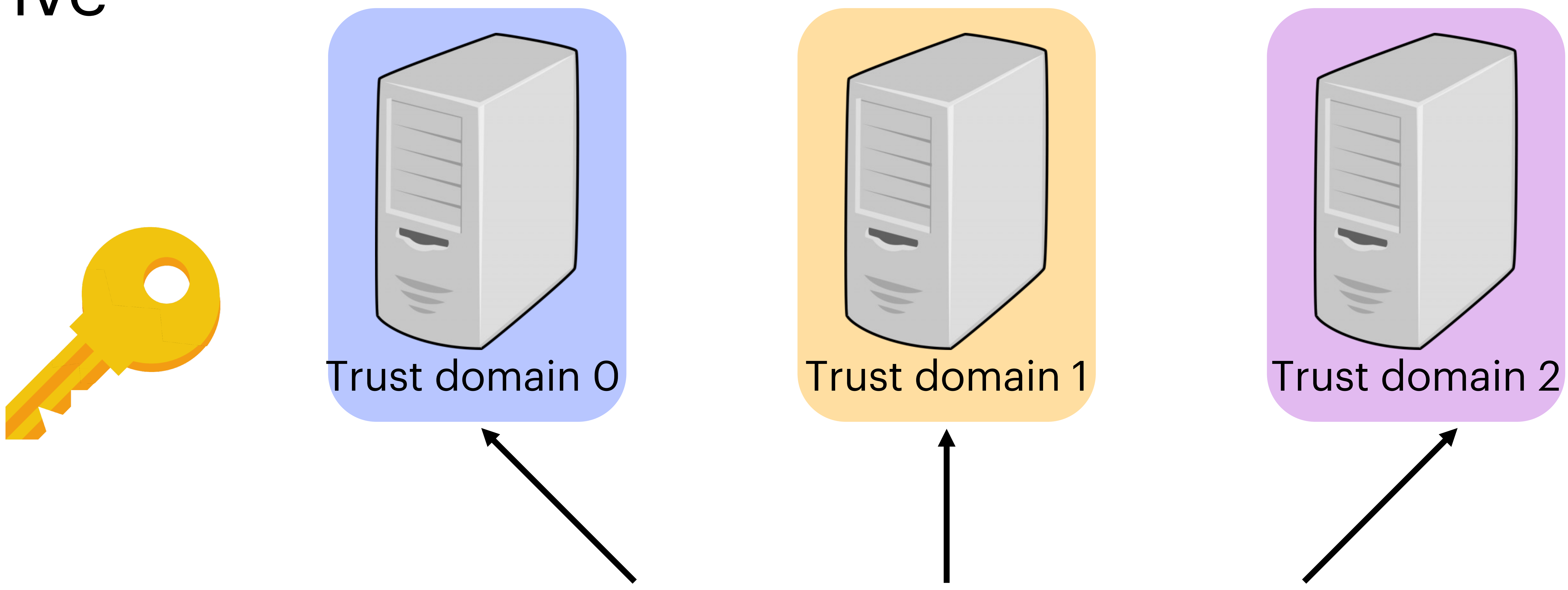


Problem: PIN can be brute-forced

Problem: Single type of secure hardware can be compromised

How to back up secret keys?

Use low-entropy PIN to derive **shares* of** secret key, with enforced guess limit.

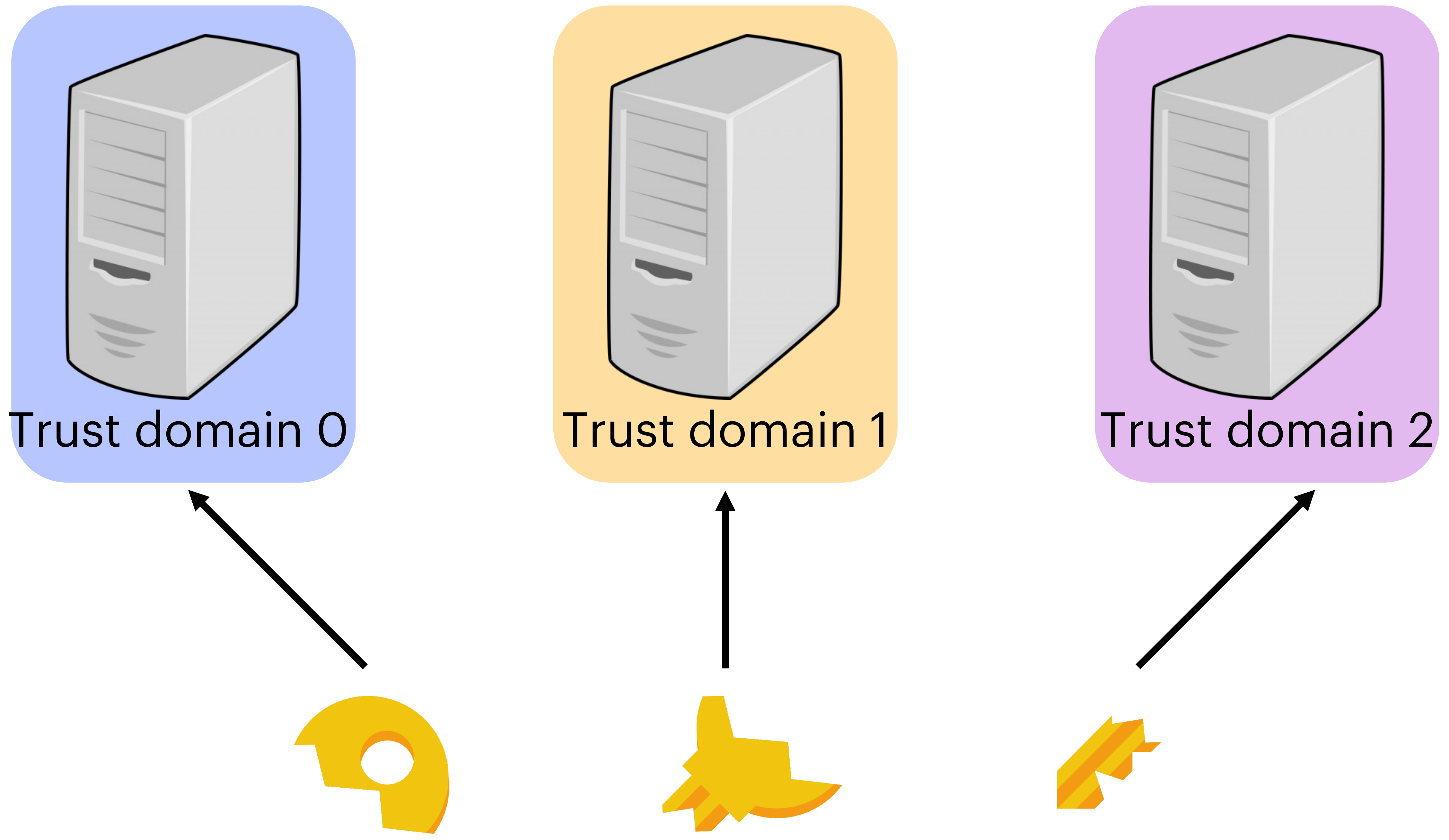


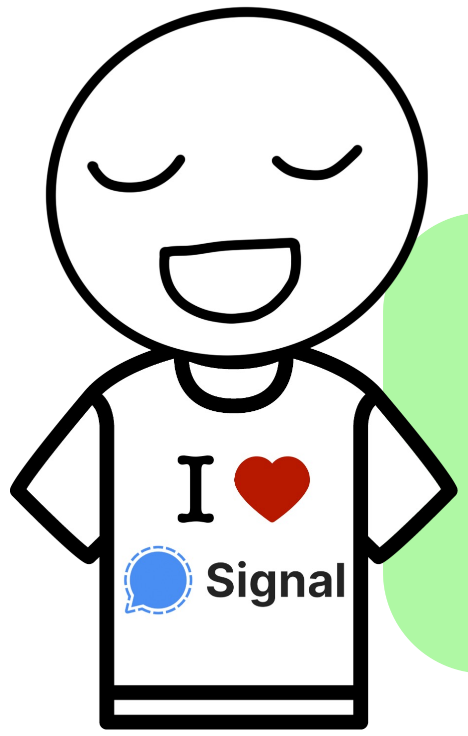
How to back up secret keys?

Use low-entropy PIN to derive **shares* of** secret key, with enforced guess limit.

**masked shares, see paper for details.*

Motivation: Heterogenous secure hardware is unlikely to be compromised *all at once*.





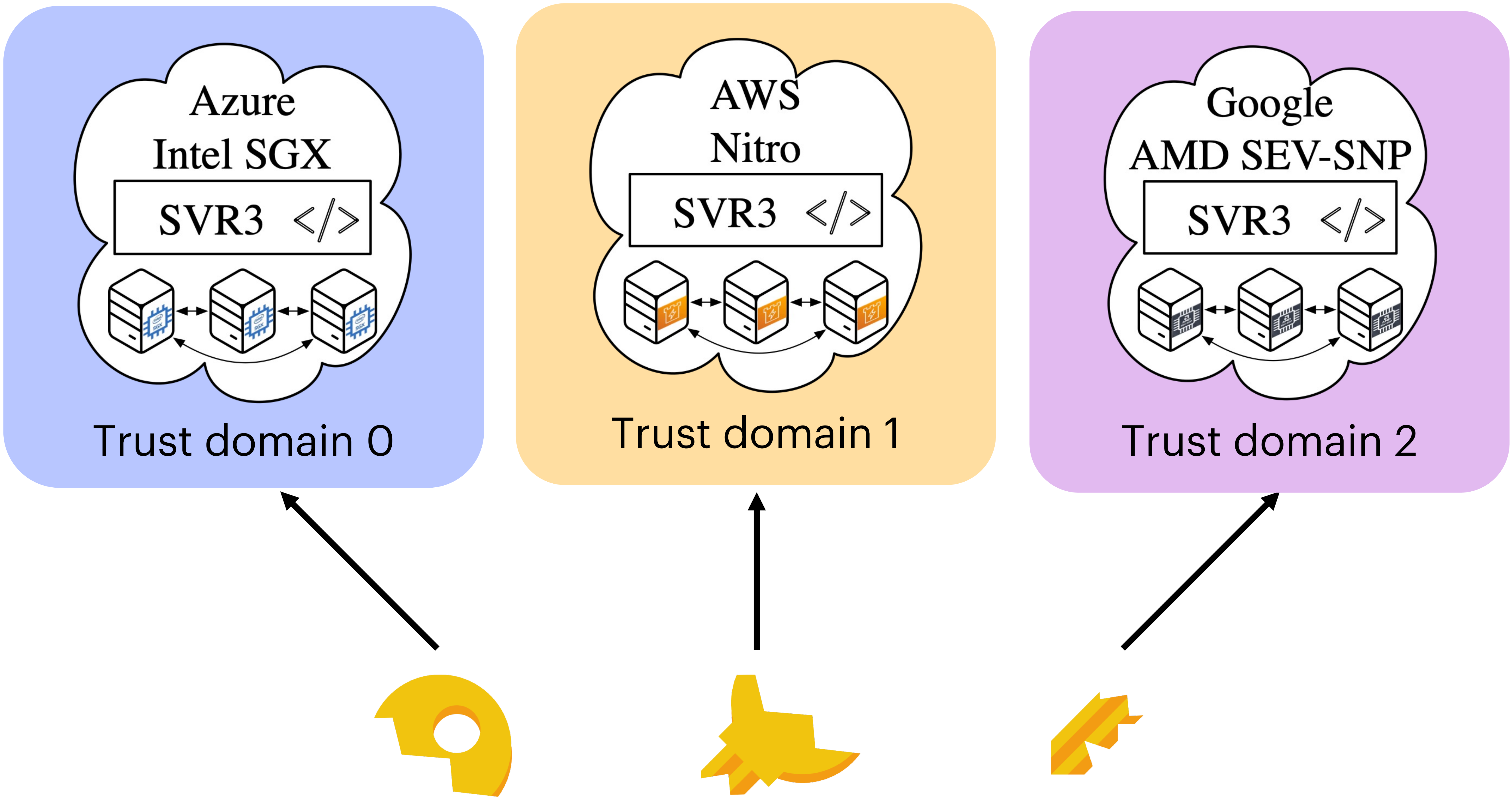
No single trust domain can compromise Bob's secret key

Secure Value Recovery 3 (SVR3)

SVR3 is the first cross-enclave, cross-cloud deployed system.

Defends against internal and external attackers.

Capacity for 500M users @ **$0.0009**/user/year.

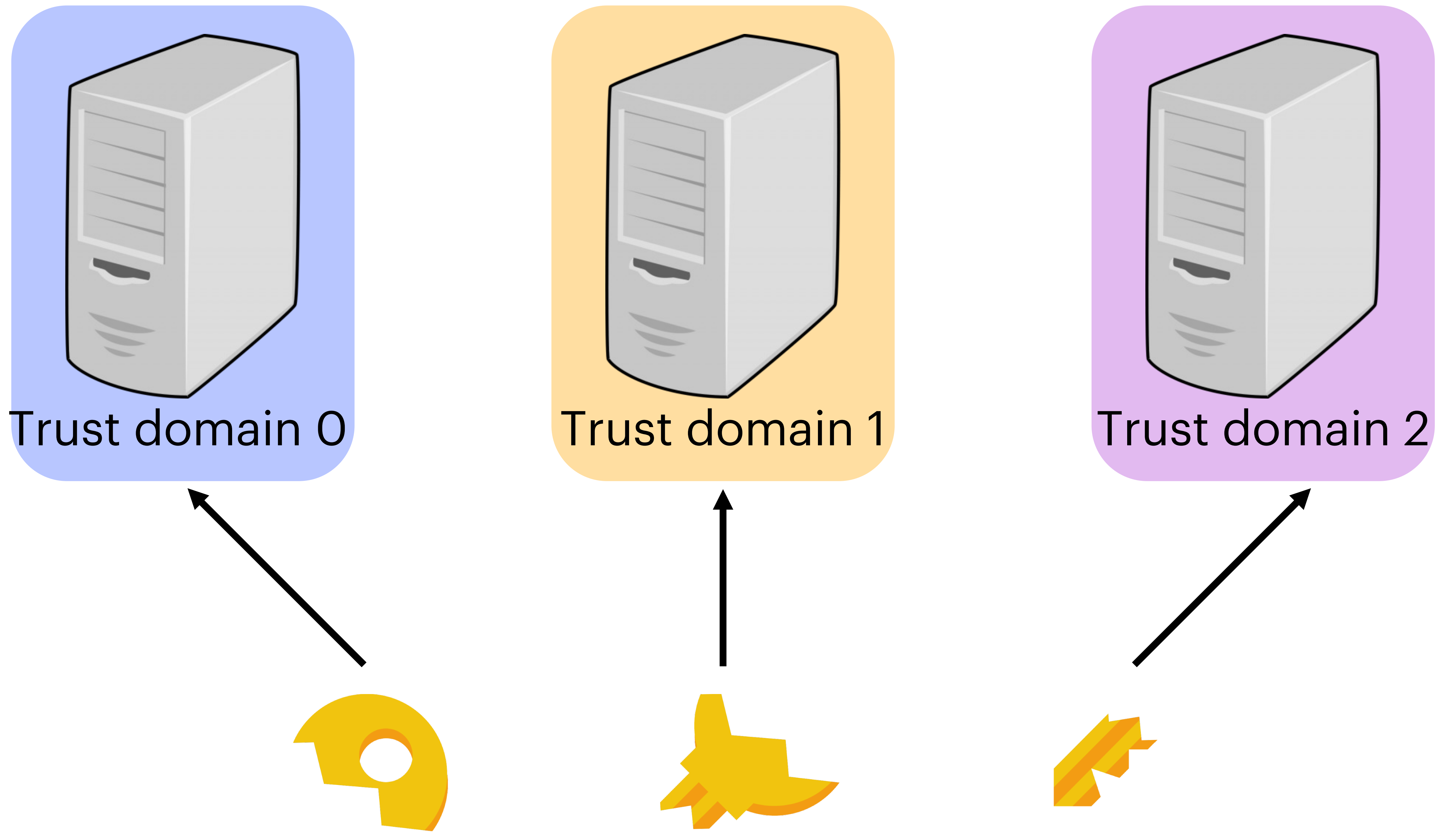


SVR3 Roadmap

Layered security guarantees

Building a SVR3 backend

Evaluation

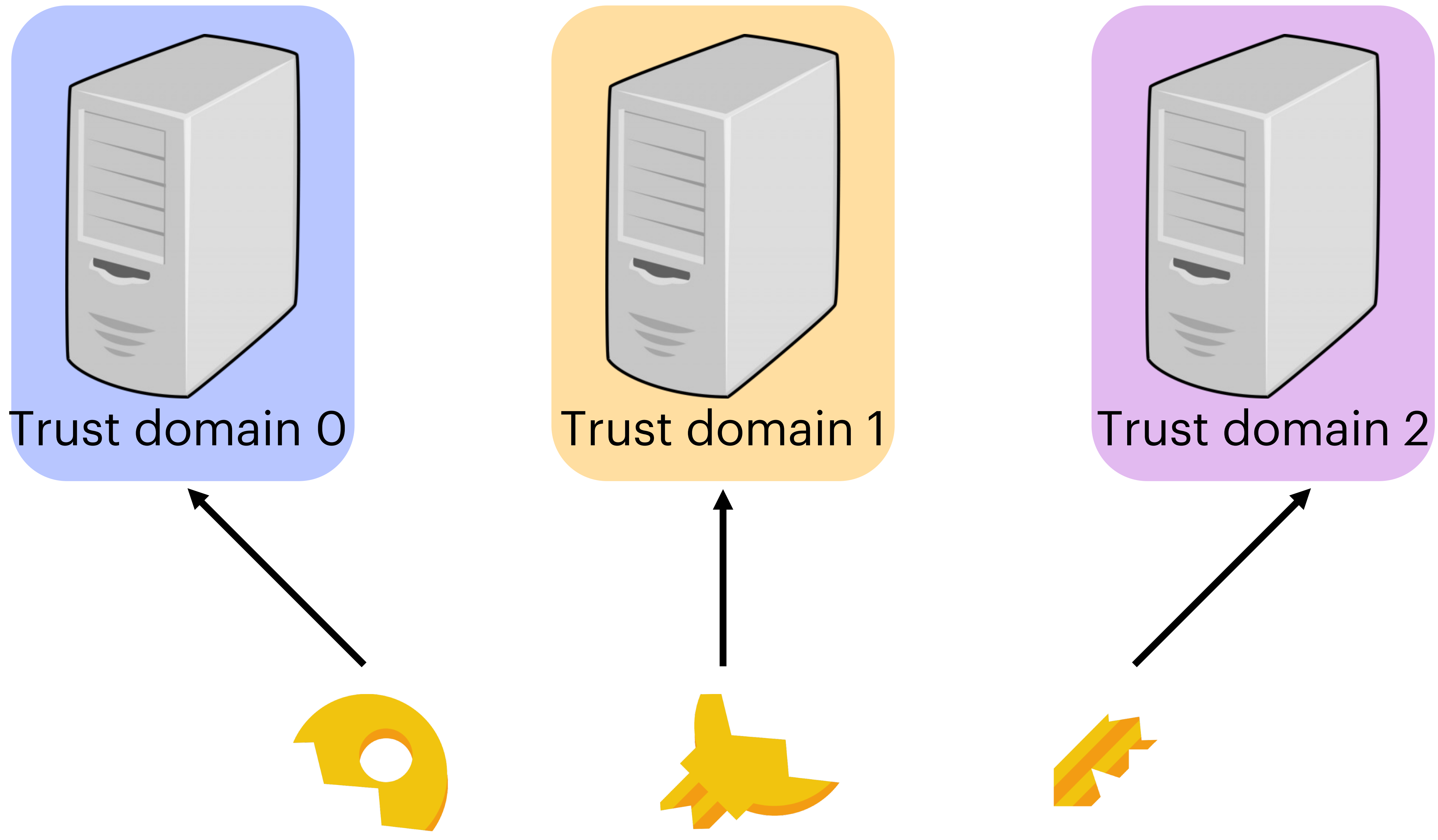


SVR3 Roadmap

→ **Layered security guarantees**

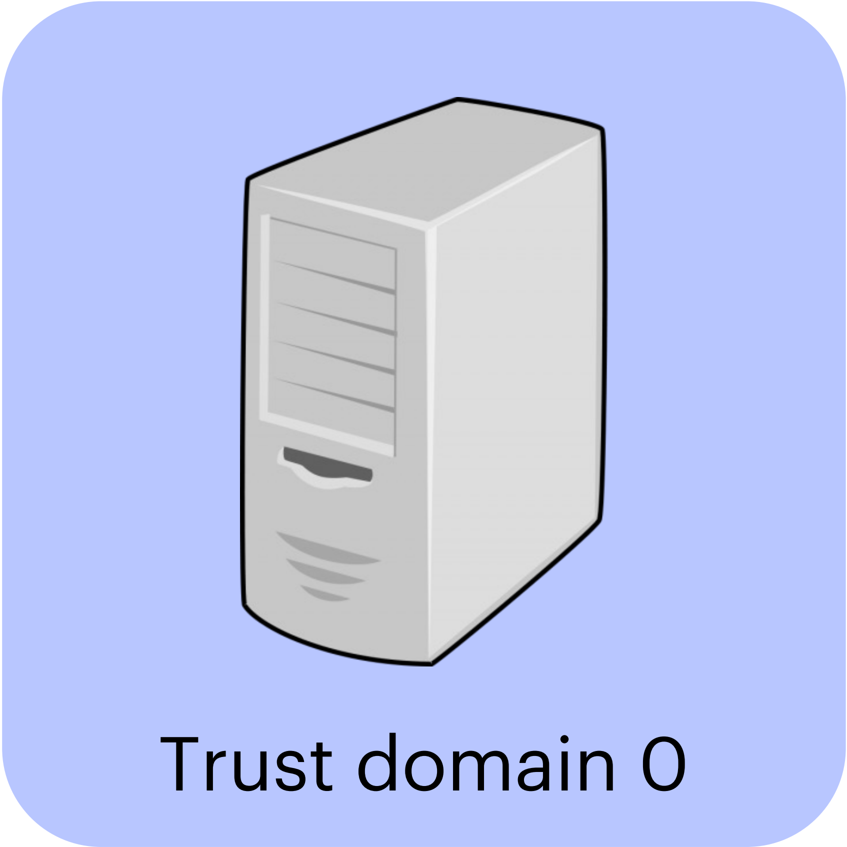
Building a SVR3 backend

Evaluation



Trust Domain

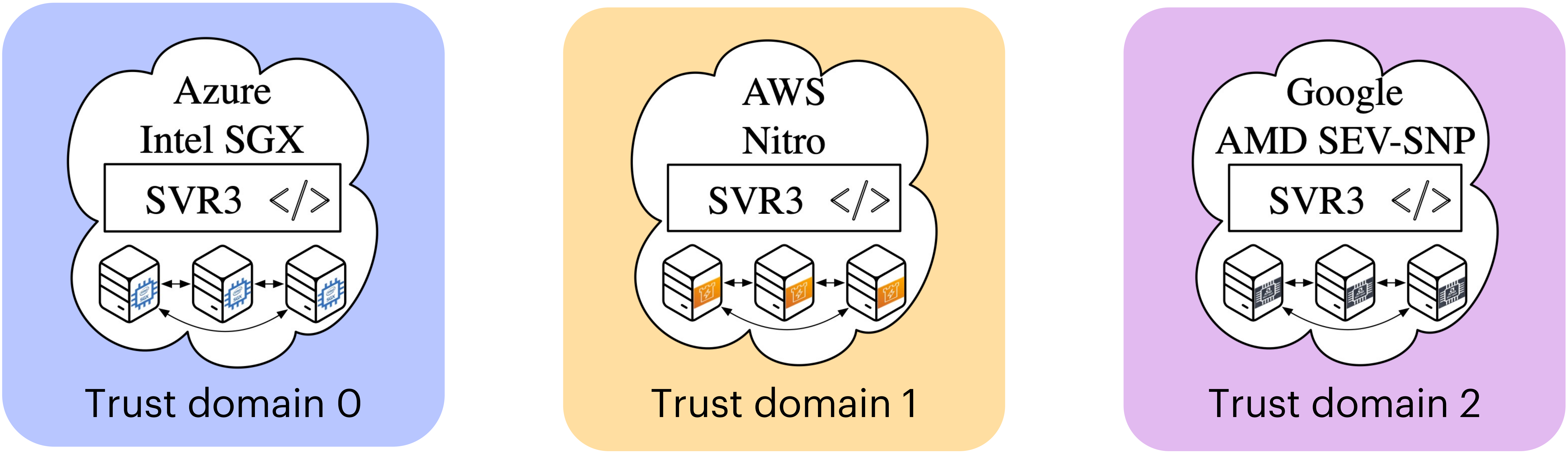
A trust domain is comprised of a **replicated enclave cluster** running on a single type of **secure hardware** on a single **cloud provider**.



Trust Domain

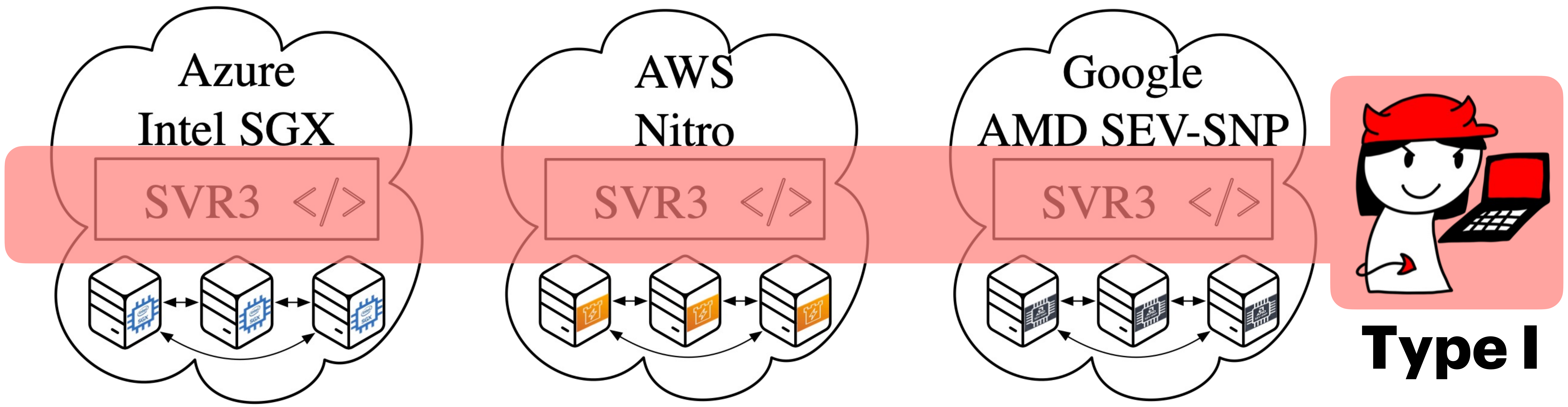
A trust domain is comprised of a **replicated enclave cluster** running on a single type of **secure hardware** on a single **cloud provider**.

Different trust domains
→ **Heterogenous** secure hardware and clouds.



Attackers SVR3 defends against

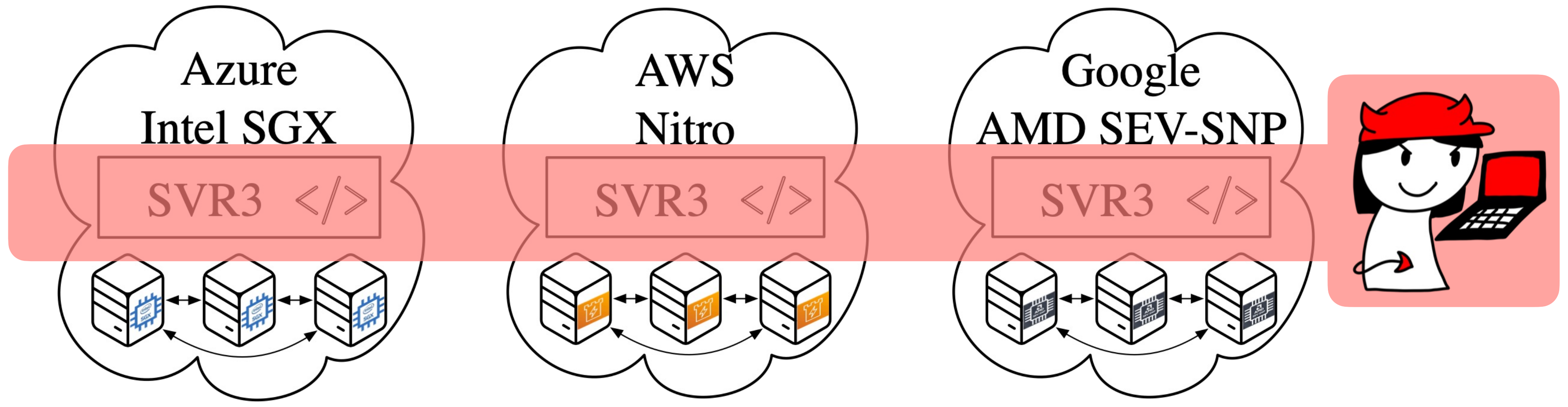
Type I: System administrators (e.g., Signal employees)



Type I Attacker

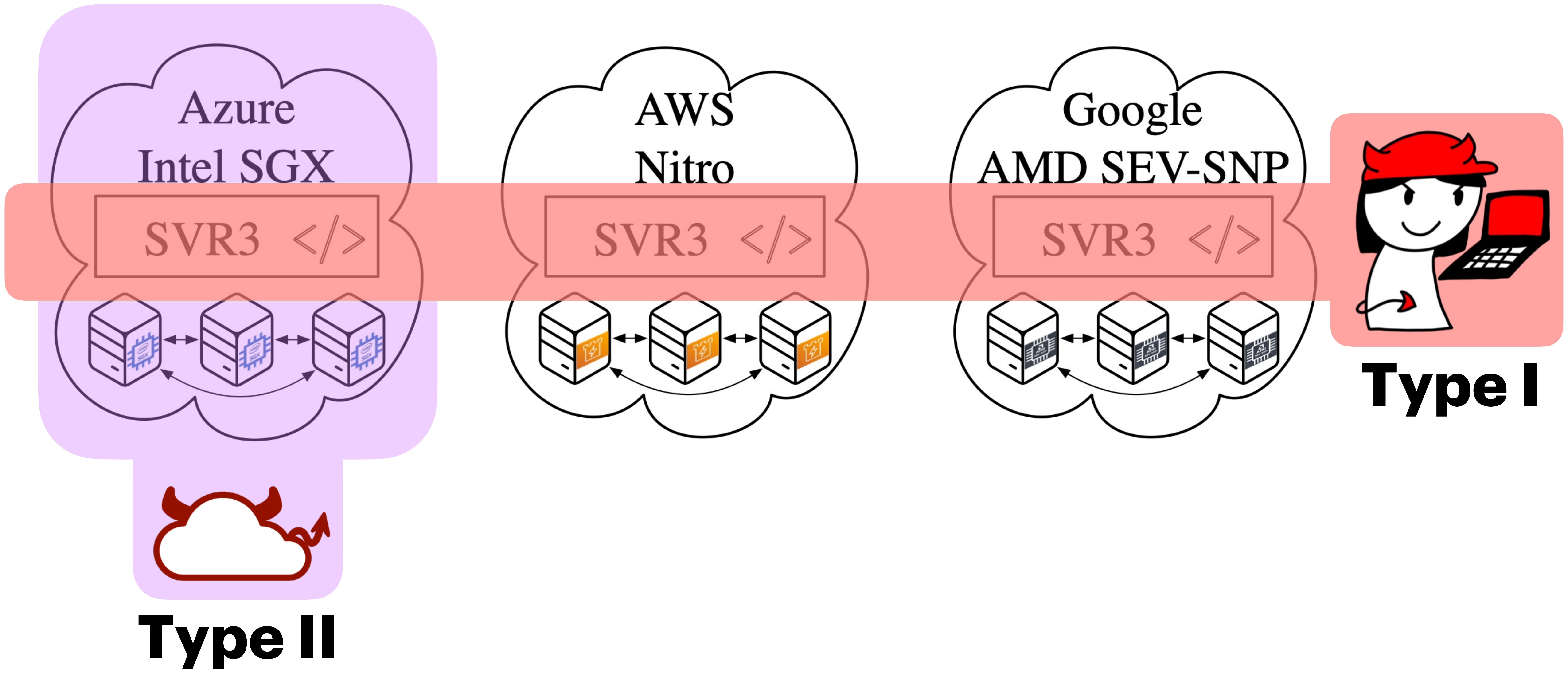
System administrators (e.g., Signal employees):

- Can compromise deployment.
- Spin up & spin down machines.
- Deploy malicious code on servers.
- No physical access to cloud machines, but has root access.



Attackers SVR3 defends against

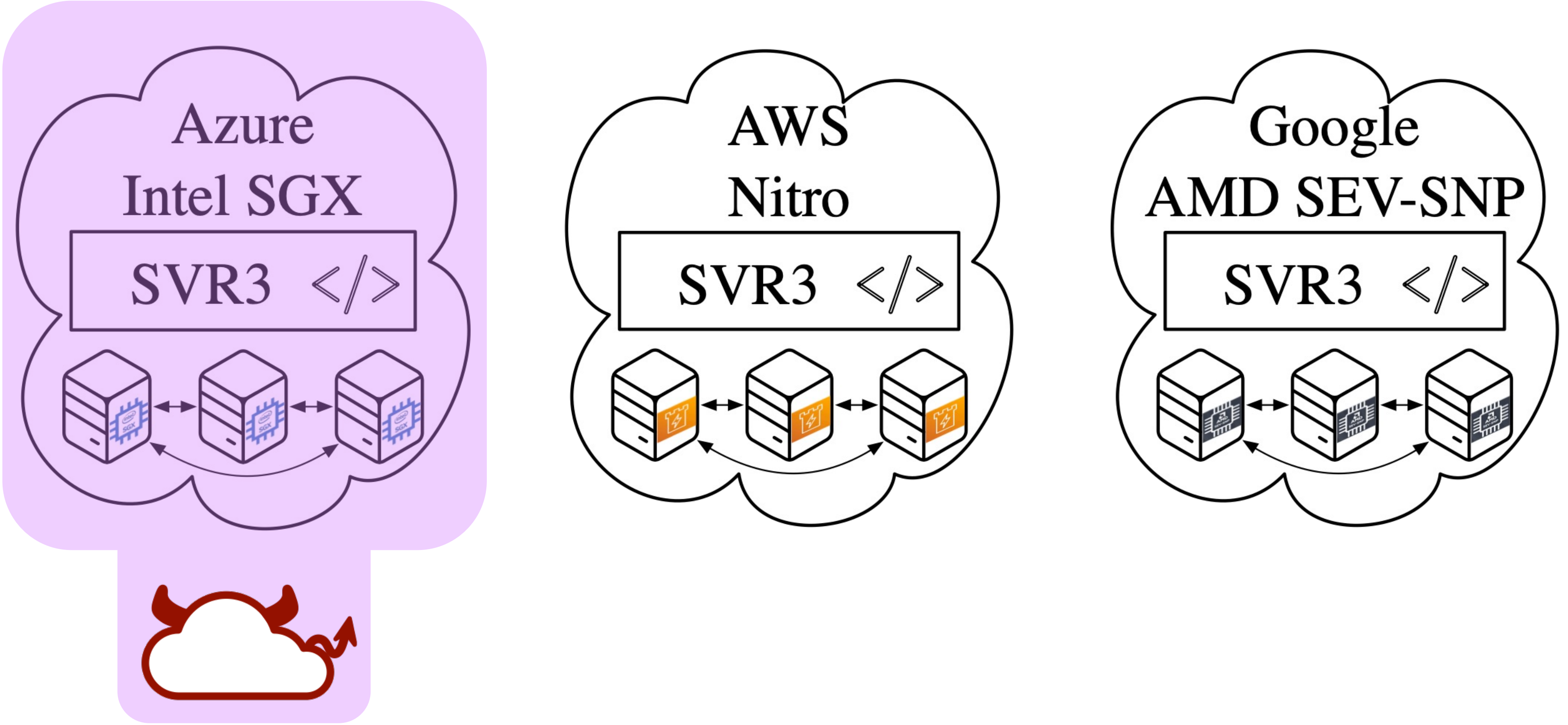
Type I: System administrators (e.g., Signal employees)
Type II: Cloud provider (e.g., Azure)

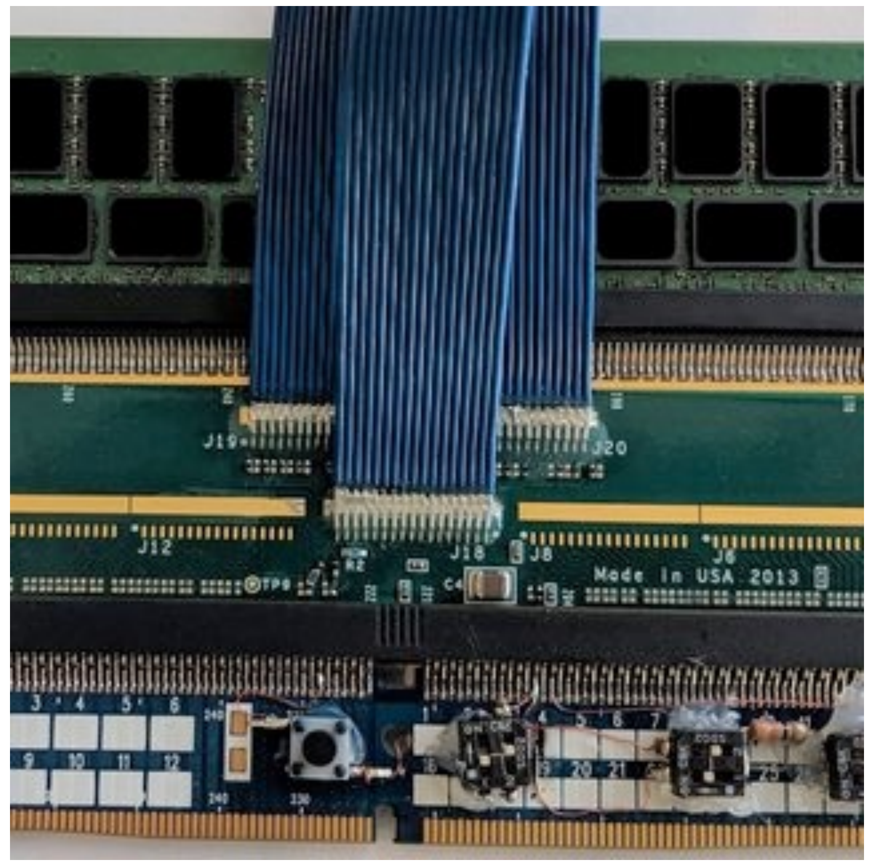


Type II Attacker

Cloud provider (e.g., Azure):

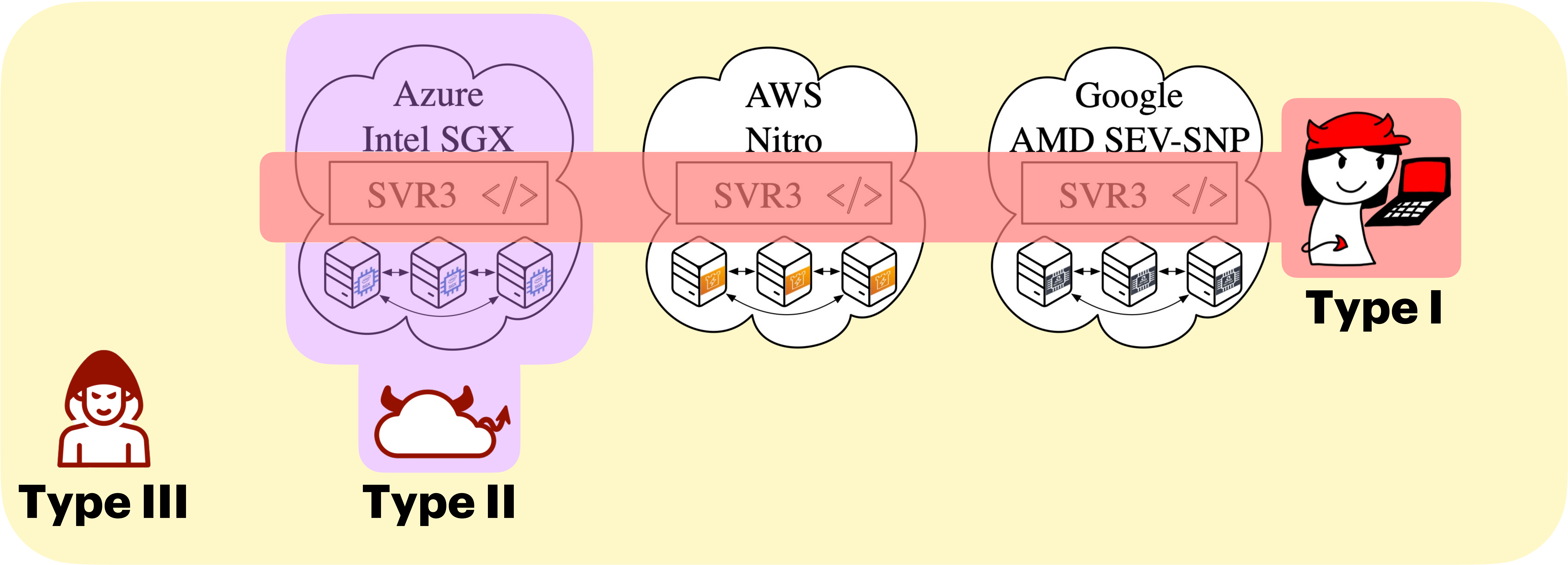
- Physical access to deployment.
- DIMM interposer attacks.
  - Can roll back enclaves.





Attackers SVR3 defends against

Type I: System administrators (e.g., Signal employees)
Type II: Cloud provider (e.g., Azure)
Type III: External (e.g., hacker)



Security guarantees

When deployed on n trust domains with m replicas per trust domain and given parameters t, s, SVR3 can, without letting an attacker compromise users' secret keys, tolerate:

Total compromise of at most t trust domains.
(Security *across* trust domains.)

Software rollback attacks, *and* at most s physical rollback attacks inside each trust domain before that trust domain is totally compromised.
(Security *within* a trust domain.)

In our deployment, $n = 3, m = 7, t = 2, s = 2.$
(3-of-3)

Availability

SVR3 provides availability when $t + 1$ trust domains are operating "correctly":

- Enclaves in the trust domain are online.
- None of the enclaves in the trust domain are under attack.

Analogous to normal operation.

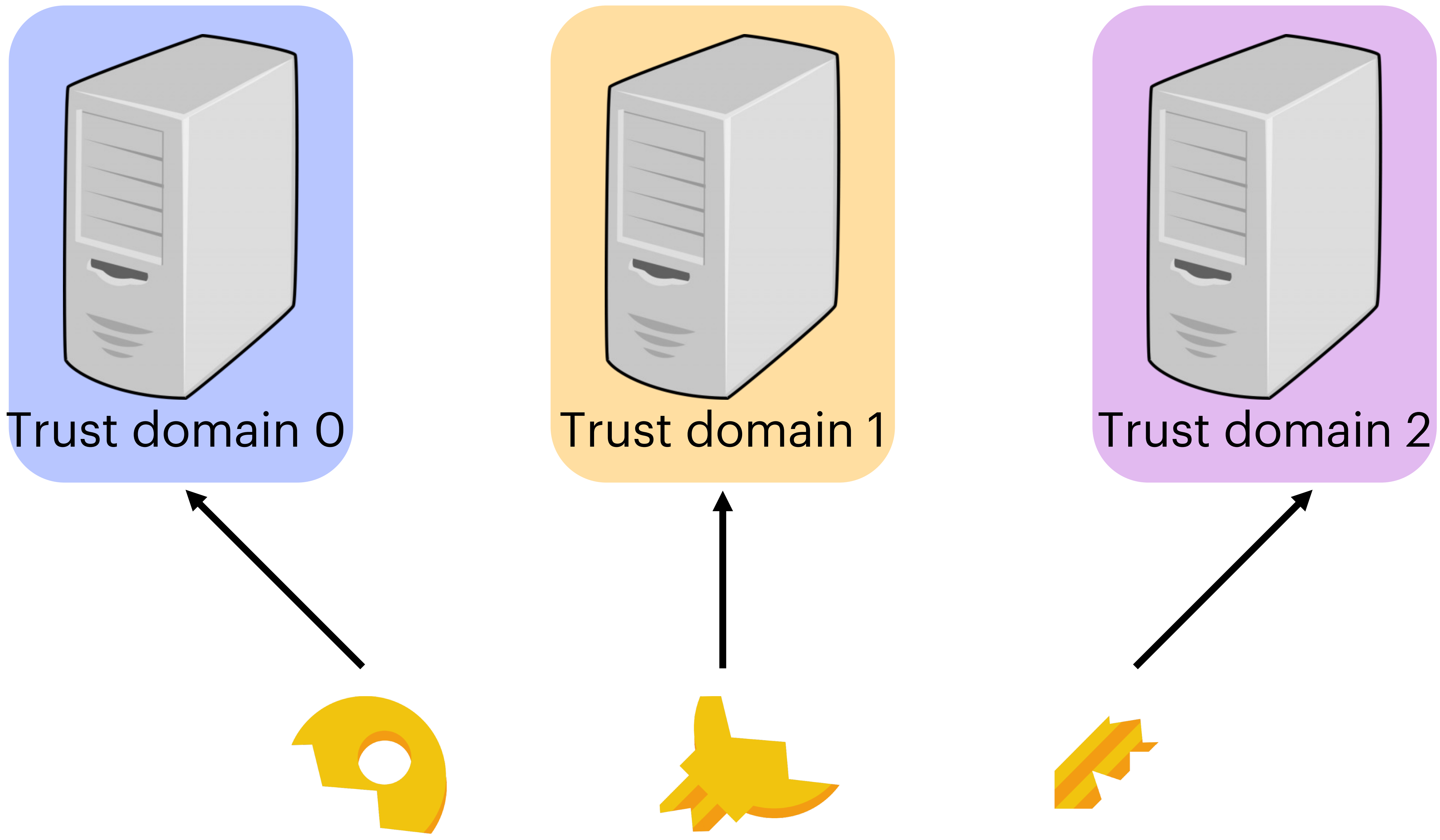
When under attack, we prioritize **safety** over availability.

SVR3 Roadmap

Layered security guarantees

→ **Building a SVR3 backend**

Evaluation



Enclave model

Application-level attestation.

Memory access control.

Attacker has page-level rollback granularity.

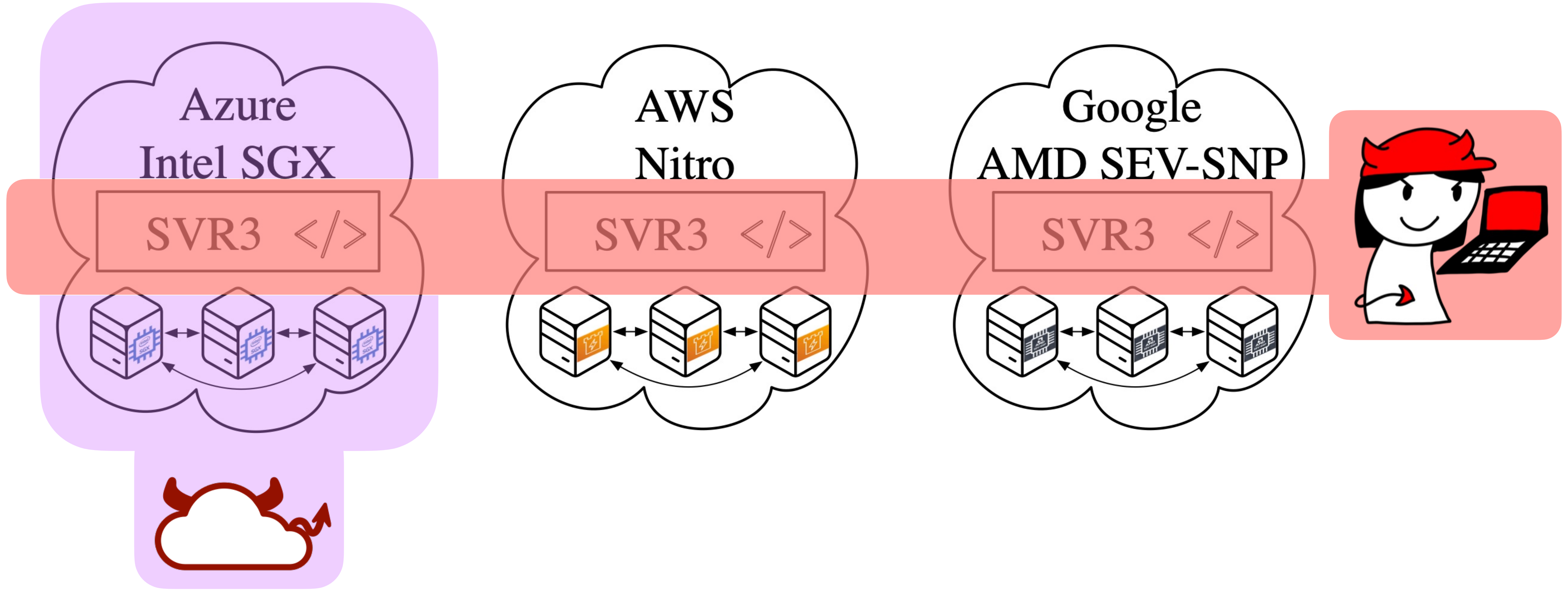
If an enclave type loses these guarantees, then the trust domain with that enclave type is considered compromised…

But SVR3 **still** protects user secrets when at most t trust domains are compromised.

Rollback attacks

Enclaves are susceptible to software and physical rollback attacks.

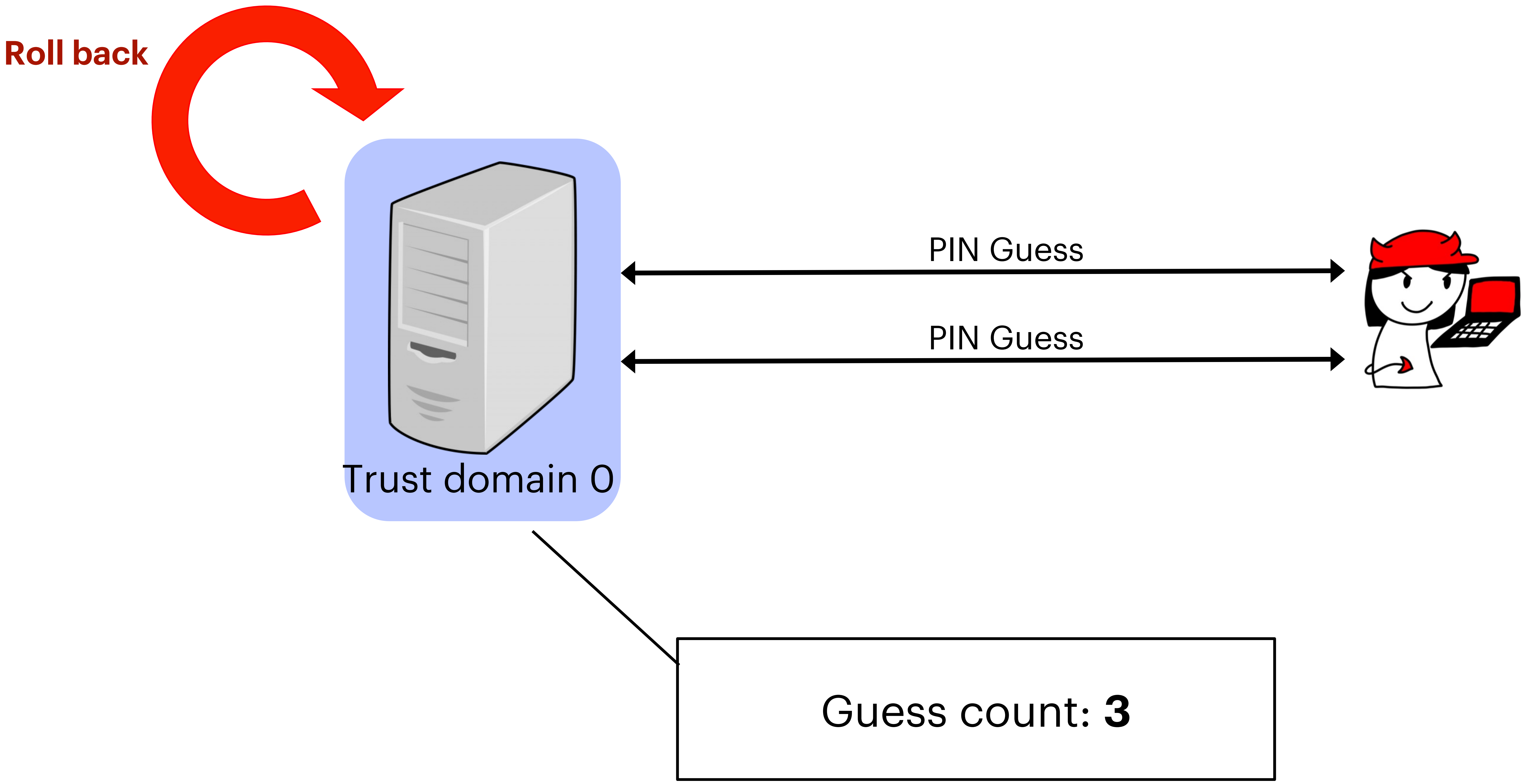
Type I **Type II**



Rollback attacks undermine guess limits

Low-entropy PIN → Need to enforce guess limit.

Rollback attacks → Attacker can get more PIN guesses.



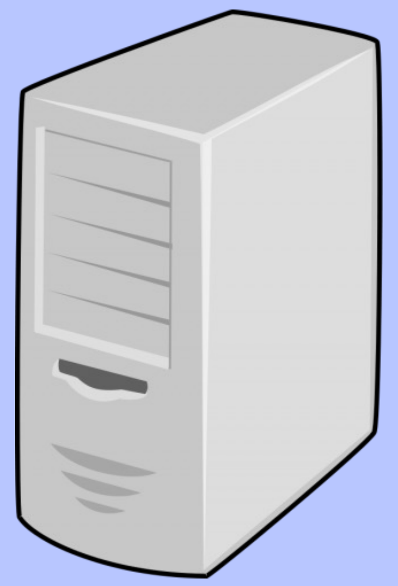
Software rollback attacks

External state stored via data sealing can be **rolled back.**

Protect against by storing entire database in memory.

Never storing external state
→ No external state can be rolled back!

Problem: If we lose a machine, we lose all its state!
→ Replicate and run cluster of enclaves using Raft inside trust domain.



Trust domain 0

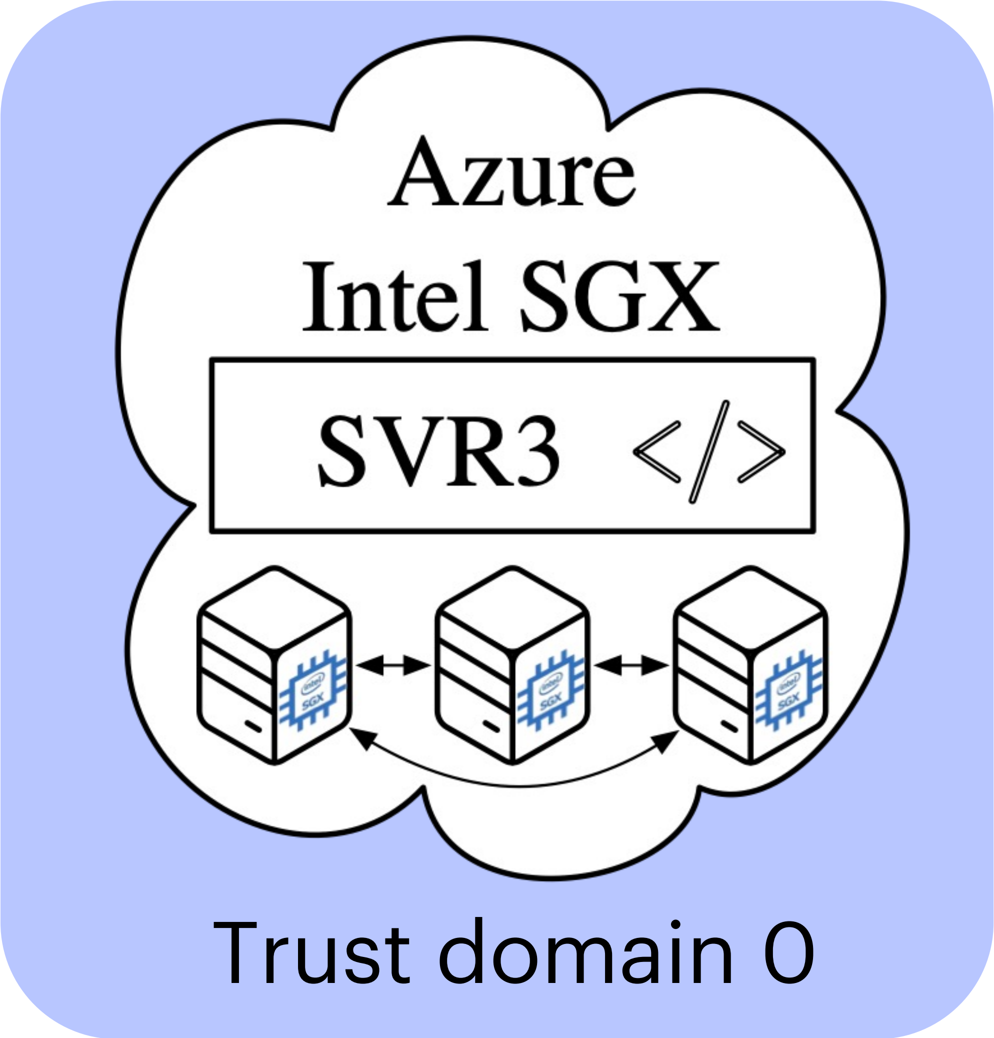
Software rollback attacks

External state stored via data sealing can be **rolled back.**

Protect against by storing entire database in memory.

Never storing external state
→ No external state can be rolled back!

Problem: If we lose a machine, we lose all its state!
→ Replicate and run cluster of enclaves using Raft inside trust domain.



Hardware rollback attacks

Roll back internal memory **during execution** by interposing on system bus.

Vanilla Raft is a crash fault tolerant protocol and **loses** safety guarantees in face of HW rollback attacks.

Observation: Physical rollback attacks are harder to carry out.

Harder to compromise trust domain if we protect against s rollback attacks in its cluster.

Raft↺ (Rollback-Resistant Raft)

Rollback resistant consensus protocol:

Hash chain verification on processing AppendEntriesRequest.

Supermajority so quorum intersection includes one non-rolled back server.

Promise round before leader proceeds with update.

Raft$^\circlearrowleft$ safety

(Informal) For every log entry that has been applied to the state machine of a server i: If the number of physically rolled back servers is $\leq s$, server i will never apply a different log entry at that log entry's position.

For safety, we require $m > s$ replicas, but s may be set smaller depending on how many crash failures to tolerate.

See paper for TLA+ specification and full safety proof.

Raft↺ liveness

Liveness when the cluster is under a physical attack is a **non-goal** for SVR3.

When operating normally (**no** physical attacks), we require

$$f_c \leq \left\lfloor \frac{m - s}{2} \right\rfloor$$

crash failures to be live under normal connectivity conditions.

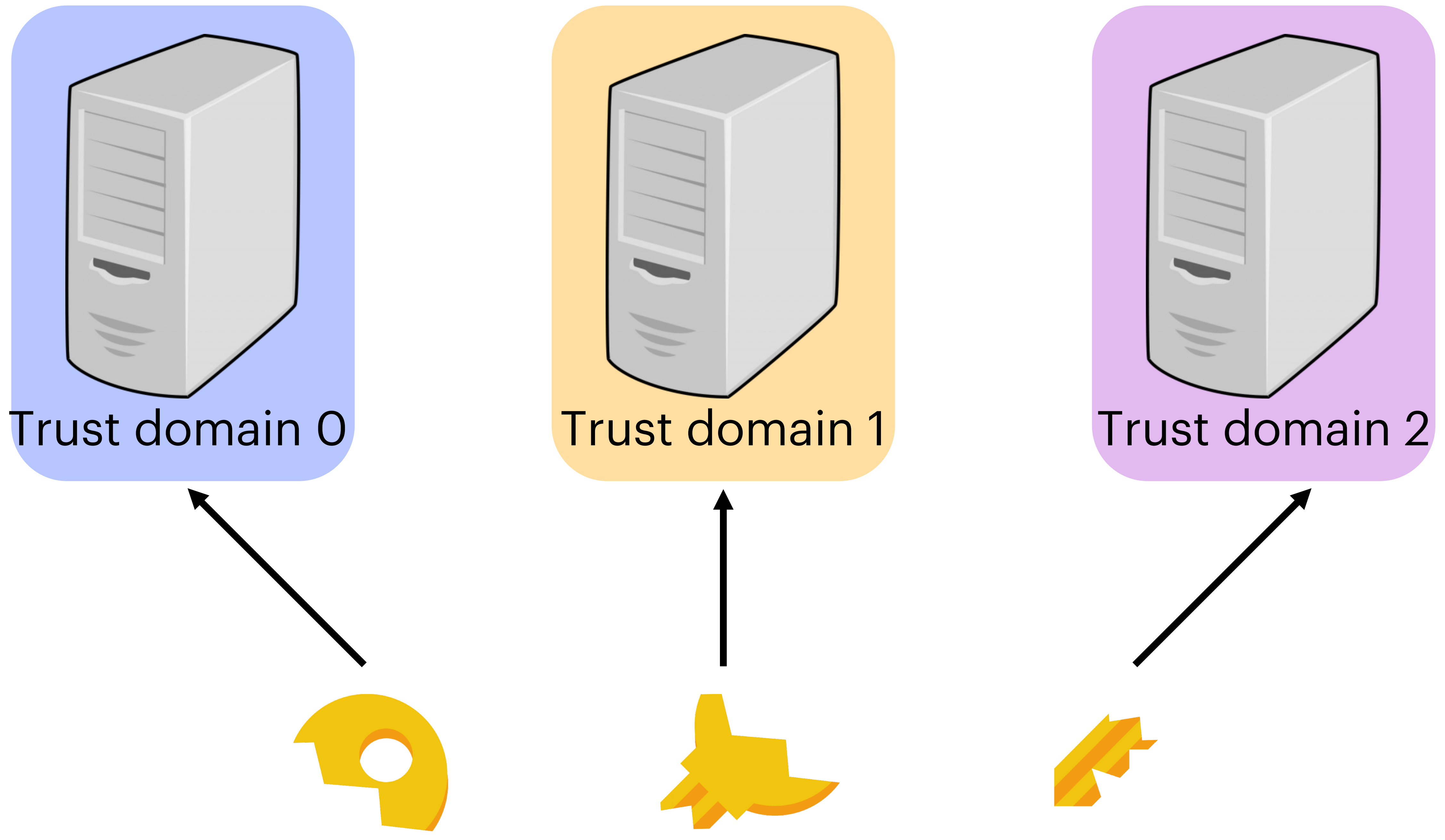
In our deployment, $m = 7, s = 2, f_c \leq 2.$

SVR3 Roadmap

Layered security guarantees

Building a SVR3 backend

→ **Evaluation**

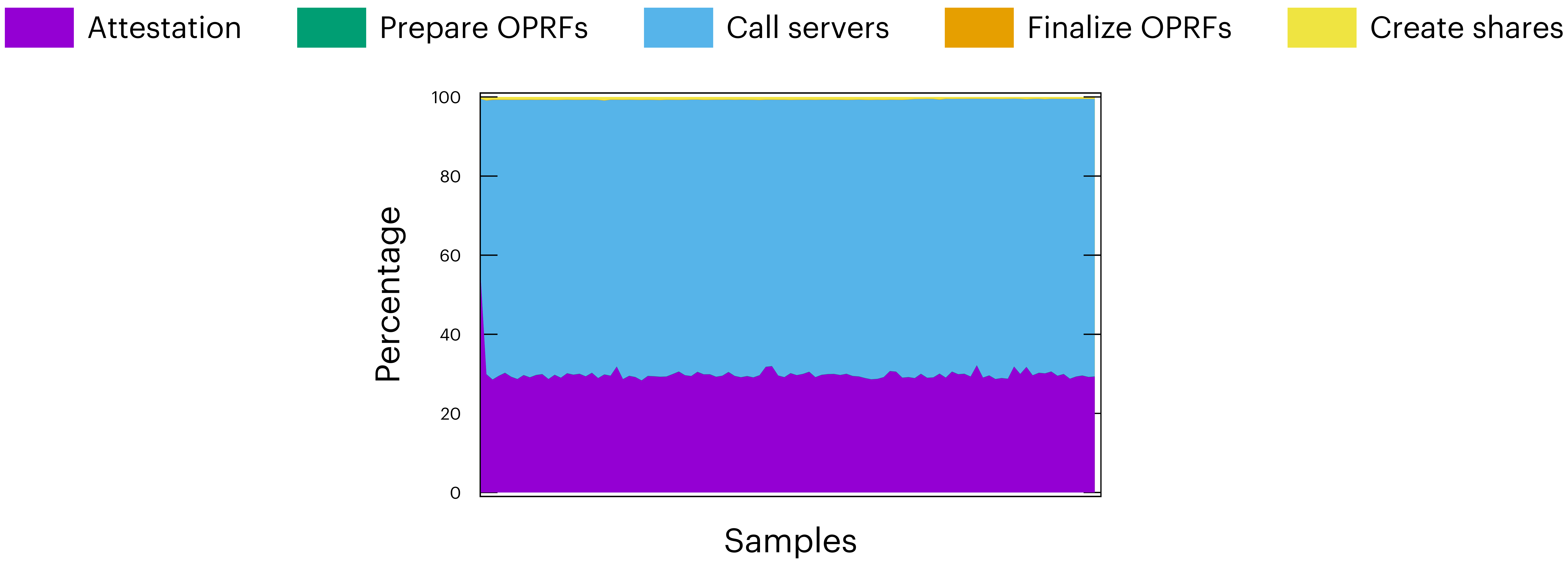


Deployment

- **3** enclave types and clouds: Azure Intel Scalable SGX, GCP AMD SEV-SNP, AWS Nitro.
- Provision for 1 req/s/1M users, ~256B RAM/user.
- Deployment supports capacity of 500M users @ **$0.0009**/user/year.
- Evaluation numbers are on staging cluster provisioned for 10M users.
  - `m5.xlarge` (2 cores, 10 GB RAM)
  - `DC2s_v3` (2 cores, 8 GB RAM)
  - `n2d-standard-2` (2 cores, 8 GB RAM)

End-to-end performance

Average end-to-end latency: **365ms**
Average throughput: **~1000 req/s**



Conclusion

SVR3 enables secret key recovery in a **real-world setting** by **distributing trust** across heterogeneous secure hardware.

Thanks!

full paper

Vivian Fang
vivian@eecs.berkeley.edu
@vivianfxng

https://eprint.iacr.org/2024/887.pdf

https://github.com/signalapp/SecureValueRecovery2

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE
ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL
ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED