When to consider KERI

argonauths

COLOSSI MEMBER

When to consider KERI?

- A need driven by GRC (Governance/Risk/Compliance)
  - “Longevity” of AIDs (key rotations as a proactive measure and recovery potential even in case of compromise)
  - Attribution to AID, not to a static key / avoids key management problem
- Building independent (“sovereign”) trust infrastructures / already existing infrastructure based on Certificate Transparency or DLT
- Looking for practical ways to achieve cryptographic agility (NIST CSWP 39) – in some cases

KERI vs. GRC

- Governance: KERI enables to model federated trust with delegated authority
  - persistent identity (cf. PKI)
  - delegated authority chains / decentralized trust flows
  - issuance doesn't stop at the edge (cf. WebPKI)
- Compliance: Full auditability
- Risk: Contextual and depending on a threat model

What is KERI?

KERI in a nutshell:

- A novel approach to digital identity infrastructure.
- Enhances systems that rely on static cryptographic identifiers in PKI and other protocols by replacing static keys with KERI AIDs. See the deep dive.
- Delivers long-lived digital identities (AIDs) designed for long-term operational continuity:

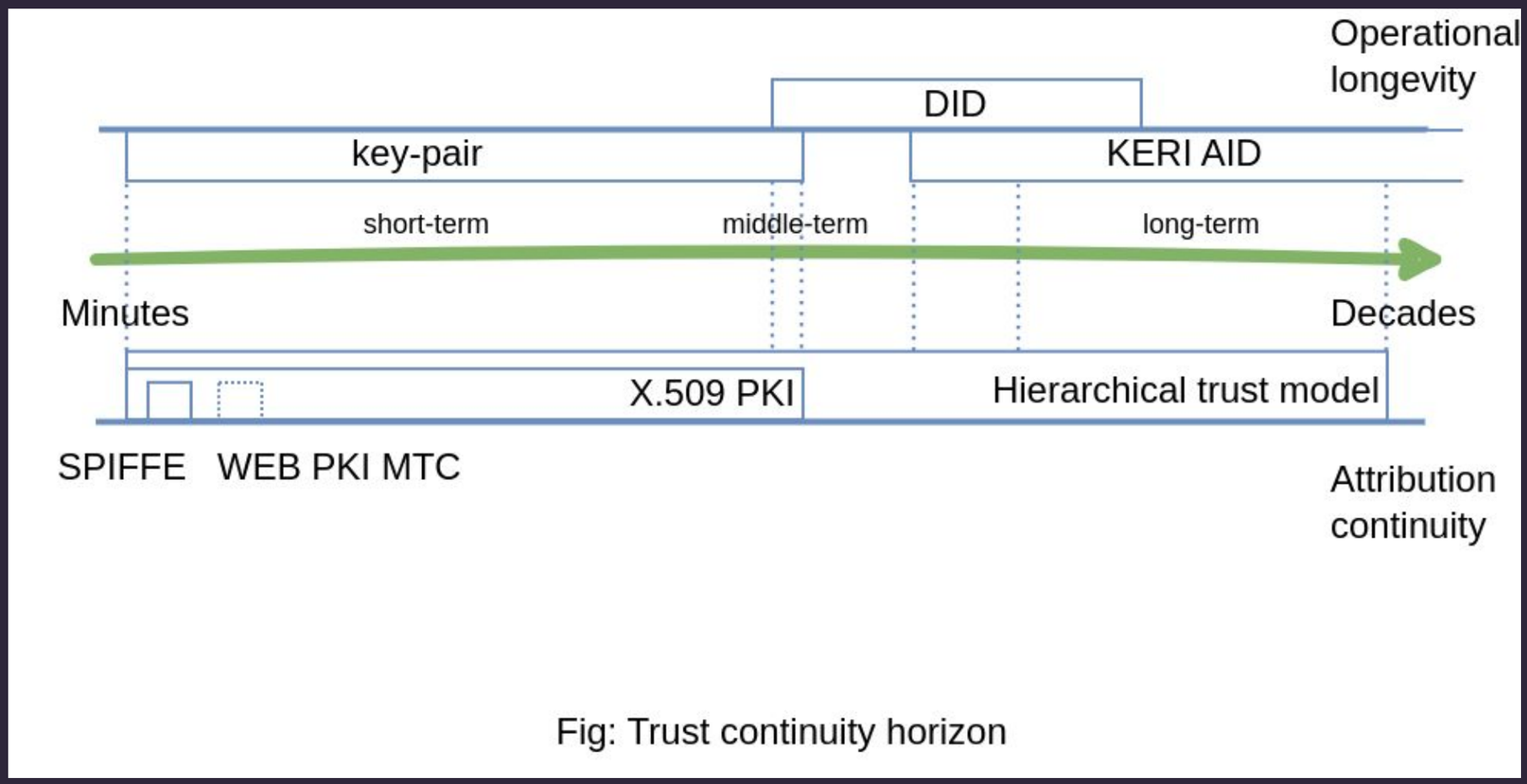
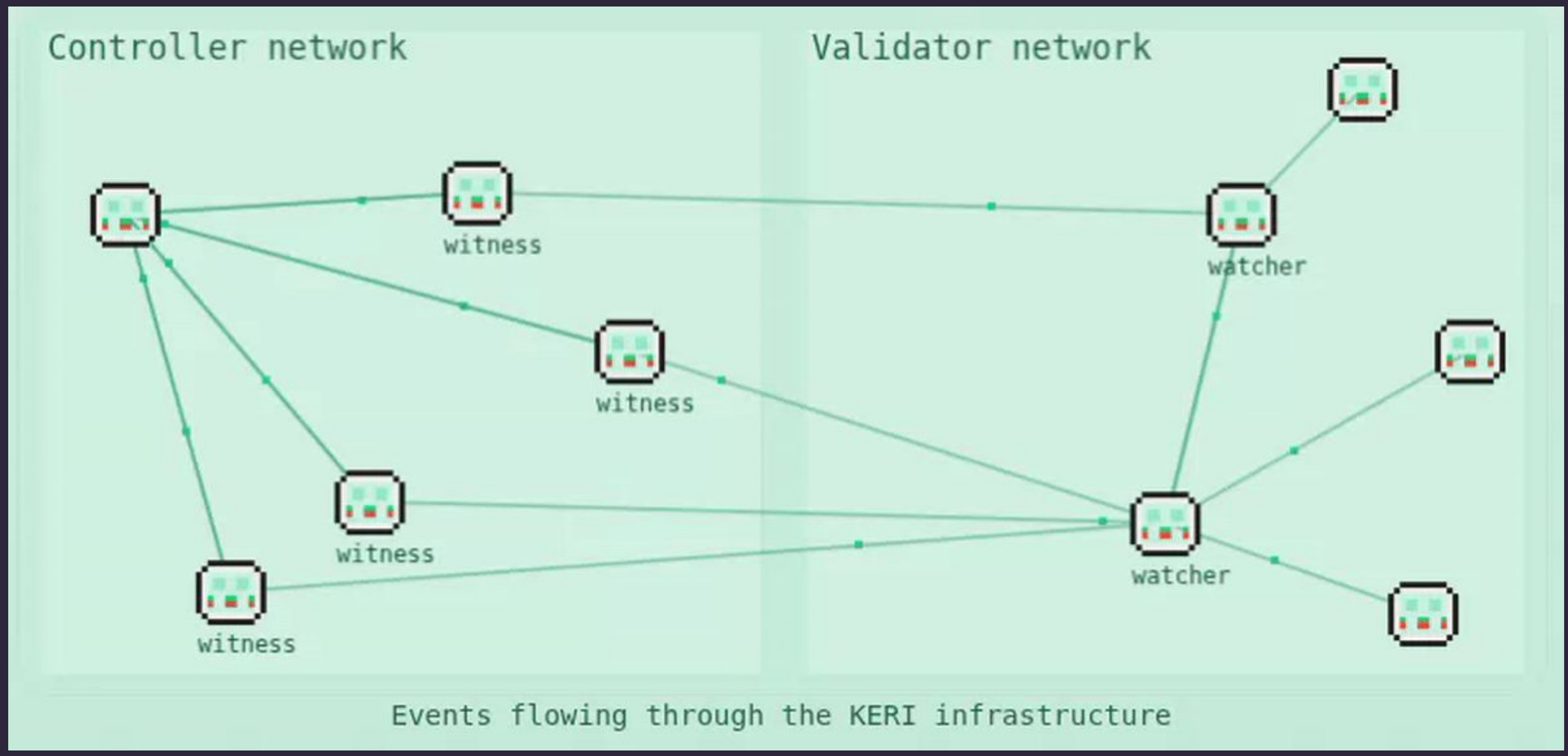


Fig: Trust continuity horizon

KERI – event distribution in the infrastructure

- KERI AID characteristics are enabled by the KERI infrastructure (which is also the main cost). Example infrastructure:



https://kerisuite.dev/

Edge nodes are clients of the infrastructure, e.g. issuers (left side) and verifiers (right side)

KERI – concept map

- KERI – Key Event Receipt Infrastructure
- AID – Autonomous Identifier
- ACDC – Authentic Chained Data Container
- KEL – Key Event Log
- CESR – Composable Event Streaming Representation

- KERI: a distributed event-based system that does not require global consensus (cf. DLT)
- KERI AID: a cryptographic identifier, derived as a hash of the first event in the KEL
- KERI AIDs carry attribution (cf. static keys in PKI)
- KEL: an immutable and non-repudiable event log (non-repudiation via digital signatures)
- Infrastructure based on Witness and Watcher nodes (although Watchers may be debatable within a single system)
- KERI ACDC ≈ equivalent of an X.509 certificate
- CESR: protocol combining signed data + cryptographic material (e.g. signatures) + metadata

KERI – DKMS implementation

- Implemented in Rust (infrastructure + client)
- License: EUPL
- Source code:
  - https://github.com/THCLab/keriox
  - https://github.com/THCLab/cesrox
  - https://github.com/THCLab/dkms-bin
  - https://github.com/THCLab/acdc-rust

Further reading

- KERI Whitepaper
- Towards KERI AIDs in Security Protocol Design
- Trust That Expires: Mapping the Longevity Horizon