W3C CCG
Cryptographic Event Logs

Agenda

1. Why Blockchains are Difficult
2. A simpler way: Cryptographic Event Logs
3. How do CELs work?
4. What Other Projects are CEL-like?
5. did:webvh and did:scid
6. CELs and "simply decentralized" DIDs
7. CELs and Social Networks
8. Next Steps

Why Blockchains are Difficult

- To order a set of events, either:
  - Burn "a lot" of energy
  - Stake "a lot" of money
  - Manage "a lot" of governance
- Blockchains are hard because it takes a lot of work to agree on "the state of the system".

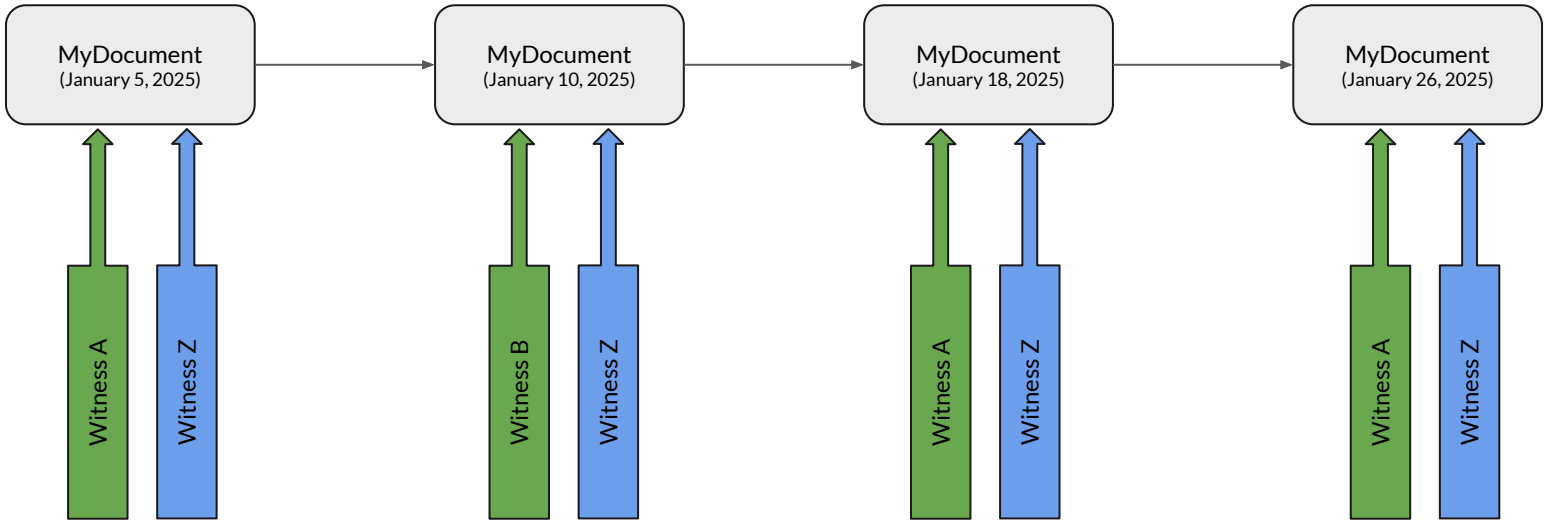
But, what if we don't have to agree on "the system"?

The Web succeeded because we allowed broken links.

What if we focused on describing ONE resource too?

What if we allowed a broken link on the blockchain?

Cryptographic Event Logs (CELs)



Cryptographic Event Logs (one interpretation)

EXAMPLE 1: An event log containing multiple events

```
{
  "log": [{
    "event": {...}, // the first event in the event log
    "proof": [...]  // a list of witness proofs securing the event
    }, {
    "event": {...}, // the second event
    "proof": [...]  // a list of witness proofs securing the second event
    }, {
    "event": {...}, // the third event
    "proof": [...]  // a list of witness proofs securing the third event
  }]
}
```

Cryptographic Event Logs (CELs)

EXAMPLE 4: An initial event log entry containing data embedded directly in the event.

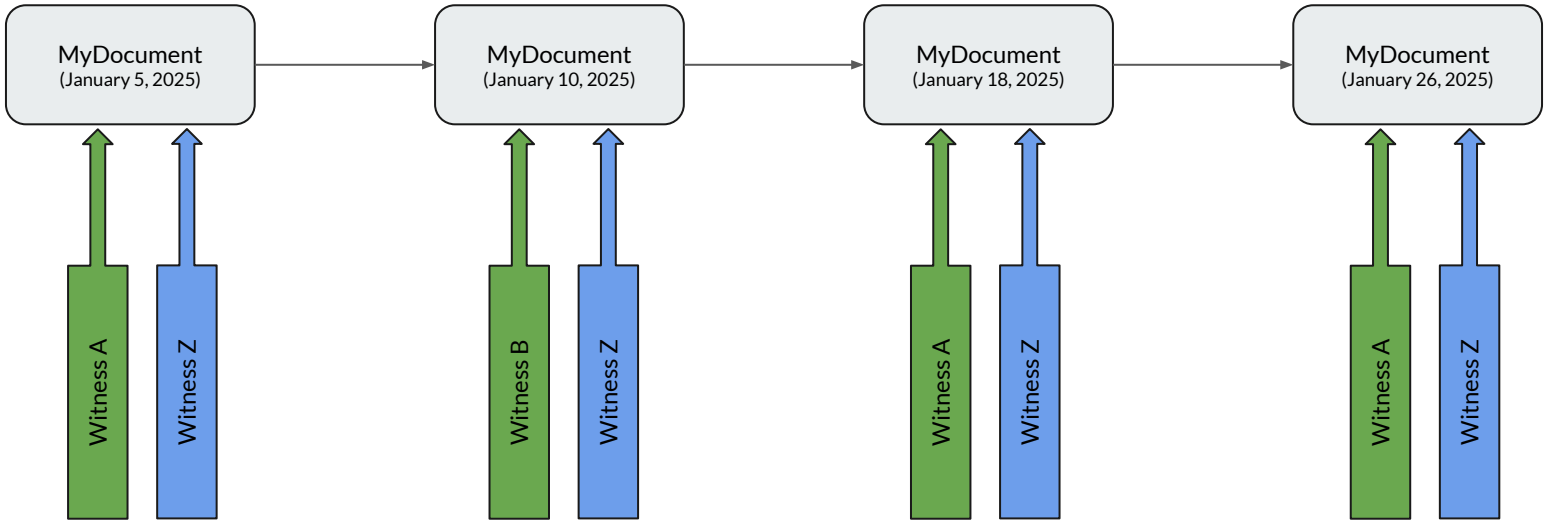
```
{
  "log": [{
    "event": {


      "operation": {
        // the type of operation observed by the event
        "type": "create",
        // the data associated with the event (in JSON format)
        "data": {
          "name": "Hello World!",
          // one or more proofs that secure the integrity of the event above
          "proof": {
            "type": "DataIntegrityProof",
            "cryptosuite": "ecdsa-jcs-2019",
            "created": "2024-11-29T13:56:28Z",
            // the cryptographic authority for the data is application specific
            // in this example, it is established by the proof in the create operation
            "verificationMethod": "https://website.example/crypto#key-1",
            "proofPurpose": "assertionMethod",
            "proofValue": "zq6PrUMCtqY5obCSsrQxuFJd...wVWgHYrXxoV93gBHqGDBtQLPFxpZxz"
          }
        }
      }
    }
  }]
}
```

Cryptographic Event Logs (CELs)

- Like Blockchains:
  - An ordered set of events
- Unlike Blockchains:
  - Focused on a single "resource"
  - Changes are blindly witnessed
  - "Truth" is the perspective of the Verifier

Cryptographic Event Logs (CELs)



Other CEL-like Projects

- This isn't a new concept
  - Blockchains
  - Certificate Transparency
  - did:webvh / did:scid

did:webvh

- Ongoing publishing of all DID Document (DIDDoc) versions for a DID instead of, or alongside a `did:web` DID/DIDDoc.
- Uses the same DID-to-HTTPS transformation as `did:web`.
- Provides resolvers the full history of the DID using a verifiable chain of updates to the DIDDoc from genesis to deactivation.
- A self-certifying identifier (SCID) for the DID that is globally unique and derived from the initial DIDDoc that enables DID portability, such as moving the DIDs web location (and so the DID string itself) while retaining the DID's history.
- DIDDoc updates include a proof signed by the DID Controller(s) *authorized* to update the DID.
- An optional mechanism for publishing "pre-rotation" keys to prevent loss of control of the DID in cases where an active private key is compromised.
- An optional mechanism for having collaborating "witnesses" that approve updates to the DID by the DID Controller before publication.
- A DID URL path `<did>/whois` that defaults to automatically returning (if published by the DID controller) a Verifiable Presentation containing Verifiable Credentials with the DID as the `credentialSubject`, signed by the DID.

Next Steps for did:webvh

- Releasing v1.0 soon
- Discuss alignment (or not) with CELs
- Potential for: CEL + web + SCID

did:scid:web?

WEB

- Ongoing publishing of all DID Document (DIDDoc) versions for a DID instead of, or alongside a `did:web` DID/DIDDoc.
- Uses the same DID-to-HTTPS transformation as `did:web`.
- A DID URL path `<did>/whois` that defaults to automatically returning (if published by the DID controller) a Verifiable Presentation containing Verifiable Credentials with the DID as the `credentialSubject`, signed by the DID.

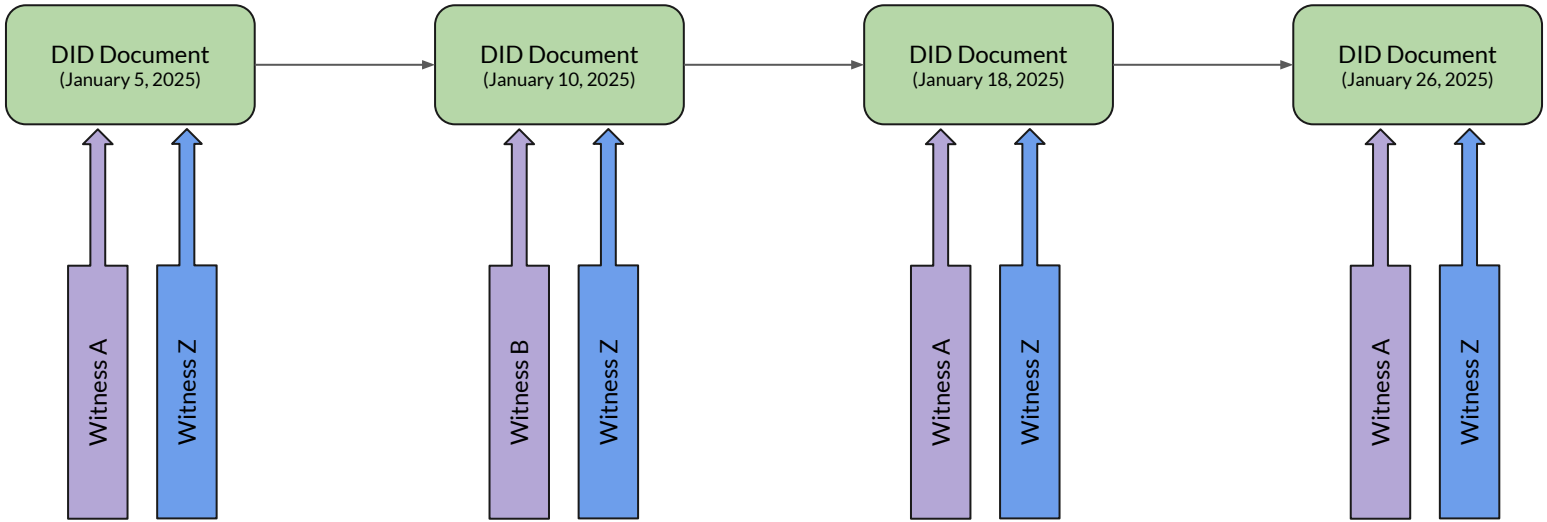
CEL

- Provides resolvers the full history of the DID using a verifiable chain of updates to the DIDDoc from genesis to deactivation.
- DIDDoc updates include a proof signed by the DID Controller(s) *authorized* to update the DID.
- An optional mechanism for having collaborating "witnesses" that approve updates to the DID by the DID Controller before publication.

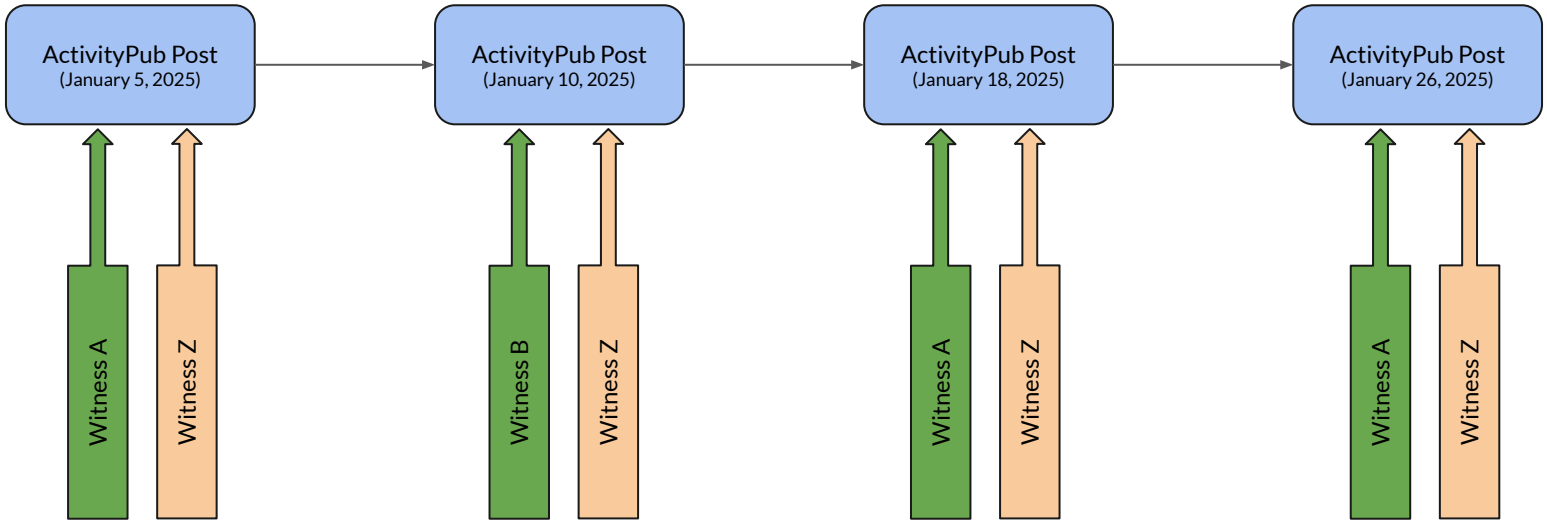
SCID

- A self-certifying identifier (SCID) for the DID that is globally unique and derived from the initial DIDDoc that enables DID portability, such as moving the DIDs web location (and so the DID string itself) while retaining the DID's history.
- An optional mechanism for publishing "pre-rotation" keys to prevent loss of control of the DID in cases where an active private key is compromised.

"Simply and Truly Decentralized" DIDs



CELs and Social Networks



Next Steps

- Participate in did:webvh group?
- Participate in DIF DID Methods Working Group?
- Join the W3C DID Methods WG when it starts?
- Create a new group for a CEL-based DID Method?
- Other ideas for collaboration?