SNAP on Solid

A Solid Base Layer for Network Ledger Technology

Network Ledger Technology

The Next Big Thing (™) in FinTech

Network Ledger Technology

- A store of value.
- A unit of value.
- A means of exchange.

Network Ledger Technology

- ~~A store of value.~~
- A unit of value.
- A means of exchange.

Network Ledger Technology

- ~~A store of value.~~
- ~~A unit of value.~~
- A means of exchange.

Network Ledger Technology

- ~~A store of value.~~
- ~~A unit of value.~~
- ~~A means of exchange.~~

So what is NLT? (1/2)

- An *interpersonal* store of value.
- An *interpersonal* unit of value.
- An *interpersonal* means of exchange.
- *Networked through multi-hop.*

So what is NLT? (2/2)

- Cryptographically triggered “I owe you” notes
- Ripple Classic Protocol (hypothetical, predates BitCoin)
- Interpersonal credit with multi-hop
- Bitcoin Lightning, Interledger, LedgerLoops

Synchronized Network Accounting

- Creditor device, debtor device, comms, nothing else (!)
- Two people can do this, in isolation
- Messages are repeated until they are acknowledged
- Supports transactions, hashlock conditions, timeouts

SNAP transaction life cycle

- > Proposing
- < Proposed
- < Accepting | < Rejecting
- > Accepted | > Rejected

Proposing

- From (string)
- To (string)
- Unit (string)
- Transaction ID (integer, incremental)
- Amount (integer)
- Optional: Condition (raw bytes, SHA256 of preimage)
- Optional: Expiry (UTC-SLS)

Proposed, Accepted, Rejected

- From (string)
- To (string)
- Unit (string)
- Transaction ID (links it to earlier Proposing message)

Accepting

- From (string)
- To (string)
- Unit (string)
- Transaction ID (links it to earlier Proposing message)
- Preimage (raw bytes, SHA256 should match condition from Proposing message, if present)

Rejecting

- From (string)
- To (string)
- Unit (string)
- Transaction ID (links it to earlier Proposing message)
- Reason (list of error codes still under construction)

No signatures (!)

Above comms but below multi-hop

Using Solid inbox technology

How Solid inboxes work

- anything-to-pod: HTTP POST (acl:Append on folder)
- pod-to-app (Solid: WebSockets-pubsub)
- app-to-user (Web Platform: Notifications API)
- bot-to-user (Web Platform: Push API)

```
@prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
@prefix as: <https://www.w3.org/TR/activitypub/#> .

<Alice> as:following <Addressbook#Friends> .
<Addressbook#Friends> a vcard:Group .
<Addressbook#Friends> vcard:hasMember <BobEntry> .
```

Solid Friends

```
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
<#owner> a acl:Authorization ;
  acl:agent <Alice> ;
  acl:accessTo <AddressBook> ;
  acl:mode acl:Read, acl:Write, acl:Control.
<#friends> a acl:Authorization ;
  acl:agentGroup <AddressBook#Friends> ;
  acl:accessTo <AddressBook> ;
  acl:mode acl:Read.
```

Solid Friends ACL

```
@prefix contacts: <https://ledgerloops.com/contacts/#> .
@prefix snap: <https://ledgerloops.com/snap/#> .

<BobEntry>
  contacts:webId <Bob> ;
  contacts:backlink <AliceEntryAtBob> ;
  contacts:nick "Bob" ;
  snap:ourInbox <BobInbox> ;
  snap:ourOutbox <BobOutbox> .
<AliceEntryAtBob>
  snap:ourInbox <AliceInboxAtBob> .
```

Solid Contacts

```
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
<#owner> a acl:Authorization ;
  acl:agent <Alice> ;
  acl:accessTo <BobInbox> ;
  acl:default <BobInbox> ;
  acl:mode acl:Read, acl:Write, acl:Control.
<#peer> a acl:Authorization ;
  acl:agent <Bob> ;
  acl:accessTo <BobInbox> ;
  acl:mode acl:Append.
```

Solid P2P Inbox ACL

NLT-KIT

- Deployed to Heroku with Redis
- NLT-KIT servers are full app servers
- Federate through WebId
- Each account on each NLT-KIT server is also a pod
- But users don’t know that :)

Solid App Architecture Layers

- Document-centric RDF manipulation (TripleDoc)
- `fetchDocument` promises are cached
- Create containers / ACLs / docs on-the-fly if missing
- Planned: WebSockets-pub-sub (with new `auth`)

SnapChecker

Double-Spend Detection

Current, Payable, Receivable

Next Steps

- User or automatic responding to a SNAP-Proposed
- Edit trust level, comms atomicity of that
- And of course …

Multi-hop! :)

Network Ledger Technology is the future ...

You heard it here first! :)

Thank you

https://github.com/ledgerloops/nlt-kit