DIDComm v2 Primer

CCG — April 2022 — Daniel Hardman — https://bit.ly/3qTuG9m

Agenda

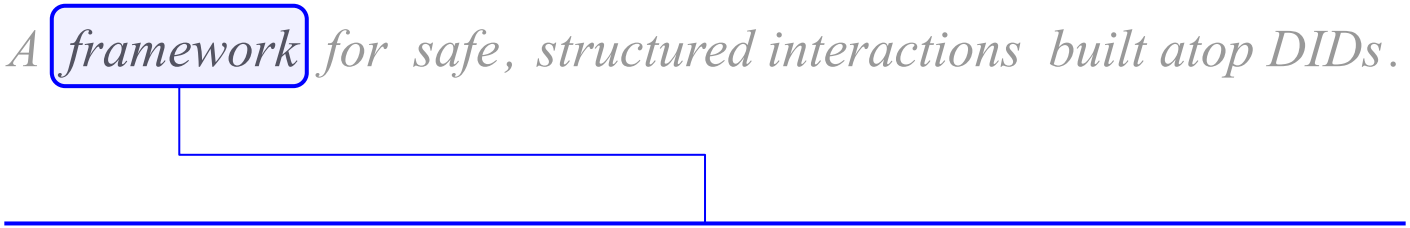
1. What it is
2. What it does and doesn't do
3. Relationship to WACI-DIDComm, CHAPI, OIDC-SIOP
4. Versions
5. Implementations

Definition

A framework for safe, structured interactions built atop DIDs.

slide for Swagger: framework for secure APIs built atop REST+TLS.

Framework

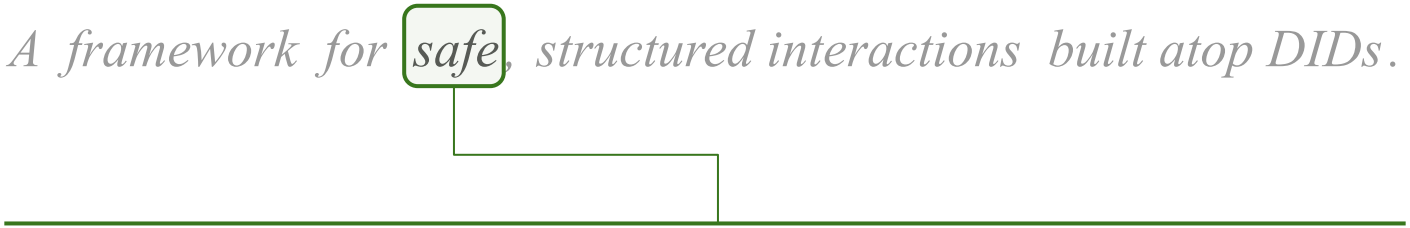


A structure for supporting or enclosing something else, especially a skeletal support used as the basis for something being constructed.

— *American Heritage Dictionary*

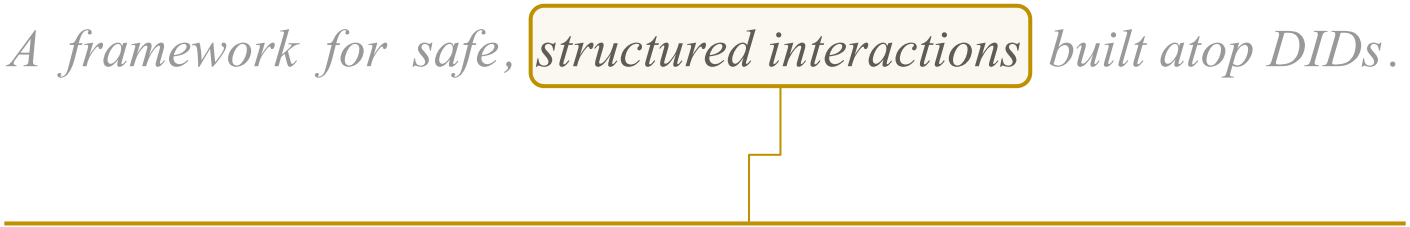
Not an API.
Not an algorithm.
Not a library.

Safe



Secure
Private
Decentralized → resilient, censorship-resistant, owned by no-one…

Structured Interactions



Not just rich chat or email (it's structured).
Not just client-server (interactions is broader than that).
Not just RPC (interactions can do more than invoking).
Not request-response (interactions can involve >2 parties at a time).

Built Atop DIDs

Fundamental properties derive from DIDs; DIDs aren't just source of keys.

Preserves whatever positive qualities DIDs claim, not just authN:

- Self-service
- Can't be siloed

DIDComm Messaging tells you how to…

- Use your DID to sign and encrypt messages for other DIDs (pairwise or n-wise), each with multiple devices having different keys
- Declare and use a DID endpoint with standard semantics
- Route a message through untrusted intermediaries, with high privacy
- Verify the sender of a message
- Use standard message headers, and declare custom ones
- Declare/handle the schema of a message
- Attach data to messages by value or by reference
- Sequence messages into a coherent thread, even with unreliable delivery
- Detect and report errors
- Discover features of other parties
- Build protocols out of these primitives

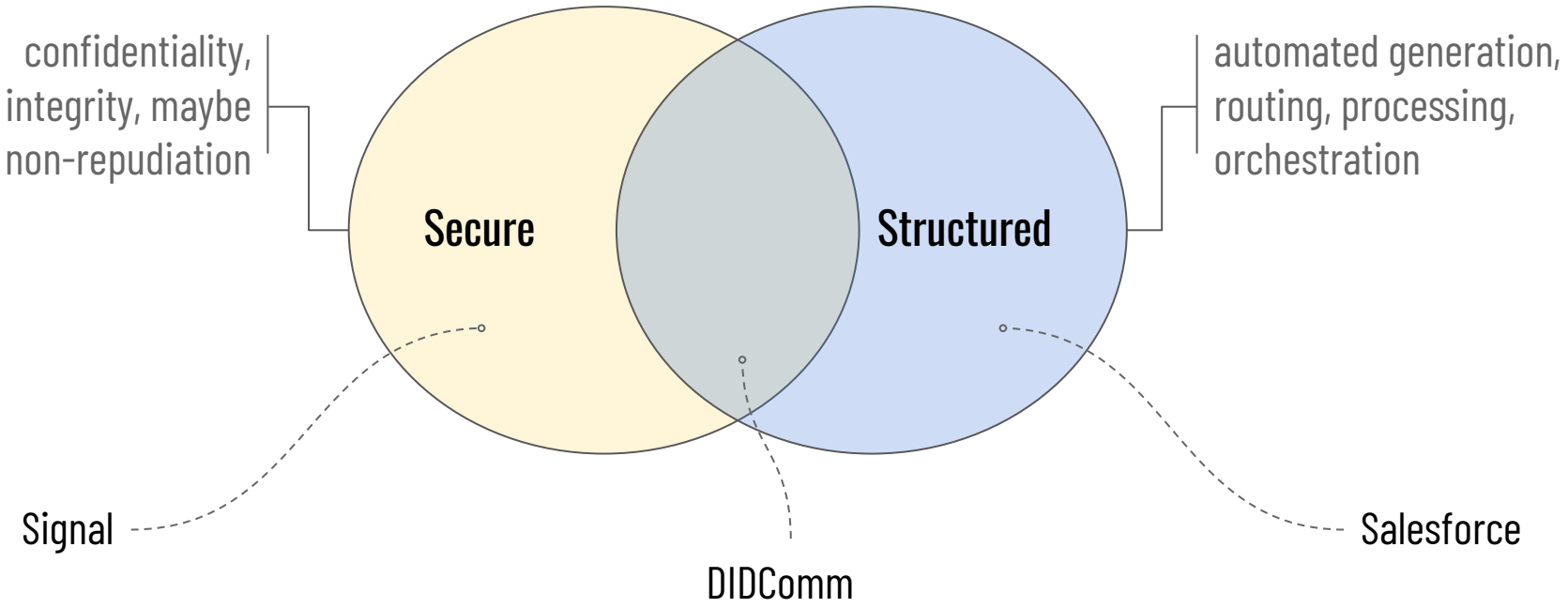
DIDComm Messaging doesn't tell you how to…

- Create or use wallets
- Work with credentials
- Associate a DID with a human (or other) identity (authN more than a DID)
- Bind a remote party to a biometric
- Move messages over a transport
- Choose DID methods or key types or blockchains
- Properly maintain relationships
- Decide whether a particular combination of behaviors will satisfy your level-of-assurance goals
- Synchronize state across multiple agents

Potential Application-Level Protocols in DIDComm

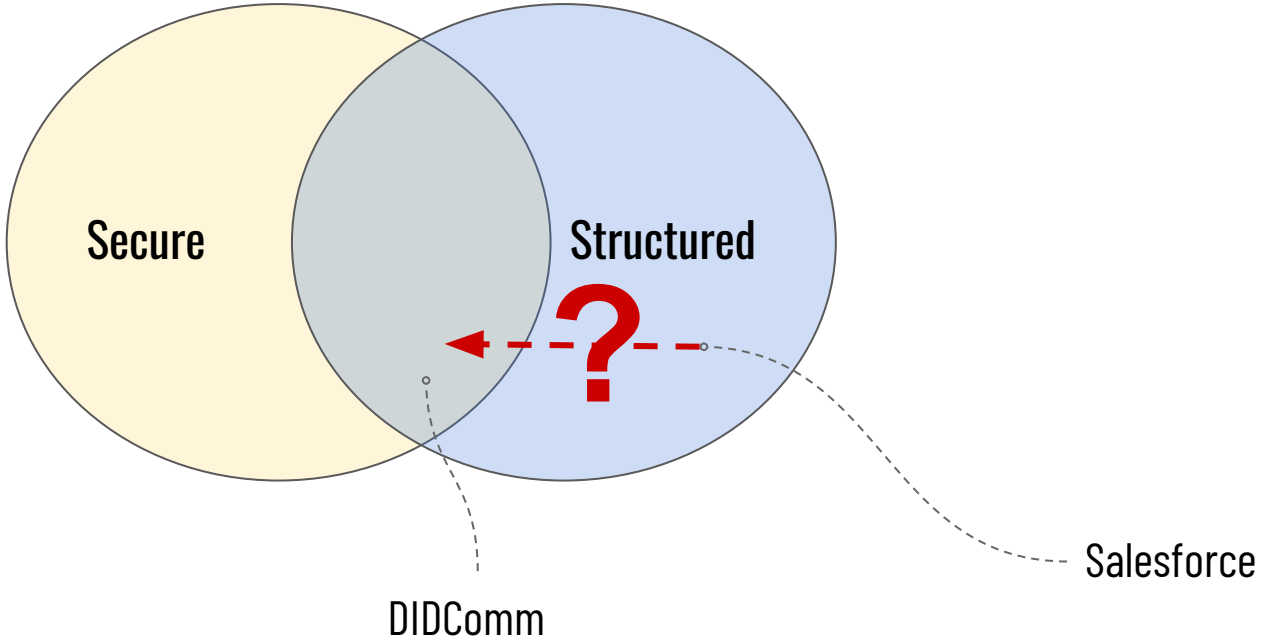
- IssueCredential
- ProveWithCredential
- Connect, Introduce, SayGoodbye
- Pay, ListForSale
- TakeTest
- ApplyForLoan, ApplyForJob
- ScheduleEvent
- Vote
- Recommend
- FlipCoin
- CheckBiometric
- HailTaxi
- BookHotel
- PlanVacation
- RichChat
- NegotiatePriceAndPaymentMethod
- ReportCrime
- RequestSupport
- FileInsuranceClaim
- PutItemInEscrow
- AskAlexa
- PostTweet

Secure vs Structured



credit: Drummond Reed

Surely it's secure?

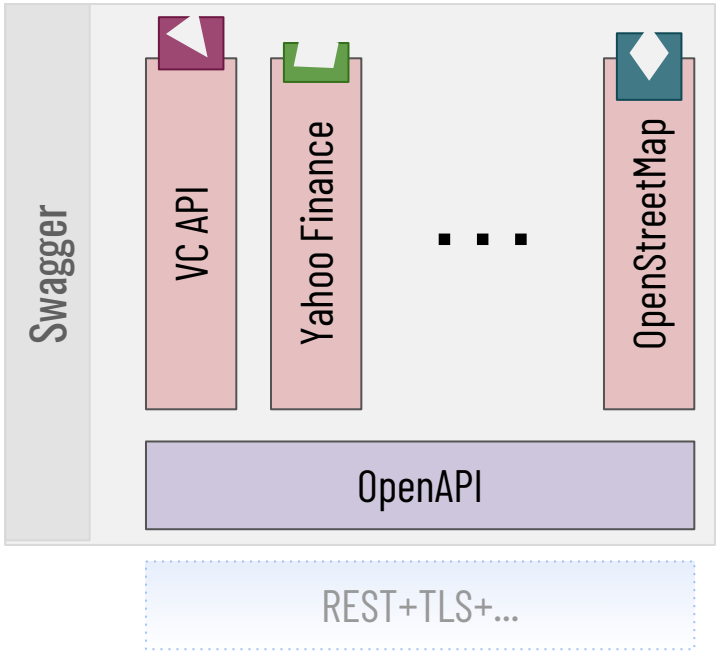


Analog?

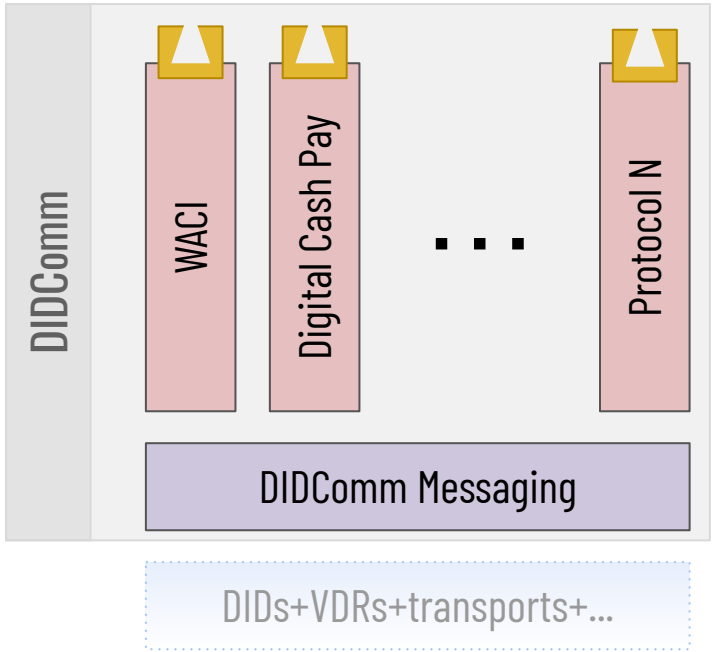
A framework for safe, structured interactions built atop DIDs.

A framework for APIs built atop REST+TLS.

No analogy is perfect, but…



client-server, web only, request-response, pairwise, each authN and URL namespacing is unique, siloed — wonderful tools and community

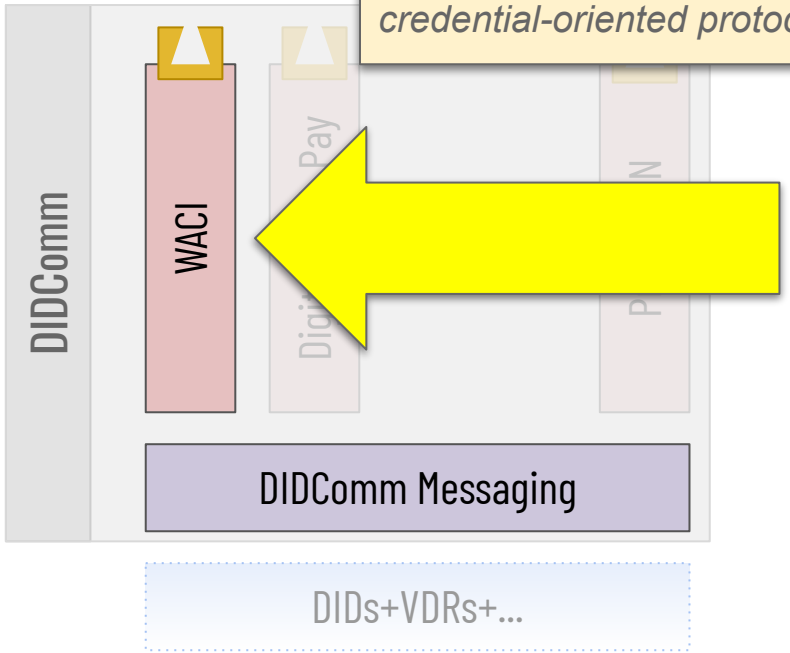


peer-to-peer or client-server, any transport, any interaction pattern, pairwise or n-wise, consistent authN and namespacing, unsiloed — immature tools and community

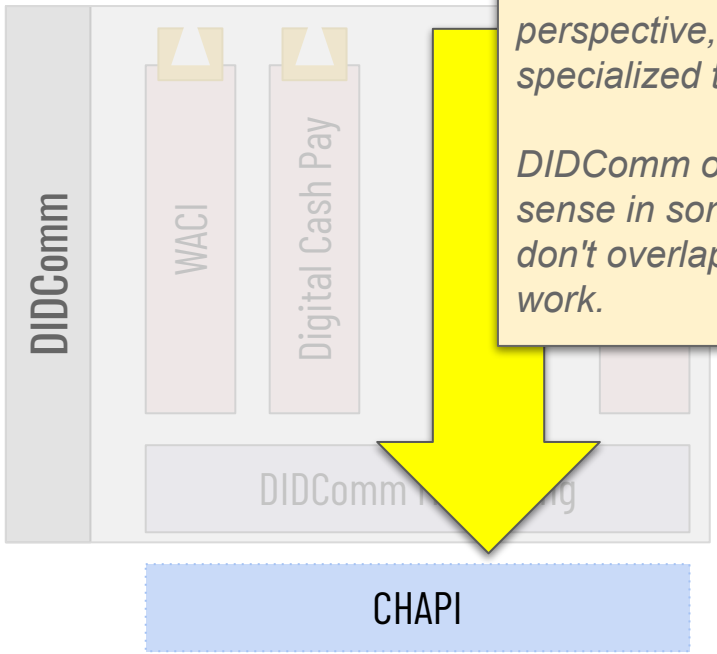
DIDComm and WACI

WACI says how to conduct wallet interactions like requesting and providing credential proof, on top of DIDComm Messaging.

WACI will eventually supersede some earlier credential-oriented protocols built on DIDComm.



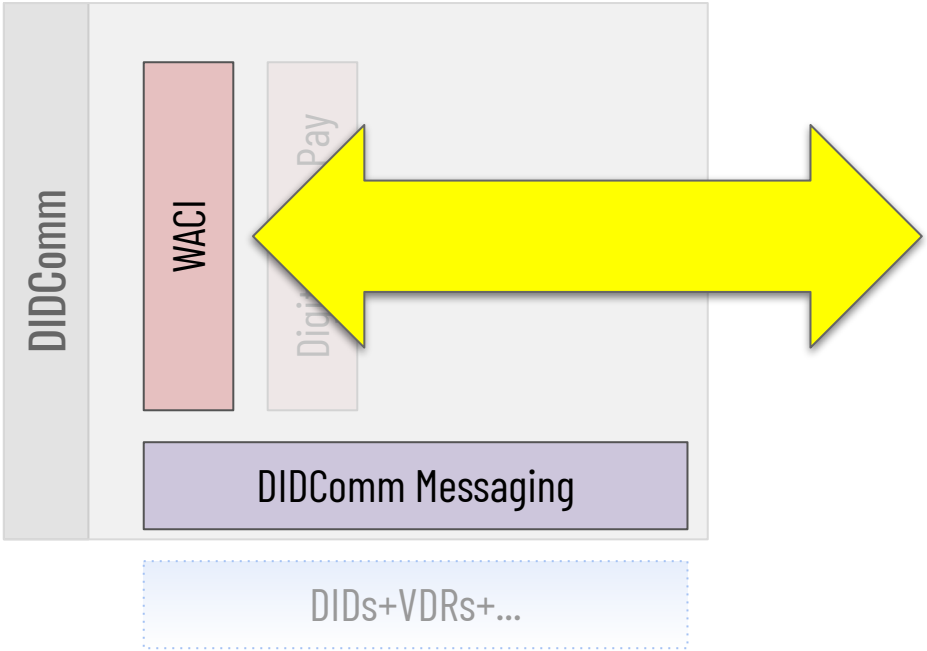
DIDComm and CHAPI

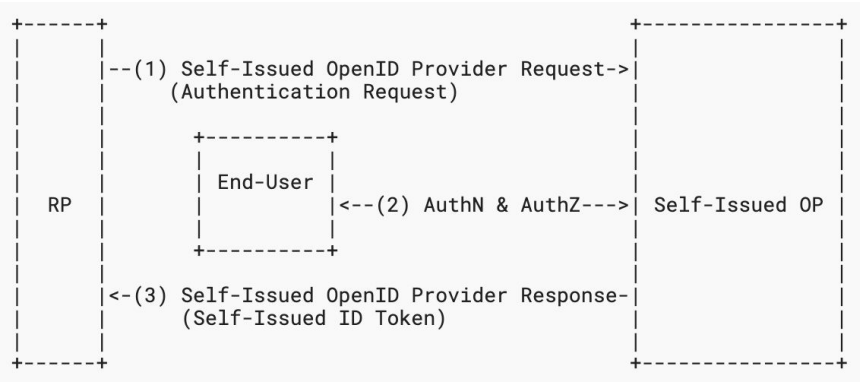


CHAPI says how to move wallet data between apps/websites hosted in the same web browser. From DIDComm's perspective, this makes CHAPI a specialized transport layer.

DIDComm on top of CHAPI may make sense in some cases, but their sweet spots don't overlap very strongly, so it's future work.

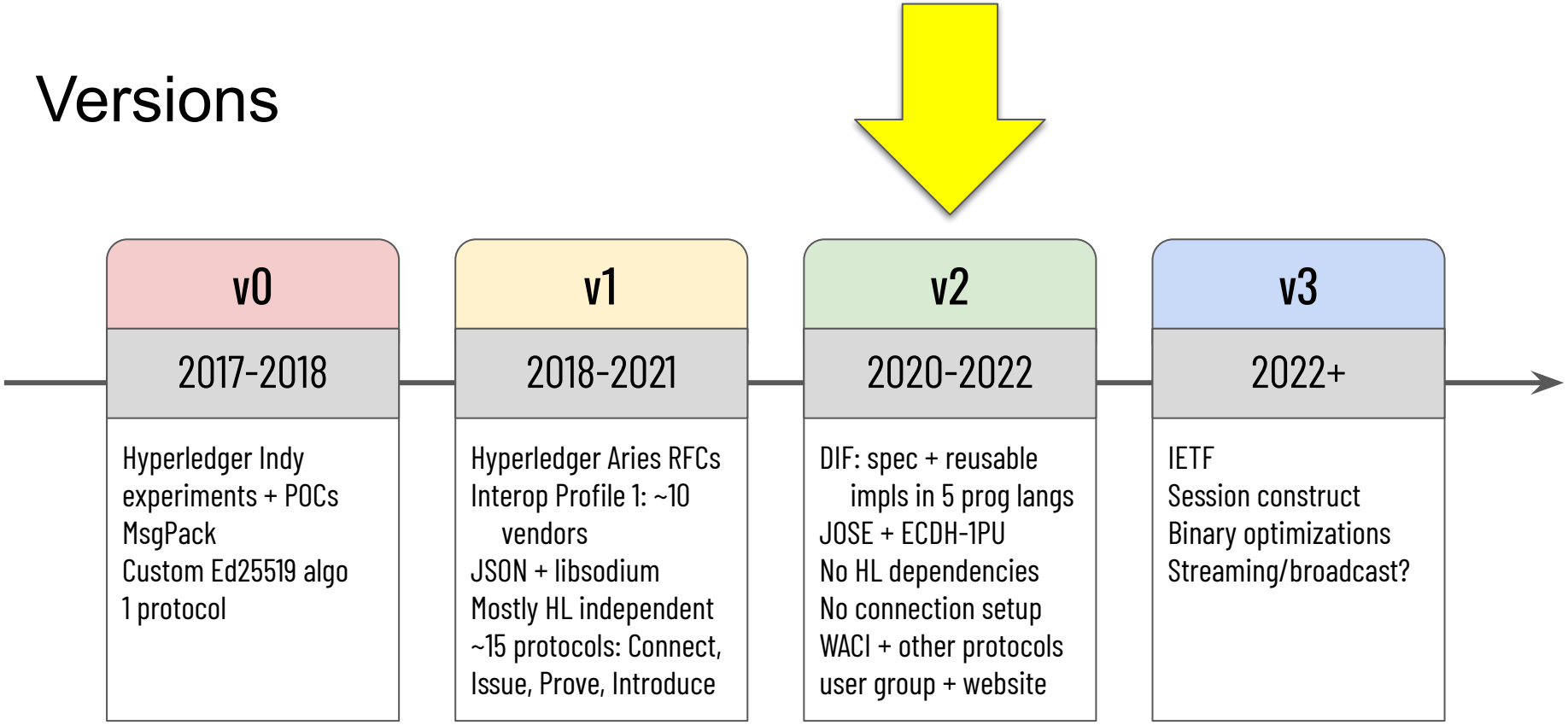
DIDComm and OIDC-SIOP

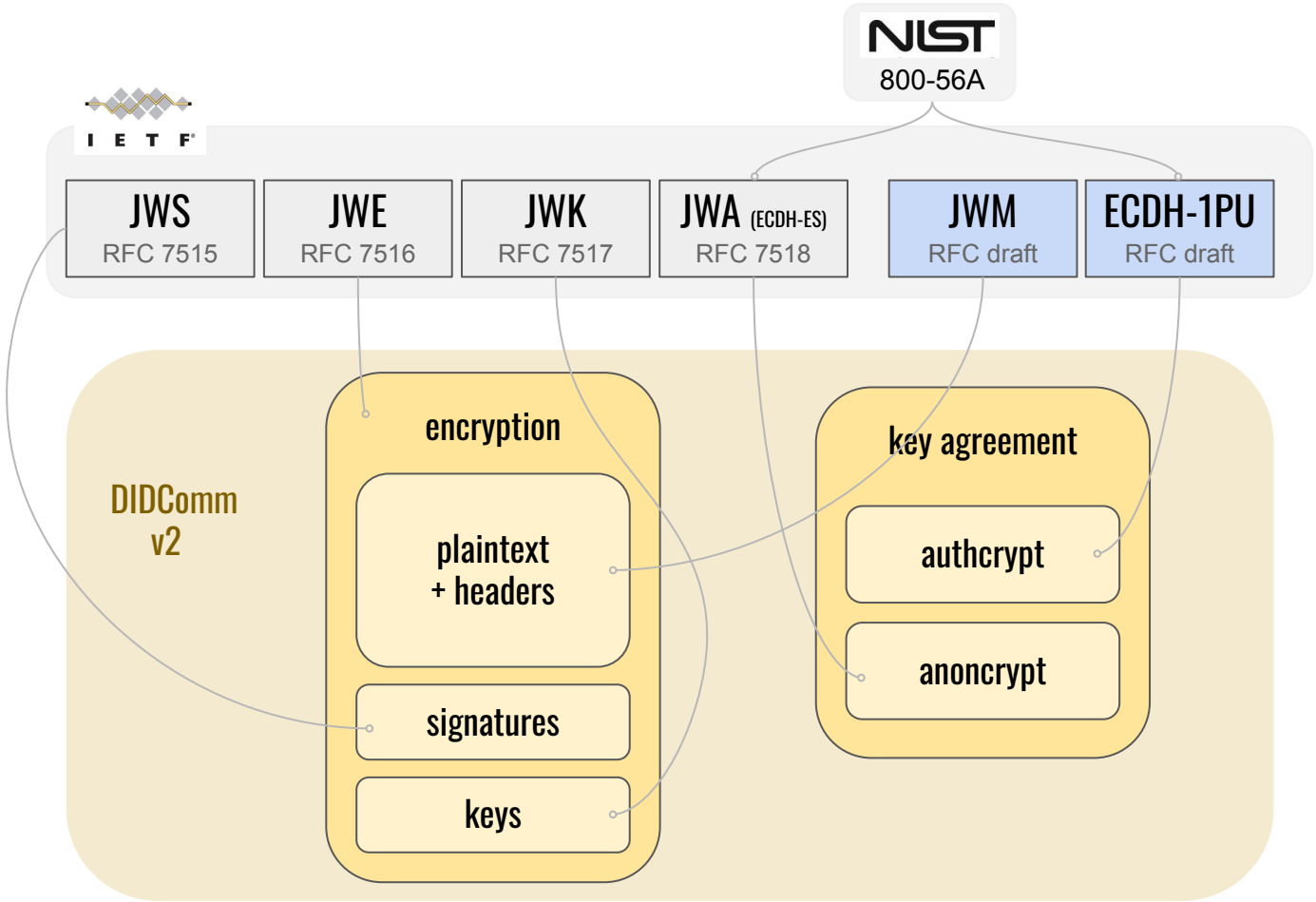




DIDComm Messaging doesn't specify authN for people, but WACI does. WACI DIDComm is a rough feature analog of OIDC-SIOP.

Versions


v0
2017-2018
Hyperledger Indy
experiments + POCs
MsgPack
Custom Ed25519 algo
1 protocol
v1
2018-2021
Hyperledger Aries RFCs
Interop Profile 1: ~10
vendors
JSON + libsodium
Mostly HL independent
~15 protocols: Connect,
Issue, Prove, Introduce
v2
2020-2022
DIF: spec + reusable
impls in 5 prog langs
JOSE + ECDH-1PU
No HL dependencies
No connection setup
WACI + other protocols
user group + website
v3
2022+
IETF
Session construct
Binary optimizations
Streaming/broadcast?


NIST
800-56A
IETF
JWS
RFC 7515
JWE
RFC 7516
JWK
RFC 7517
JWA (ECDH-ES)
RFC 7518
JWM
RFC draft
ECDH-1PU
RFC draft
DIDComm
v2
encryption
plaintext
+ headers
signatures
keys
key agreement
authcrypt
anoncrypt

Implementations (v2 only; all open source)

Javascript / Typescript

- https://github.com/uport-project/veramo
- https://github.com/aviarytech/didcomm

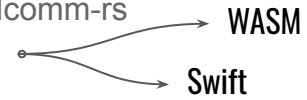
```
npm install didcomm
```

Go

- https://github.com/hyperledger/aries-framework-go/tree/main/pkg/didcomm/packer

Rust

- https://github.com/decentralized-identity/didcomm-rs
- https://github.com/sicpa-dlab/didcomm-rust
- https://github.com/idp2p/idp2p




```
[dependencies]
didcomm = "0.3"
```

Python

- https://github.com/sicpa-dlab/didcomm-python
- hyperledger/aries-cloudagent-python#1331

```
pip install didcomm
```

Java

- https://github.com/sicpa-dlab/didcomm-jvm

```
gradle: implementation 'org.didcommx:didcomm:0.3.0'
```