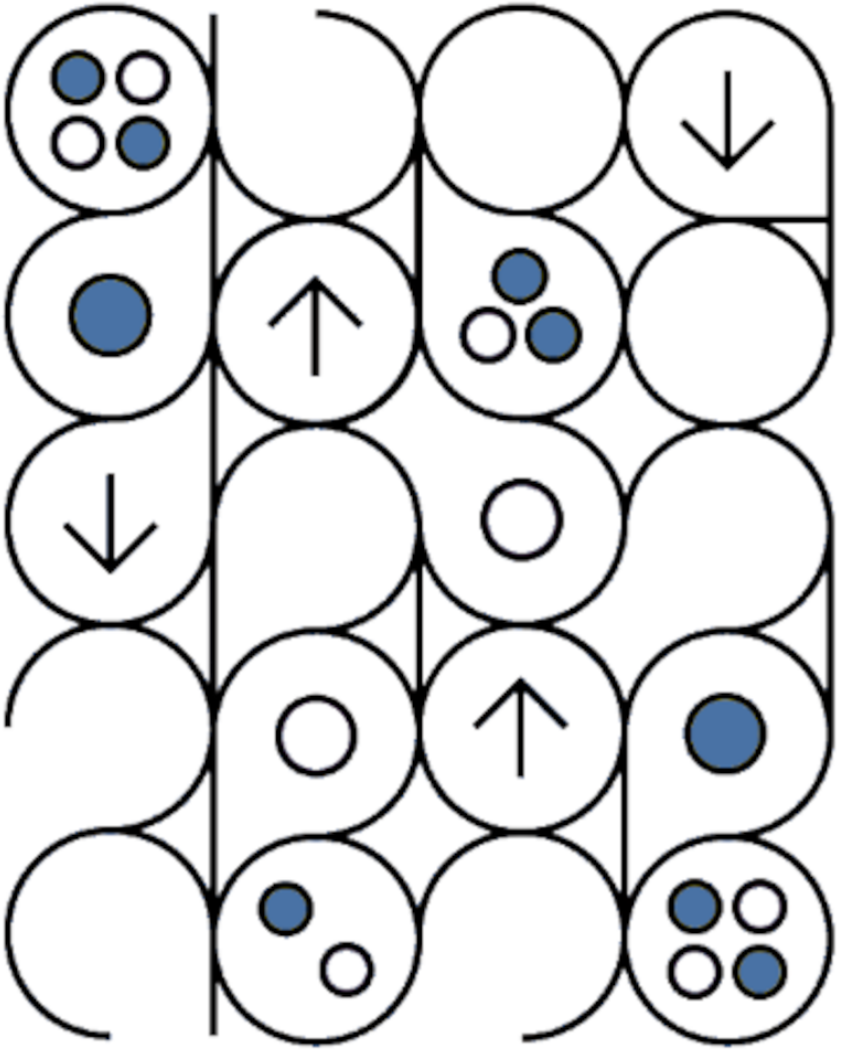


NextGraph
.org

Niko Bonnieure

European Commission
DG CNECT

NGI ASSURE

NextGraph
.org

NGI ZERO COMMONS

nlnet
FOUNDATION

A story about data

A story about data

Where is my data today ?

A story about data

Where is my data today ?

 **Big Tech**

👎 Big Tech

Myriad of accounts

👎 Big Tech

Myriad of accounts

No access to raw data

👎 Big Tech

Myriad of accounts

No access to raw data

Plaintext in their cloud

Open Source Alternatives

Open Source Alternatives

👎 Myriad of accounts

Open Source Alternatives

 Myriad of accounts

 **Incompatible APIs**

Open Source Alternatives

👎 Myriad of accounts

👎 Incompatible APIs

👎 **Plaintext in their server**

I want my data

I want my data
I want all my data

I want my data

I want all my data

I want all my data in one place

All my data in one place **control**

All my data in one place

control

ownership

All my data in one place

control

ownership

availability

control
ownership
availability
security & privacy

control
ownership
availability
security & privacy
share & collaborate*

I want a central place for all my data

With E2EE

Server = store & forward

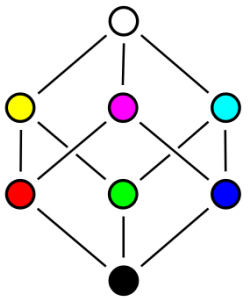
I want a central place for all my data

With E2EE

Server = store & forward

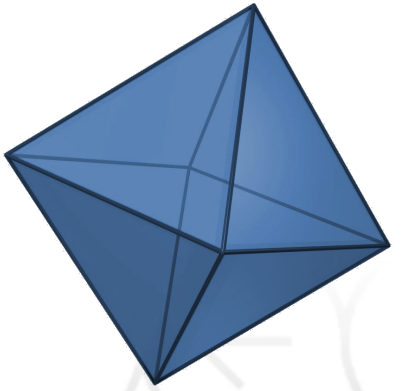
Local First & CRDT

Conflict-free Replicated Data Types



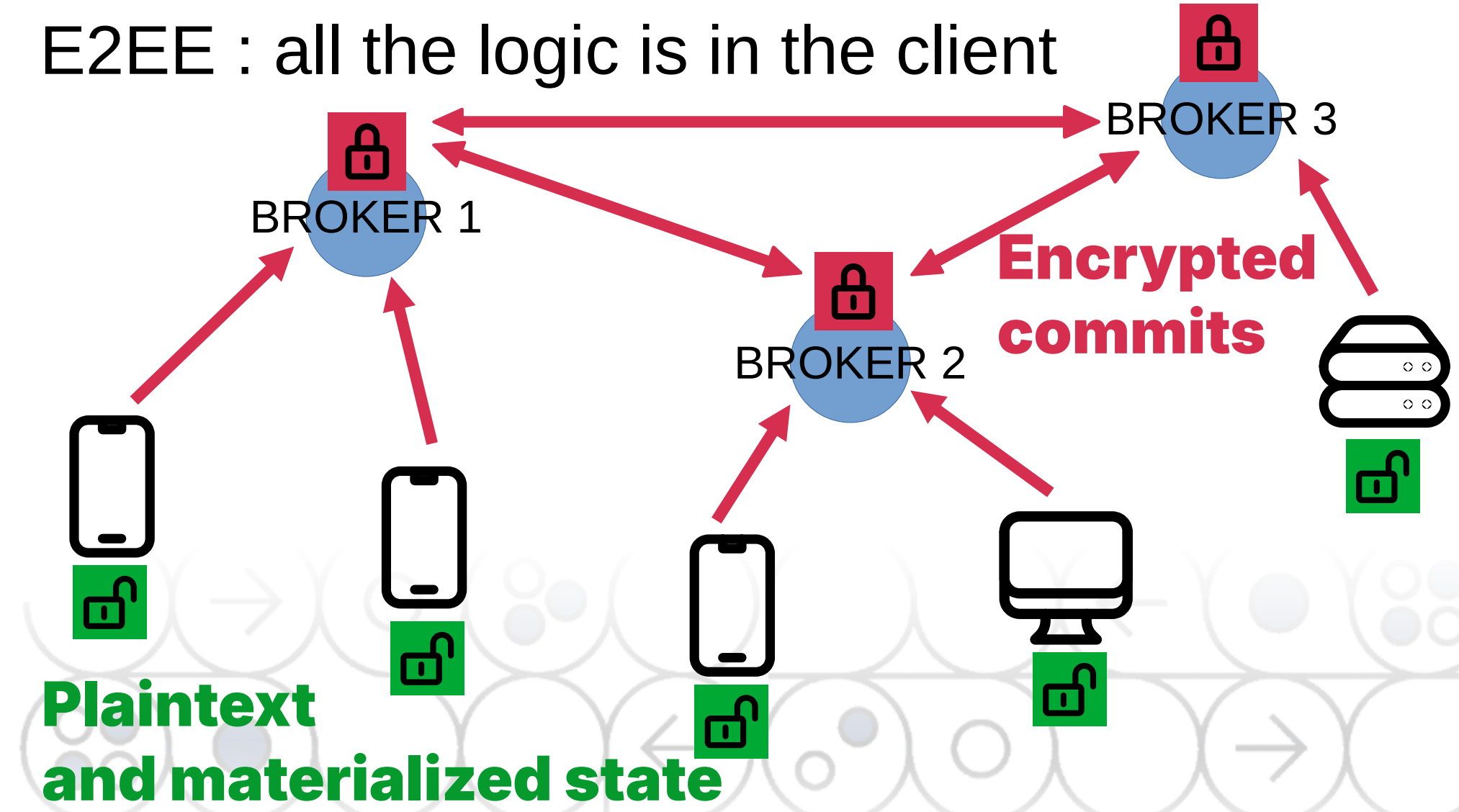
CRDT.tech

lofi.so



Open Local First .org

E2EE : all the logic is in the client



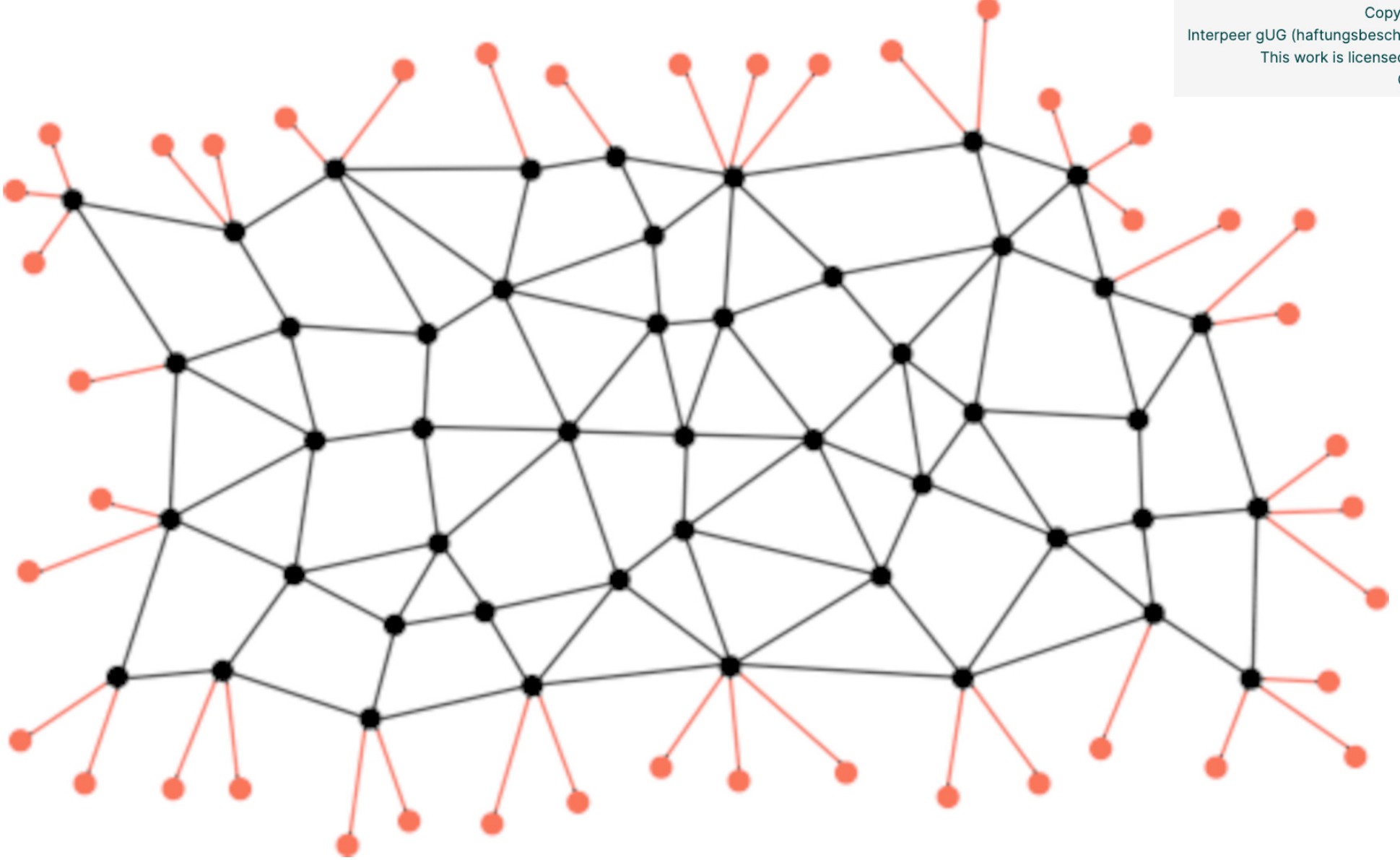
I want a central place for all my data

How many servers?

I want a central place for all my data

How many servers?

A federation!



I want a central place for all my data

How many servers?

A federation!

Synced in a pub/sub

I want a central place for all my data

Replication on all devices

I want a central place for all my data

Replication on all devices

And on brokers

I want a central place for all my data

Replication on all devices

And on brokers

Full decentralization

I want a completely decentralized central place for all my data

And that's

NG proto Specialized for E2EE and CRDT

NG proto

Specialized for E2EE and CRDT

Can sync any kind of CRDT

NG proto

Access control with Cryptographic capabilities

NG proto

DID decentralized identifiers

did:ng:

NG proto

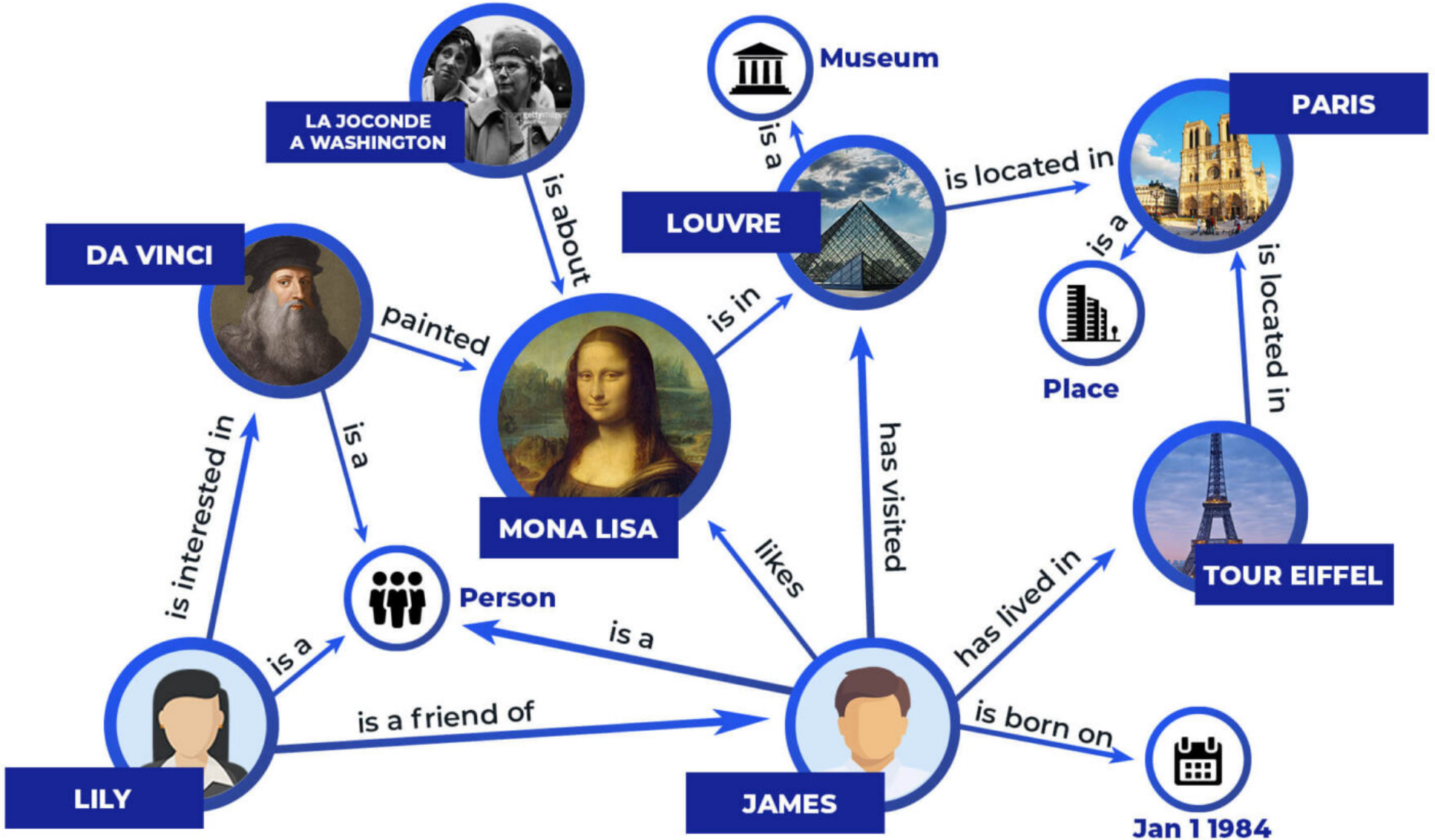
upload/download binary files, with streaming, content addressing, and chunking

NG proto
Supported CRDTs

RDF
aka Linked Data
W3C standard
triples
SPARQL

RDF

Interoperability
Malleable software


LA JOCONDE A WASHINGTON
is about
Museum
is a
LOUVRE
is located in
PARIS
is a
Place
is located in
DA VINCI
painted
is in
MONA LISA
is interested in
is a
Person
is a
likes
has visited
has lived in
TOUR EIFFEL
LILY
is a
is a friend of
JAMES
is born on
Jan 1 1984

RDF

Link and reference other documents.

It is a database

RDF

Automatic joins
no foreign key
all data joinable

RDF

Global IDs for each record, using URI. did:ng:...

RDF

A global database!

Tim Berners-Lee used to call it the Giant Global Graph

NextGraph engine :

Sync protocol

CRDT agnostic

Graph database

Encryption at rest

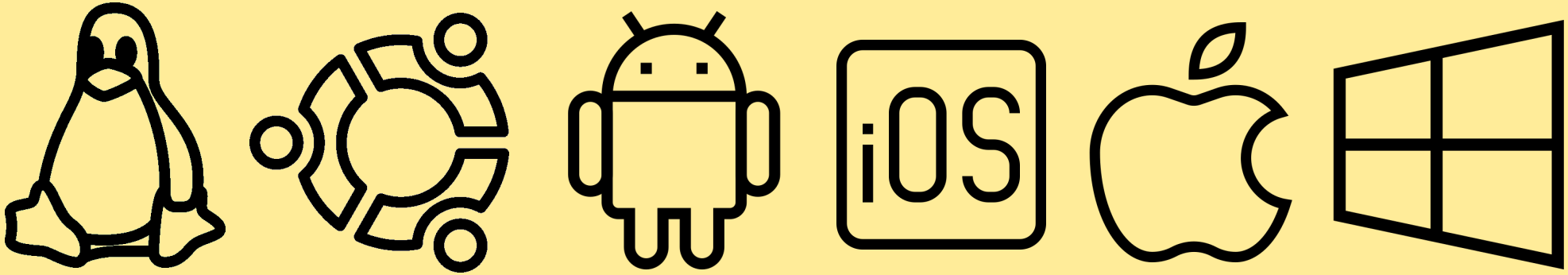
NextGraph SDK

Web (WASM)

Native (Tauri, webview)

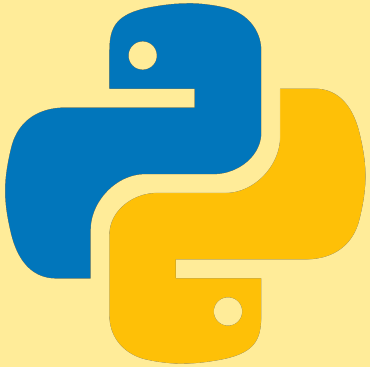
Rust, Python, Nodejs

NextGraph SDK



NextGraph SDK



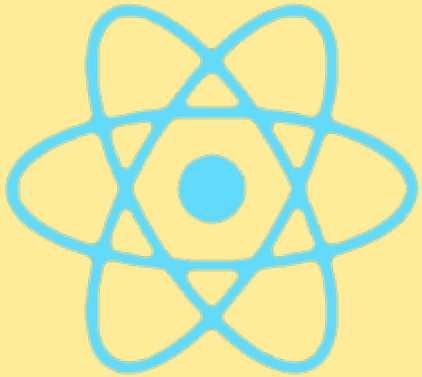



NextGraph SDK

Raw APIs: subscribe, update CRDT binary blobs, or RDF triples

ORM APIs: higher level. bind your POJOs to reactive frontend components

NextGraph SDK



React

SVELTE

Vue.js

NextGraph Framework

History

revision, signature, immutability, audit trail, Authenticity, time travel

Permissions

grant access: read, write, subscribe, pull
Create groups, share with others

NextGraph Framework

Search

full text search, compond indexes

SPARQL

Local queries on all documents, or a subset, or a store.

Reactive Queries

Incremental View Maintenance (IVM) (soon)

NextGraph Framework

Comments

on any document, within apps too

Chat

And group chat

Notifications

With OS integration

NextGraph Framework

Stores / Drive

Public, Protected, Private stores, and Groups

Wallet

User Identity manager, password, keys

App store

Install and publish apps, all decentralized

NextGraph Framework

Social Network

P2P web of trust private social network

Smart contracts

ACID transactions

with strong consistency. Paxos coordination

NextGraph Ecosystem

Web Apps

in third party mode

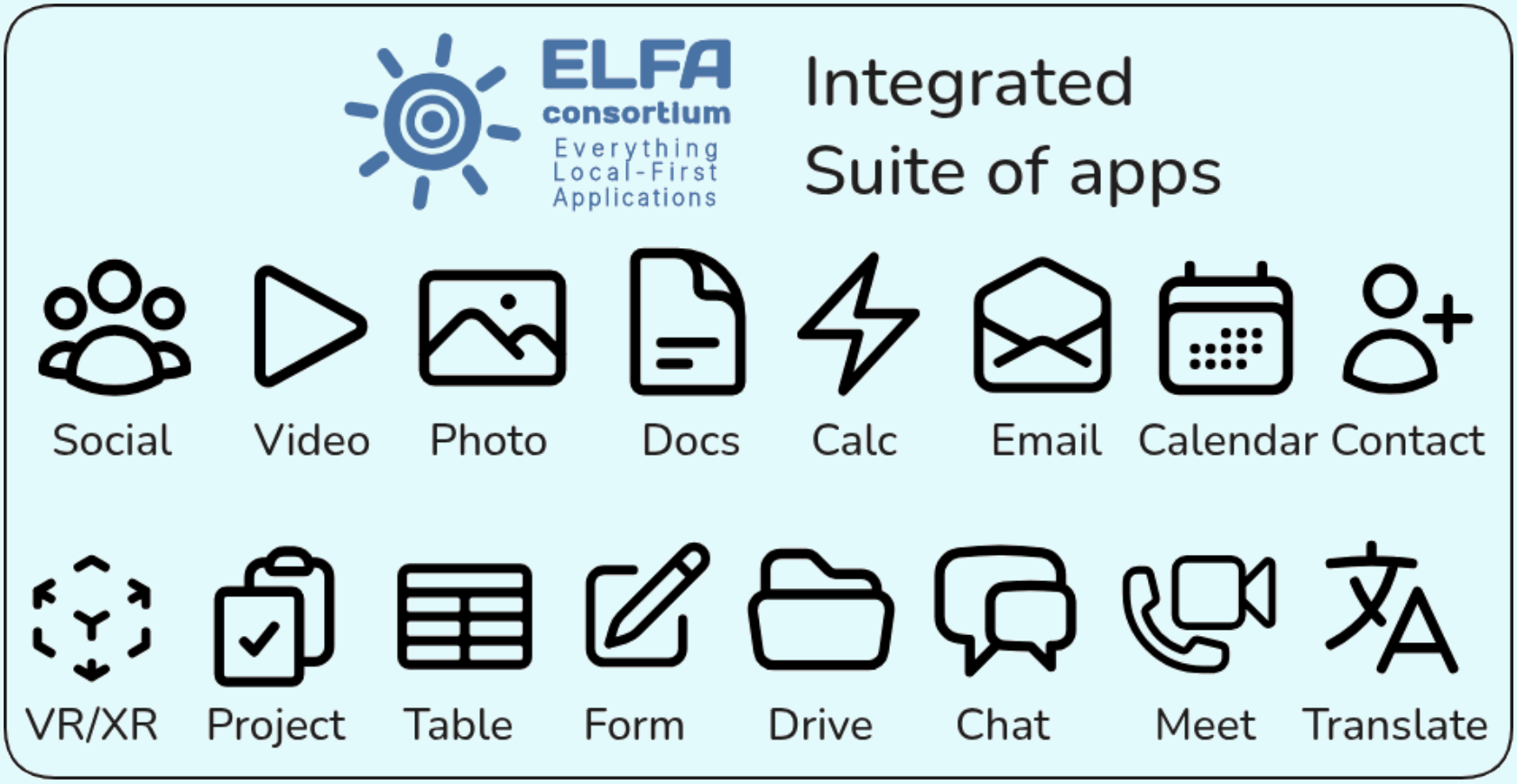
omyn.ai

Native Apps

Services

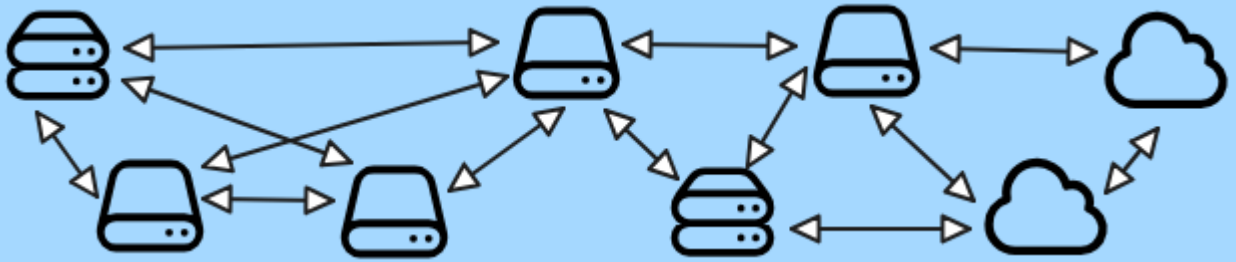
AtomicServer

Everything Local First Applications



NextGraph stack

OPEN PLATFORM : a federation of brokers



At home/office, On Premise, Self-hosted, community hosting, hosters, integrators, edge, cloud

NextGraph stack

NG PROTO : an E2EE encrypted sync protocol
NG ENGINE : local first database

NG OPEN PLATFORM : a federation of brokers

NextGraph stack

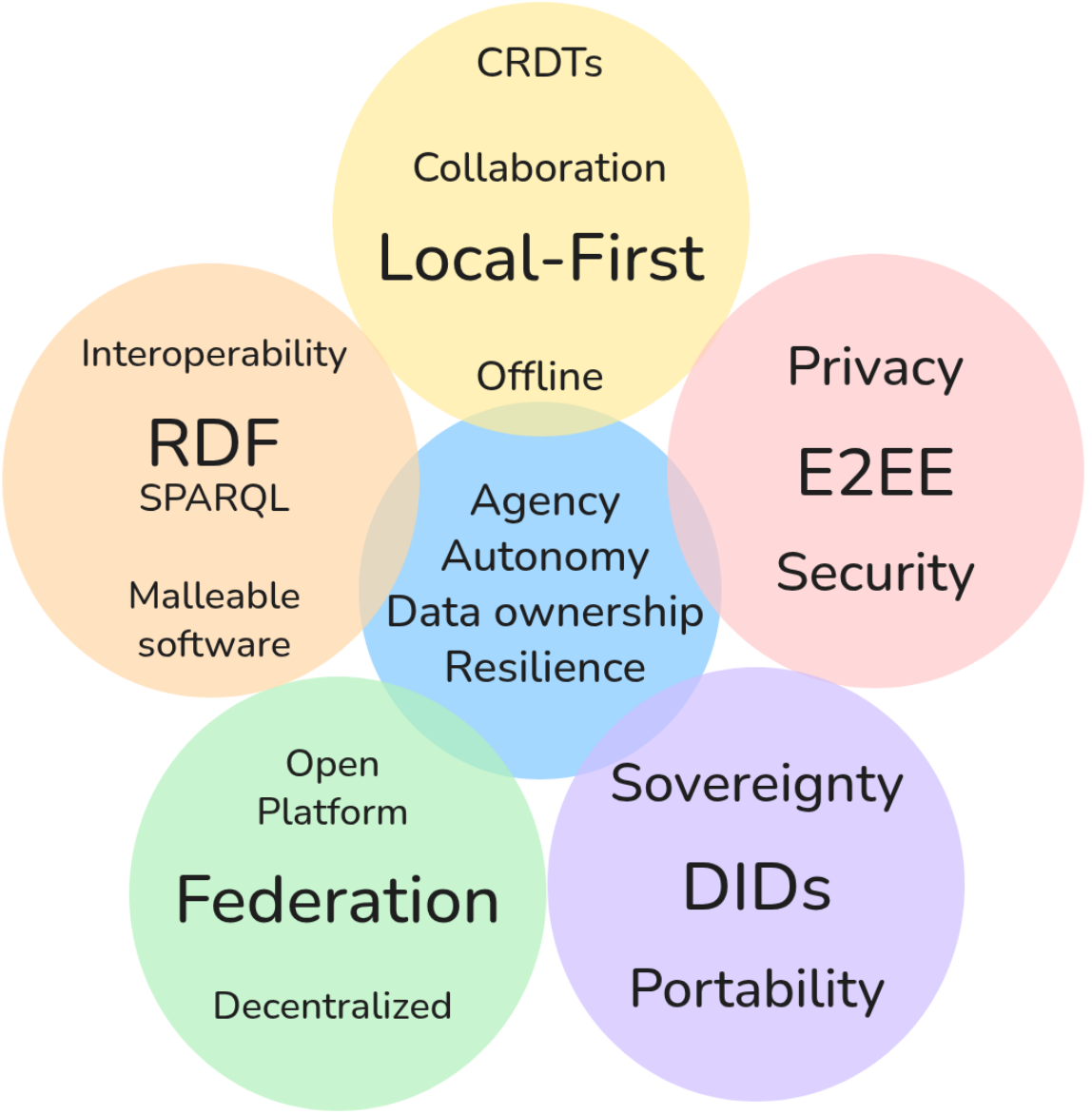
NG ECOSYSTEM : apps and services, ELFA productivity suite, social network, bridges, etc...

NG FRAMEWORK : common features & facilities

NG SDK : build web & native apps, services

NG PROTO & NG ENGINE : E2EE sync database

NG OPEN PLATFORM : a federation of brokers


CRDTs
Collaboration
Local-First
Offline
Interoperability
RDF
SPARQL
Malleable software
Agency
Autonomy
Data ownership
Resilience
Privacy
E2EE
Security
Open
Platform
Federation
Decentralized
Sovereignty
DIDs
Portability

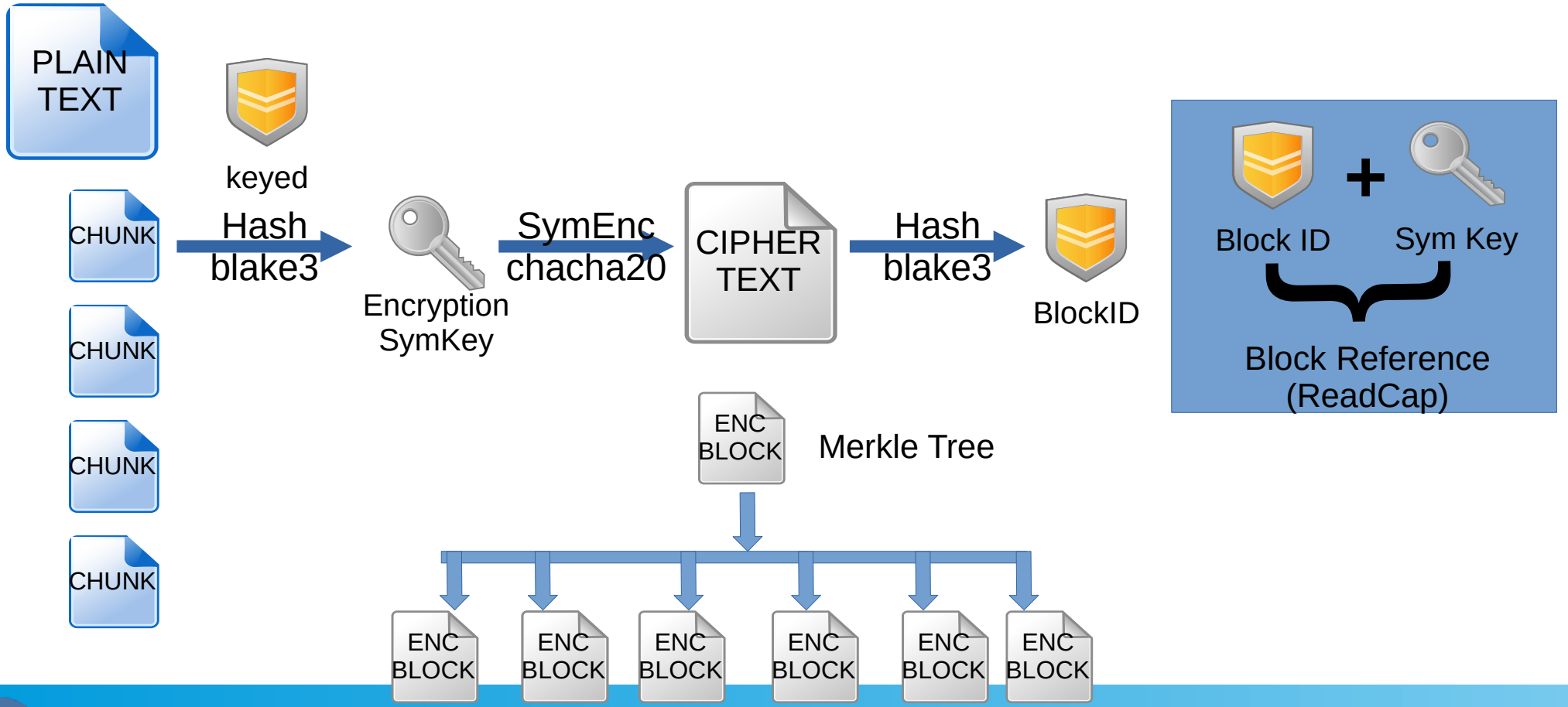
Still in alpha. Next steps

performance
Tauri plugin
App store
framework
core protocol

Permissions & Capabilities

- NextGraph is private by default
- User gets 3 stores: Public, Protected, Private
- Can create more stores: Groups and Organizations
- In addition to E2EE, all materialized state encrypted at rest
- And in transit (Noise protocol inside websocket)
- Permission’s granularity: Document
- Permission inheritance: by Store, and transitive permissions when capability inserted into document

Blocks and References



Encrypted DAG

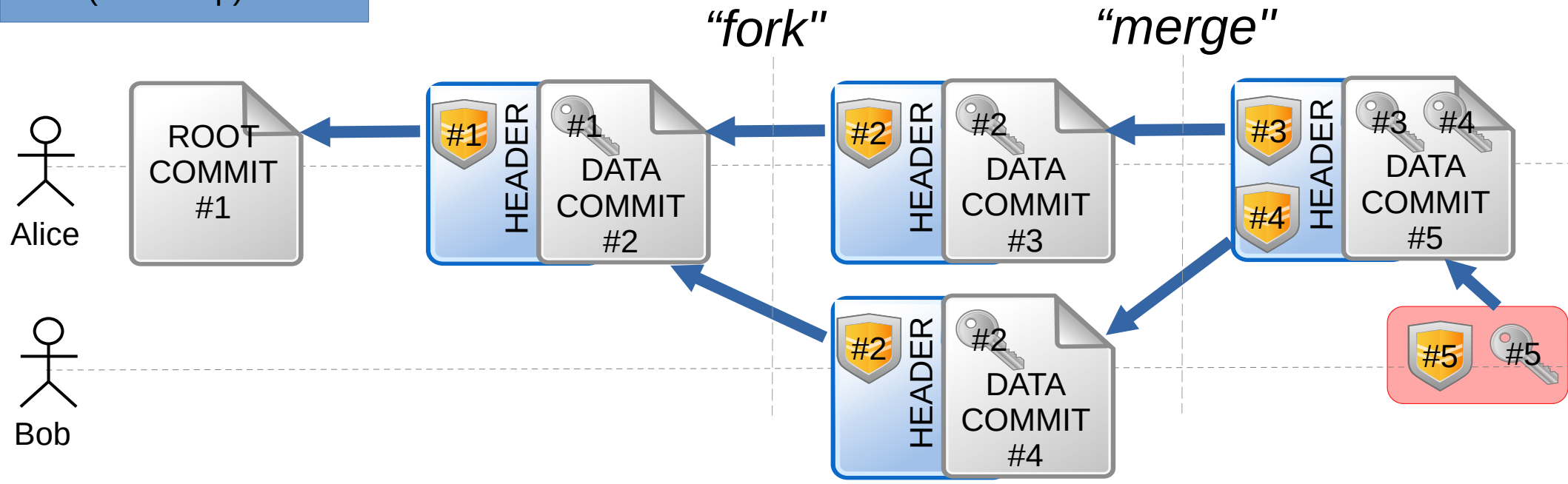
Commit ID + Commit Key

Commit Reference (ReadCap)

TIME

PLAIN TEXT

CIPHER TEXT



Branches

- Each Document is composed of several branches (DAGs)
  - A root branch
  - A main branch
  - Any additional branch: forking a branch or a subdocument
- The PUB/SUB is organized in Topics.
- Branch ReadCap : gives read access to a topic.
- TopicID can be refreshed when ReadCap must be revoked

Sync Protocol

- Sync Protocol is based on BEC paper of Martin Kleppmann et al. adapted to the Broker architecture. Uses bloom filters.
- Causal partial order is preserved => Pub/Sub of NextGraph is a reliable causal broadcast.
- Brokers maintain connectivity between each other and reroute paths if needed, in a general undirected graph network topology. They maintain routing tables.

Sync Protocol

- The Commit Reference #5, the last one in the DAG, is sent in the Pub/Sub Topic, in an Event encrypted with the Branch ReadCap
- Only the readers of this Topic/Branch can decrypt this event. The brokers can’t.
- Clients subscribe to Topics and tell their Home Broker to maintain the subscription for when they go offline.
- When back online, Clients use sync protocol with Broker to get all the missing events.
- Thanks to Brokers, data is available even if no peer is online.

NURI and sharing capabilities

- Capabilities are of a new kind, that we call “cryptographic capabilities”. They contain a symkey (for ReadCap) or a private key (for WriteCap).
- Capabilities are encoded as NURI (DID based NextGraph URIs) and can be included inside a document, making them transitive.

Pull Control & Overlays

- Having access to encrypted events is a risk. We limit to the minimum the capability to obtain encrypted blocks from brokers.
- Users that interact within a Store are isolated at the network level inside an Overlay
- Readers need access to the Outer Overlay
- While Writers need access to the Inner Overlay.
- Access to both is refreshed when needed

External Signatures

- Signatures of individual commits are only useful for other editors or for readers that have access to the root branch, which is not the case of external readers.
- In order to prove authenticity of data to external readers, we have implemented a threshold signature mechanism
- Signers gather asynchronously in a quorum in order to compute a group signature for each repo’s commit
- This signature is verifiable from the Document’s ID (DID) and following a chain of certificates

Finality

- Eventual consistency is great for Document oriented data, aiming at local-first collaboration.
- It is less interesting for use cases like avoiding double spend, ACID properties of transactions, and all cases when the strong consistency of transaction must be assured
- Based on the threshold signature mechanism, we offer the option to mark some segment of a document as “synchronous”. Those will only be modified after a group signature is obtained, enforcing finality and strong consistency

Synchronous Transactions

- Coordination and strong consistency with Paxos
- Synchronous transactions can implement Smart Contracts, cross-document transactions, atomic transactions, decentralized naming system, and e-commerce applications
- It complements well the CRDT/eventual consistent model. Each document/application can benefit from both world.

NextGraph.org

mastodon:

@nextgraph@fosstodon.org

+ forum, newsletter

Come say hi