



Interoperable Data Management for a Decentralized World

✉ john@source.network

🌐 source.network

Stagnant Evolution of Data Management

Fundamental characteristics of hardware have been revolutionized, yet how we manage data has not and persists with structures that date back to a bygone era.



Past

Digital paper
filing



80's & 90's

PC



2000's

Cloud



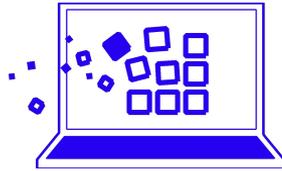
Future

Decentralized



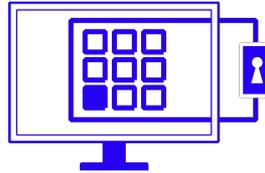
Current Data Management Is Broken

Constantly replicating the same problems and limitations; further magnified by the decentralization of web



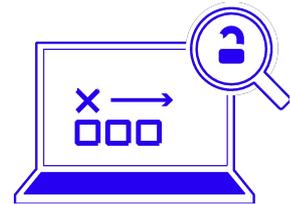
Fragmentation

Unstructured and scattered across applications, devices, and file management system silos.



Imprisonment

Locked behind walled gardens, file containers, or application specific data models.



Vulnerability

At the mercy of bad actors as its impossible to ensure data privacy and security



Decentralized data management needs to

Augment the advancements being made in protocols and decentralized infrastructure; only then will we be able to reach the full potential of Web3



Interoperable

Accessible using open data models in any application or device.



Controllable

True ownership and authority over their information.



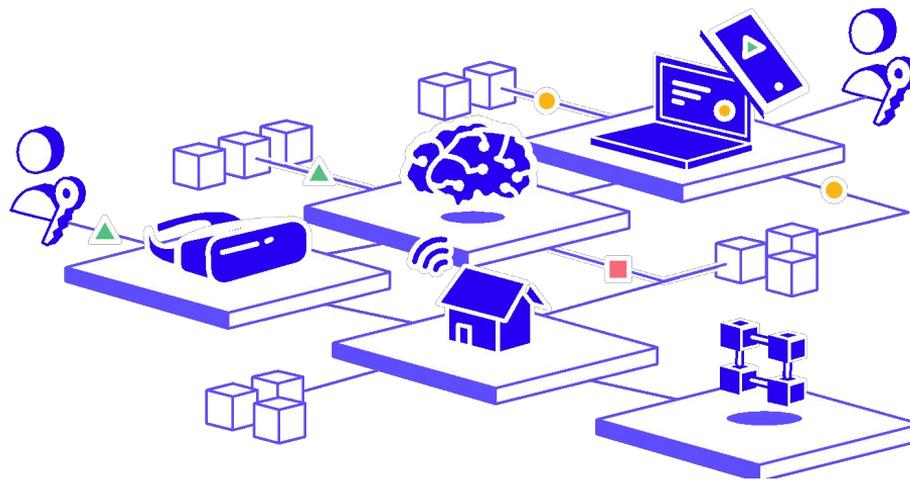
Secured

Anonymization and cryptographic protocols ensures trust.





**Source is the data management layer
that powers the connected future**



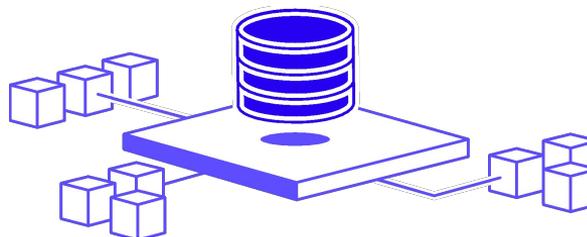


DEFRA · DB

The (DE)centralized In(FRA)structure database.

DefraDB is a peer to peer user-centric Edge Database designed for privacy focused applications.

- Decentralized
- NoSQL Document Store
- Merkle CRDTs
- IPFS Technologies, IPLD & LibP2P
- GraphQL



Why do we need Another database?

The current landscape of databases fails to fulfill the requirements of a modern, privacy focused data management ecosystem



Traditional

Designed for single DB single application use cases.



Cloud Native

Optimized for cloud deployments, newer generation DBs compared to Traditional.



Peer-to-Peer

Take an extreme approach to user ownership, only useful in some applications.



What Sets DefraDB Apart



Extensible Deployment Models

DefraDB can be deployed in a variety of environments including edge servers, user devices, and high performance data center clusters. All connected via Peer-to-Peer networking.



User Centric, Developer Friendly

User centric database means that the user is in control of their data regardless of where it lives. Applications request access to read/write user data through a NoSQL document interface using GraphQL,



Semantically Linked and Content Identifiable

DefraDB relies on semantically linked schemas, and Content Identifiable Data models to realize its goal of interoperability. Using a combination of JSON-LD, JSON-Schema, and IPFS CIDs.



Mutable CRDT Structures

Internally, all data is structured as a mutable link of immutable DAG links. Using a new generation of structures, Merkle CRDTs. Allowing multiple actors to freely and independently make changes, even offline.



The Tech Behind the DB

We've made a series of advancements in database, cryptographic, and distributed technologies.



Merkle CRDTs

A new CRDT system, optimized for peer-to-peer, decentralized applications.



Efficient State Sync & Verification

Using Zero Knowledge Set Membership, Recursive SNARKS, and Bloom Filters to quickly synchronize and verify DAG state



Secure Indexes

Utilizing encrypted indexes on remote peers to maintain user privacy.

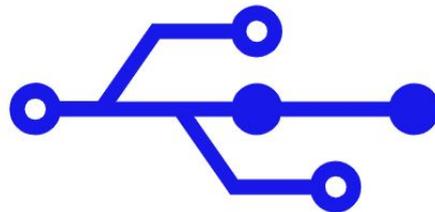


Merkle DAG CRDTs

Merkle CRDTs combine the power of Merkle DAGs and CRDTs.

They rely on a new mechanism to provide ordering of events, called MerkleClocks. MerkleClocks are optimized for high churn, large participant networks.

Together with Content Addressable Identifiers, MerkleCRDTs allow deterministic verifiable convergence of state.



Content Identifiable Data (CID)



01010101
01000101
00101000

01001010
01010001
111110

01010010
10000111
100101

Chunk

da99e818
432bb526
5b41b

fdbcdda0
47eb248a
44eede

e975420f
856b8cc6
4dd

Digest

Qmemejfe
22Gy...

Qma1kuVp
Z9MU...

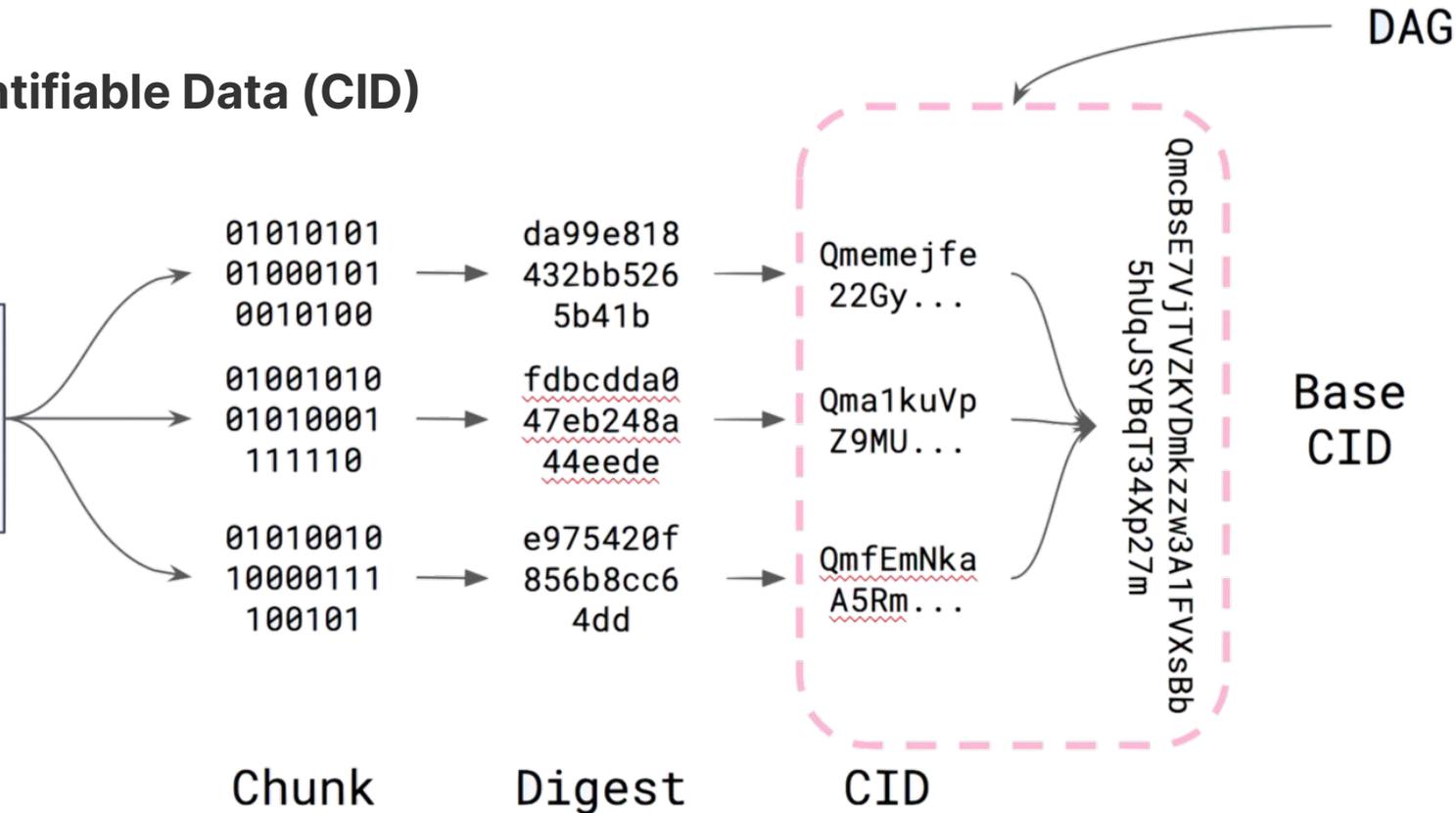
QmfEmNka
A5Rm...

CID

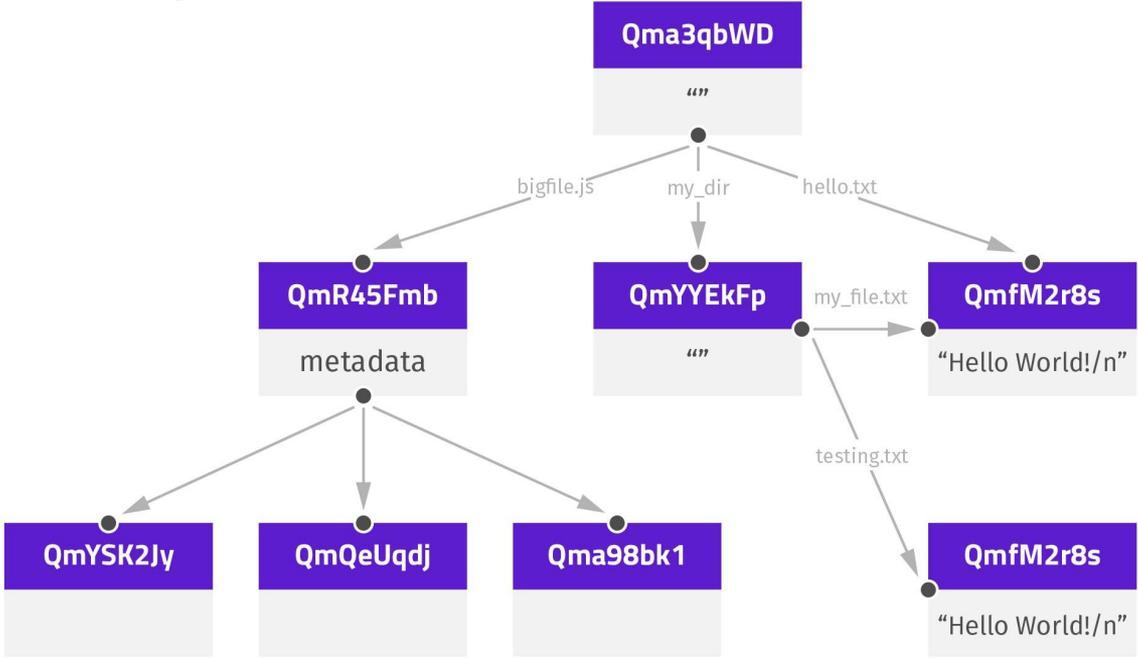
QmcBSE7VjTVZKYDmkzZw3A1FVXSbB
5hUqJSYBqT34Xp27m

Base
CID

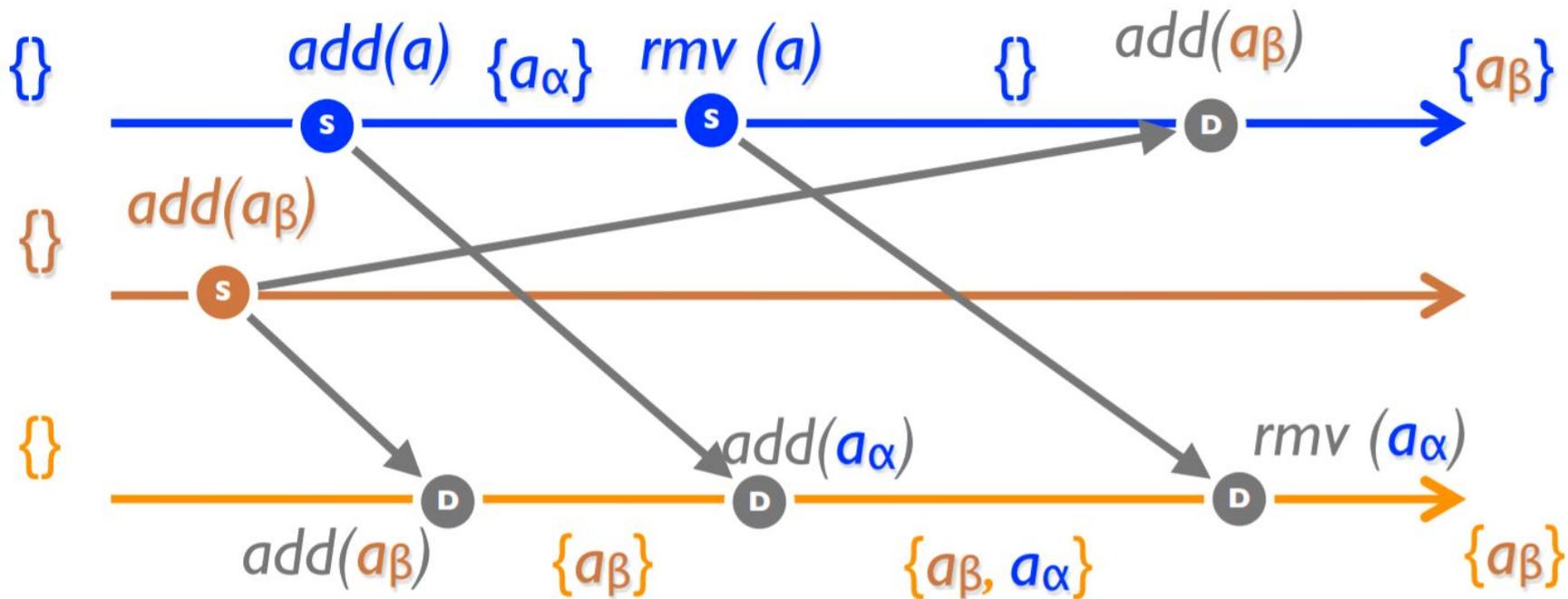
DAG



Directed Acyclic Graph (DAG)

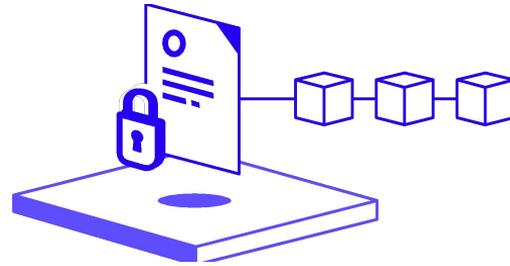


Conflict-Free Replicated Data (CRDT)



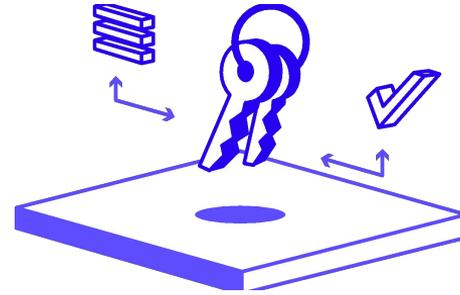
Global Trust Layer Blockchain

A high throughput Tendermint-powered Blockchain, providing access control, policy enforcement, identity management, secret management and audit trailing. Powering the trust layer of the Source Ecosystem.



Zero Knowledge Key Management System

Industry leading On-Chain Key Management System offering powerful user authentication & authorization. Founded on developer friendly standards like DID and OpenID. Enabled by Zero Knowledge Proofs, and Multi-Party Computation. Providing all the simplicities of Cloud login, with the benefits of decentralized security.

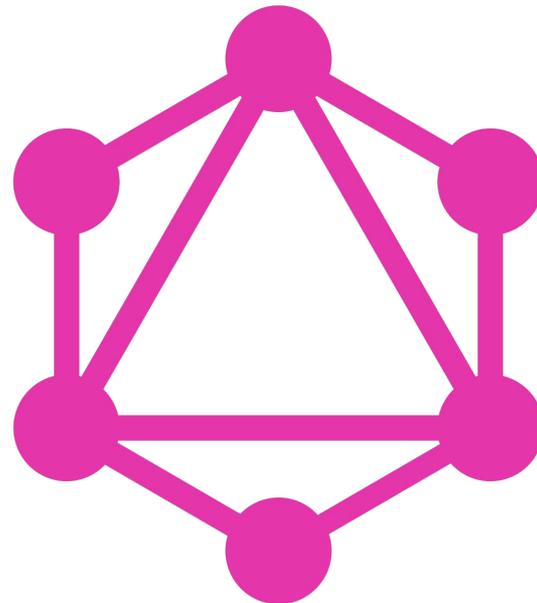


GraphQL: Native Database Query Language

DefraDB supports the Industry leading query language, GraphQL. GraphQL has been blazing the trail and redefining modern API interfaces.

DefraDB is one of the first databases to natively utilize GraphQL as its primary Query Language.

This approach reduces the friction between application, and data.



```
type User {
  name: String
  age: Int
  verified: Boolean
  address: Address
  friends: [User]
}

type Address {
  street: String
  number: Int
  city: String
  country: String
}

// Query for all users who are older than 18, and has
// more than 5 friends.
//
// Return the name, age, and friends from the result set.

query {
  users(filter: {age: {_gt: 18}, {friends: {_length: {_gt: 5}}}) {
    name
    age
    friends {
      city
      country
    }
  }
}
```

```
type Commit {
  cid: String      // CID of the node in the graph
  height: Int      // height of the node
  delta: String    // delta payload that generated the node
  links: [CommitLink] // and links this node refers to
}

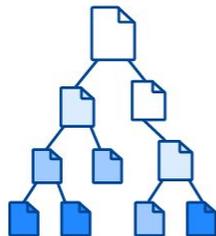
type CommitLink {
  cid: String // CID of the target link
  name: String // name of the link (refers to a field or a HEAD)
}

// Query for all users who's name contains 'Smith', order by the 'age' field
// in ascending order, limit to the first 10.

query {
  users(filter: {name: {_like: "Smith"}}, limit: 10, order: {age: ASC}) {
    name
    isVerified: verified // alias to isVerified

    // query the generated _version field
    // which lets us traverse the MerkleCRDT graph.
    _version {
      cid
      height
      links {
        name
        cid
      }
    }
  }
}
```

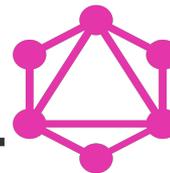
Document



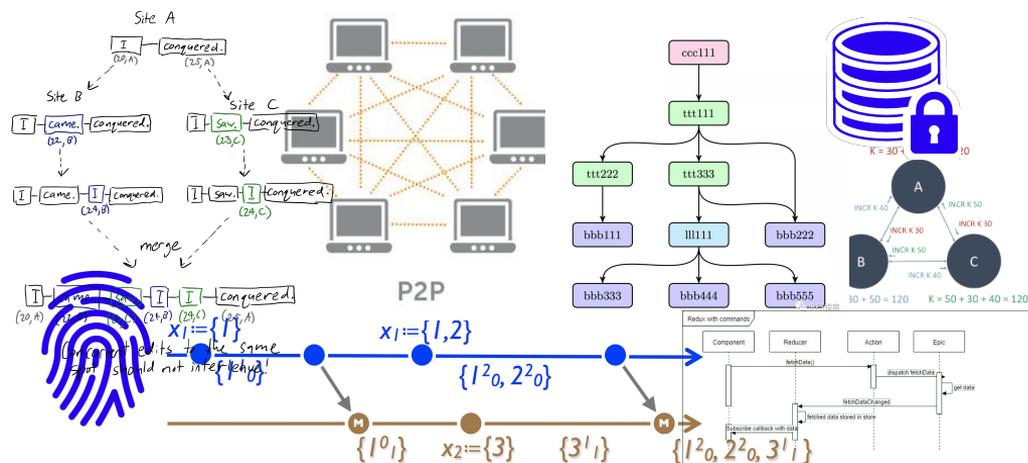
NoSQL



GraphQL



DEFRA · DB



Early Results

Don't take our word for it

We have partnered with some of the most innovative and valuable protocols in the space to bring Source to their ecosystem and are executing on integrations.

Here is what they are saying:



“

“Your vision for data management is exactly what is needed for Web3.”



“

“Once you hear what you guys are doing, you wonder why has it taken this long.”



“

“We need to make Source available at our hackathon.”



“

“Advances you guys are making with CRDTs are much needed for Web3.”



“

“We need this to work with our protocol asap.”





**Let's build the interoperable
Web3.**

✉ john@source.network

🌐 source.network