The Fluree Solution

Fluree is an open-source, protocol-based, Linked Data platform consisting of ledgers that record tamper-proof updates to data and horizontally scalable query services that materialize a ledger, or a combination of ledgers, as time-traveling graph databases that maintain provable provenance.

An individual Fluree Ledger is agnostic to both dataset size and underlying storage. Fluree supports both micro-ledgers (e.g. a Verifiable Credential or Decentralized IDentifier) through large databases at TB or PB scale. Ledger storage is pluggable with support for a local file system, cloud providers (i.e. S3), decentralized storage networks (like IPFS), public blockchains, or combinations thereof.

A horizontally scalable Fluree Graph Database with sub-millisecond query speed can be materialized from a single Fluree Ledger, or by merging multiple ledgers regardless of owner or underlying storage. Small datasets enable this to happen entirely in-memory, or a trusted Fluree indexing service can be run to make large databases scalable even to small devices.

In addition to merging ledgers from multiple, Fluree can also federate management of a single ledger to multiple parties through the deployment of a Federated Ledger Server that operates under programmable policies called Fluree SmartFunctions. Like the storage layer, consensus in a Ledger Group is designed to be pluggable and today runs a Proof of Authority model over the Raft consensus protocol.

Next, we'll cover the three foundational components of the Fluree system: Linked Data, Ledgers, and Query.

Crypto-data protocol

Storage agnostic

Linked/semantic data standards

Data can be big or small

Custom data policies

Unlimited scale

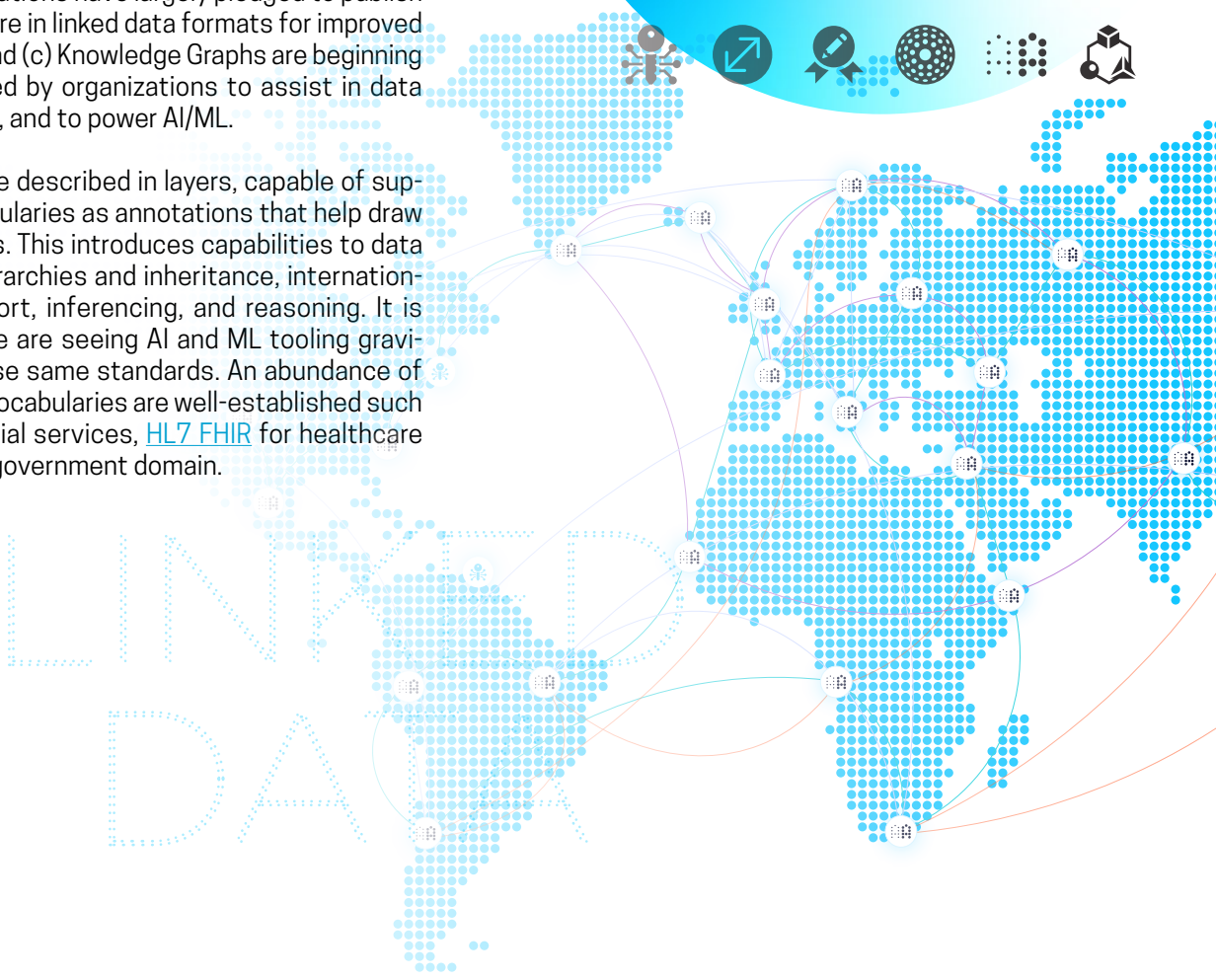
Privacy-Preserving

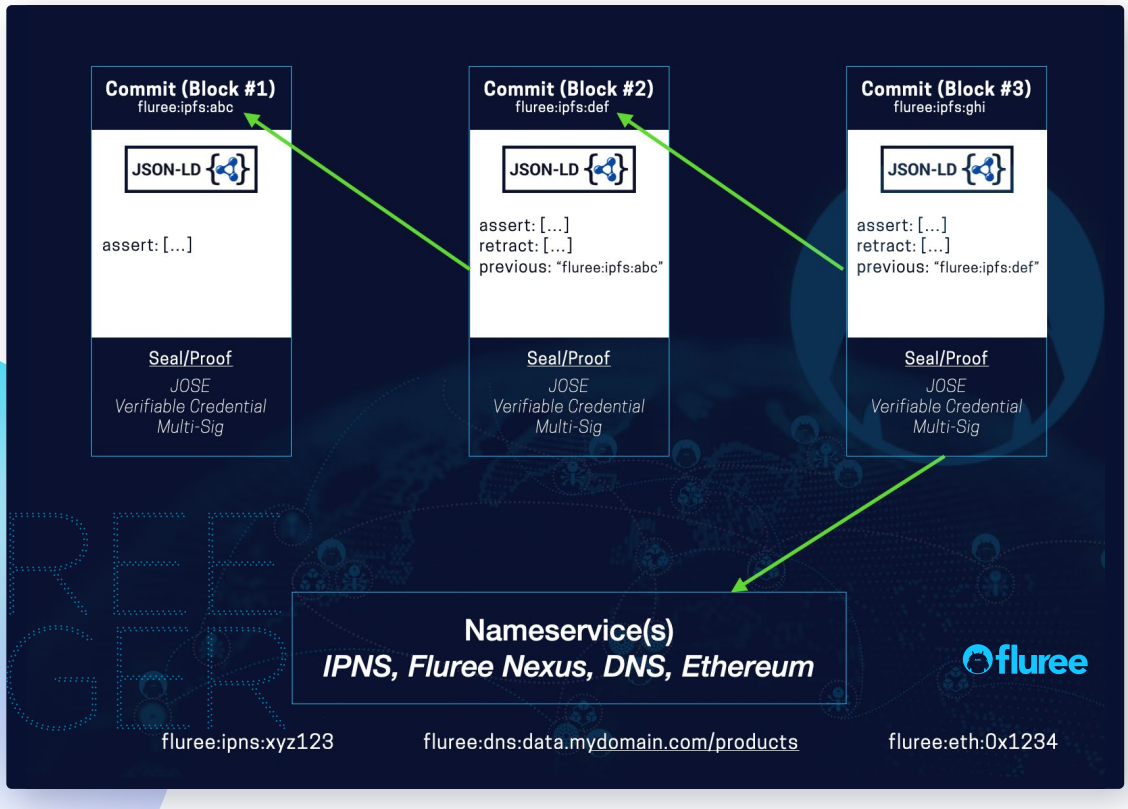
LINKED DATA

Fluree data is managed using standards-based linked data. Interaction with Fluree, be it updates/transactions or database query results, is formatted with JSON-LD. JSON-LD is a way of using JSON serialization to represent W3C RDF data, a simple data format that can store any conceivable fact in a manner that is self-describing.

Linking gives to data what HTML linking gave to the web - decentralized references. For this reason JSON-LD is the basis for decentralized data standards like Verifiable Credentials and Decentralized Identifiers. There is also an abundance of Linked Data that already exists primarily from three domains: (a) over 10 million websites contain embedded linked data to improve search relevance and other features through initiatives like schema.org, (b) research institutions have largely pledged to publish all data in the future in linked data formats for improved interoperability, and (c) Knowledge Graphs are beginning to be implemented by organizations to assist in data agility, integration, and to power AI/ML.

Linked data can be described in layers, capable of supporting rich vocabularies as annotations that help draw additional insights. This introduces capabilities to data such as class hierarchies and inheritance, international language support, inferencing, and reasoning. It is for this reason we are seeing AI and ML tooling gravitate towards these same standards. An abundance of domain-specific vocabularies are well-established such as FIBO for financial services, HL7 FHIR for healthcare and NIEM for the government domain.





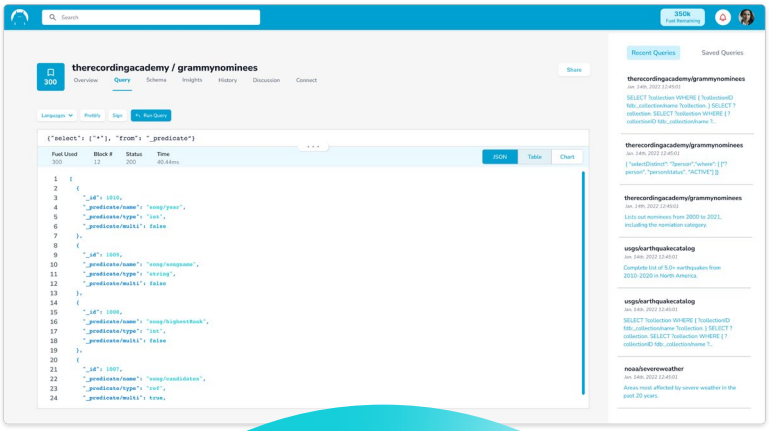
THE FLUREE LEDGER

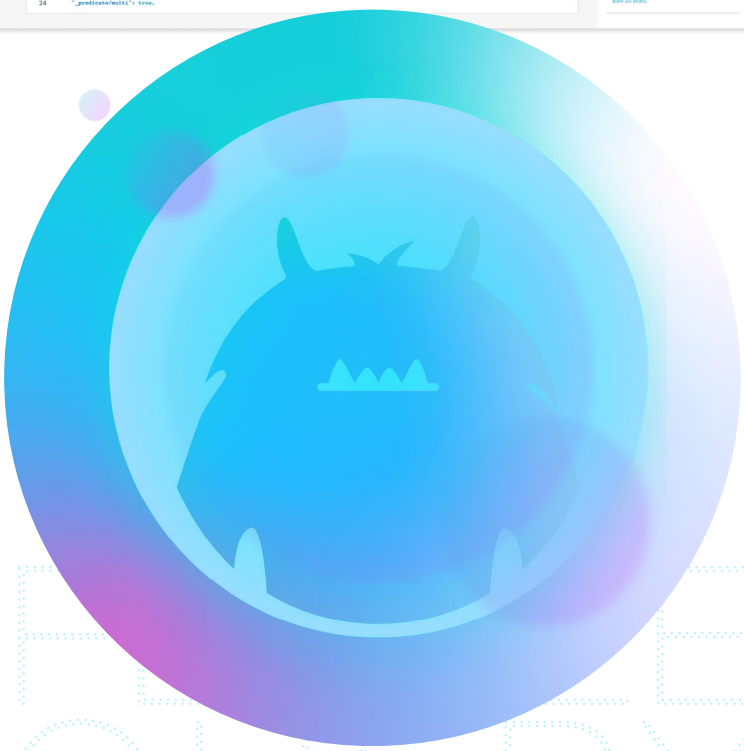
The Fluree Ledger creates cryptographically secured, traceable updates (transactions) backed by a pluggable storage service (currently local disk, IPFS, S3, and memory - Cardano storage in development). Because ledgers are encouraged to use linked-data standards for interoperability, any number of ledgers across any number of organizations/entities can write in parallel and resulting graphs can be merged at query time with no limitation of write volume for the overall network. All ledger updates, and therefore every historical database version, becomes content-addressable and is stored in a highly-optimized form that records only deltas and provenance information.

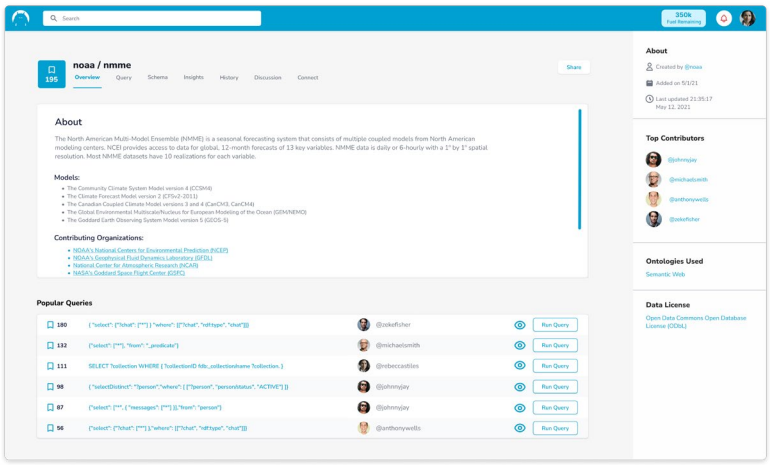
The capability of dynamically merging data not only provides immense flexibility, but also opens up the new realm of micro-ledgers, which might record ongoing updates to a student transcript as a Verifiable Credential issued by a university, or updates to an organization's DID (Decentralized Identifier) Document, recording cryptographic key updates. While Fluree can support very large datasets, it is legacy database limitations that have generally required these large data-sets to exist in the first place by necessitating all data be housed together to operate under a single query. Fluree does not suffer from that same limitation, and regardless of the use case can manage data at a micro, or macro scale.

While any organization, edge-device, or individual can publish a dataset and ongoing updates independently with provable provenance, it is possible to run a Ledger Group across independent entities governed by rules-as-data called Fluree SmartFunctions. Ledger servers in this configuration utilize Proof of Authority and communicate via the Raft protocol, although the consensus protocol is designed to be pluggable. An ongoing project funded by the Cardano allows a micro-ledger consisting of the Fluree SmartFunction governance to be managed on-chain (due to space and cost limitations of Cardano), and the bulk of the data to be stored off-chain while being governed by on-chain interactions. Fluree's ability to merge graphs means that data intended for a single application need not all exist in the same physical place. This capability can also be leveraged for segmenting privacy-sensitive data like PII.

Being storage-agnostic creates flexibility not only for users of Fluree but also the Fluree protocol itself, as it is not tied to any specific technology in a rapidly evolving space.







THE FLUREE QUERY SERVICE

A Fluree Query service represents the ledger data as a queryable database, designed to scale to the edge without any limitations. Instead of thinking of a database as a centralized and limited resource, Fluree's "database-as-a-variable" concept brings the database co-resident with the application code as a library, be it server-based, mobile, edge, or directly inside of a web-based app, e.g. React.

While small ledger-backed databases can easily run fully embedded inside apps, large databases cannot practically do so. Any user with ledger access can run one or more Fluree Indexing services, which generate small (~100kb) index chunks leveraging a form of persisted immutable data structures that tracks new ledger updates via novelty that can be merged client-side between occasional indexing jobs. This allows TB-scale DBs to be served in small memory footprints and also exposes HTTP query endpoints in addition to the embeddable libraries.

Apps can query for data using multiple languages, the primary of which of FlureeQL which is a JSON-LD based query-as-data interface that combines the immensely powerful analytical capability of the standards-based SPARQL graph query language and graph-crawling strengths of GraphQL - both of which are also natively supported along with a limited set of SQL for for those completely new to graph.

Private ledger data can also be exposed in a permission-compliant manner by running a Fluree Index Service that is publicly accessible. The index service will require all queries to be cryptographically signed to prove identity, and the same Fluree SmartFunctions capability that secures policy-driven ledger updates can be used to create "virtual databases" that only contain data the user can view.

This allows permissioned consumers the ability to use common query languages to get exactly the data they want in exactly the format desired without ever having to build or consume custom API services. This approach we call data defending itself gives more flexibility at substantially lower costs while improving security over traditional API development.