Fluree is made up of two components: FlureeDL, a custom distributed ledger that handles all updates, and FlureeDB, a graph database optimized to build smart applications on top of FlureeDL. In both systems, Fluree developers use sophisticated logic (SmartFunctions™) to enforce custom read/write permissions and rules directly at the source as meta-data.

### FlureeDL

- **Immutable Data**: Every update is recorded as a block that includes the prior block’s hash, and itself is hashed using SHA 3-256.
- **Pluggable Consensus**: Optionally decentralize your ledger across a network of participants with RAID or PBFT consensus.
- **Configurable Governance**: Create your custom data schema and enforce its rules for ledger updates with custom read/write SmartFunctions™.
- **Partition-Friendly**: Partition data and storage for privacy and compliance (and FlureeDB can bring it together in query).
- **Standard and Versatile Data Format**: FlureeDL outputs ‘block’ data in W3C standardized RDF format for versatile and semantic integration.

### FlureeDB

- **Graph Database**: Optimized to work with FlureeDL RDF data, FlureeDB is a rich database capable of powerful aggregations.
- **Time Travel**: Query against any historical version of your immutable database and instantly retrieve results.
- **Familiar Language Support**: Query support for SPARQL, GraphQL, and FlureeQL (JSON REST API) make FlureeDB an easy tool to pick up.
- **Join Data in Query**: Query across data stores (either Fluree or otherwise), securely joining RDF data for broader access.
- **ACID Compliant**: Provides transactional consistency so your applications can scale with reliable data.