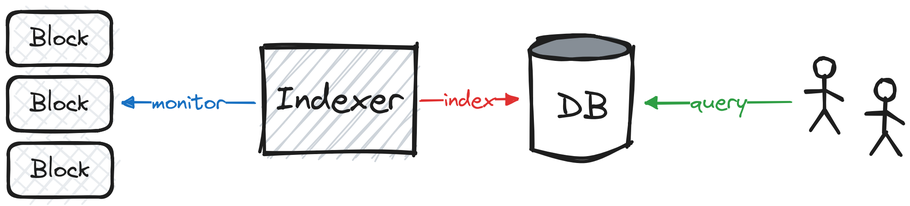
Cardano Indexer Cheat Sheet

v1

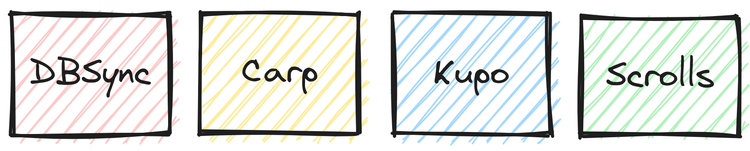
About Blockchain Indexers

Blockchains are NOT databases. We need "indexers" to monitor the chain and store data in a query-friendly format



Cardano Ecosystem

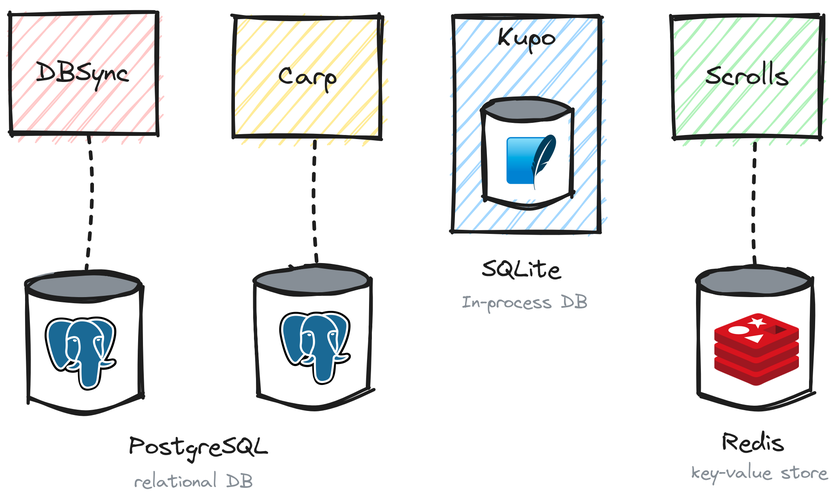
Cardano has a very diverse ecosystem of chain indexers, each one with different trade-offs



⚠ //TODO: research the following tools and add them to the cheat sheet:

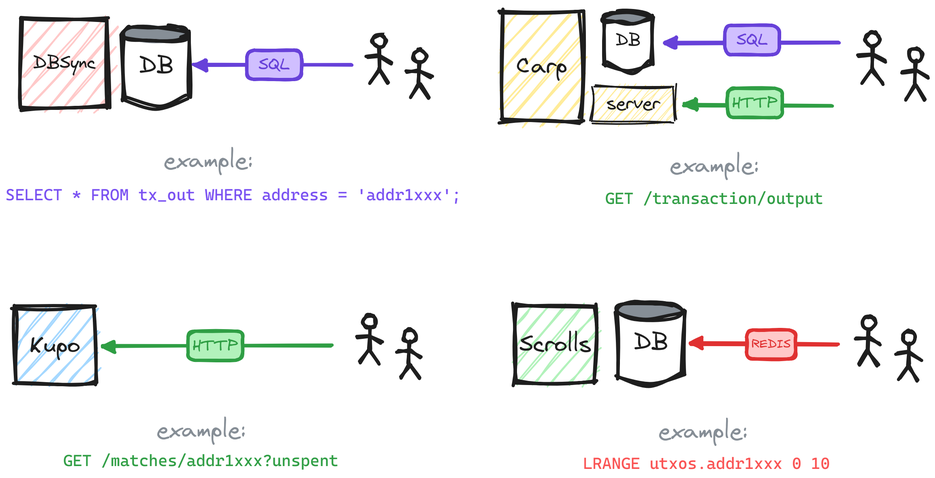
Storage Backends

Indexers need a place to store the data, usually a database of some sort



Query Mechanism

Your app needs to query the indexed data somehow, each indexer provides access in a different way



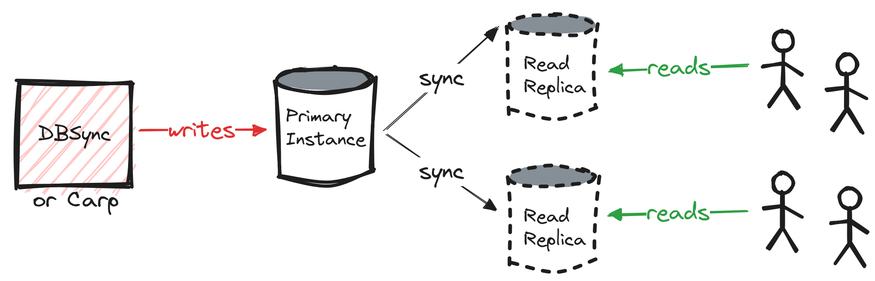
Horizontal Scalability

When traffic increases, your might need to scale horizontally to sustain the load.

Different strategies are used by each indexer

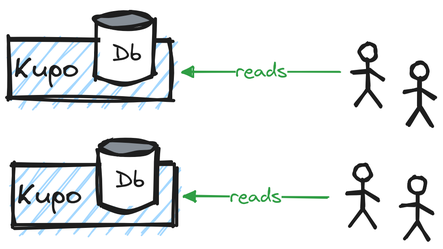
Read-only Replicas

PostgreSQL can run with a primary that handles write ops and multiple read-replicas that are kept in sync. Readers can be load-balanced across replicas for scaling.



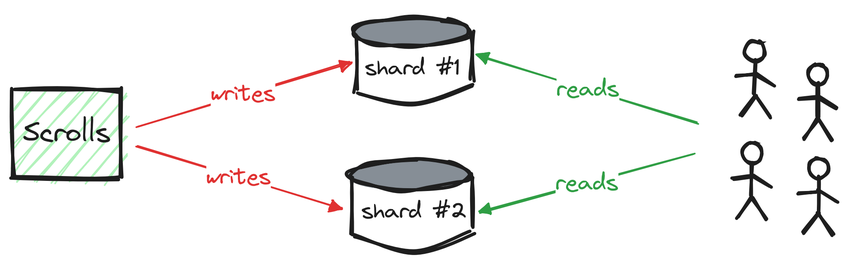
Full Replicas

Multiple replicas of the indexer process + data are created side-by-side. Queries are load-balanced across available instances.



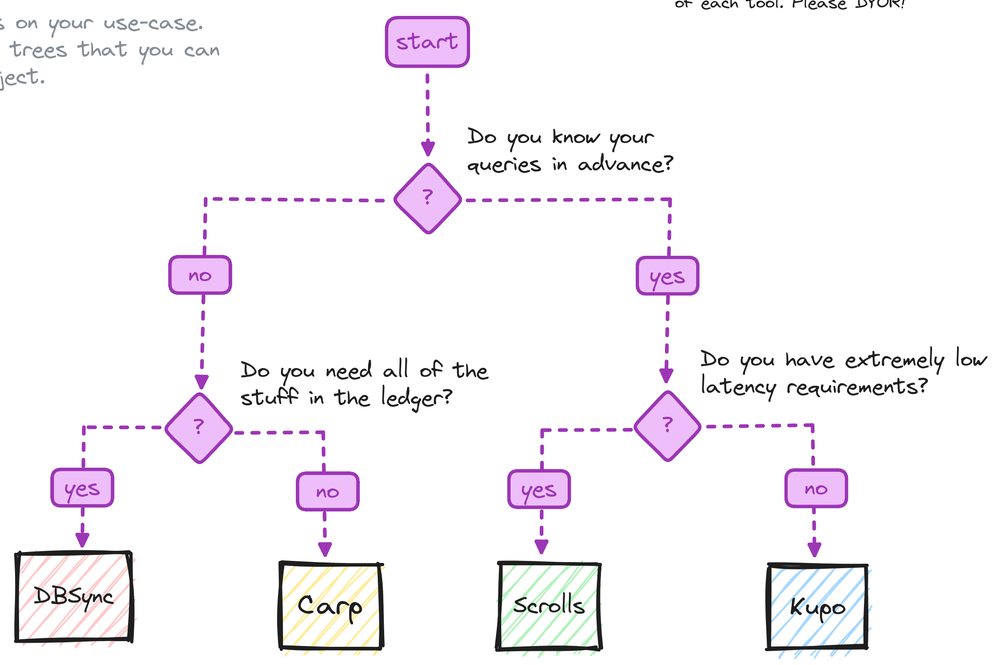
Sharding

Consistent hashing is used to distribute data across different data partitions. Both writers and readers know how to route their calls to the correct shard.



How to pick one

Which indexer to pick depends on your use-case. Here's one (of many) decision trees that you can use when starting a new project.

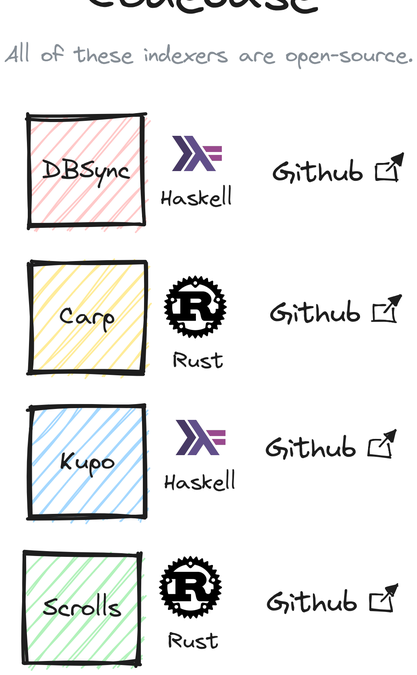


Overall simplicity trophy goes to Kupo. If you're not sure what you need, start with Kupo.

⚠ DISCLAIMER: This is a very crude characterization of the trade-offs of each tool. Please DYOR!

Codebase

All of these indexers are open-source.



This is an open-source cheat sheet.
If you want to contribute, send a PR to:

https://github.com/txpipe/cheat-sheets

made with ♥ by TxPipe