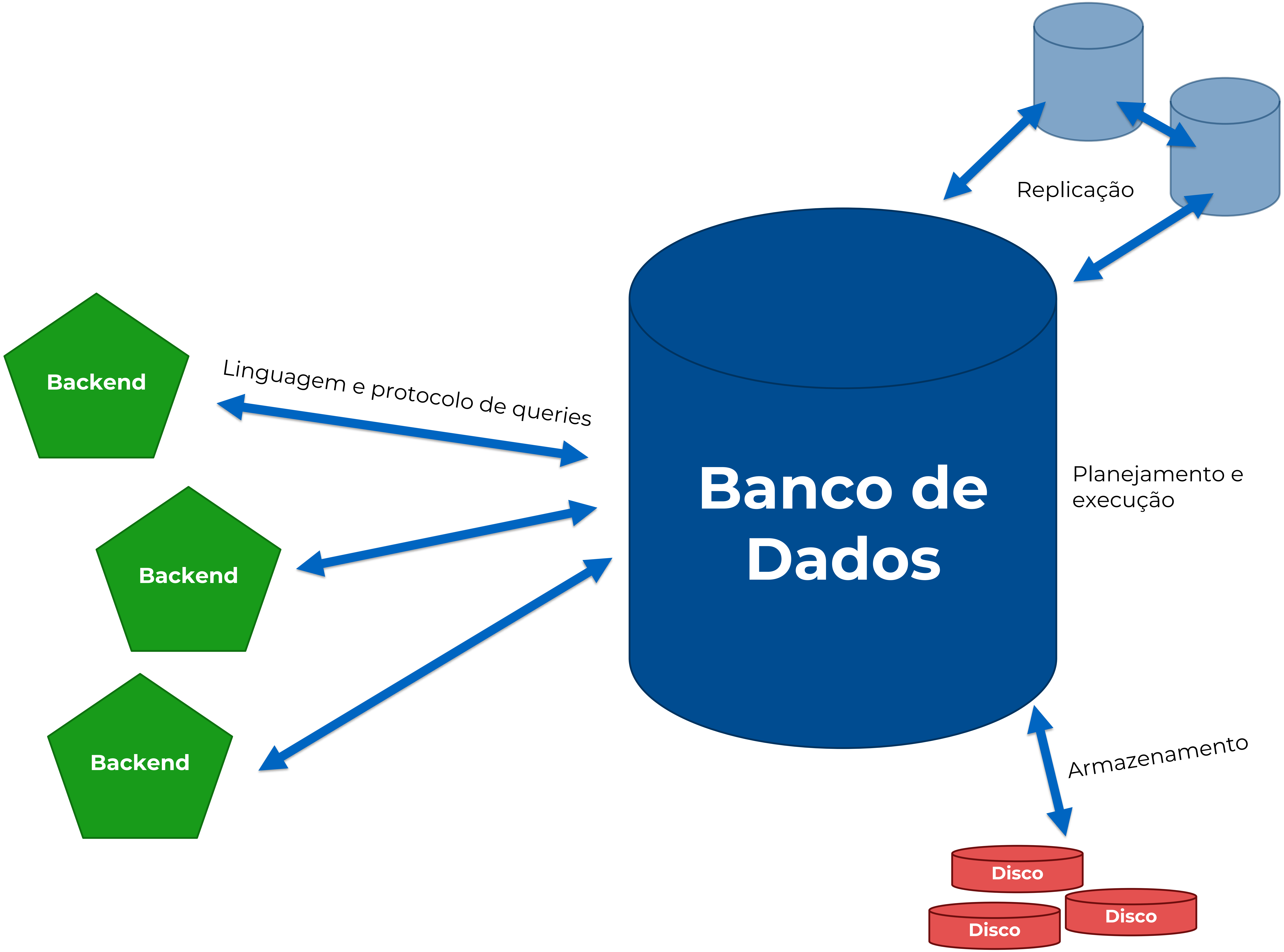
nosql:ba

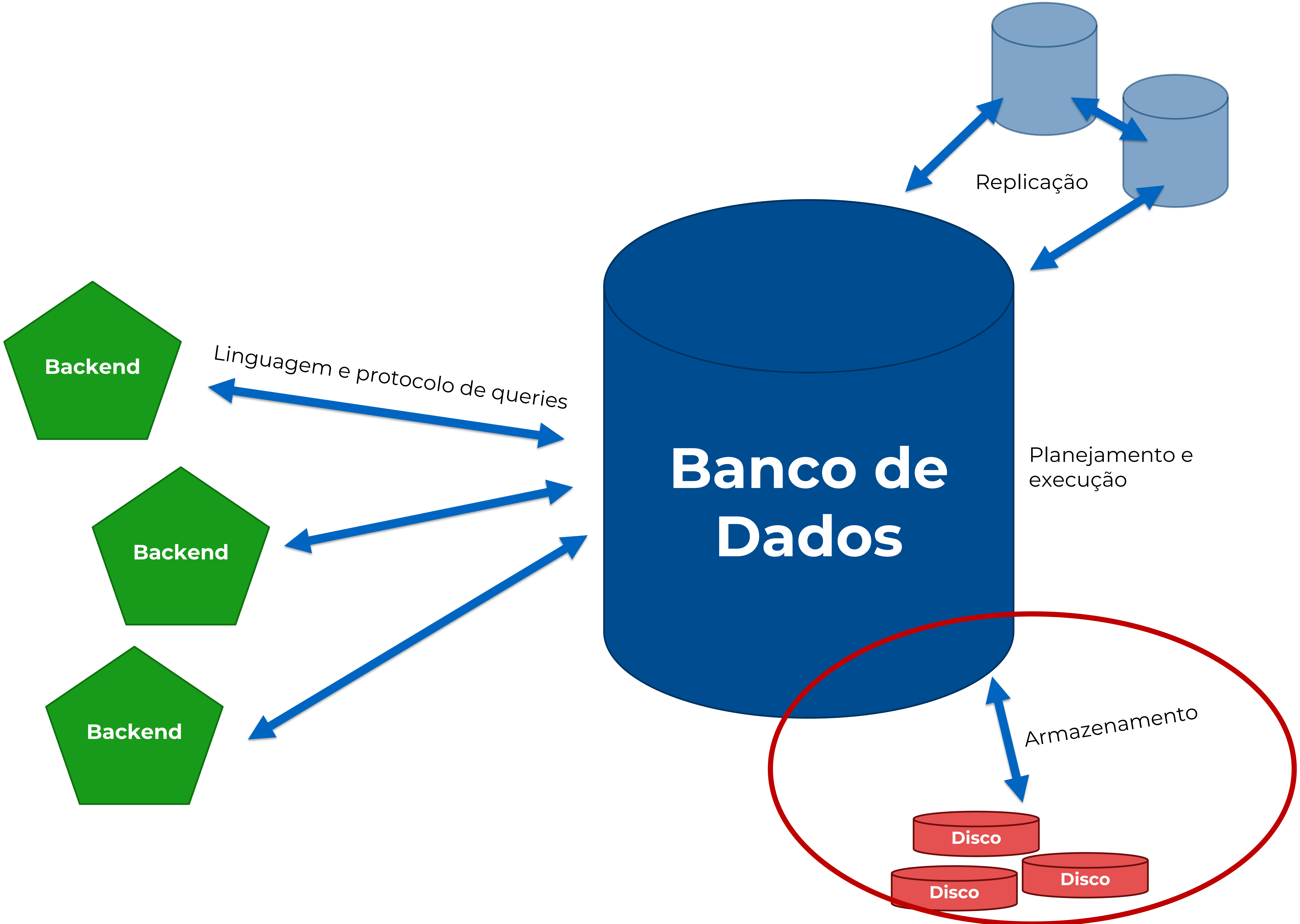
Construindo um banco NoSQL do zero

cubos

Do zero? Tá doido?

- Existem muitos bancos diferentes
- Vantagens e desvantagens
- Garantias de consistência
- Performance


Replicação
Backend
Linguagem e protocolo de queries
Backend
Backend
Banco de Dados
Planejamento e execução
Armazenamento
Disco
Disco
Disco


Replicação
Backend
Linguagem e protocolo de queries
Backend
Banco de Dados
Planejamento e execução
Backend
Armazenamento
Disco
Disco
Disco

Camada de Armazenamento

Camada de Armazenamento

Objetivo primário é prover **Persistência** (ou **Durabilidade**).

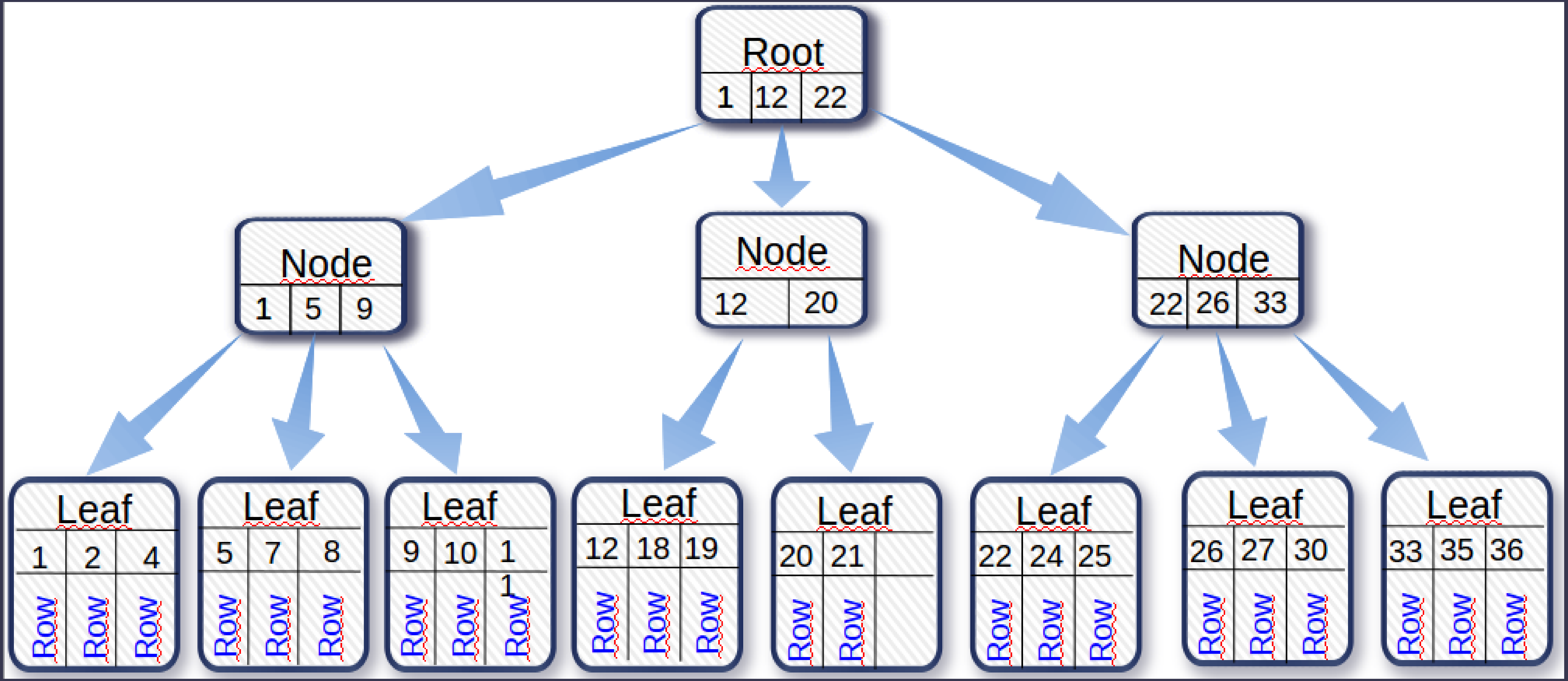
Algumas características importantes são a latência de uma leitura ou escrita, o throughput sustentado, a amplificação, o consumo de espaço e a capacidade de realizar escritas atômicas.

Os dados podem ser armazenados com algumas estruturas diferentes, otimizadas para trabalhar em blocos do disco. Pode utilizar uma **B-tree**, ou **LSM tree**, ou **Hash table**, **String table** ou **Log**.

A maioria dos sistemas utilizam um **WAL** (Write Ahead Log) para atingir consistência.

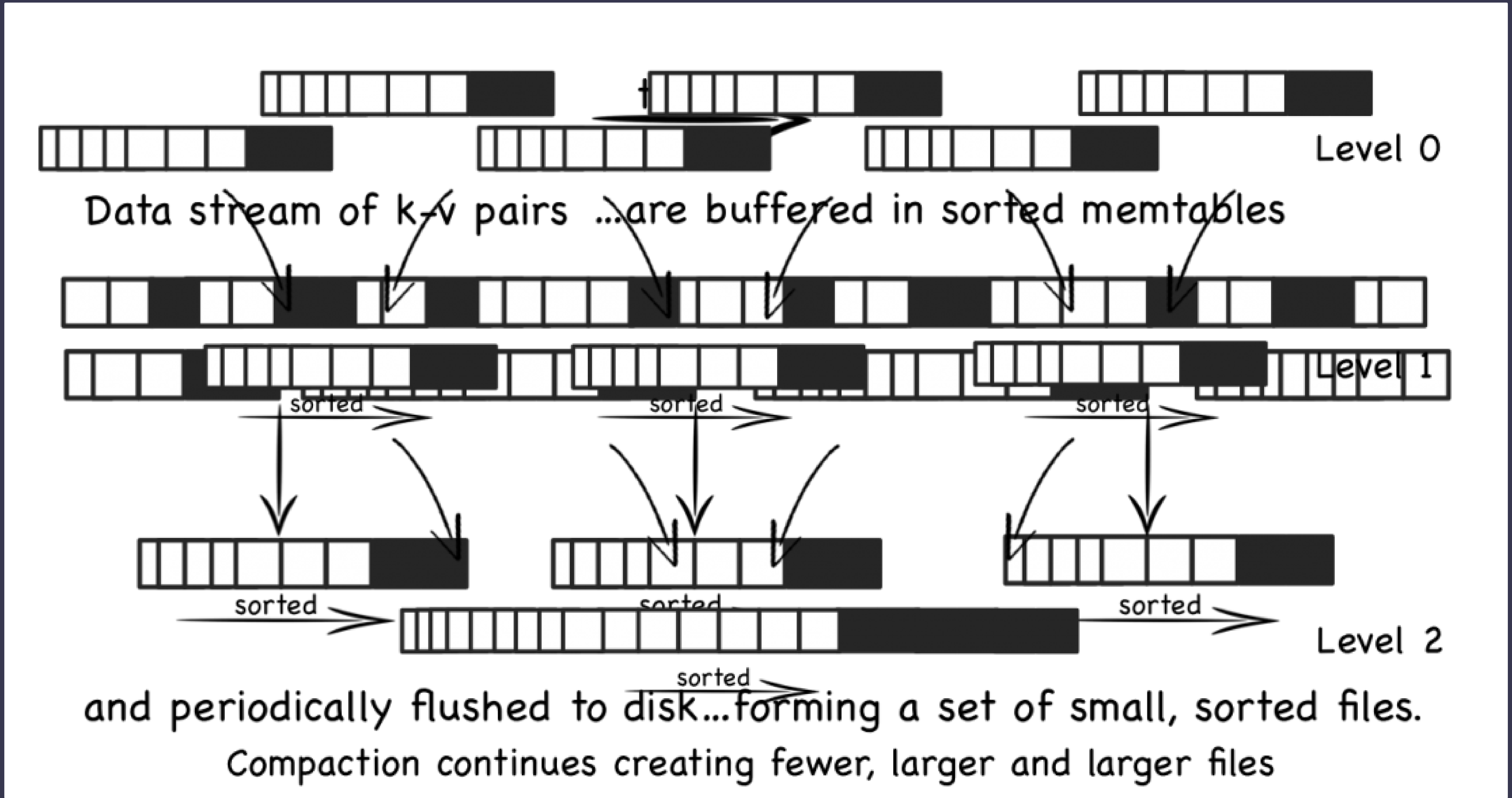
Camada de Armazenamento

B-tree:



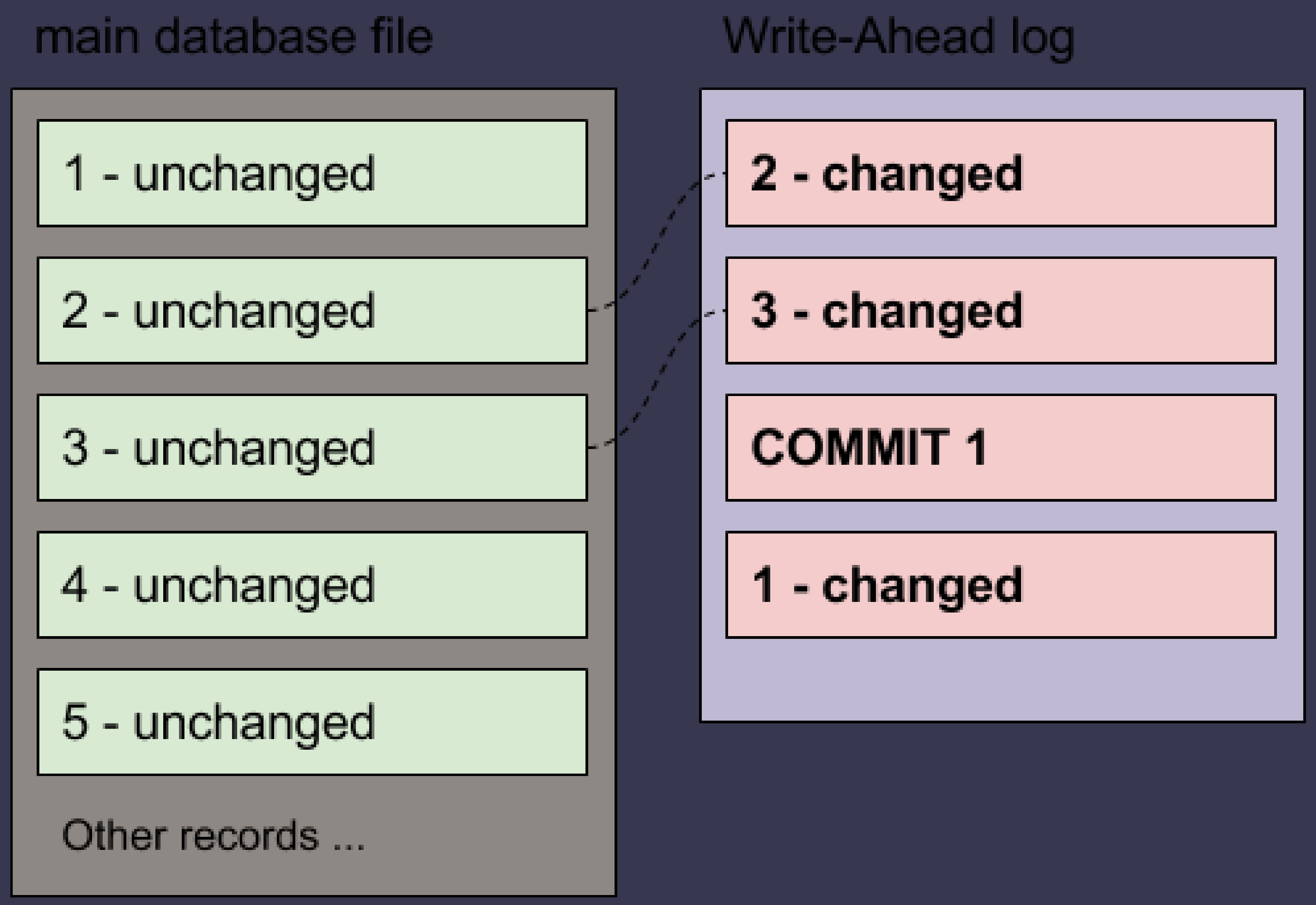
Camada de Armazenamento

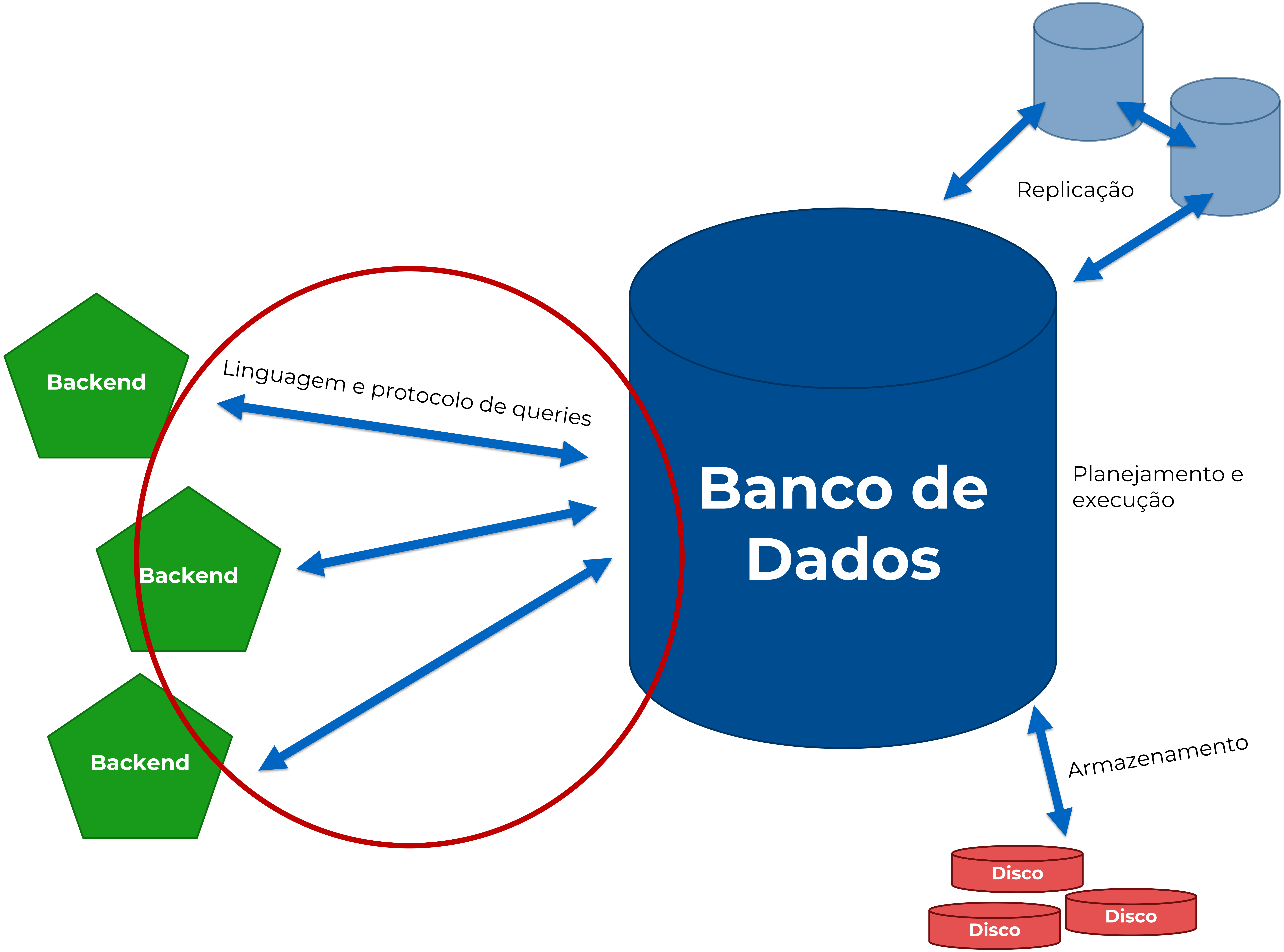
Log Structured Merge Tree:


Level 0
Data stream of k-v pairs ...are buffered in sorted memtables
Level 1
sorted
sorted
sorted
sorted
sorted
sorted
Level 2
sorted
and periodically flushed to disk...forming a set of small, sorted files.
Compaction continues creating fewer, larger and larger files

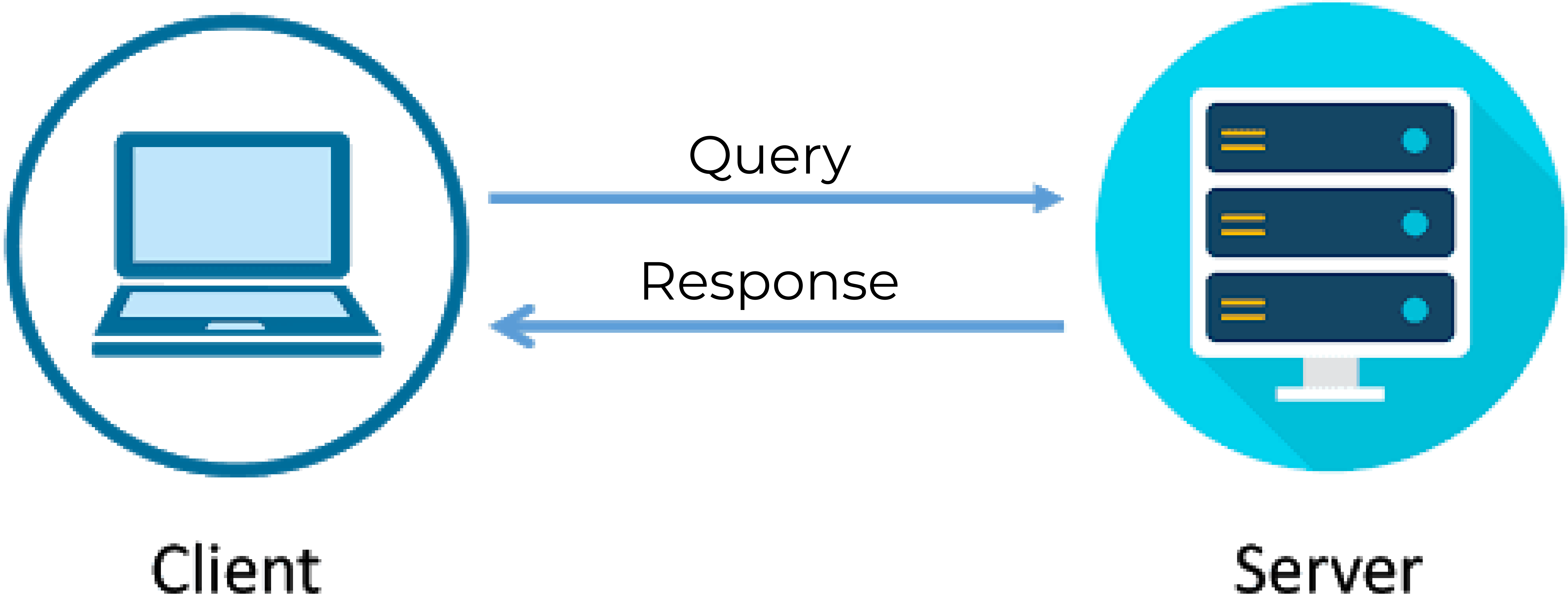
Camada de Armazenamento

Write-Ahead Log:




Replicação
Backend
Linguagem e protocolo de queries
Banco de Dados
Planejamento e execução
Backend
Backend
Armazenamento
Disco
Disco
Disco

Camada de Queries



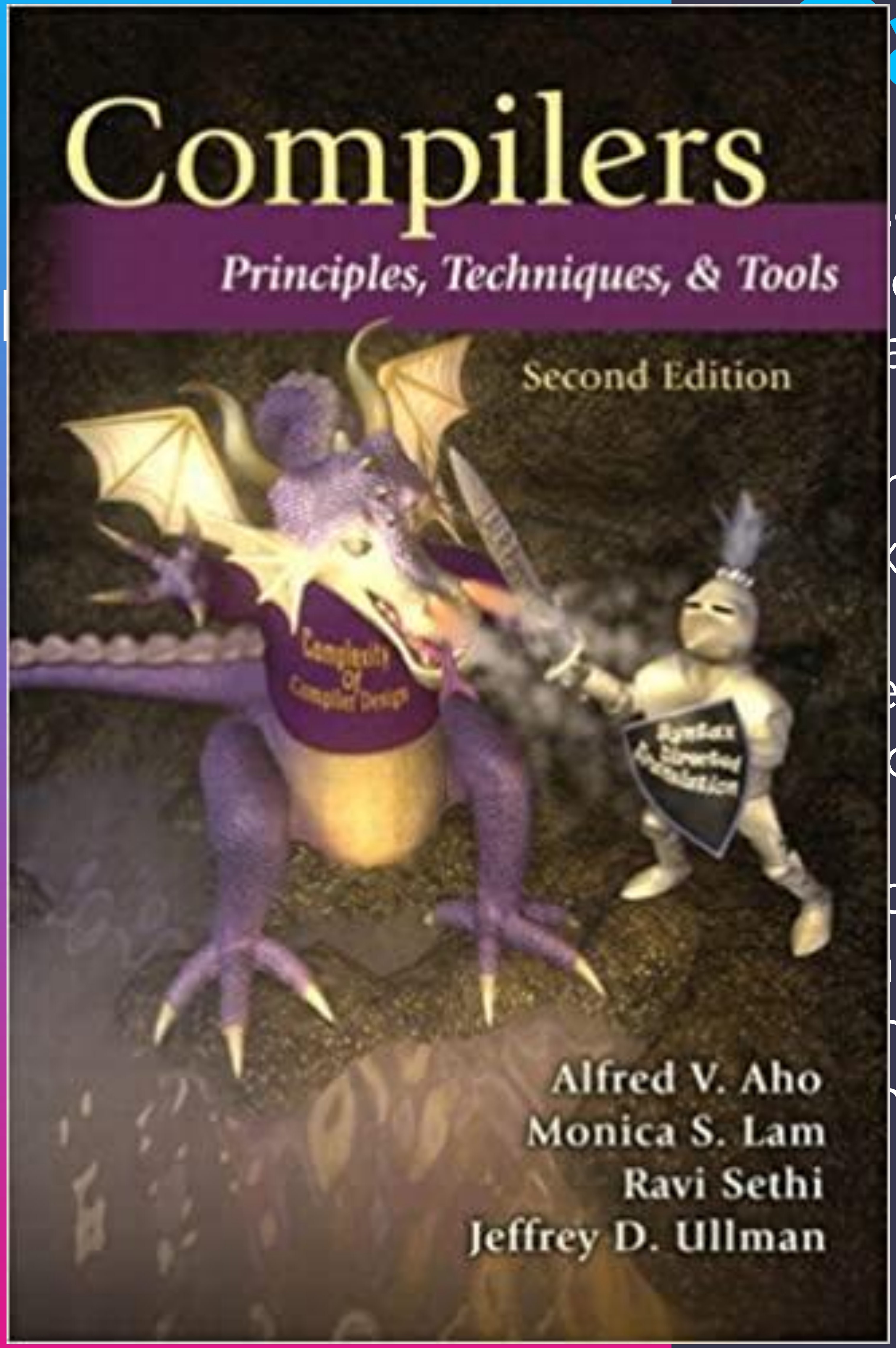
Query

Response

Client

Server

Camada de Queries

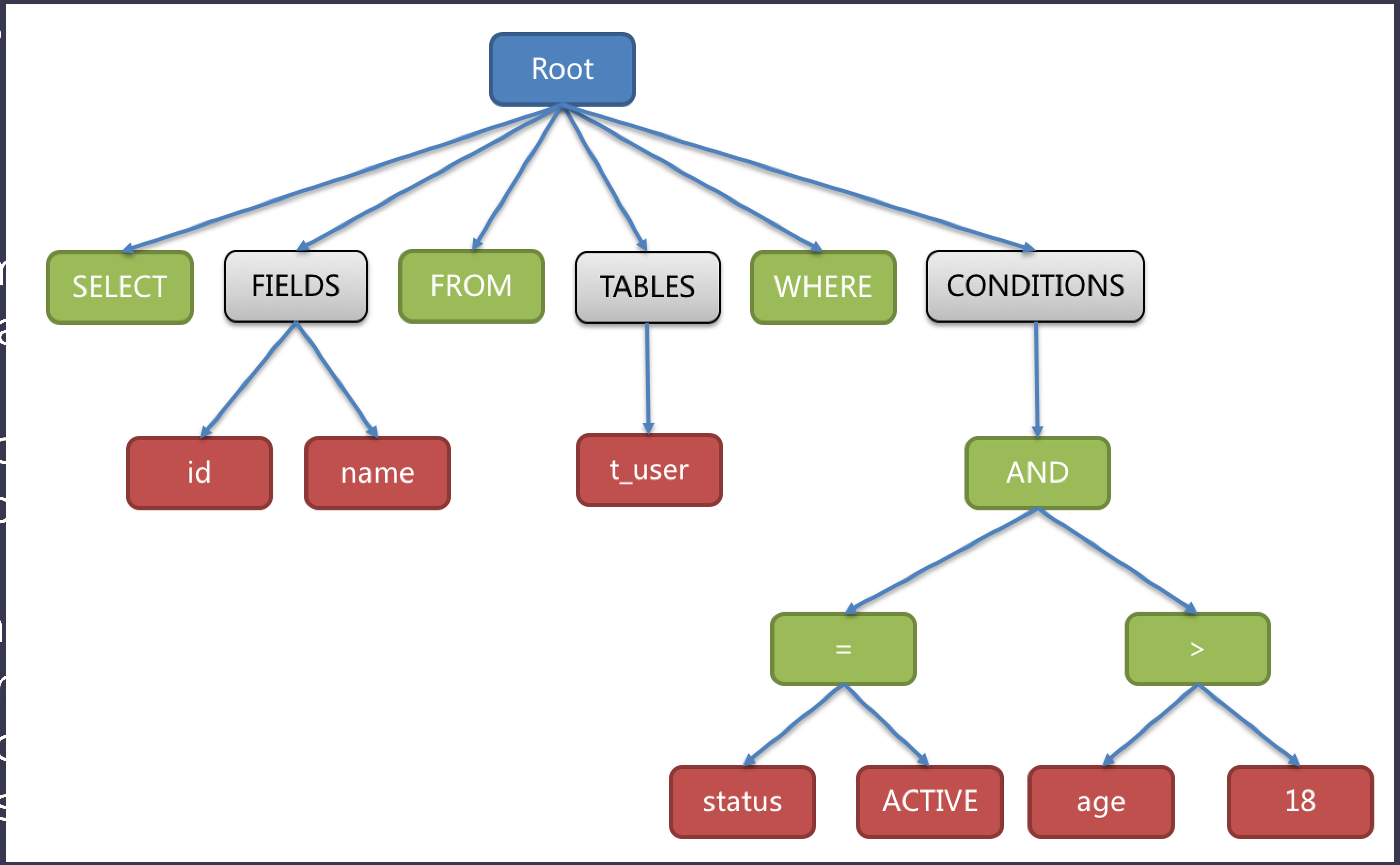


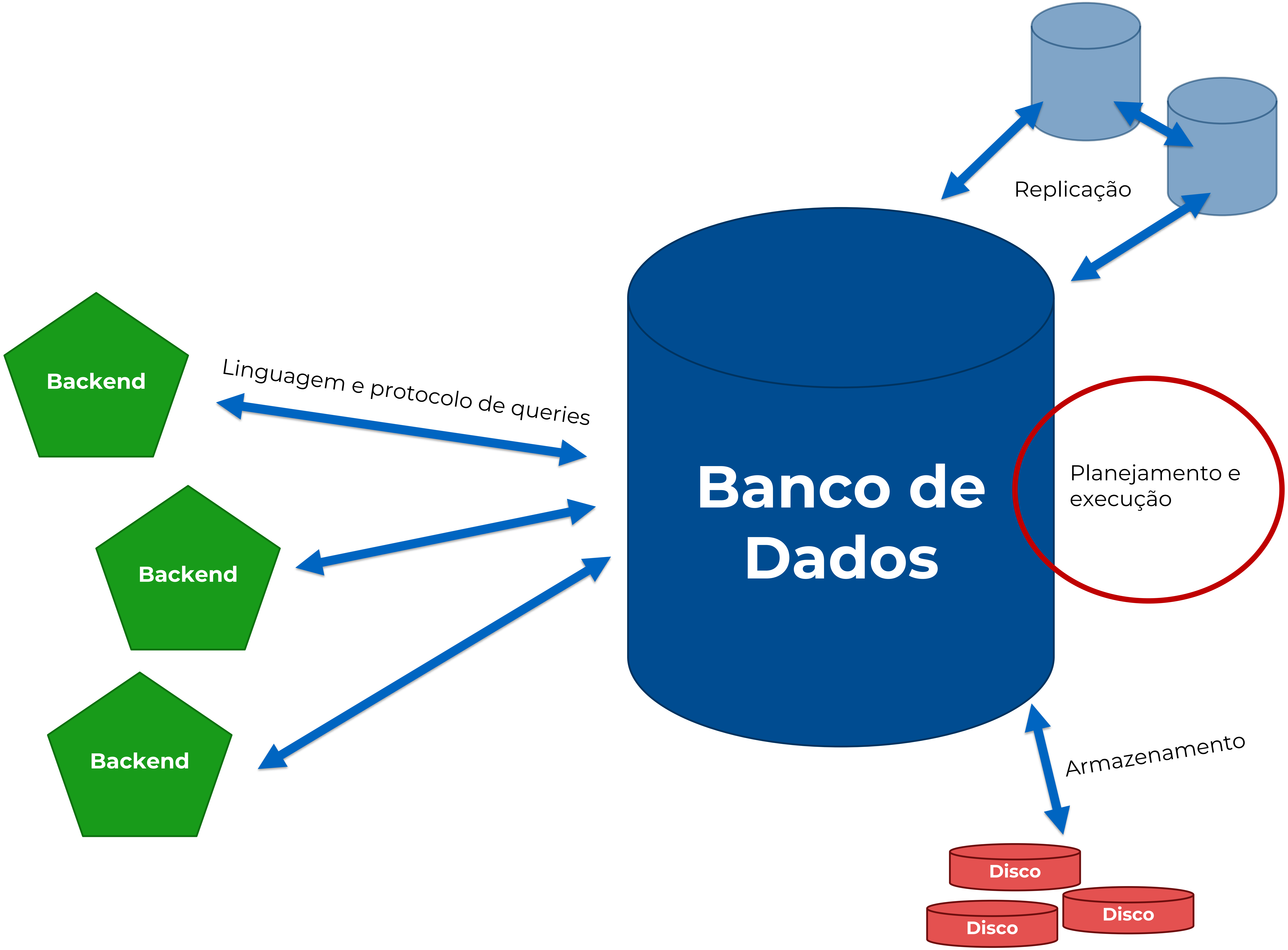
ervido
query
alquer

nhecim
xer + Pa

ecisa lic
om o p

da ban
is expr
nversac
nplistas





Replicação
Backend
Linguagem e protocolo de queries
Backend
Backend
Banco de Dados
Planejamento e execução
Armazenamento
Disco
Disco
Disco

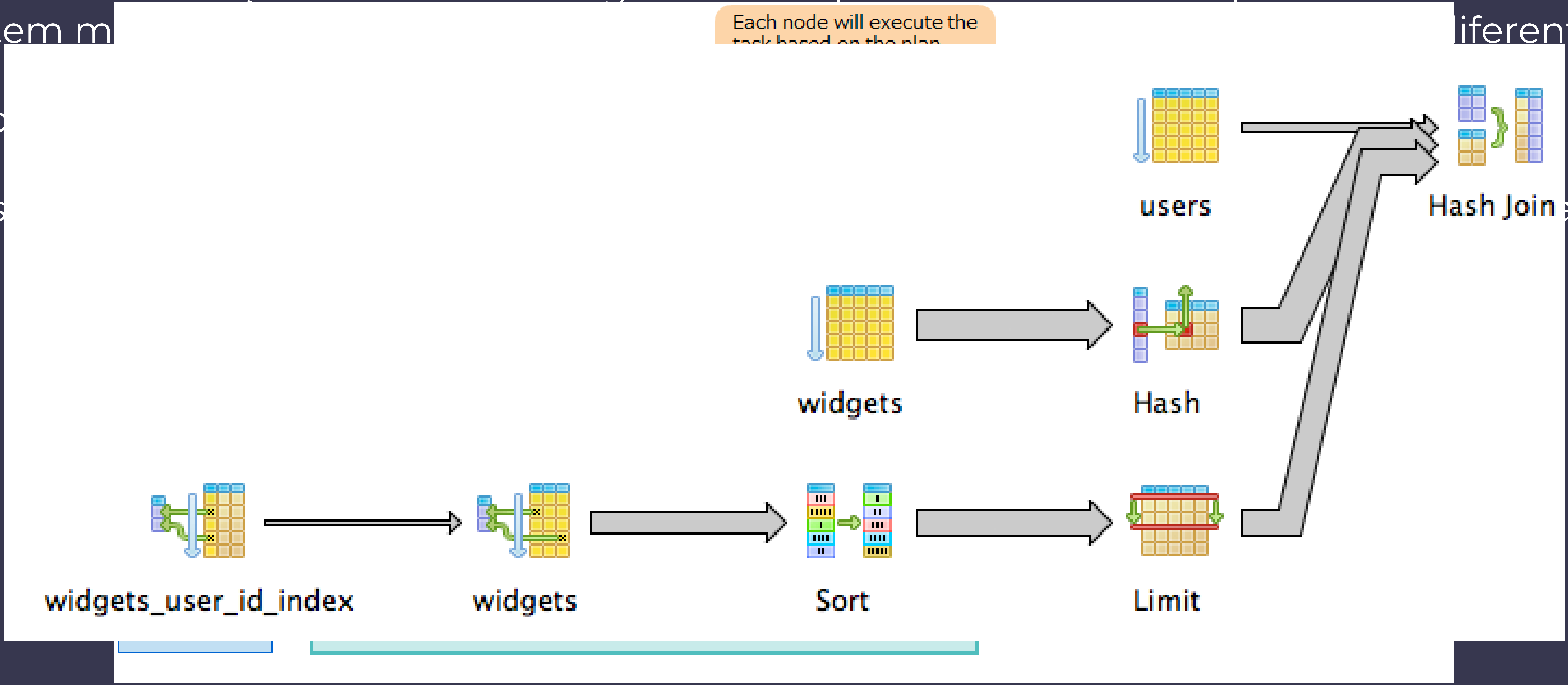
Planejamento e Execução

Planejamento e Execução

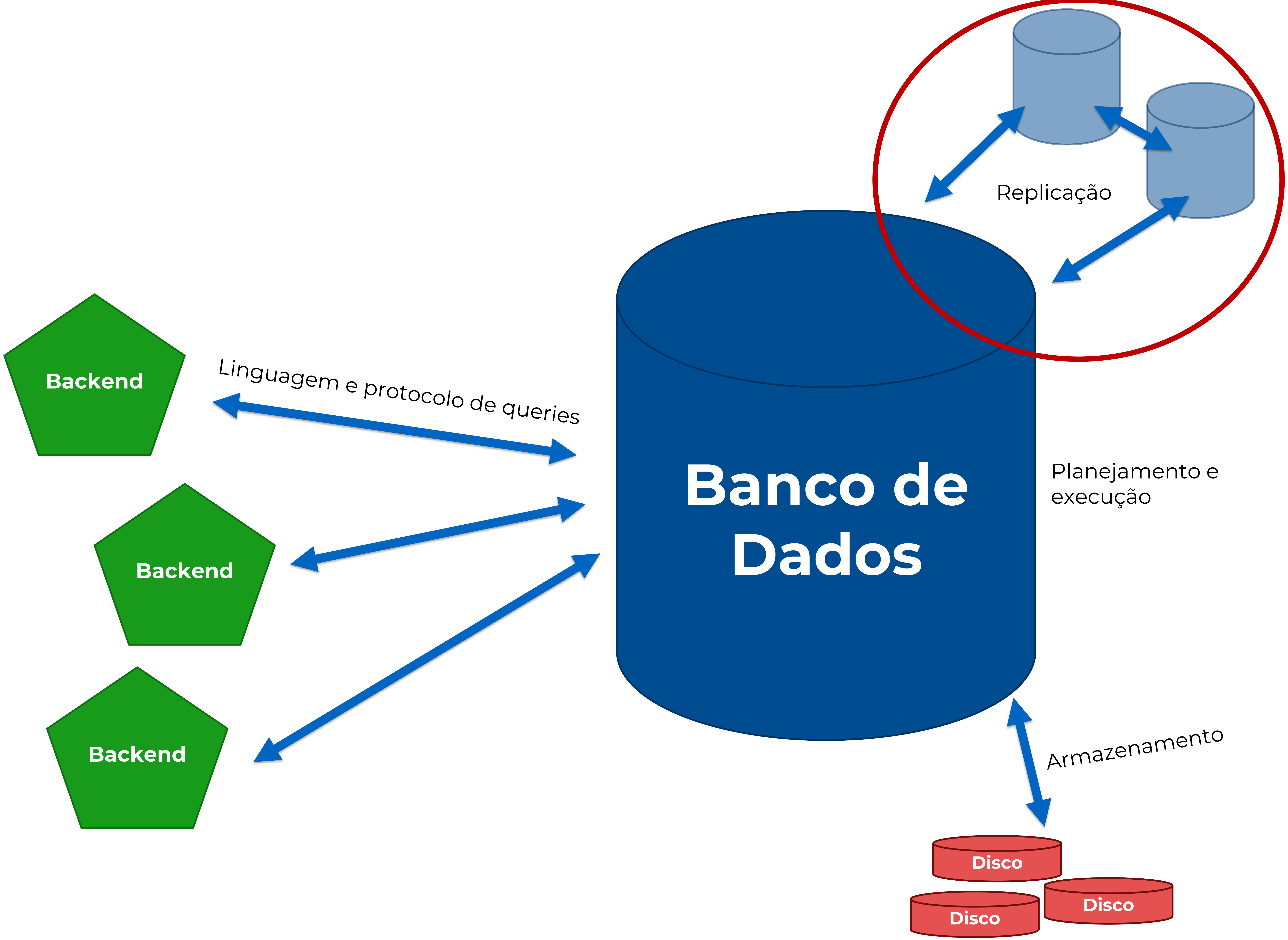
O plano de execução define a estratégia adotada pelo banco de dados para obter o resultado.
Existem m iferente.

Índic

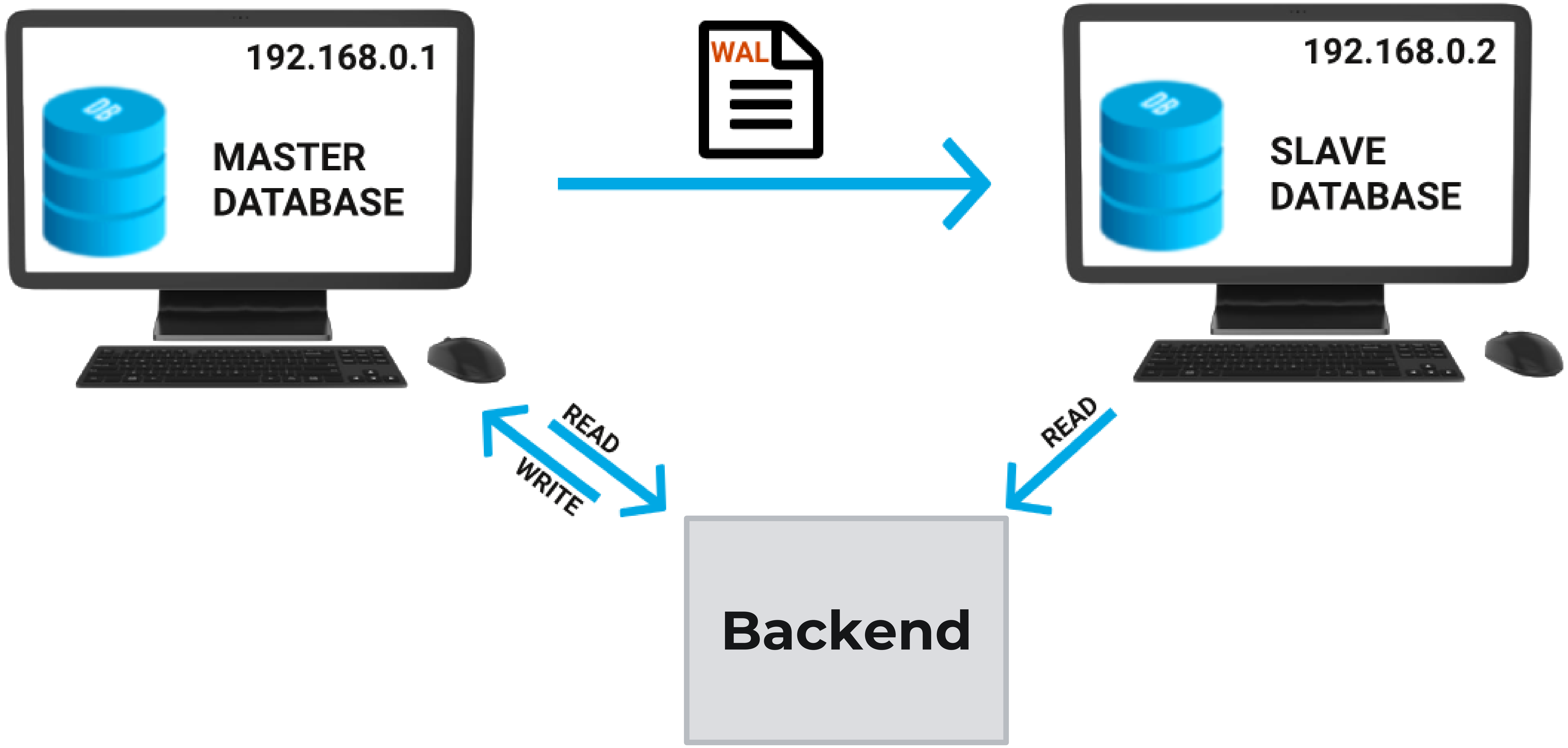
Em s a.



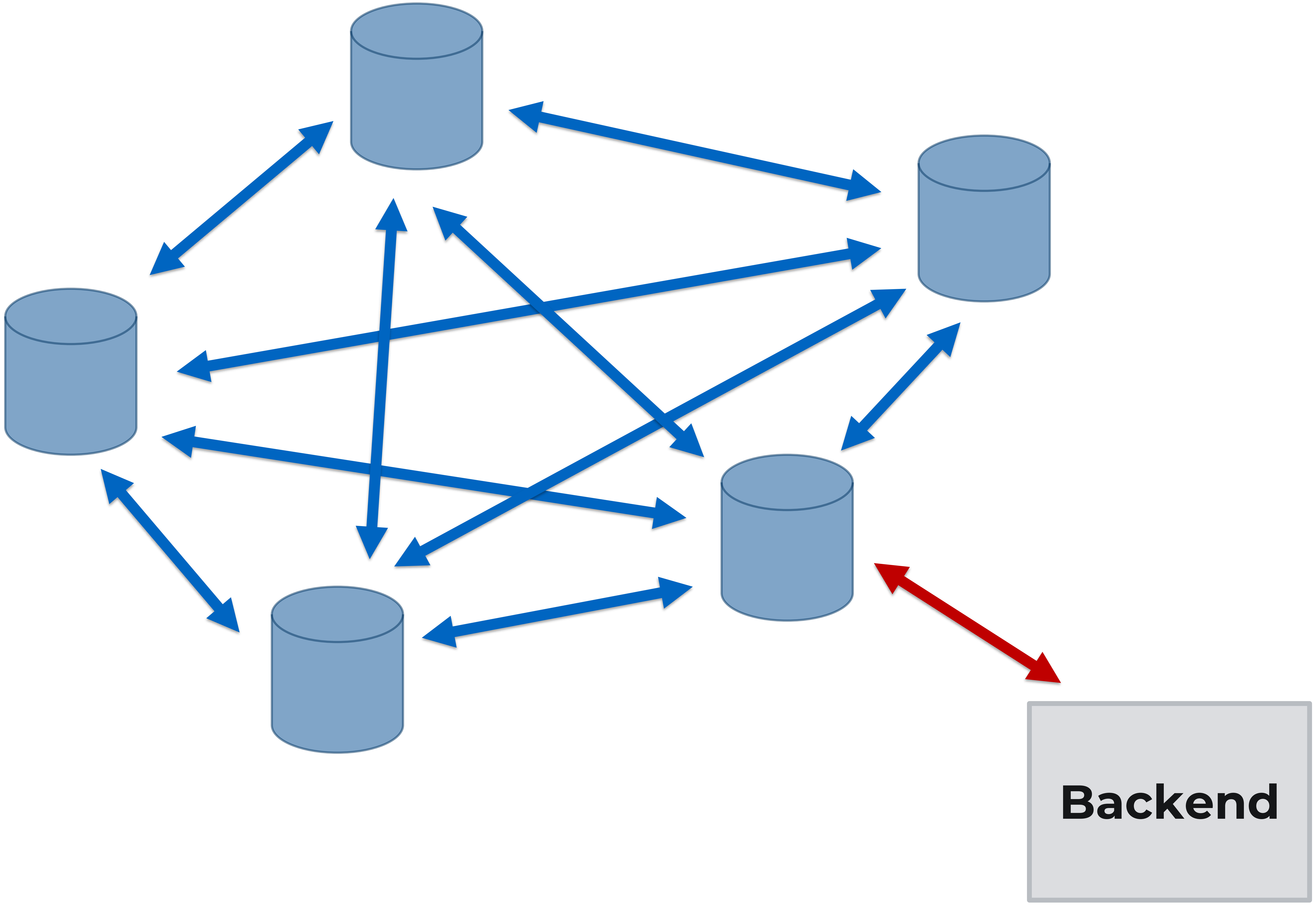
Each node will execute the task based on the plan


Replicação
Backend
Linguagem e protocolo de queries
Banco de Dados
Planejamento e execução
Backend
Backend
Armazenamento
Disco
Disco
Disco

Replicação

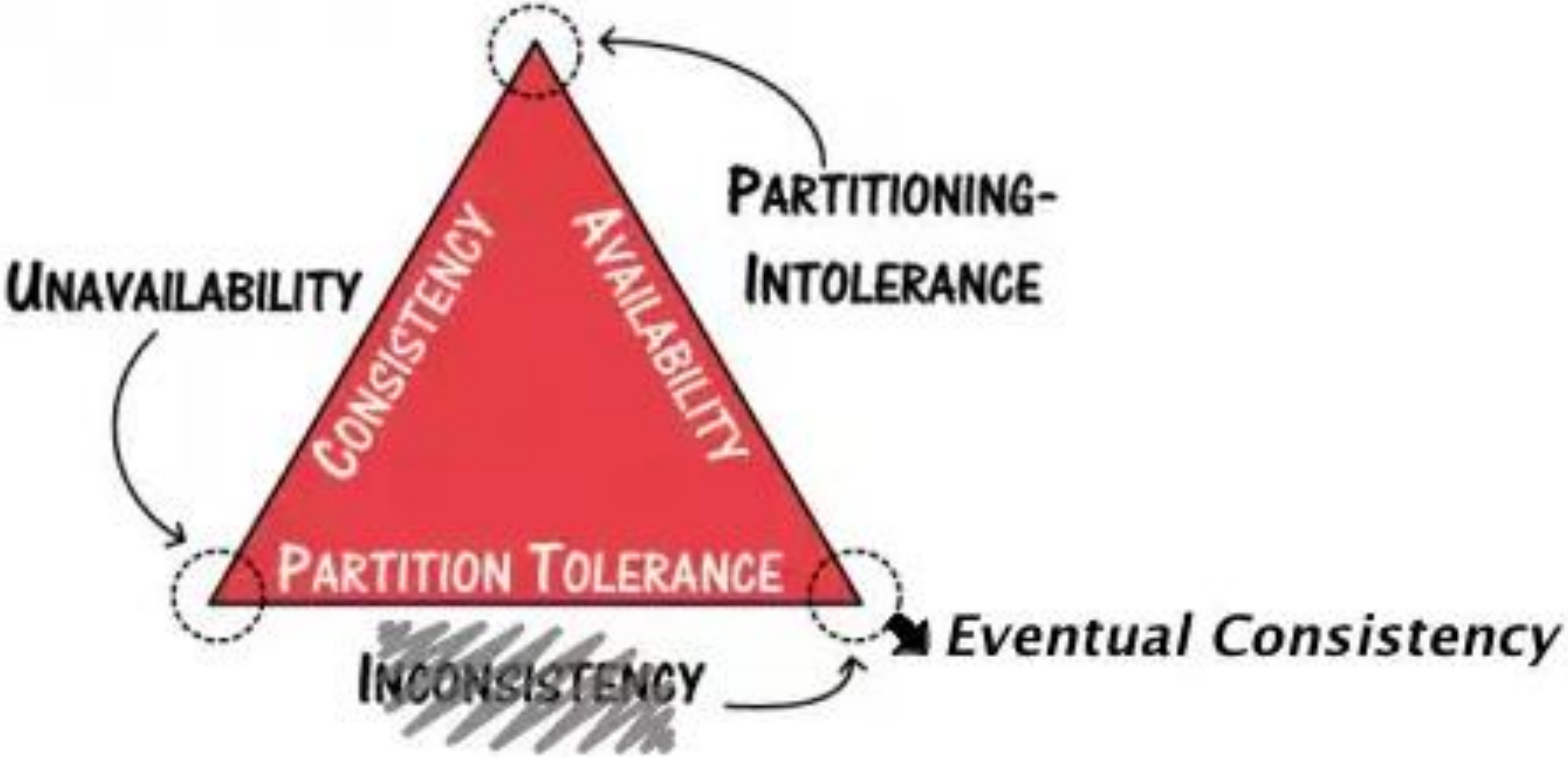


192.168.0.1
DB
MASTER
DATABASE
WAL
192.168.0.2
DB
SLAVE
DATABASE
READ
WRITE
READ
Backend

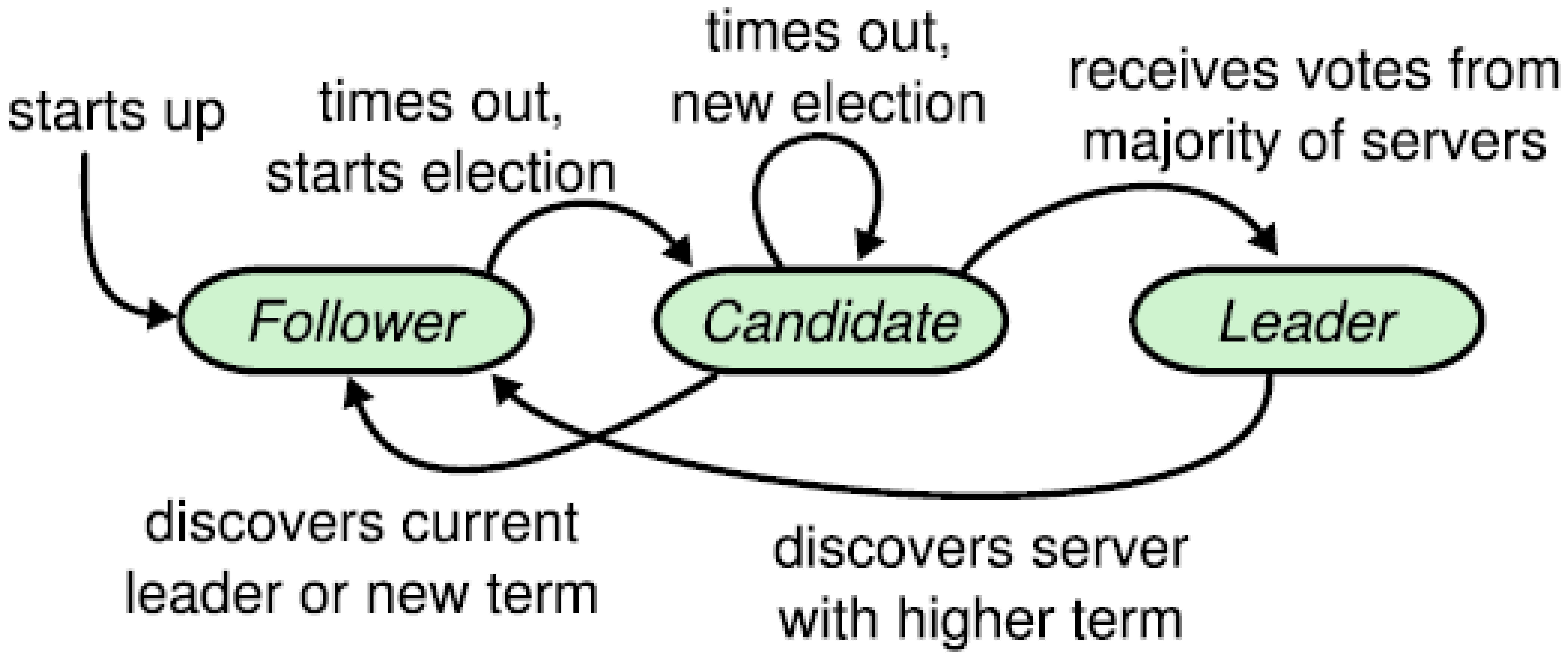


Consistência?

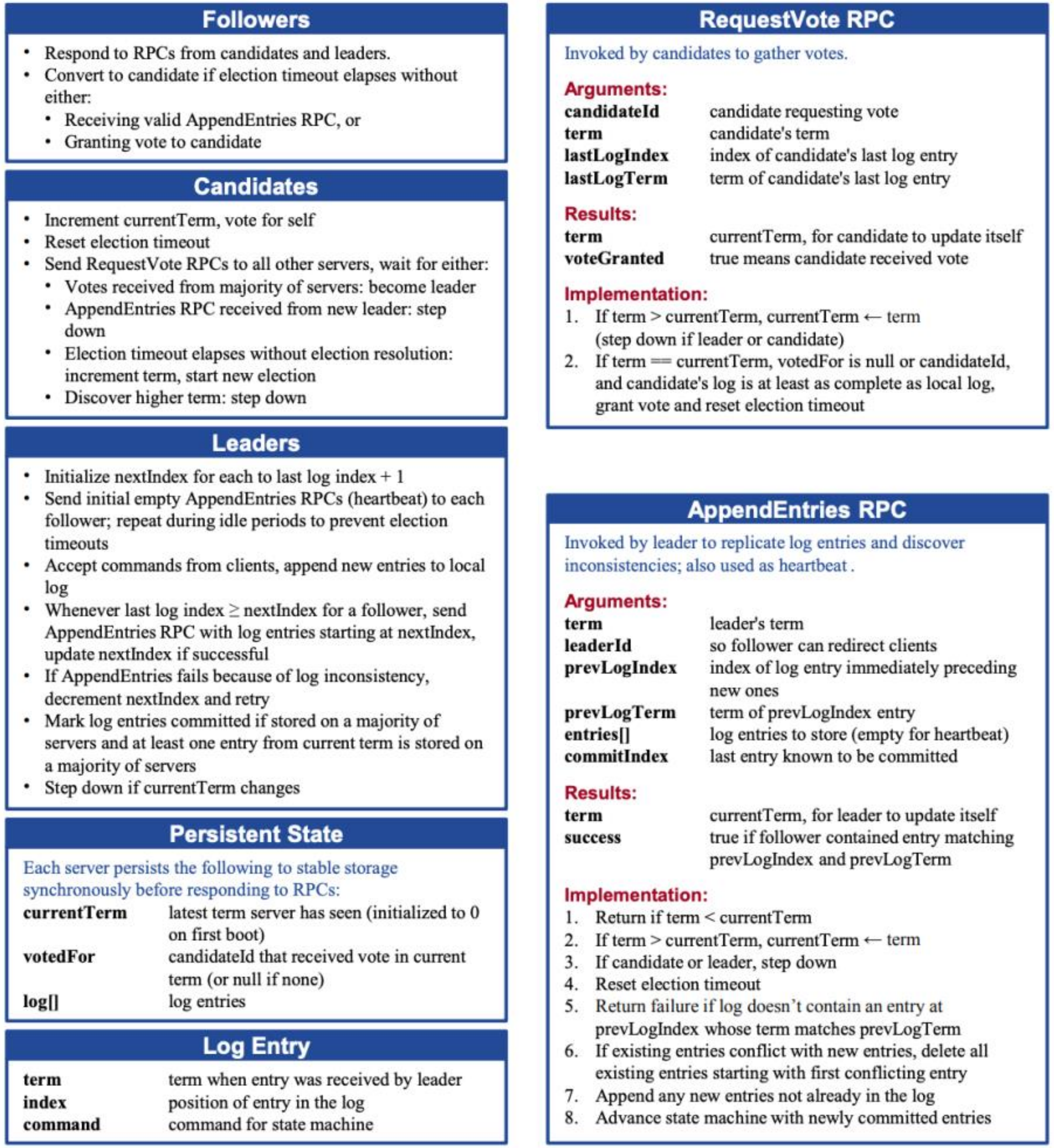
Muitos bancos de dados são “eventualmente consistentes”


Partitioning-
Intolerance
Unavailability
Consistency
Availability
Partition Tolerance
Inconsistency
Eventual Consistency

Raft!


starts up
times out,
starts election
times out,
new election
receives votes from
majority of servers
Follower
Candidate
Leader
discovers current
leader or new term
discovers server
with higher term

Raft Protocol Summary



Followers

- Respond to RPCs from candidates and leaders.
- Convert to candidate if election timeout elapses without either:
  - Receiving valid AppendEntries RPC, or
  - Granting vote to candidate

Candidates

- Increment currentTerm, vote for self
- Reset election timeout
- Send RequestVote RPCs to all other servers, wait for either:
  - Votes received from majority of servers: become leader
  - AppendEntries RPC received from new leader: step down
  - Election timeout elapses without election resolution: increment term, start new election
  - Discover higher term: step down

Leaders

- Initialize nextIndex for each to last log index + 1
- Send initial empty AppendEntries RPCs (heartbeat) to each follower; repeat during idle periods to prevent election timeouts
- Accept commands from clients, append new entries to local log
- Whenever last log index ≥ nextIndex for a follower, send AppendEntries RPC with log entries starting at nextIndex, update nextIndex if successful
- If AppendEntries fails because of log inconsistency, decrement nextIndex and retry
- Mark log entries committed if stored on a majority of servers and at least one entry from current term is stored on a majority of servers
- Step down if currentTerm changes

Persistent State

Each server persists the following to stable storage synchronously before responding to RPCs:

currentTerm	latest term server has seen (initialized to 0 on first boot)
votedFor	candidateId that received vote in current term (or null if none)
log[]	log entries

Log Entry

term	term when entry was received by leader
index	position of entry in the log
command	command for state machine

RequestVote RPC

Invoked by candidates to gather votes.

Arguments:

candidateId	candidate requesting vote
term	candidate's term
lastLogIndex	index of candidate's last log entry
lastLogTerm	term of candidate's last log entry

Results:

term	currentTerm, for candidate to update itself
voteGranted	true means candidate received vote

Implementation:

1. If term > currentTerm, currentTerm ← term (step down if leader or candidate)
2. If term == currentTerm, votedFor is null or candidateId, and candidate's log is at least as complete as local log, grant vote and reset election timeout

AppendEntries RPC

Invoked by leader to replicate log entries and discover inconsistencies; also used as heartbeat .

Arguments:

term	leader's term
leaderId	so follower can redirect clients
prevLogIndex	index of log entry immediately preceding new ones
prevLogTerm	term of prevLogIndex entry
entries[]	log entries to store (empty for heartbeat)
commitIndex	last entry known to be committed

Results:

term	currentTerm, for leader to update itself
success	true if follower contained entry matching prevLogIndex and prevLogTerm

Implementation:

1. Return if term < currentTerm
2. If term > currentTerm, currentTerm ← term
3. If candidate or leader, step down
4. Reset election timeout
5. Return failure if log doesn't contain an entry at prevLogIndex whose term matches prevLogTerm
6. If existing entries conflict with new entries, delete all existing entries starting with first conflicting entry
7. Append any new entries not already in the log
8. Advance state machine with newly committed entries

Outros assuntos

- Transações
- Indexação
- Alterações de schema online
- Sharding
- Relações
- Balanceamento de carga

https://github.com/lbguilherme/rethinkdb-lite

Search or jump to...
Pull requests
Issues
Marketplace
Explore
Unwatch 4
Star 41
Fork 3
Code
Issues
Pull requests
Actions
Projects
Wiki
Security 1
Insights
Settings
master
Go to file
Add file
Code

lbguilherme / rethinkdb-lite

2 branches 0 tags

lbguilherme fix: disable SimplifyVariablesTransformer 1fb3af1 on 21 Jul 460 commits

.github/workflows	fix: Upgrade to crystal 0.35.1	2 months ago
spec	Create AST transformers. Here they are used to convert group....ungro...	6 months ago
src	fix: disable SimplifyVariablesTransformer	2 months ago
vendor/rethinkdb-webui	use npm lockfile on webui	9 months ago
.dockerignore	Add Dockerfile with a release build	9 months ago
.editorconfig	Initial commit	3 years ago
.gitignore	implement table.index_list	9 months ago
API_STATUS.md	Add db_list and table_list	7 months ago
CHANGELOG.md	Upgrade to Crystal 0.33.0	7 months ago
Dockerfile	fix: Upgrade to crystal 0.35.1	2 months ago
LICENSE	Initial commit	3 years ago
README.md	fix: Upgrade to crystal 0.35.1	2 months ago
coverage.sh	add coverage script based on kcov	9 months ago
shard.lock	fix: upgrade for Crystal 0.35.1	2 months ago
shard.yml	fix: Upgrade to crystal 0.35.1	2 months ago

README.md

crystal 0.35.1

RethinkDB-lite

This is a personal project aiming at reimplementing everything RethinkDB currently does. At the same time, it is also a driver capable of connecting to a database and sending queries.

First use case: Database driver

You can connect to a running RethinkDB instance and send queries. Methods are pretty much equal to the official

About

A RethinkDB-compatible database written in Crystal

crystal rethinkdb database rocksdb

Readme

MIT License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

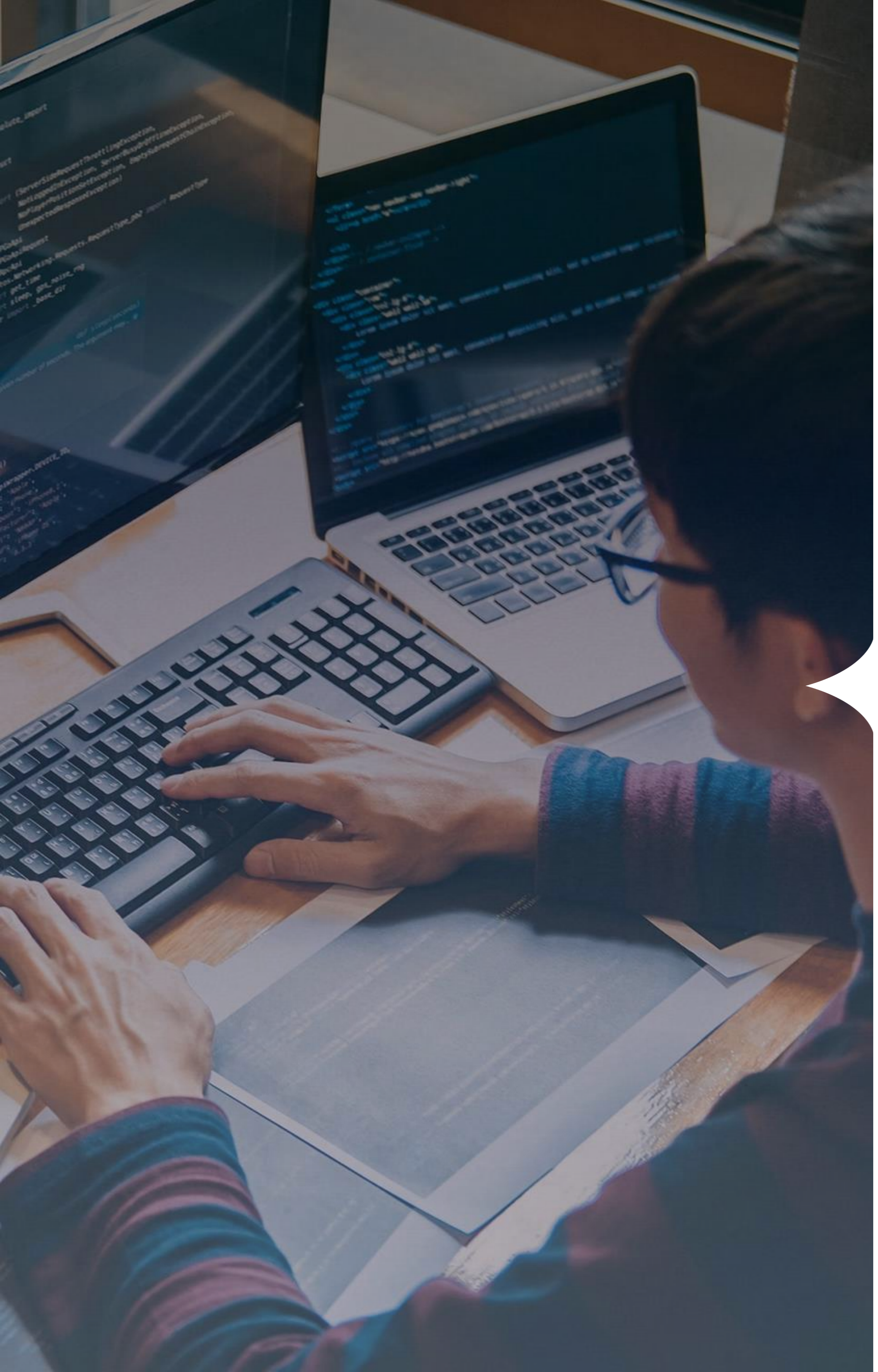
lbguilherme Guilherme Bernal
joshuapassos Joshua Passos

Languages

● Crystal 99.4% ● Other 0.6%

cubos

//academy//



Cursos de Extensão

PARA QUEM JÁ ESTÁ NA ÁREA DE TECNOLOGIA

Os cursos de extensão funcionam para quem já atua na área e deseja aprofundar seus conhecimentos em algum tópico.

As aulas acontecem com uma frequência menor, mas possuem um conhecimento técnico mais profundo.

https://cubos.academy/

- Cursos mais detalhistas e de profundeza técnica
- Geralmente noturno, 2 vezes na semana ou aos sábados
- Aulas online ao vivo
- Turmas menores, maior valor agregado

Guilherme Bernal
CTO na Cubos
guilherme@cubos.io

cubos

Estamos contratando!
jobs.cubos.io

cubos.io f in