Open Source Relational Databases

Open Source Relational Databases

Open Source Relational Databases

by @3manuek

Who and what is about?

- Emanuel Calvo, currently at OnGres as a PostgreSQL Consultant and ayres.io as _*root*_.
- Working on Modern techniques for DBRE.
- What is the current status of the Open Source SQL databases per component?
- What's the good, the bad and the ugly in the market?

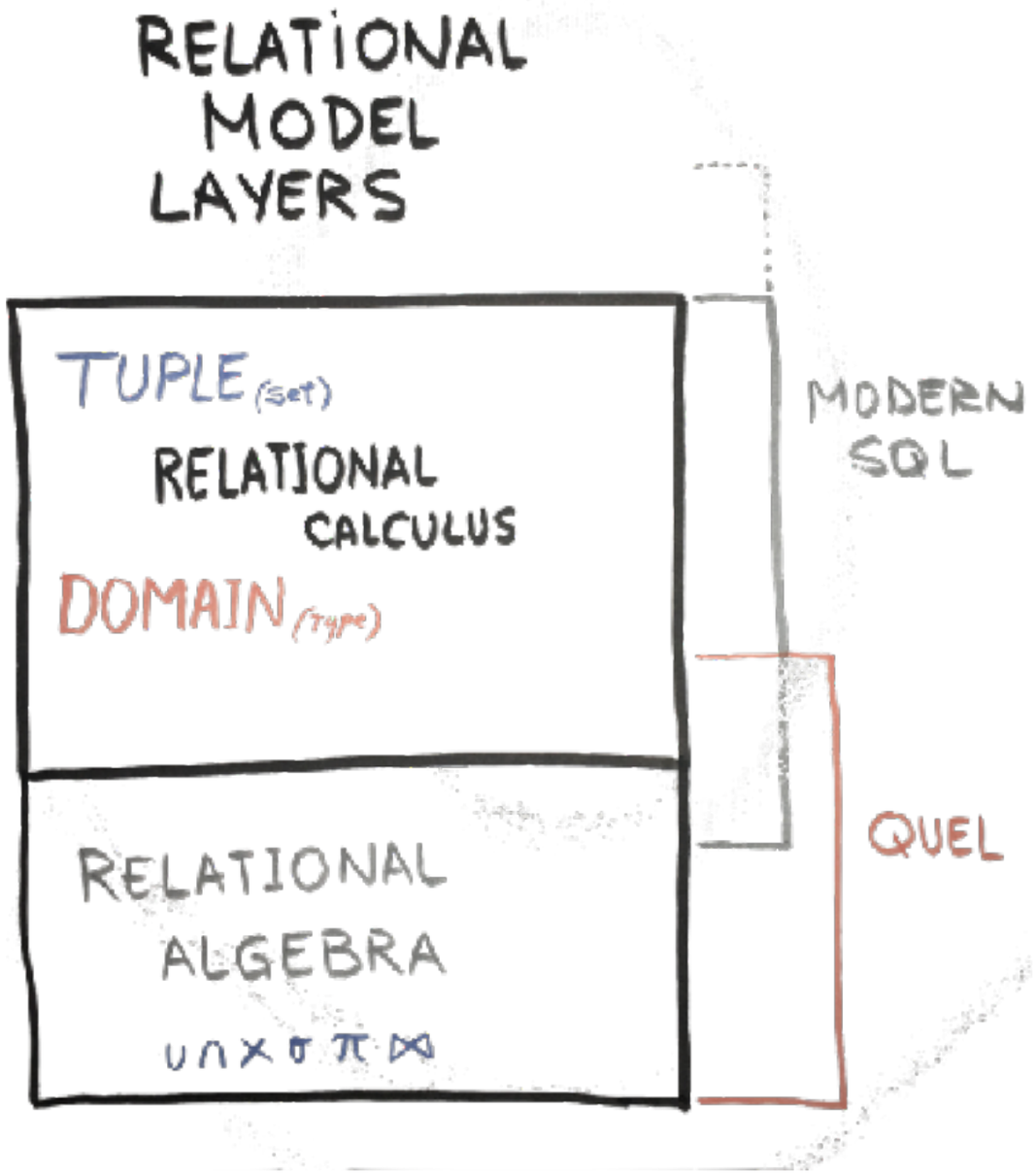
ER model

Entity-Relationship and why SQL isn’t considered so.

At least in its pure state.

The ER Map

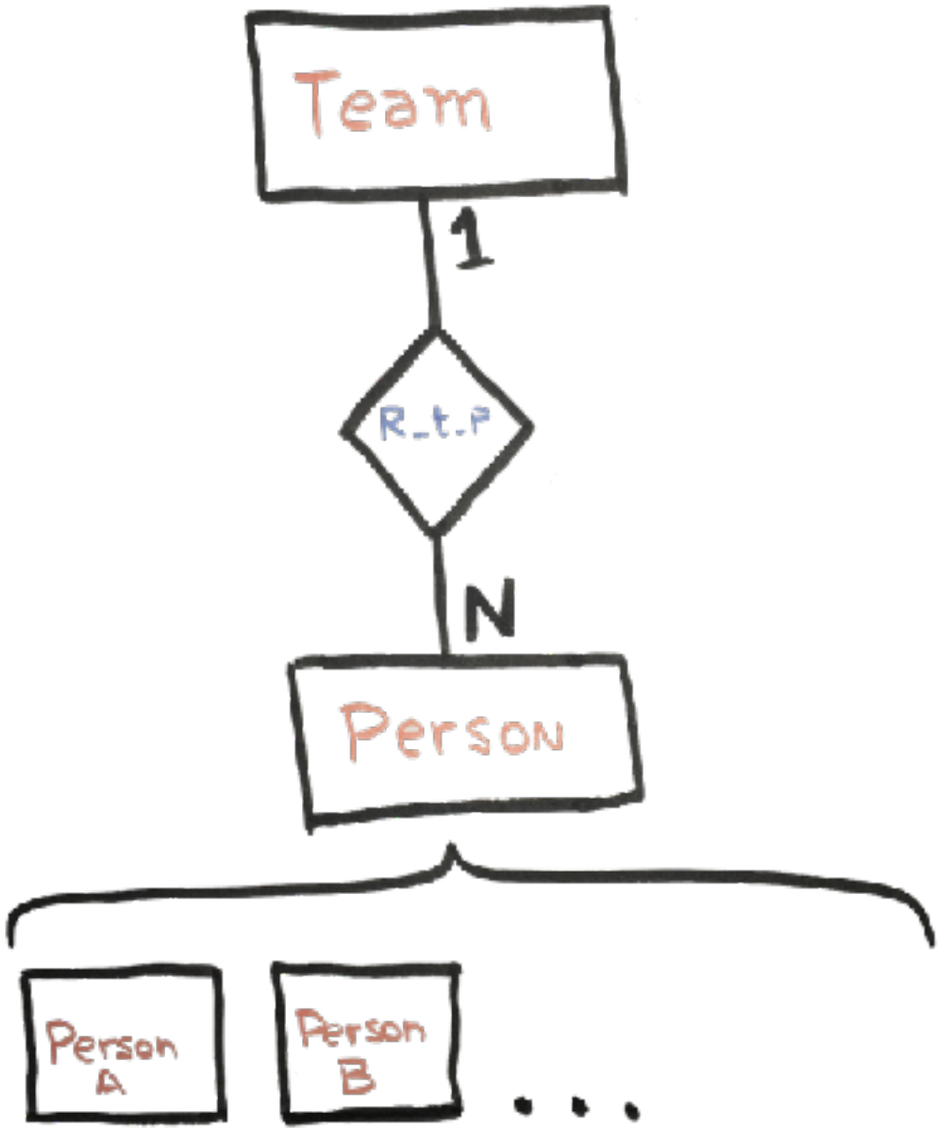
- Needs a First-Order logic language for retrieving data.
- Relational Algebra
- Tuple and Domain Relational Calculus.



The model example

- Obscures everything behind the complexity of the storage.
- It is represented as relational algebra, but is hidden from you.
- How to select the names of the people of "Black" team?

$$\Pi_{\substack{FName \\ LName}}\left(Person \bowtie_{Id=IdPerson} (R\_t\_p) \bowtie_{\substack{Id_{team}= \\ Id}} \sigma_{TName='Black'}(Team)\right)$$



Some SQL:2011 tangent distinctions

- Support NULLs
- Support SubQueries
- Column precedence affects (horizontal alignment) depending on the engine
- SQL/MED

- Is a declarative language
- Hides all the complexity of the executions to the end user
- Planners were very advanced already.

The Transaction Model

Concurrency, consistency and availability.

The Entity Consistency

- CAP Theorem (Consistency, Availability and Partition Tolerance). PACELC adds to choose between [L]atency and [C]onsistency.
- ACID (Atomicity, **Consistency**, Isolation and Durability)
- BASE (Basically Available, Soft State, **Eventual consistency**)

The chosen

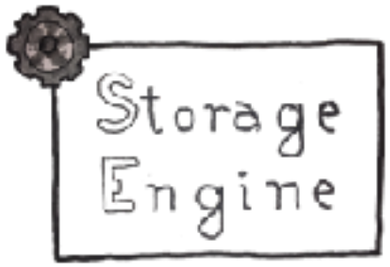
We grab them by the storage and use them wisely without paying money to Oracle.

- CockroachDB
- PostgreSQL
- MySQL / MariaDB
- Clickhouse
- MongoDB

Components

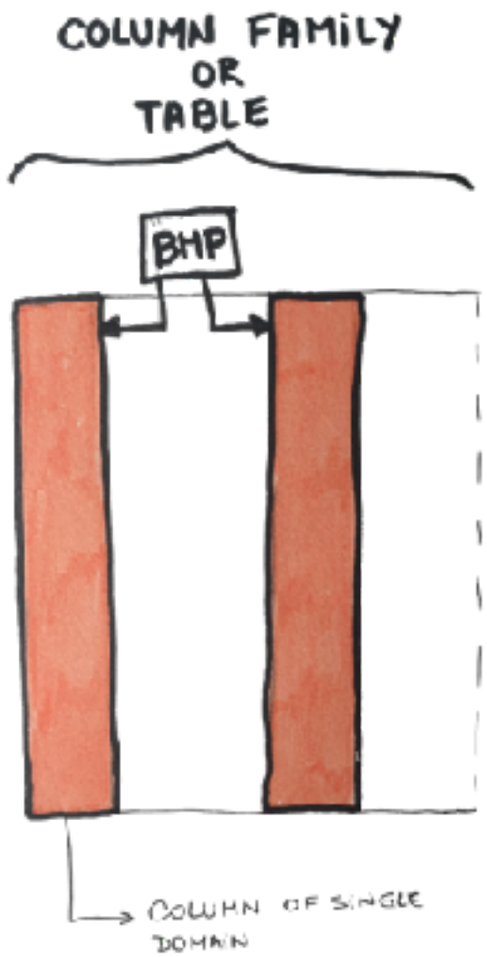
The Lego

- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

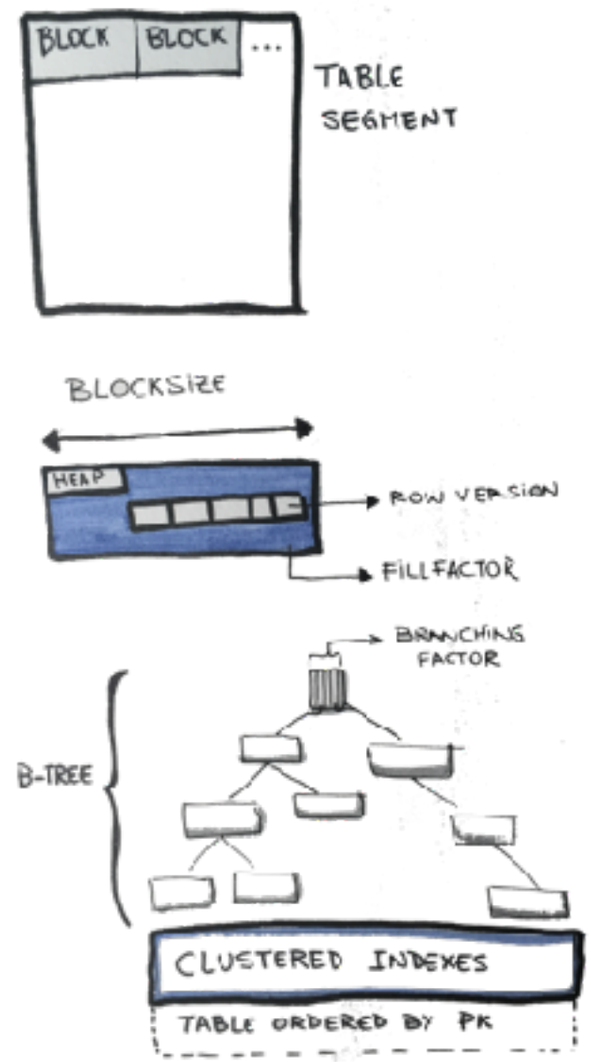


- Buffer Management
- IO method (Direct/io, fsync)
- Transaction Management (storage layer)
- Point in Time Recovery and Undo Log
- For distributed engines you want to read Jepsen tests.
- **Is the sauce**

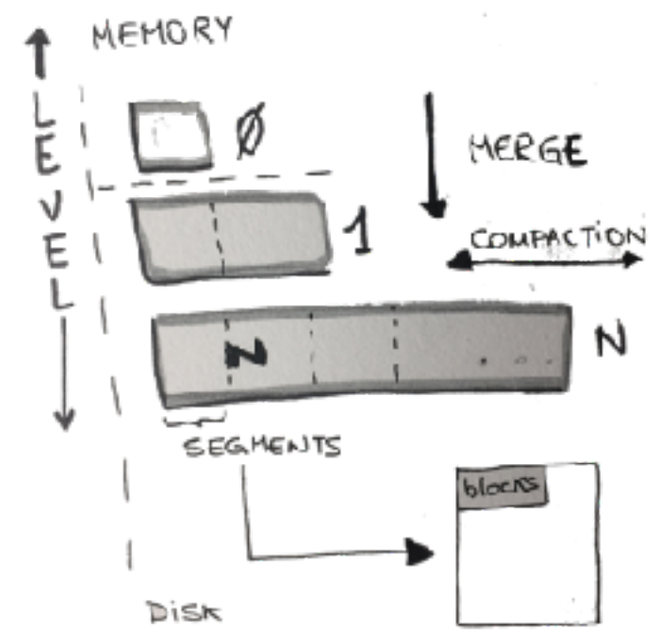
Wide-range Storage Engine Map



Columnar Based

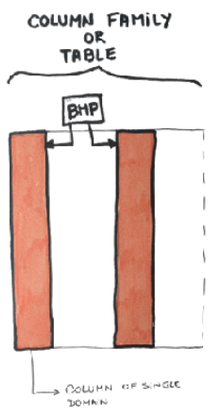


Tuple Based

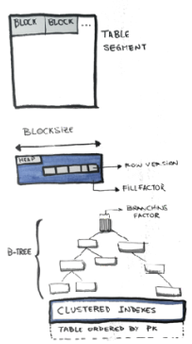


Leveled Structured Map Tree

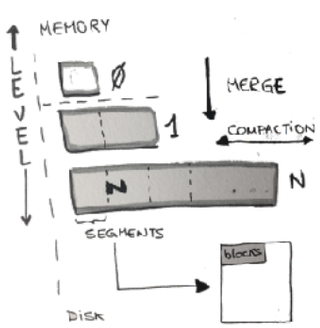
Quick cherry pick



- Fast for aggregations
- Easy for parallelization
- Better compression due to ColBased
- Better to scale massive amount of data



- Better for concurrency
- Hard to scale
- Better when manipulating entities atomically
- Balance between performance and concurrency.



- Bloom filters
- Sparse indexes by design
- Avoid Write Amplification
- Index-based storage
- More disk efficient, more CPU

“Relational databases require a Query Optimizer/ Query Planner for translating the first-order logic language to relational algebra and other optimizations. The result is called *Execution Plan*.”

– Jorge de Lanús Oeste (maneja Uber pero sabe mucho de Bases de Datos)

- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

- Heuristic
- Cost based {Parametric, MO, MOP}
- Mixed

- Planner, Resolver, Opmitizer, Executor

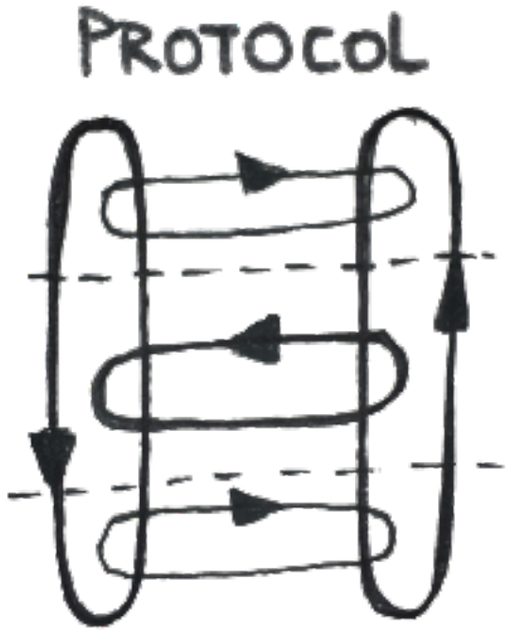
- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

- Heuristic
- Cost based {Parametric, MO, MOP}
- Mixed

- Planner, Resolver, Opmitizer, Executor

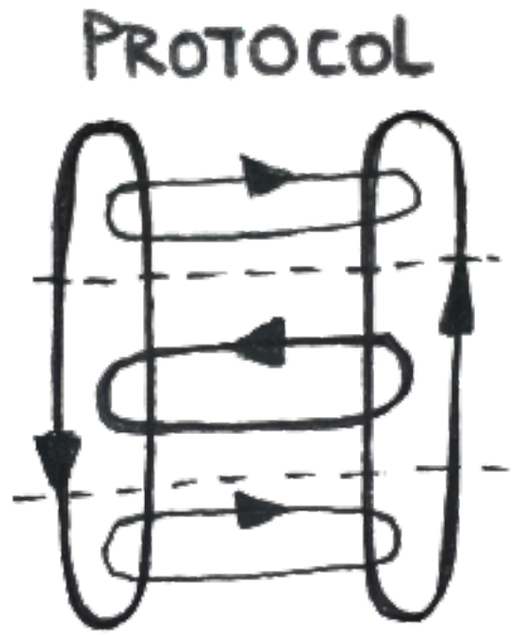
- MySQL has also Condition Pushdown
- PostgreSQL has a rich planner
- MySQL plan information lacks of information
- PostgreSQL does not provide additional tools for plan reading.

- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.



- Client Protocol
- Replication Protocol
  - Logical/Binary
- Coordination Protocol
- HA protocol
- Gossip
- Consensus {RAFT, Paxos}
- ...

- Storage Engine
- Planner
- **Protocol**
- Language
- Ecosystem
- Framework
- WAL
- Source Code availability, documentation both user and internal, community, etc.



- Client Protocol
- Replication Protocol
- Coordination Protocol
- HA protocol
- Gossip
- Consensus {RAFT, paxos}
- ...

- No standard
- JSON is becoming more present (thankfully)
- Absence of internal consensus

- Storage Engine
- Planner
- Protocol
- Language

- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

- Abstract all relation algebra
- SQL != Relational
  - NULLs
  - Column Alignment
  - Subquery
  - Mixed implementations
  - Relational is conceptually unable to return more than 1 result set.

- Storage Engine
- Planner
- Protocol
- Language

- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

- Abstract all relation algebra
- SQL != Relational
  - NULLs
  - Column Alignment
  - Subquery
  - Mixed implementations
  - Relational is conceptually unable to return more than 1 result set.

- Standard
- Backward Compatibility
- Modern

“Postgres original implementation was in QUEL and its organization resembles to many of the concepts of the original ER model. COPY is a inherited piece from this prior implementation.”

Postgres95 -> PostgreSQL

- Storage Engine
- Planner
- Protocol
- Language
- **Ecosystem**
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

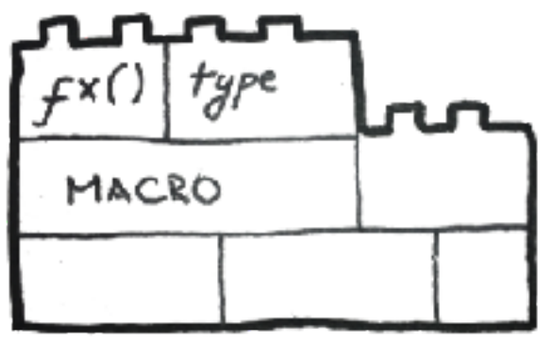
- Single Provider or *fake* Open source
- Community contribution or *Social Entropy Experiment*
- Satellite companies building tools
- Satellite companies building forks
- Satellite coders copy pasting
- Tons of under-proven libraries

- Storage Engine
- Planner
- Protocol
- Language
- **Ecosystem**
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

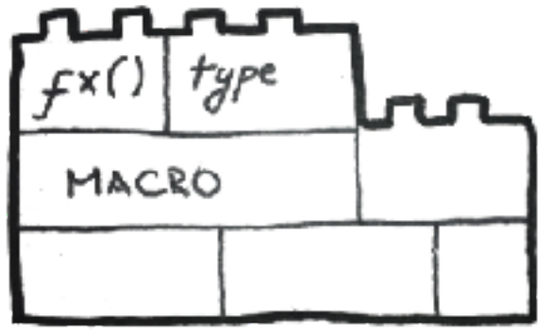
- Multi-database tools tend to fail awesomely
- Choose tools that are integrated with the core and that have frequent updates
- Bug fixing tied to community times
  - bugs.mysql.com
  - Postgres uses mailing list
  - Clickhouse/Cockroach use GH

- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- **Framework**
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

- Core extensibility plugins or extensions
- Customize Planner
- Manage protocol
- Creating workers
- Creating own types



- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- **Framework**
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

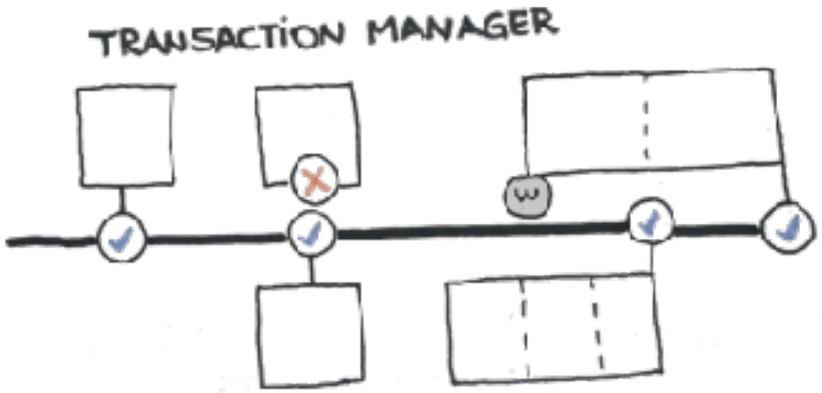


- Complex, generally in C.
- Multi-provider packages.

- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.

- WAL or Redo
- MySQL has undo log, but only for rollback space.
- Postgres has extensions for rewind (pg_rewind)
- It can reside on the Storage Engine or higher layers
- It’s local and provides consistency and durability
- Distributed WALs or Certification log could be in this group, although there will be always a WAL.

- Storage Engine
- Planner
- Protocol
- Language
- Ecosystem
- Framework
- WAL
- Transaction Manager
- Source Code availability, documentation both user and internal, community, etc.



- It can be at node level or cluster level
- Concept of source and origin
  - Group Replication
  - Logical Replication
- Concept of Global Id
- Centralized Commits are possible through Kafka brokers
- Functional sharing must relay on node try level
- Serializable only supported by Postgres
- Uncommitted only supported by InnoDB

Other components or capabilities

- Access Methods (B-Tree, L-Tree, Reverse, Hash)
- FTS (Full Text Search) and advanced search
- Geo capabilities

Entity Consistency at Scale

Replication, Sharding and HA.

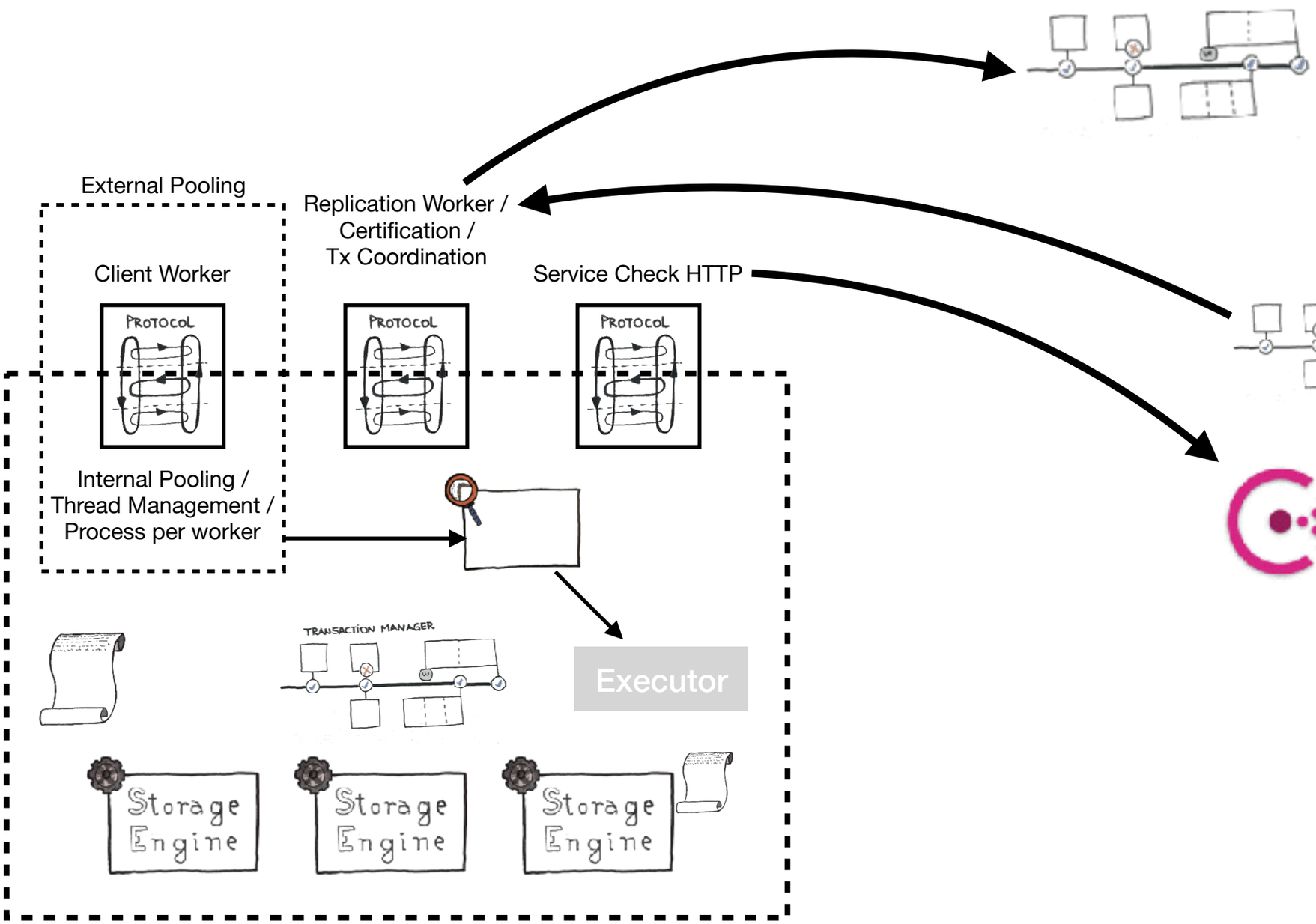
What is in the land of single leader engines?

- Async
- Semi-synchronous replication
  - First node response, as in MySQL.
- Simple Synchronous replication
- Quorum Synchronous
  - Postgres

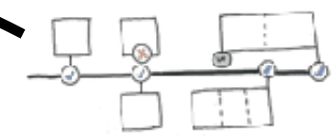
What is in the land of distributed/ multileader[less] engines?

- Asynchronous Multi Leader replication
  - BDR
- Snapshot Isolation
  - Galera (MySQL layer on top InnoDB)
- Serializability
  - CockroachDB (2PC to a consensus group, with Hybrid Logical Clock, *not strict serial*)
  - VoltDB
- External consistency
  - Google Spanner (through True Time clocks).

The [full] architecture



- Write Quorum
  - Single Leader
  - Multi Leader
- Group Replication
- Inter node coordination
- Distributed Transactions
- Conflict-Free Replicated Datatype (LWW, 2PC set, etc)



- Centralized Commit

- Consensus for HA
- Also in the entry points if external

The status of horizontal scalability in OSDBs

- Non native support for distributed consensus.
- Only MySQL has Global identifiers and recently supported Group Replication.
- There are extensions/forks for providing sharding in Postgres and MySQL.

SandBox

- https://gitlab.com/3manuek/HA_PoC
- https://gitlab.com/ongresinc/testing-pg-ha-solutions/merge_requests/1

References

- Designing Data-Intensive Applications (Martin Kleppmann)
- Database Reliability Engineering (L. Campbell/C. Majors)

Thank you!

@3manuek
3manuek [at] gmail {dot} com