

Переосмысление Picodata в качестве cluster-first СУБД

Ярослав Дынников

Picodata



**Saint
HighLoad++**

Слайды: <https://rosik.github.io/2024-shl>

План доклада

1. Питч
2. Архитектура и алгоритмы
3. Конкурентные отличия
4. Расширение функциональности

Picodata — это

- Distributed SQL

Picodata — это

- Distributed SQL
- Sharding, replication

Picodata — это

- Distributed SQL
- Sharding, replication
- In-memory

Picodata — это

- Distributed SQL
- Sharding, replication
- In-memory, single-threaded

Picodata — это

- Distributed SQL
- Sharding, replication
- In-memory, single-threaded
- Горизонтальное масштабирование

Picodata — это

- Distributed SQL
- Sharding, replication
- In-memory, single-threaded
- Горизонтальное масштабирование
- ...



Маленькие, быстрые данные

Катакомбы Picodata

Обзорная экскурсия



**Saint
HighLoad++**

Что было до

Что было до

Tarantool

In-memory СУБД и сервер приложений на Lua

Get your data in RAM. Get compute close to data. Enjoy the performance

Что было до

Tarantool

In-memory СУБД и сервер приложений на Lua
Get your data in RAM. Get compute close to data. Enjoy the performance

Vshard

Модуль шардирования на основе виртуальных бакетов

Что было до

Tarantool

In-memory СУБД и сервер приложений на Lua
Get your data in RAM. Get compute close to data. Enjoy the performance

Vshard

Модуль шардирования на основе виртуальных бакетов

Cartridge

Фреймворк для разработки распределенных приложений

Особенности экосистемы

Performance

Быстро, но не всегда предсказуемо (LuaJIT, GC)

Особенности экосистемы

Performance

Быстро, но не всегда предсказуемо (LuaJIT, GC)

Функциональные

SQL есть, но в рамках одного узла

Особенности экосистемы

Performance

Быстро, но не всегда предсказуемо (LuaJIT, GC)

Функциональные

SQL есть, но в рамках одного узла

Разработка

Очень интересно, но сложно

Особенности экосистемы

Performance

Быстро, но не всегда предсказуемо (LuaJIT, GC)

Функциональные

SQL есть, но в рамках одного узла

Разработка

Очень интересно, но сложно

Эксплуатация

То слишком гибко, то слишком жестко (two-phase commit)

План

1. Заменить two-phase commit на Raft
2. Реализовать распределенный SQL
3. И все это на Rust
4. ???
5. Profit

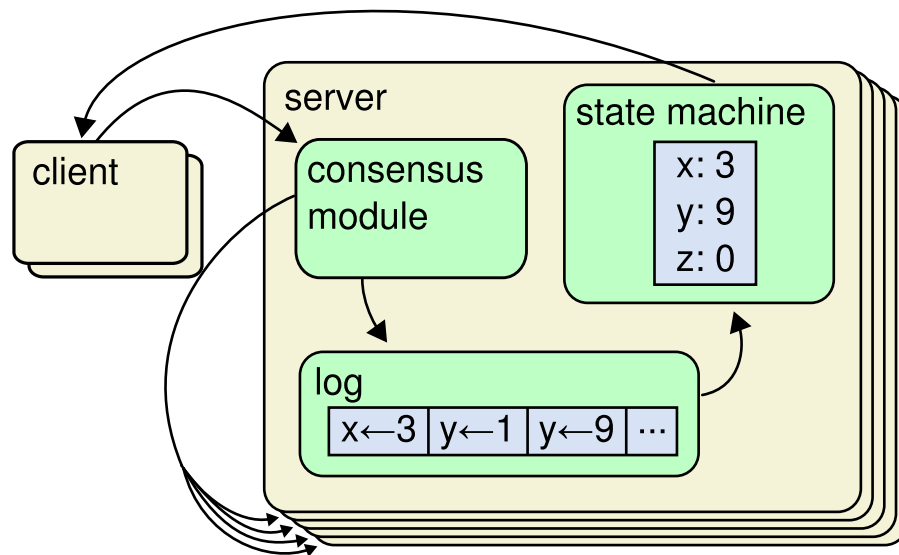
Управление кластером



**Saint
HighLoad++**

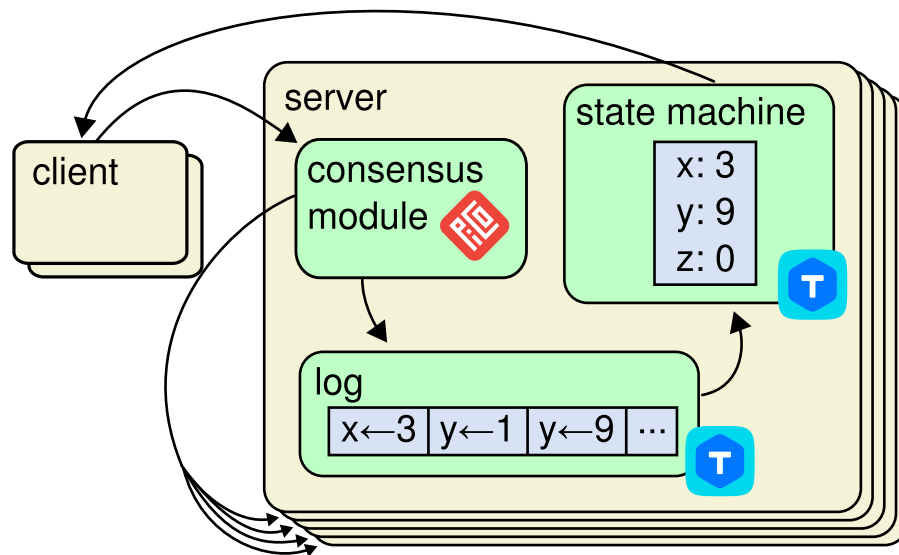
Raft

Алгоритм для решения задач консенсуса
в сети ненадежных вычислений

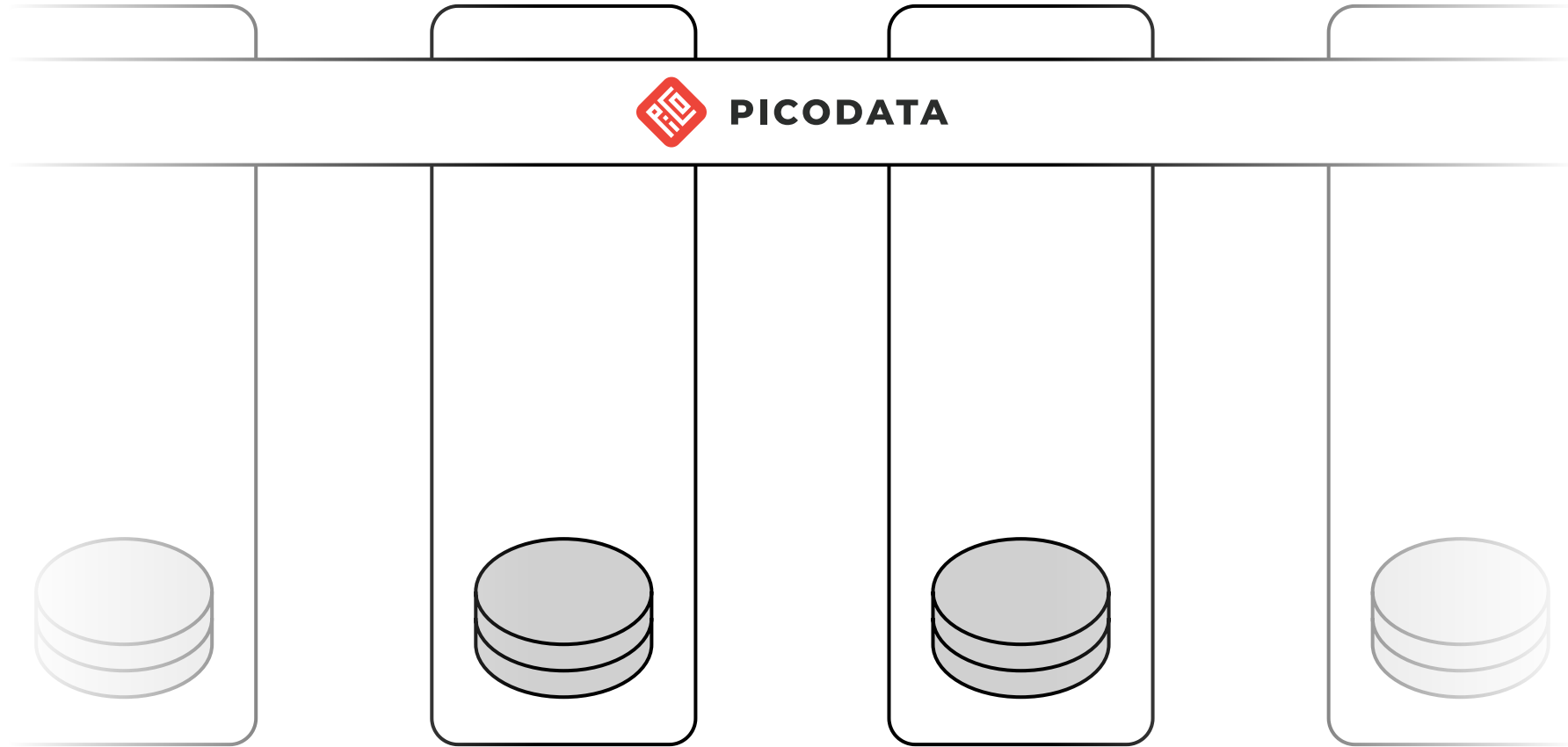


Raft

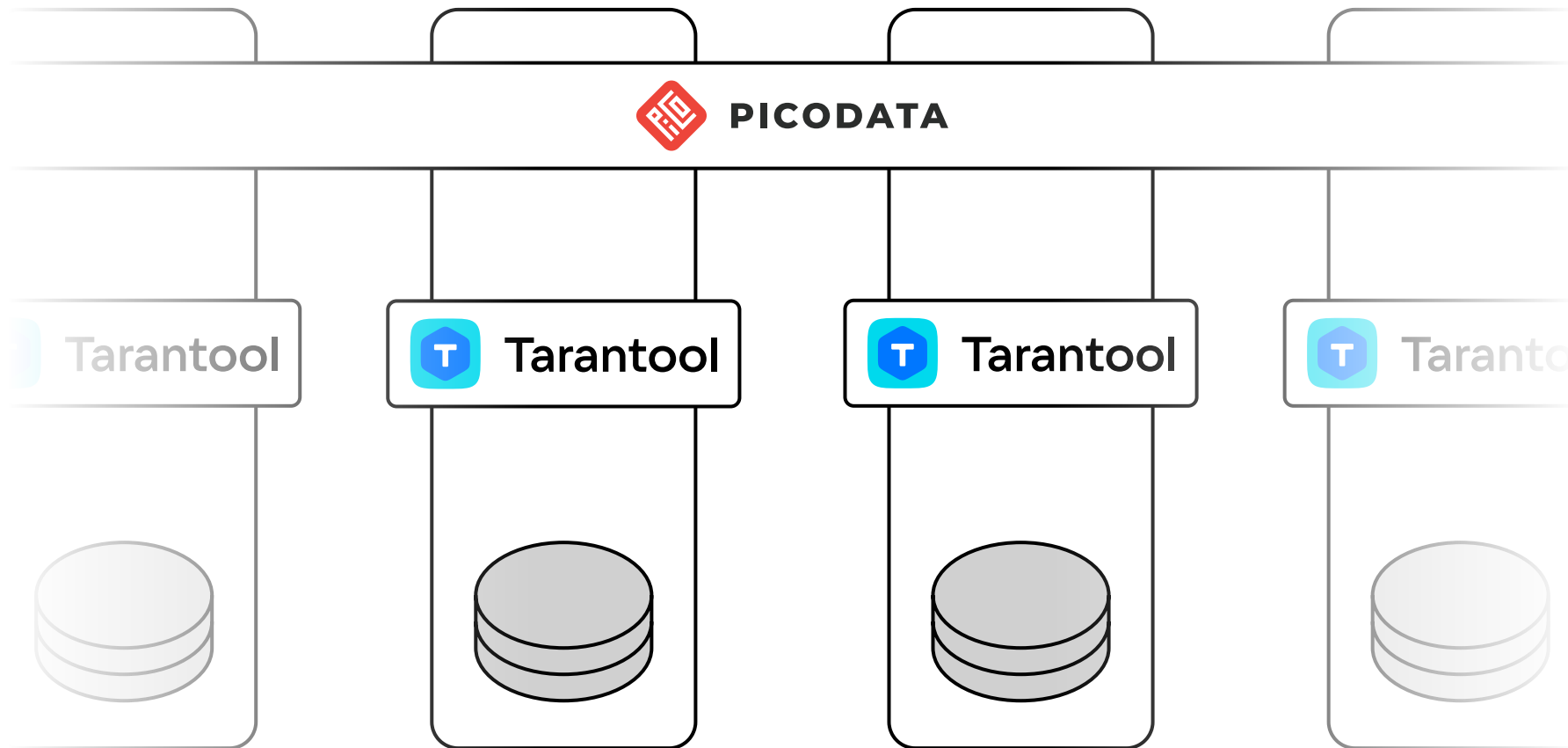
Алгоритм для решения задач консенсуса
в сети ненадежных вычислений



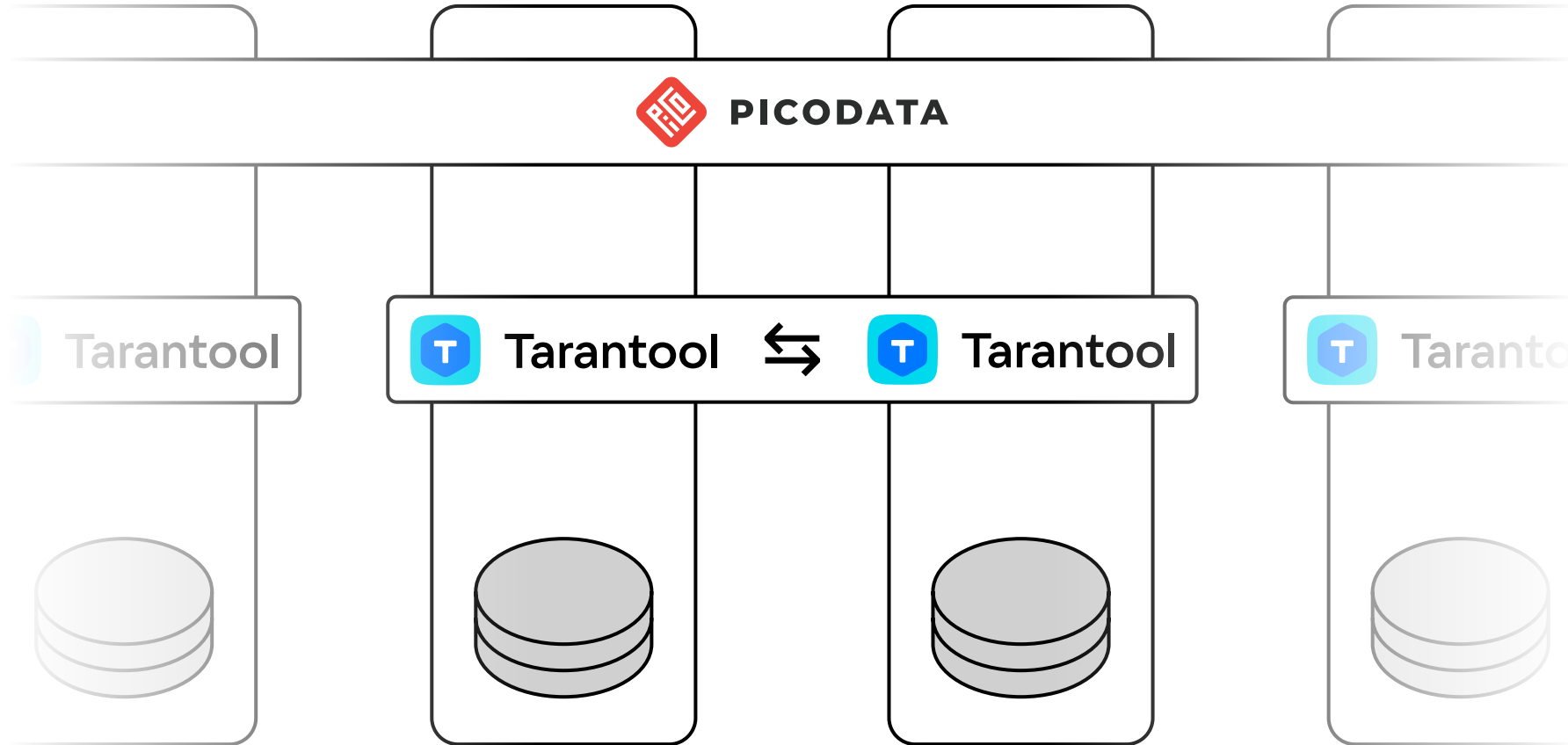
Cluster manager



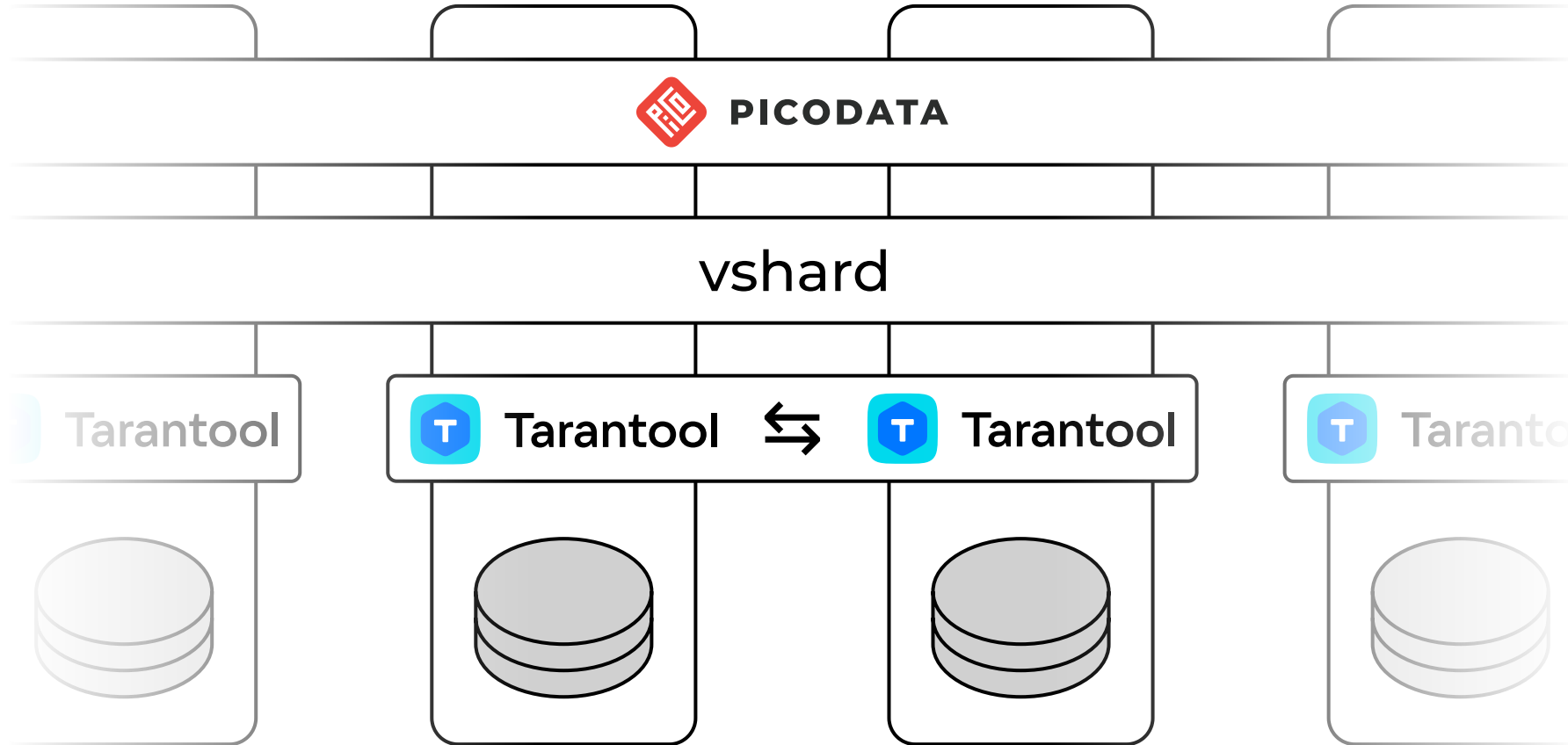
Cluster manager



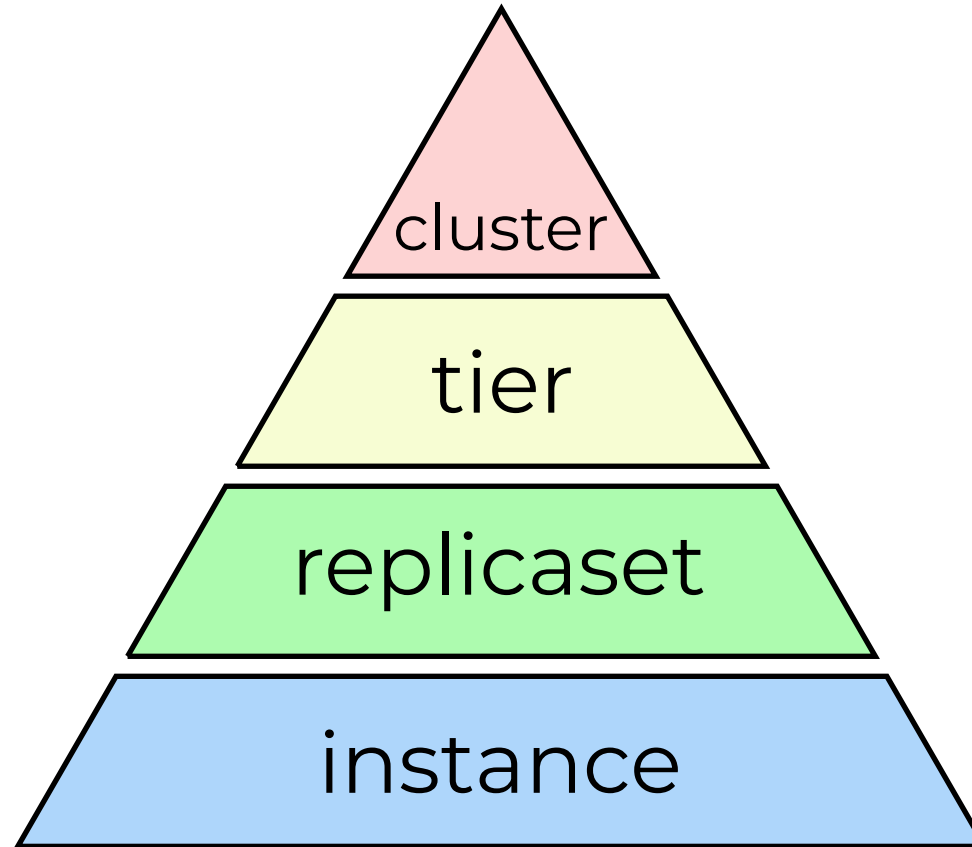
Cluster manager



Cluster manager



Иерархия



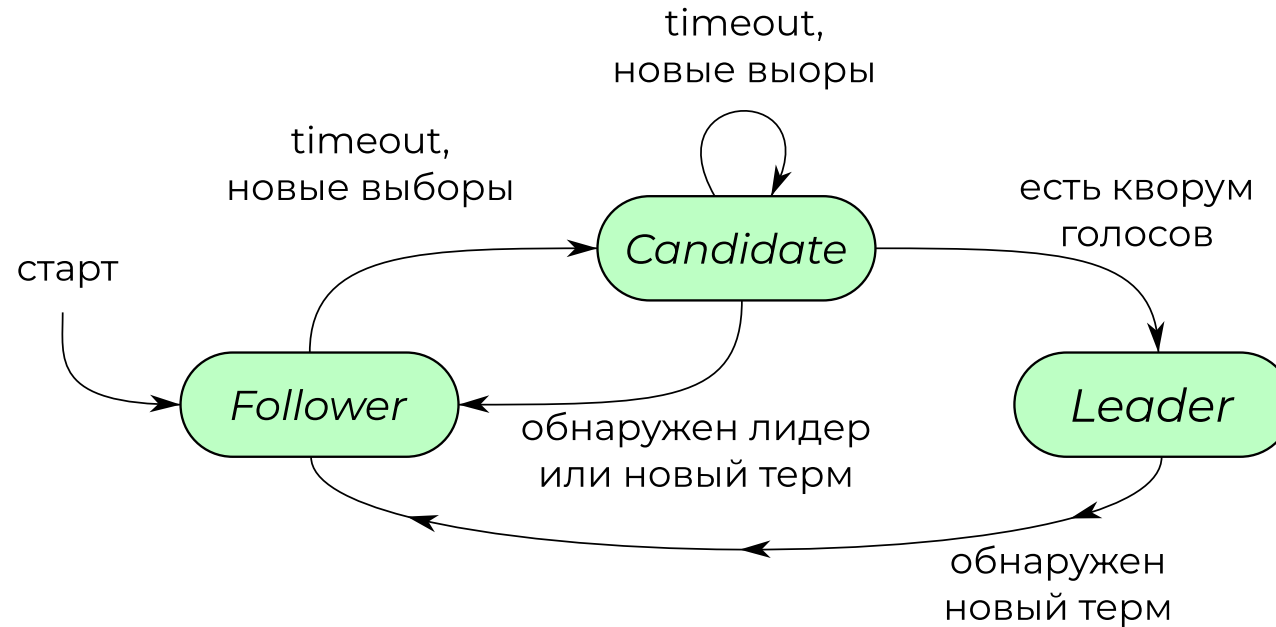
Описание структуры

INSTANCE	
	raft_id
PK	instance_id
FK	replicaset_id
	current_state
	target_state

State: Online \leftrightarrow Offline; generation

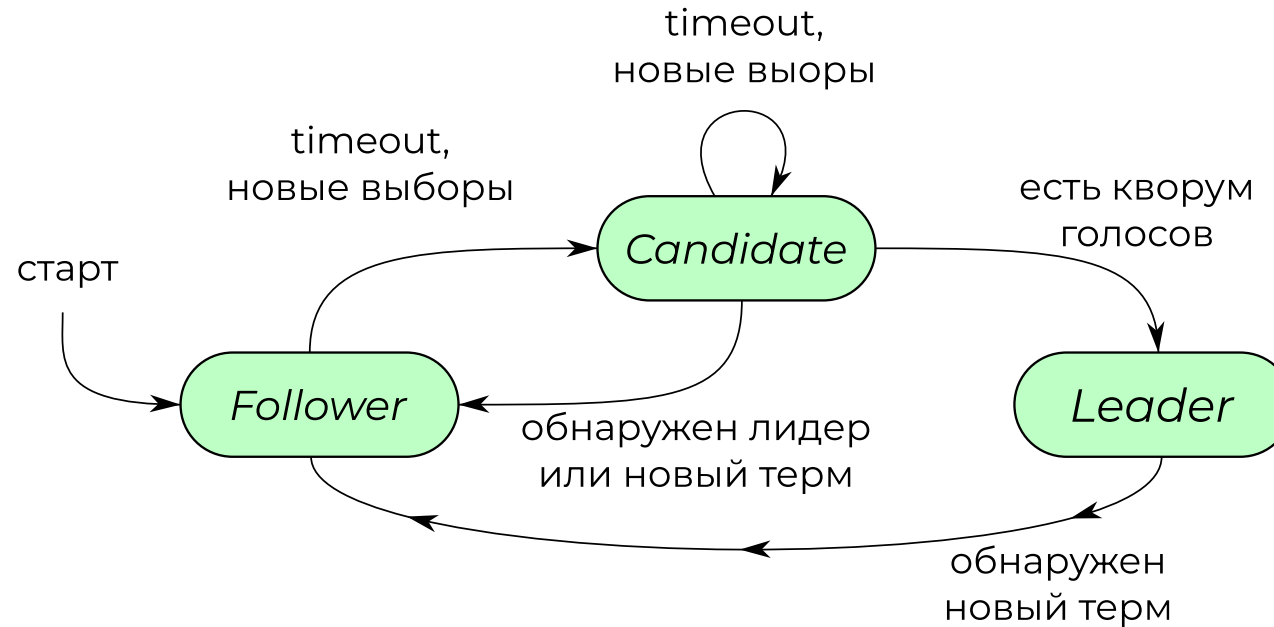
Важно: не status, а state!

Лидер, фолловер, кандидат



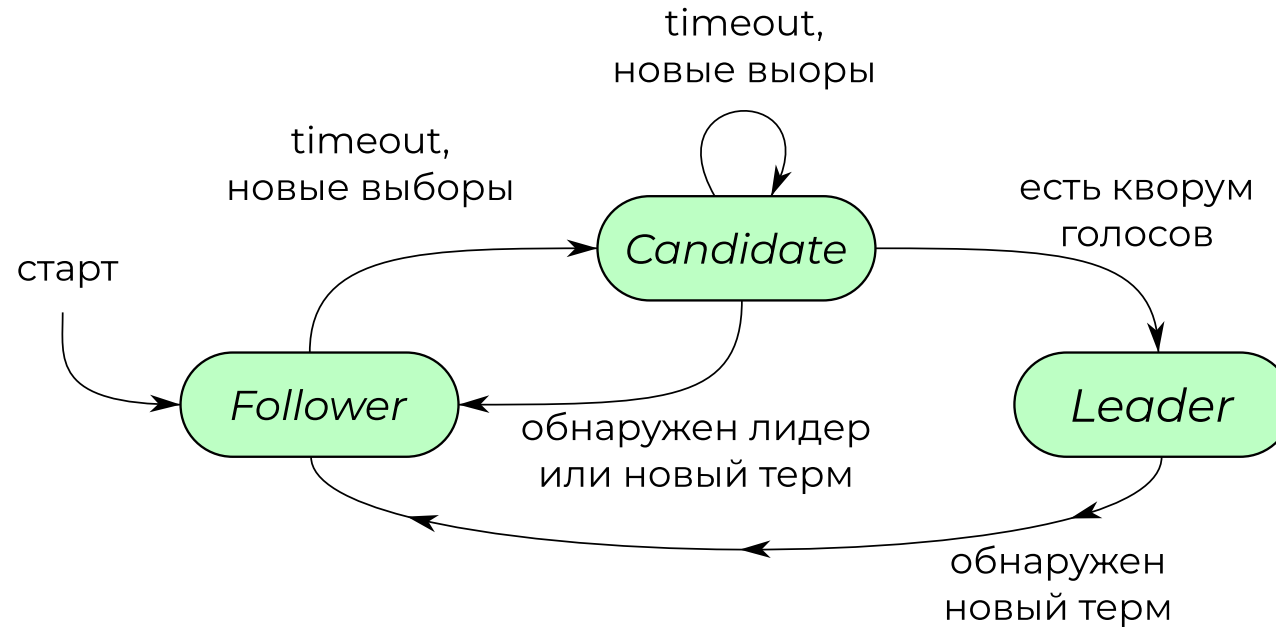
Leader единственный пишет в журнал + пингует окружающих

Лидер, фолловер, кандидат



Leader единственный пишет в журнал + пингует окружающих
Follower пассивен, не отправляет никаких запросов

Лидер, фолловер, кандидат



Leader единственный пишет в журнал + пингует окружающих
Follower пассивен, не отправляет никаких запросов
Candidate проводит голосование

Отказоустойчивость

- Сотни инстансов в кластере
- 5 из них *voters*, остальные *learners*
- 1 из них *Leader*, остальные *Follower*

Работа с данными



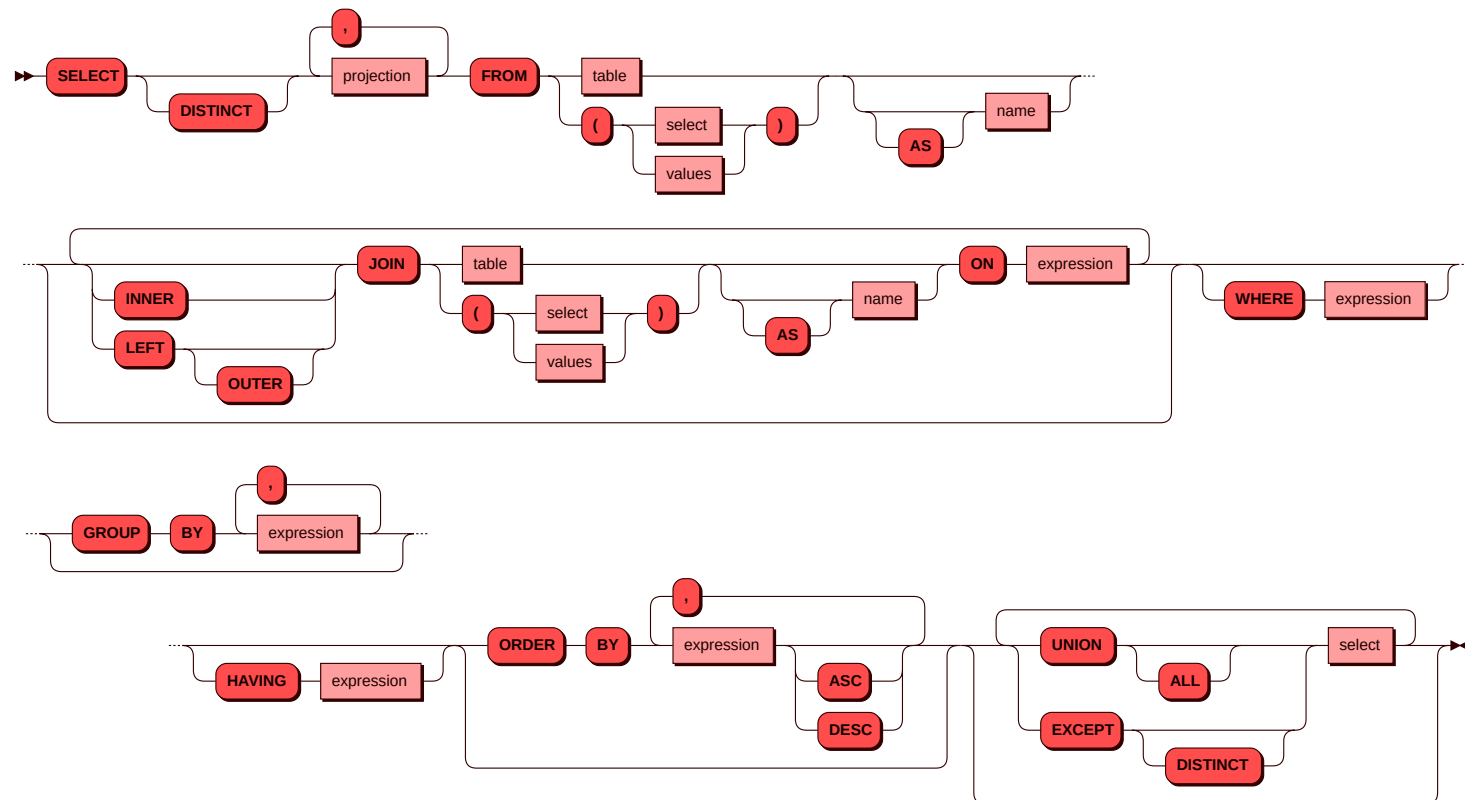
**Saint
HighLoad++**

Хранение данных

- Таблицы бывают *шардированные* и *глобальные*
- Схема данных единая на весь кластер
- Доступ к данным посредством языка SQL

Распределенный SQL

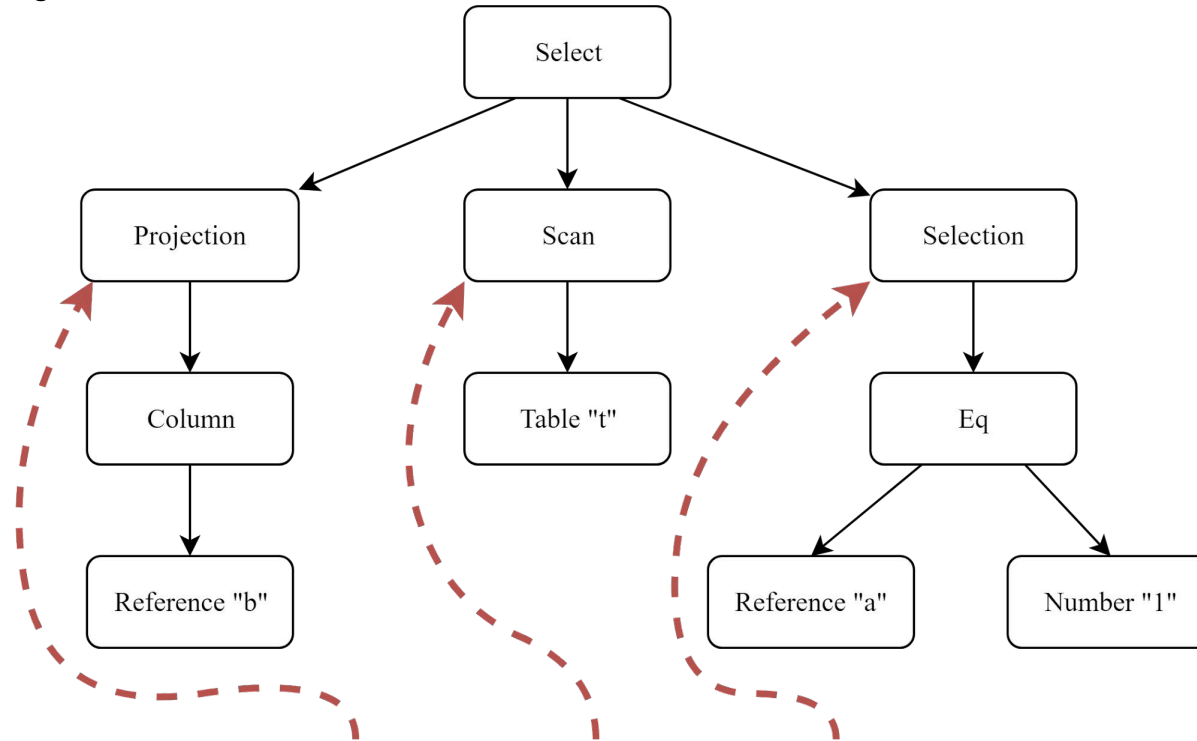
INNER JOIN
 LEFT OUTER JOIN
 WHERE
 WHERE ... IN
 GROUP BY
 HAVING
 ORDER BY
 UNION ALL
 DISTINCT
 EXCEPT DISTINCT



https://docs.picodata.io/picodata/24.4/sql_index/

AST

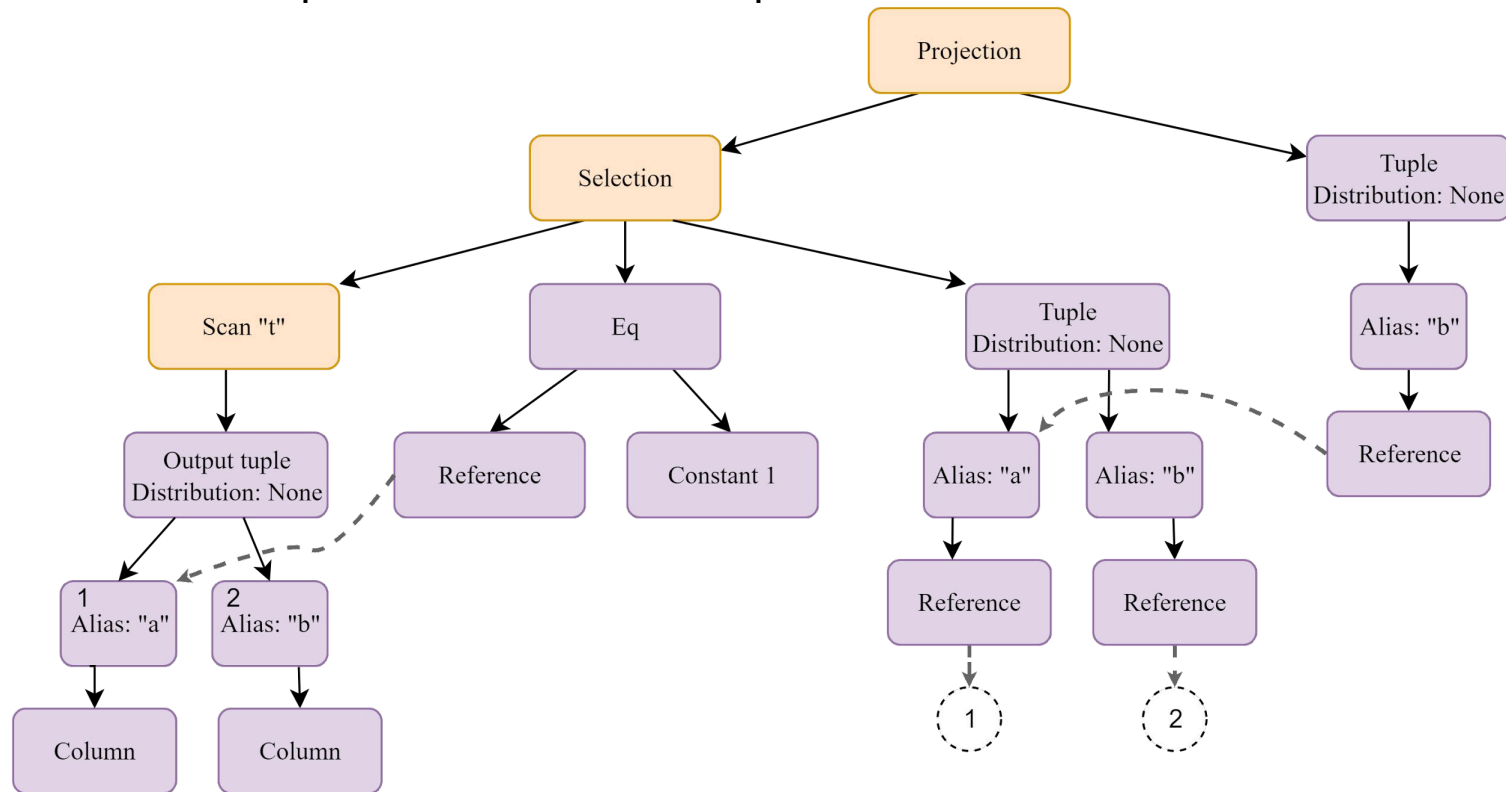
AST == abstract syntax tree



SELECT b FROM t WHERE a = 1

План запроса (IR)

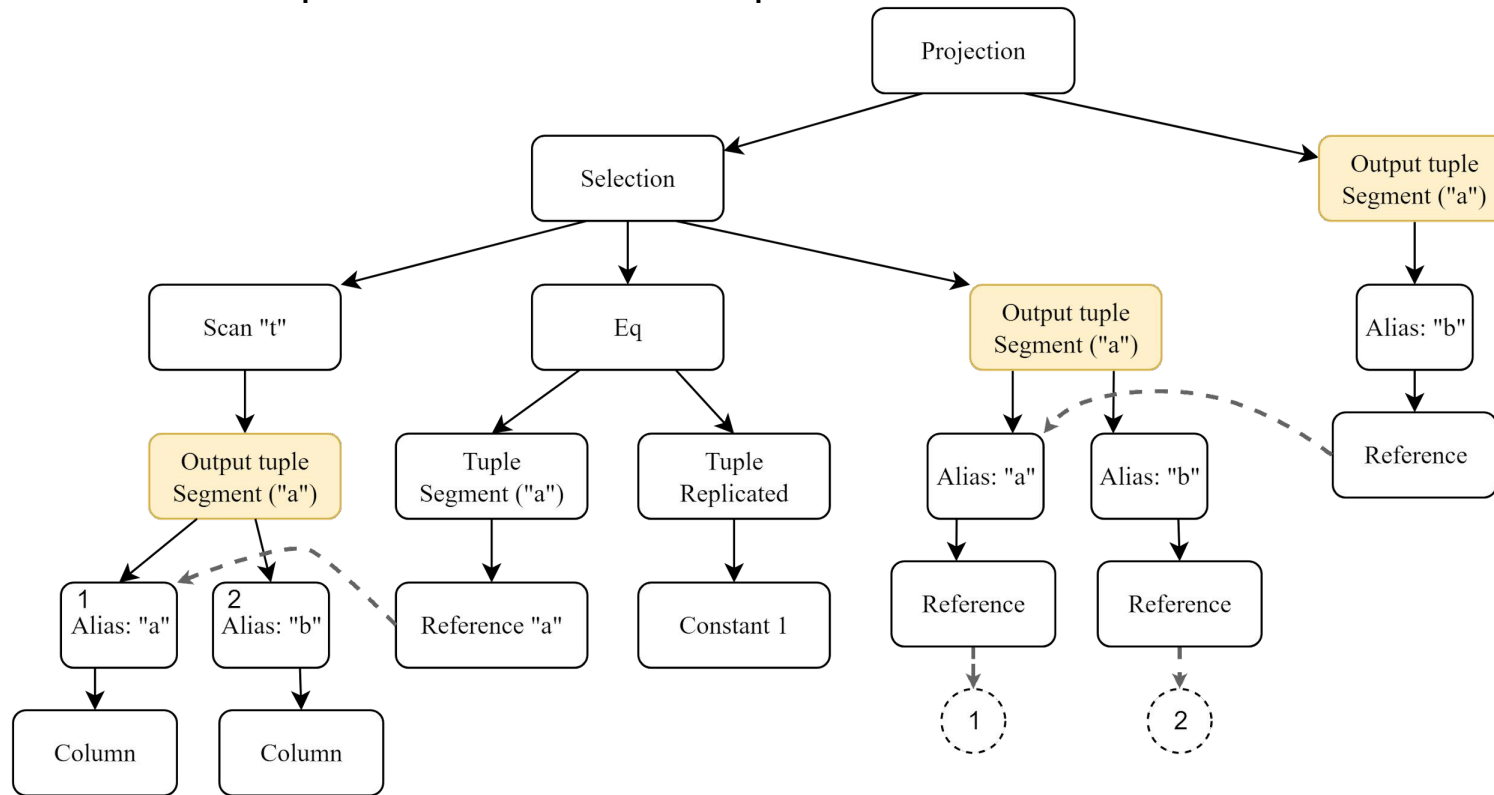
IR == intermediate representation == plan



SELECT b FROM t WHERE a = 1

План запроса

IR == intermediate representation == plan



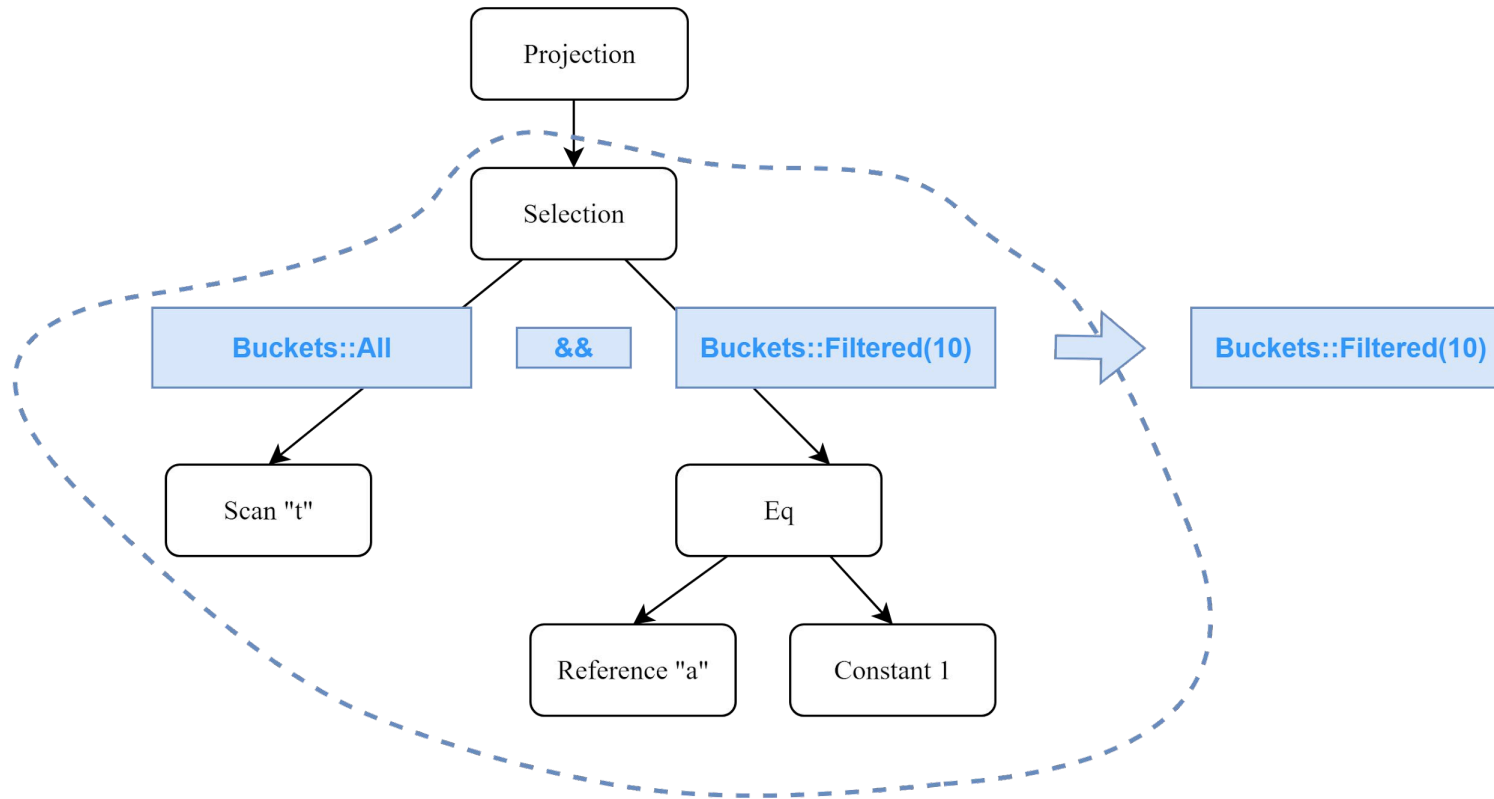
SELECT b FROM t WHERE a = 1

Обработка запроса (1)

Router

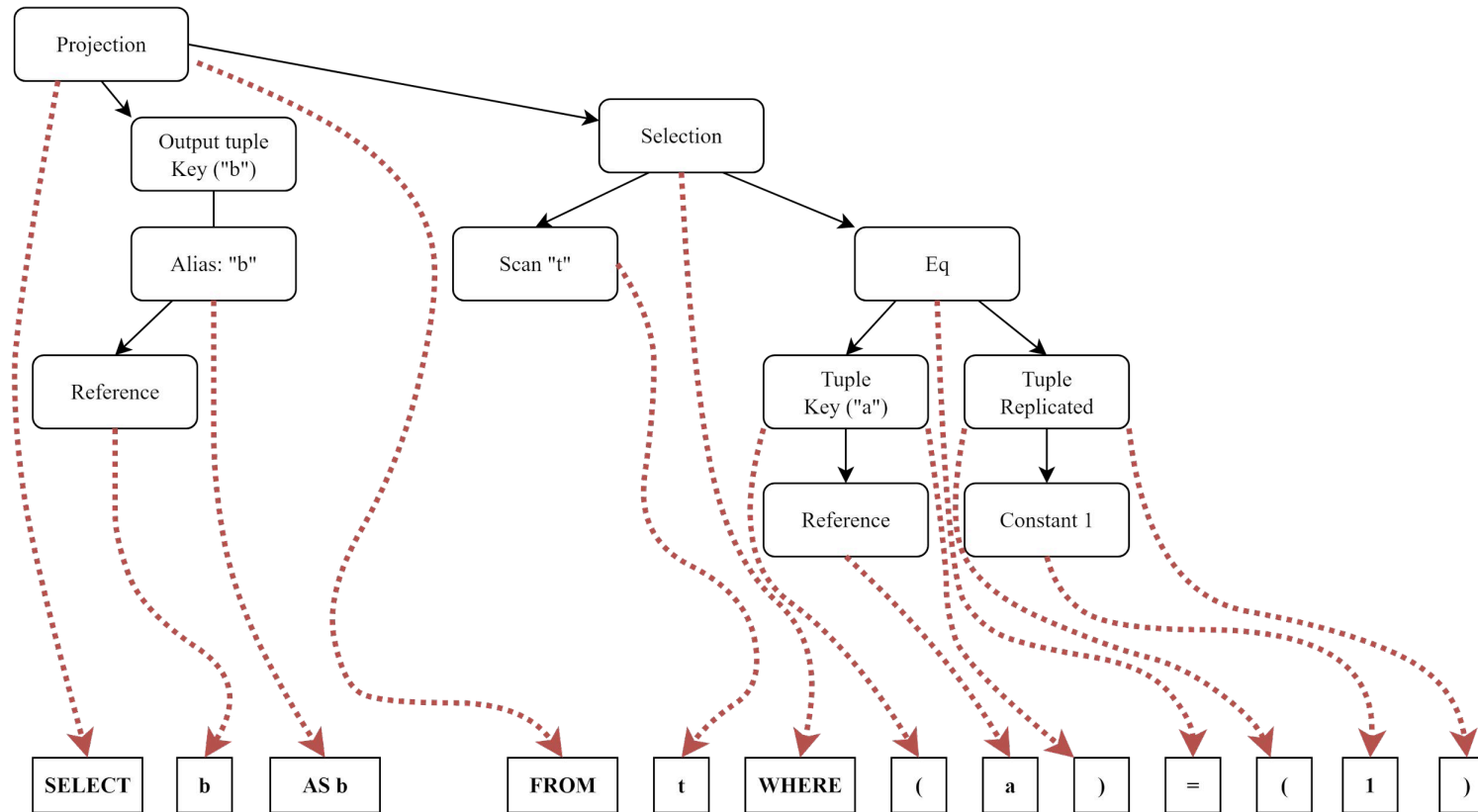
SQL → AST → IR → *execute*

Bucket discovery



SELECT b FROM t WHERE a = 1

IR → VDBE



SELECT b FROM t WHERE a = 1

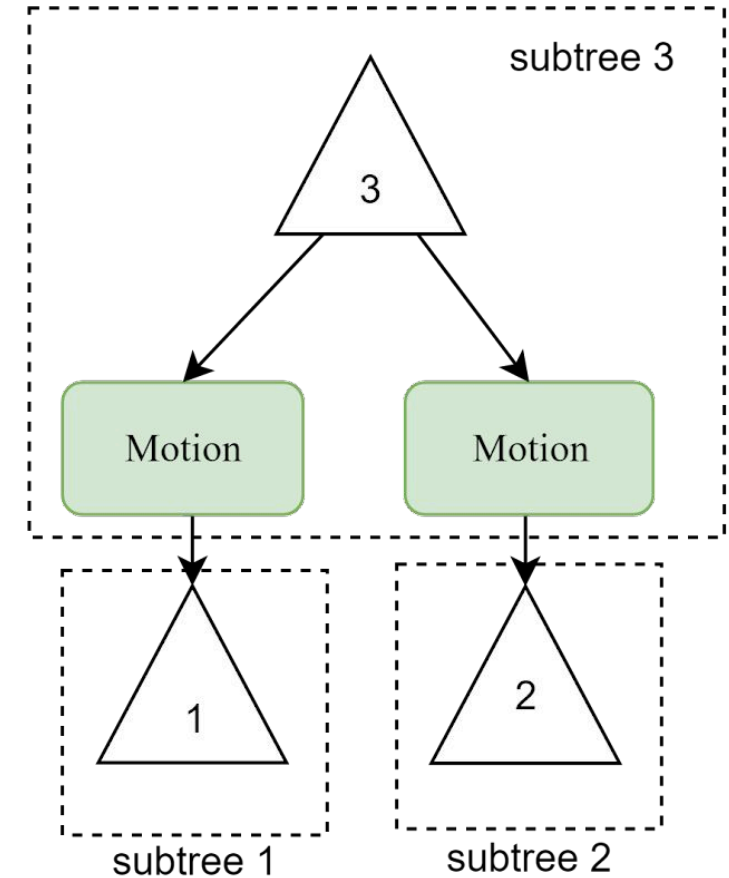
Обработка запроса (2)

Router

SQL → AST → IR → execute

Executor

IR subtree → bucket discovery →
→ map → VDBE ops → reduce



Итого

История Tarantool

Алгоритм Raft

Иерархия — инстансы, репликasetы, тирь, кластер

He status, a state

Лидер, фолловер, кандидат

Динамическое переключение голосующих узлов

Распределенный SQL

Материалы

- Слайды: <https://rosik.github.io/2024-shl>
- Picodata: <https://picodata.io/>
- Telegram: [@picodataru](https://t.me/picodataru)

Обратная связь:

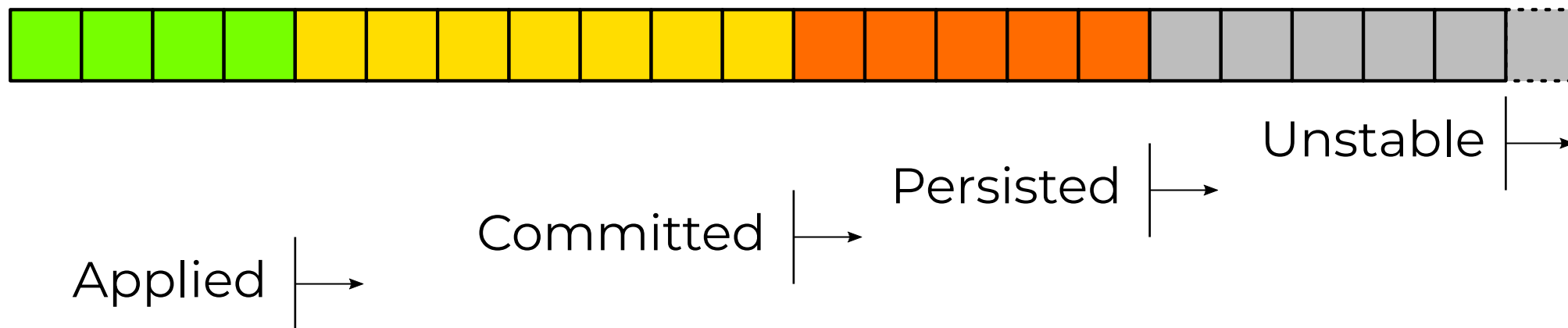


<https://conf.ontico.ru/online/shl2024/details/5492379>



**Saint
HighLoad++**

Состояние записей в raft-журнале



Виды операций в raft-журнале

```
pub enum Op {  
    Nop,  
    Dml(Dml),  
    DdlPrepare { schema_version: u64, ddl: Ddl },  
    DdlCommit,  
    DdlAbort,  
    Acl(Acl),  
}
```