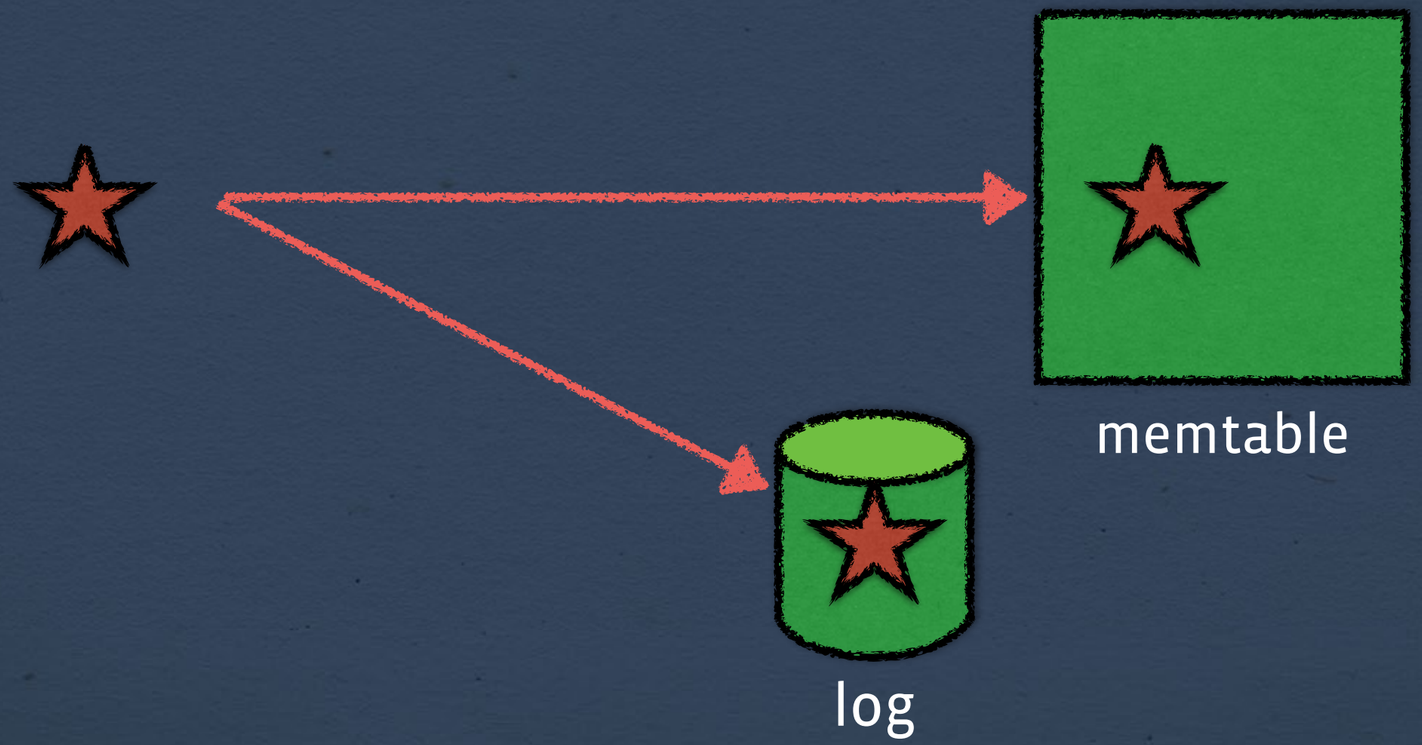
Column Families in RocksDB

Igor Canadi
Database Engineering @ Facebook

Overview

- Motivation
- API
- Implementation

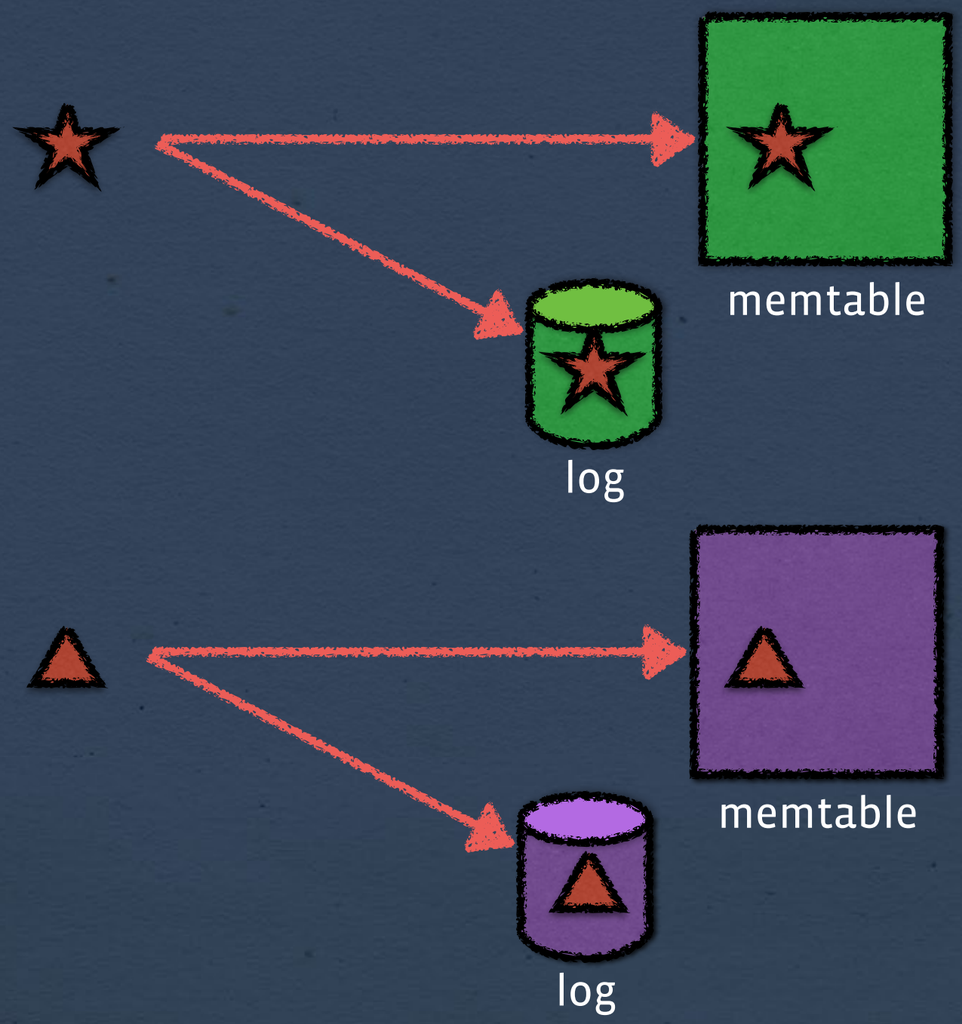
Motivation



Motivation

- Data you might want to manage separately:
  - data, secondary index
  - data, metadata

Separated everything approach

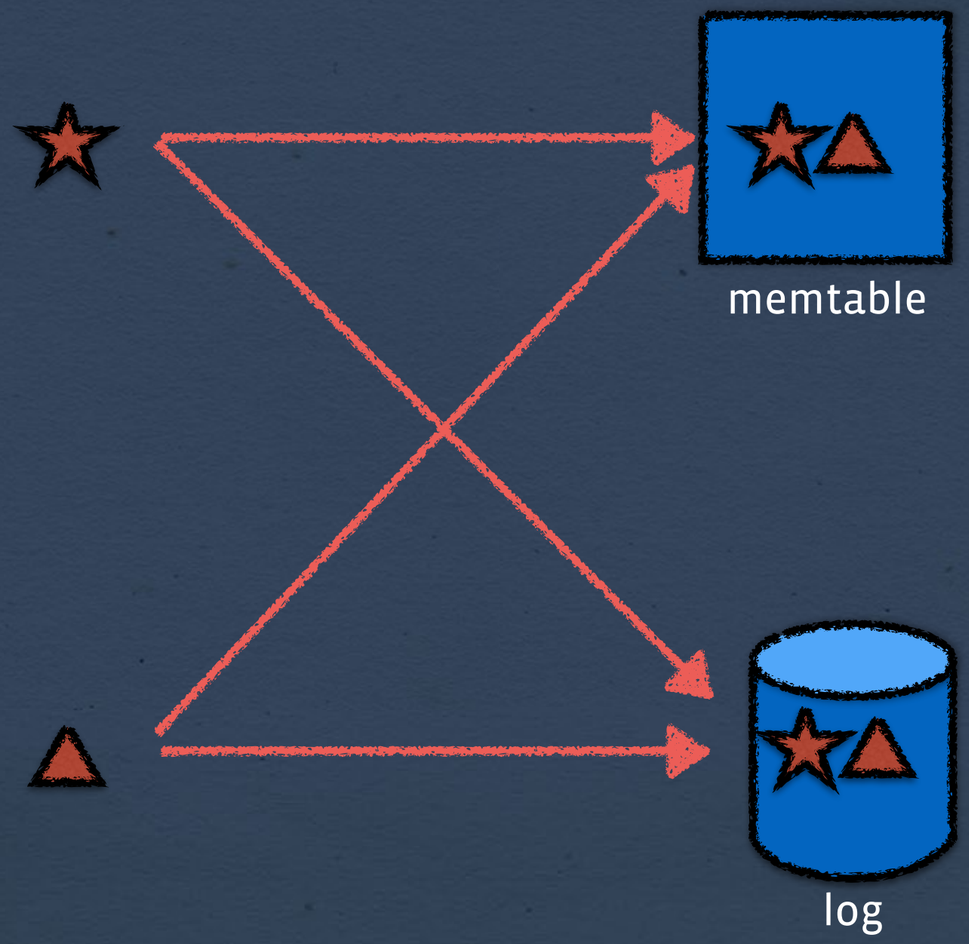


Separated everything approach

- Separated data
  - Better compressibility
  - More configuration options
  - Fast deletion of the whole data set

- Shared write-ahead log
  - Atomic operations
  - More effective group commits

Shared everything approach

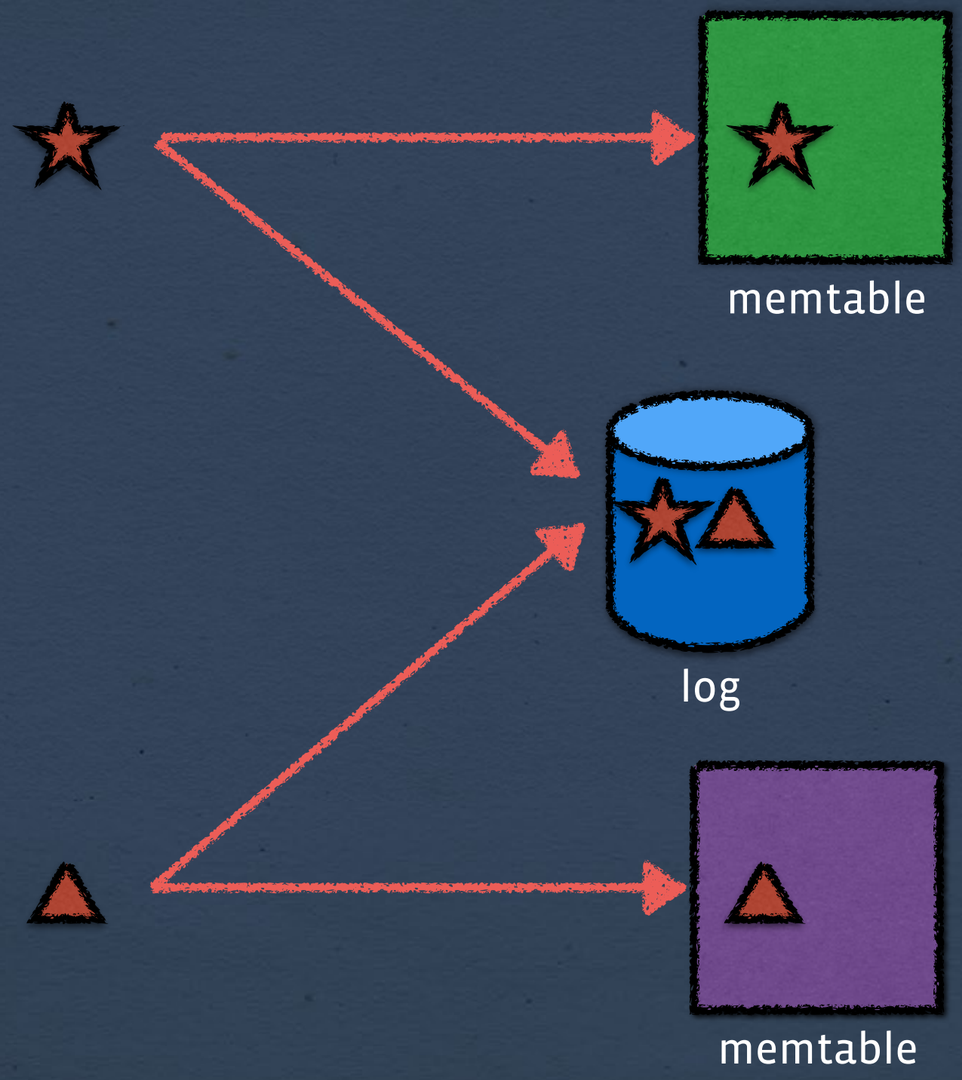


Shared everything approach

- Separated data
  - Better compressibility
  - More configuration options
  - Fast deletion of the whole data set

- Shared write-ahead log
  - Atomic operations
  - More effective group commits

Column families approach

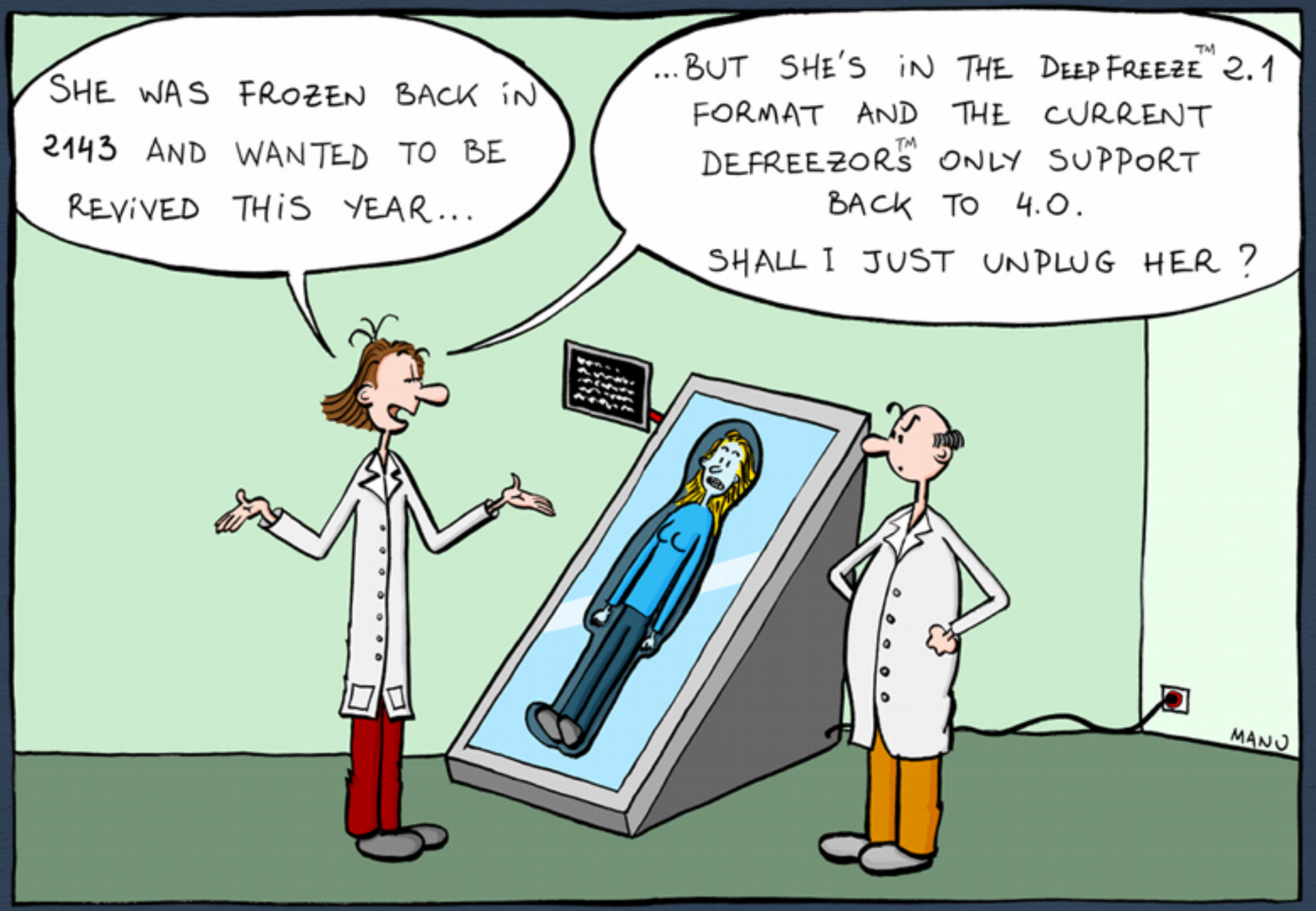


Column families approach

- Separated data
  - Better compressibility
  - More configuration options
  - Fast deletion of the whole data set

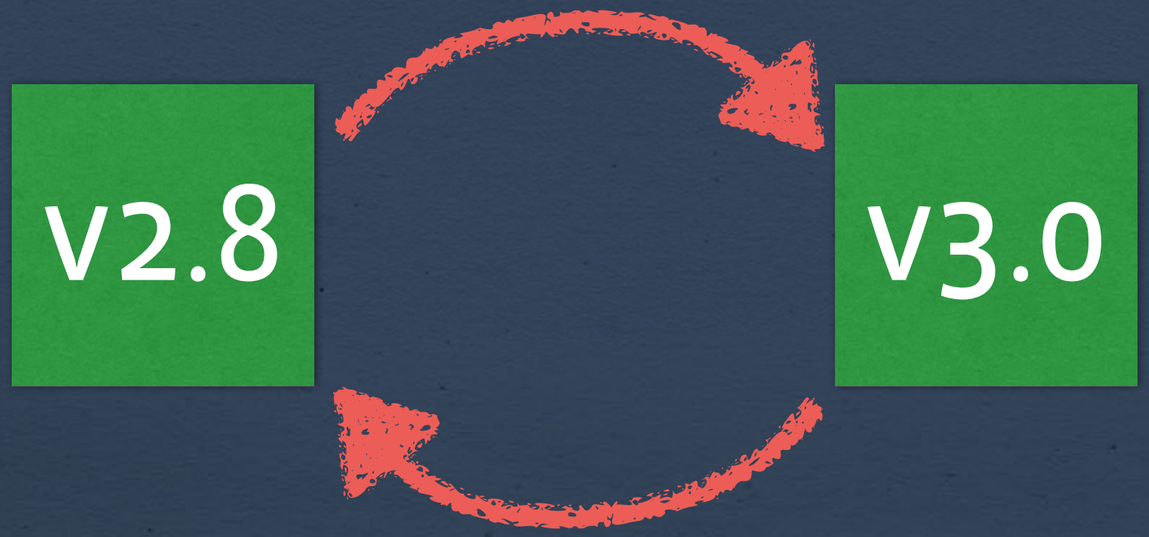
- Shared write-ahead log
  - Atomic operations
  - More effective group commits

API

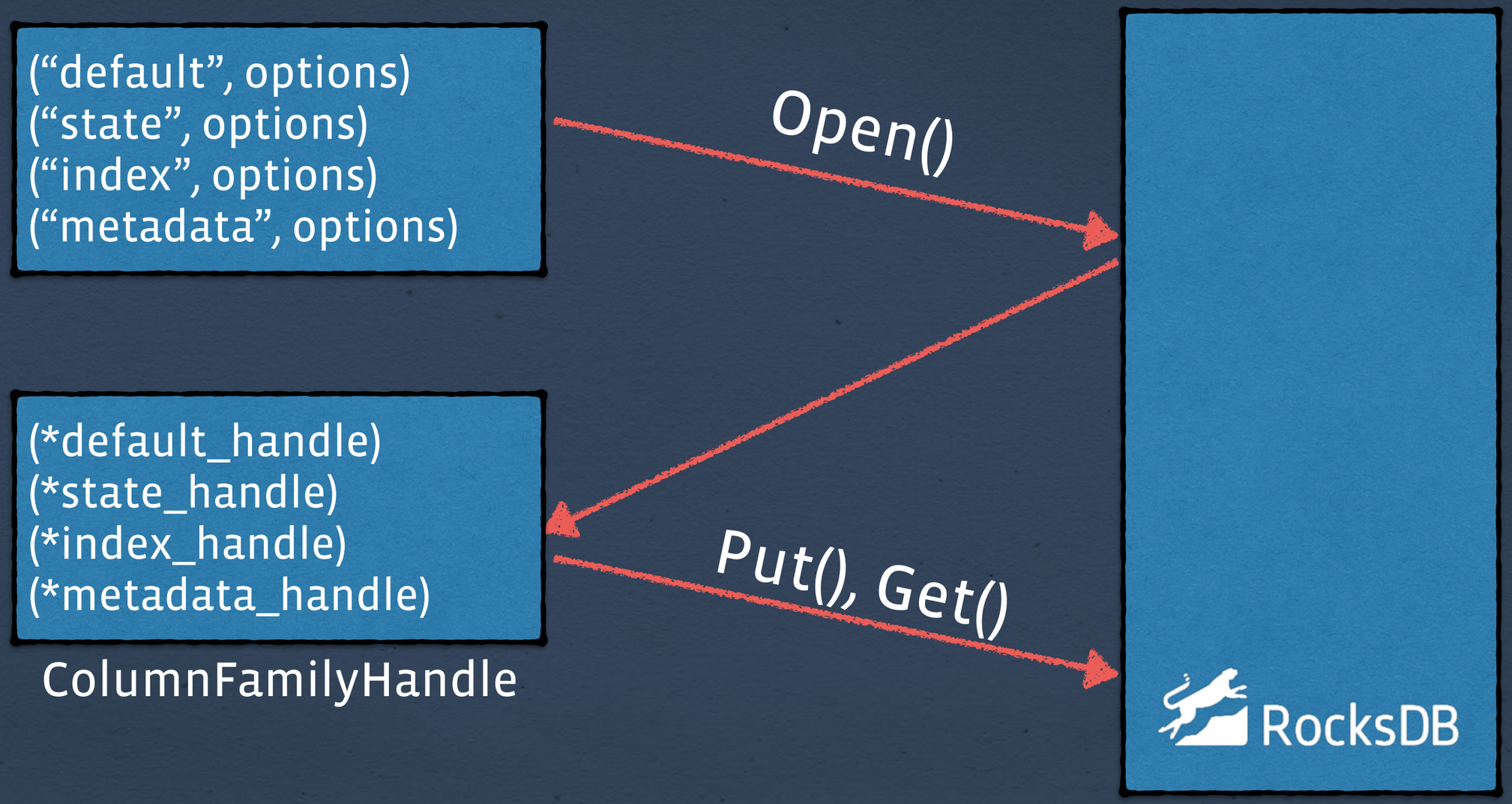

SHE WAS FROZEN BACK IN 2143 AND WANTED TO BE REVIVED THIS YEAR...
...BUT SHE'S IN THE DEEP FREEZE™ 2.1 FORMAT AND THE CURRENT DEFREEZORS™ ONLY SUPPORT BACK TO 4.0.
SHALL I JUST UNPLUG HER ?
MANU

API

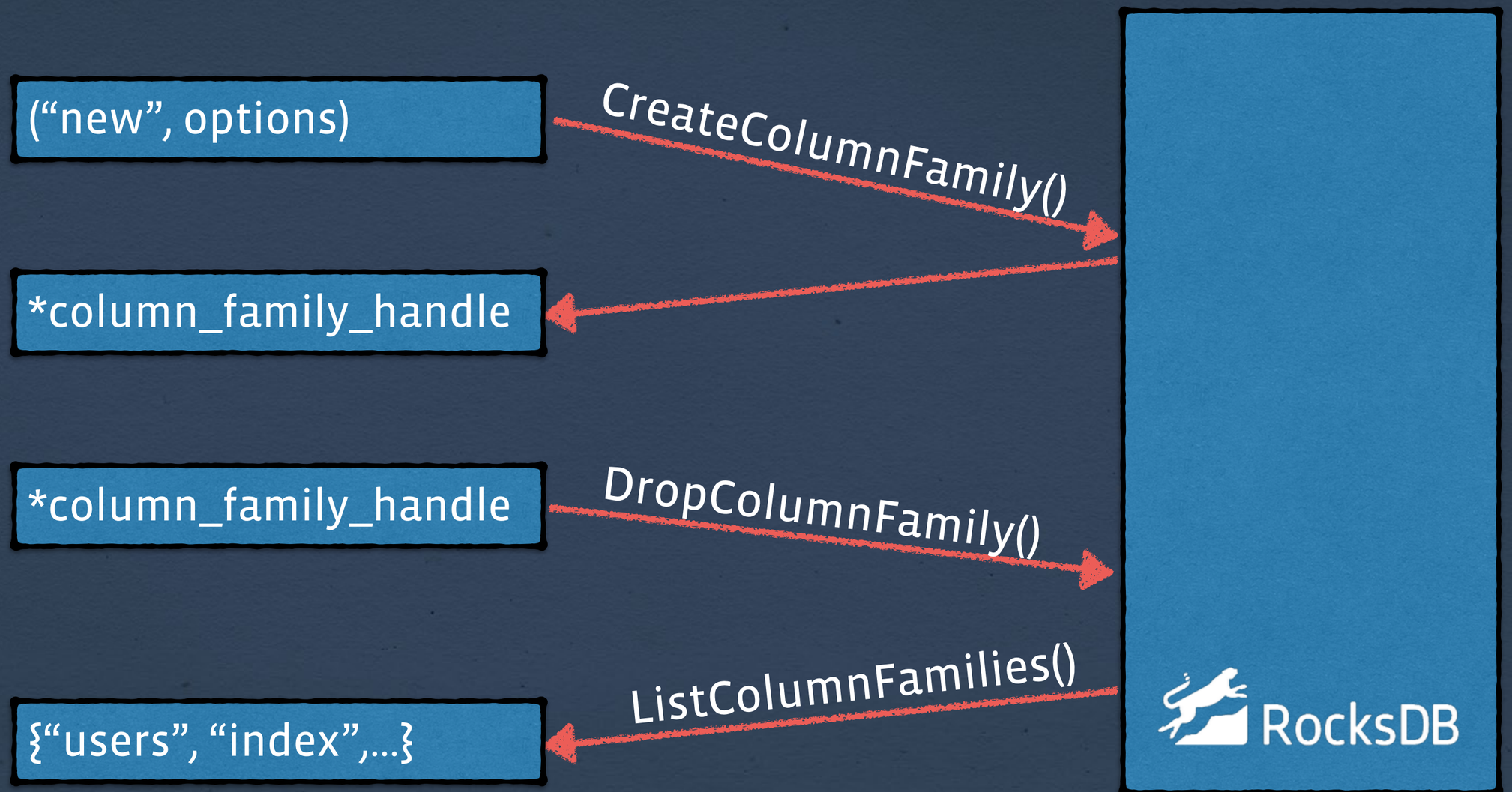
- Backward compatibility
- Forward compatibility (if not created new column family)



API


("default", options)
("state", options)
("index", options)
("metadata", options)
Open()
(*default_handle)
(*state_handle)
(*index_handle)
(*metadata_handle)
ColumnFamilyHandle
Put(), Get()
RocksDB

API


("new", options)
CreateColumnFamily()
*column_family_handle
*column_family_handle
DropColumnFamily()
ListColumnFamilies()
{"users", "index",...}
RocksDB

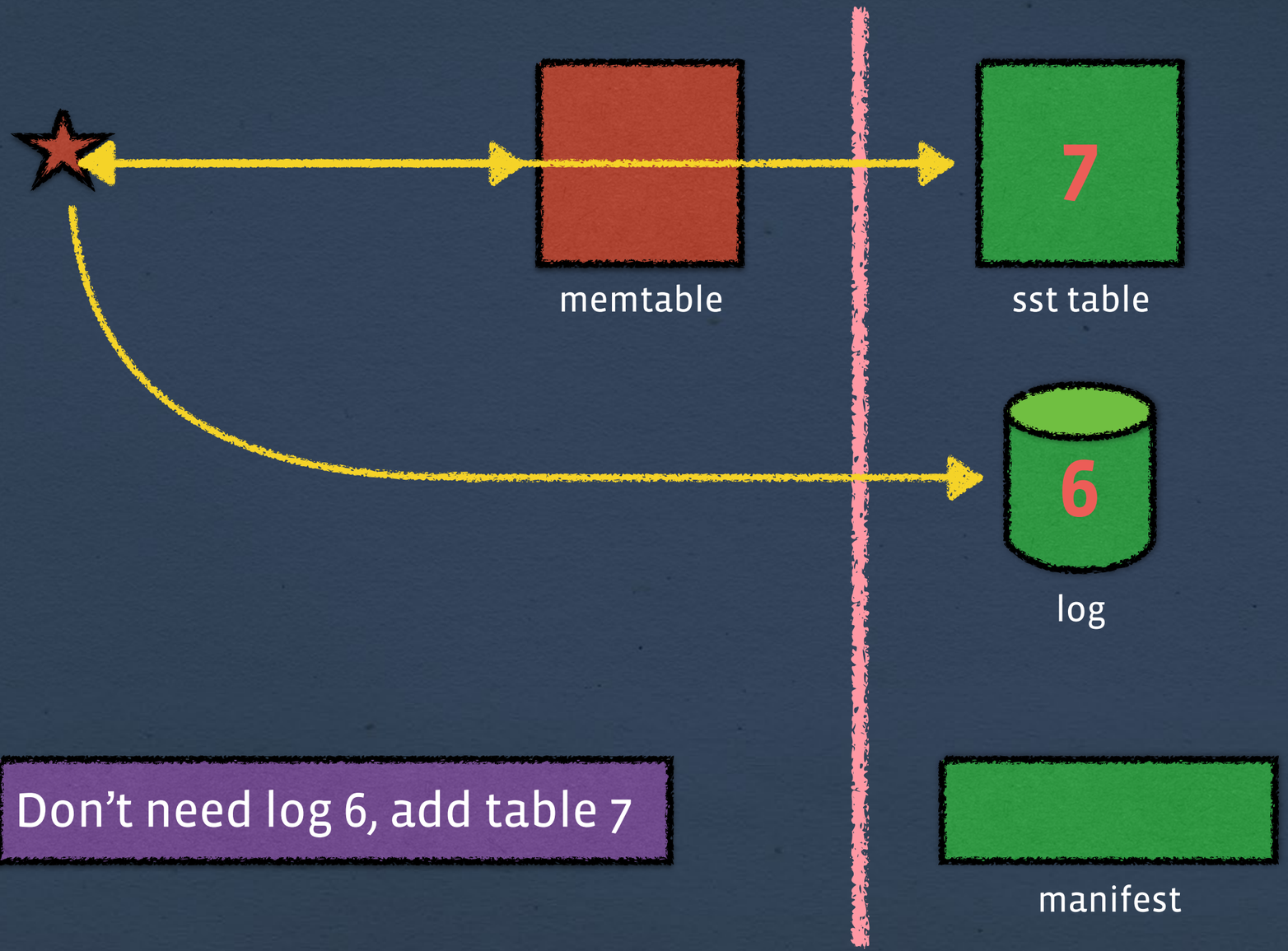
Implementation

- Rolling log files

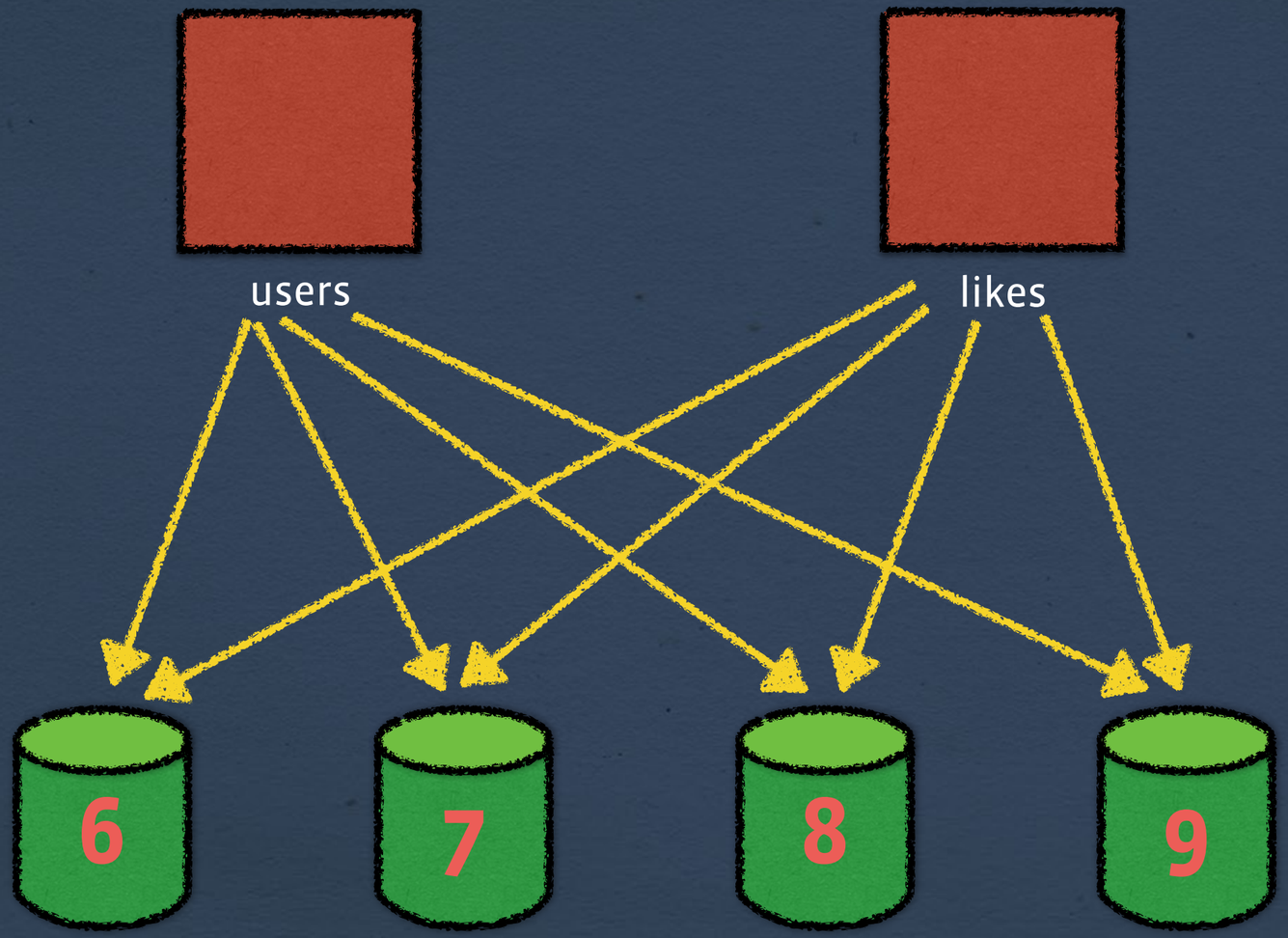
Background — Manifest

- DB state is a set of files
- DB metadata — mainly list of files
- Manifest — transactional log for DB metadata
- Atomic transactions

Rolling log files - single column family



Rolling log files - two column families



manifest

Create column family likes
Create column family user
likes — persisted logs up to 6
likes — persisted logs up to 7
users — persisted logs up to 8

Summary

- A way to store data separately, but write to the same write-ahead log
- Coming in RocksDB 3.0

Thank **you!**