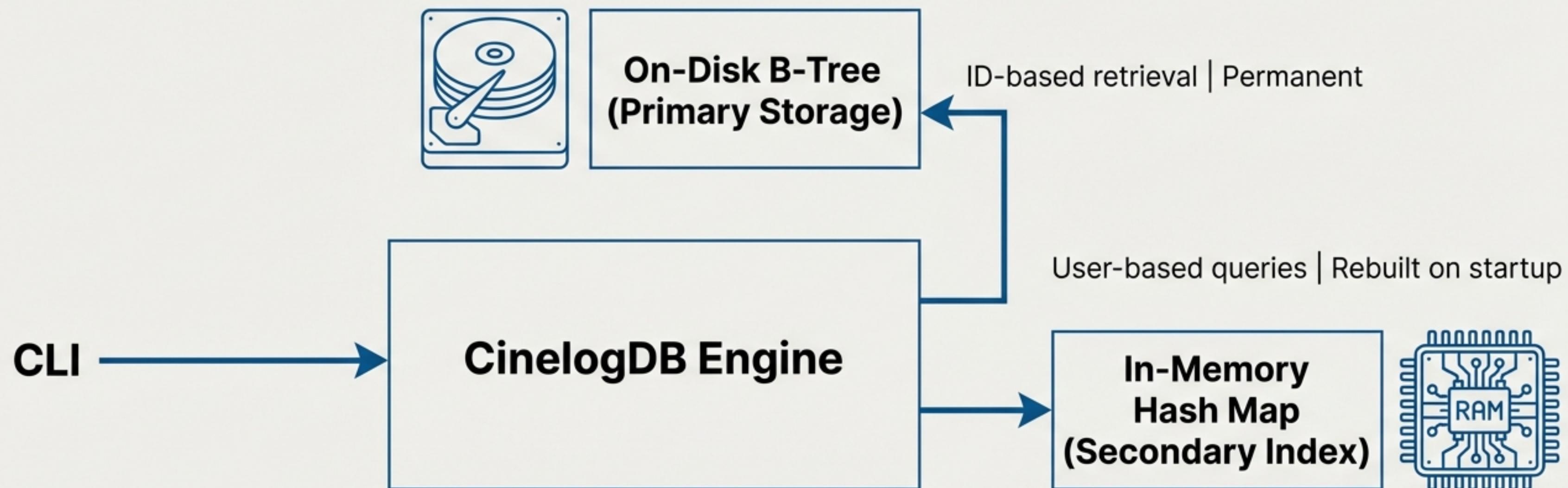




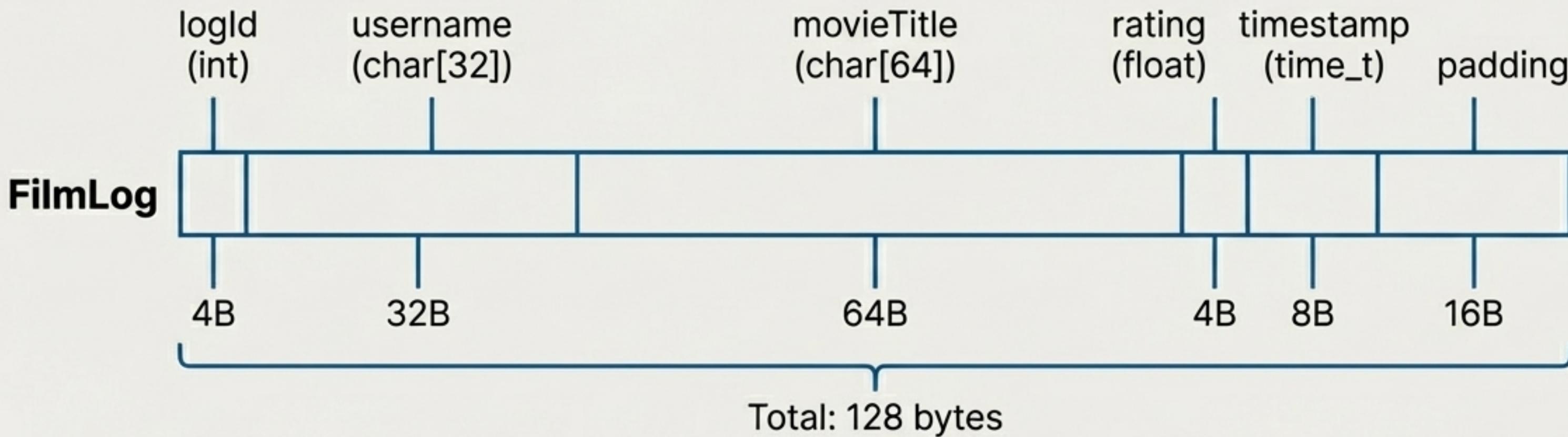
Cinelog

Architecture of a Custom Data Engine

A Dual-Index Architecture for Persistence and Speed



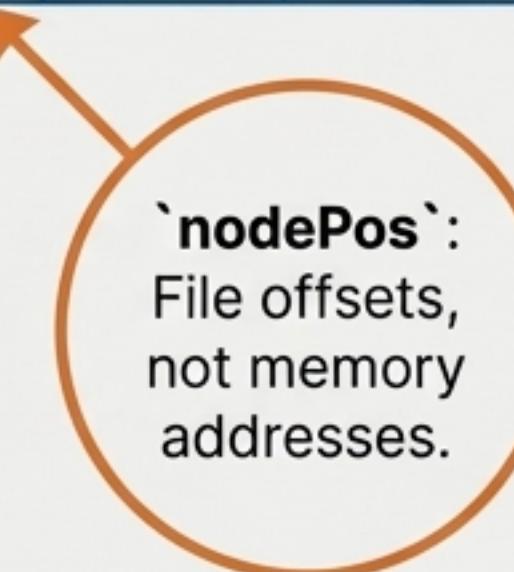
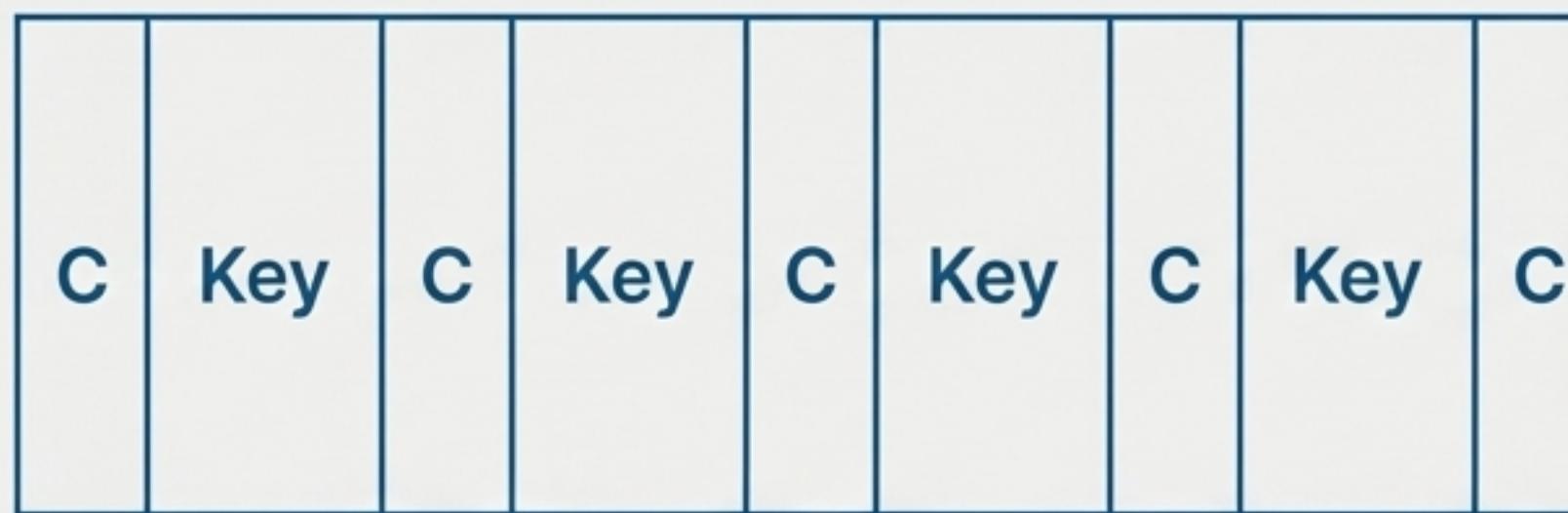
The Core Unit: A 128-Byte POD Struct.



Plain Old Data (POD) for direct binary serialization.

No JSON, no CSV. We use `'memcpy'` to write the raw struct directly to a binary buffer for maximum efficiency.

The Foundation: An On-Disk B-Tree for Persistent, Ordered Storage

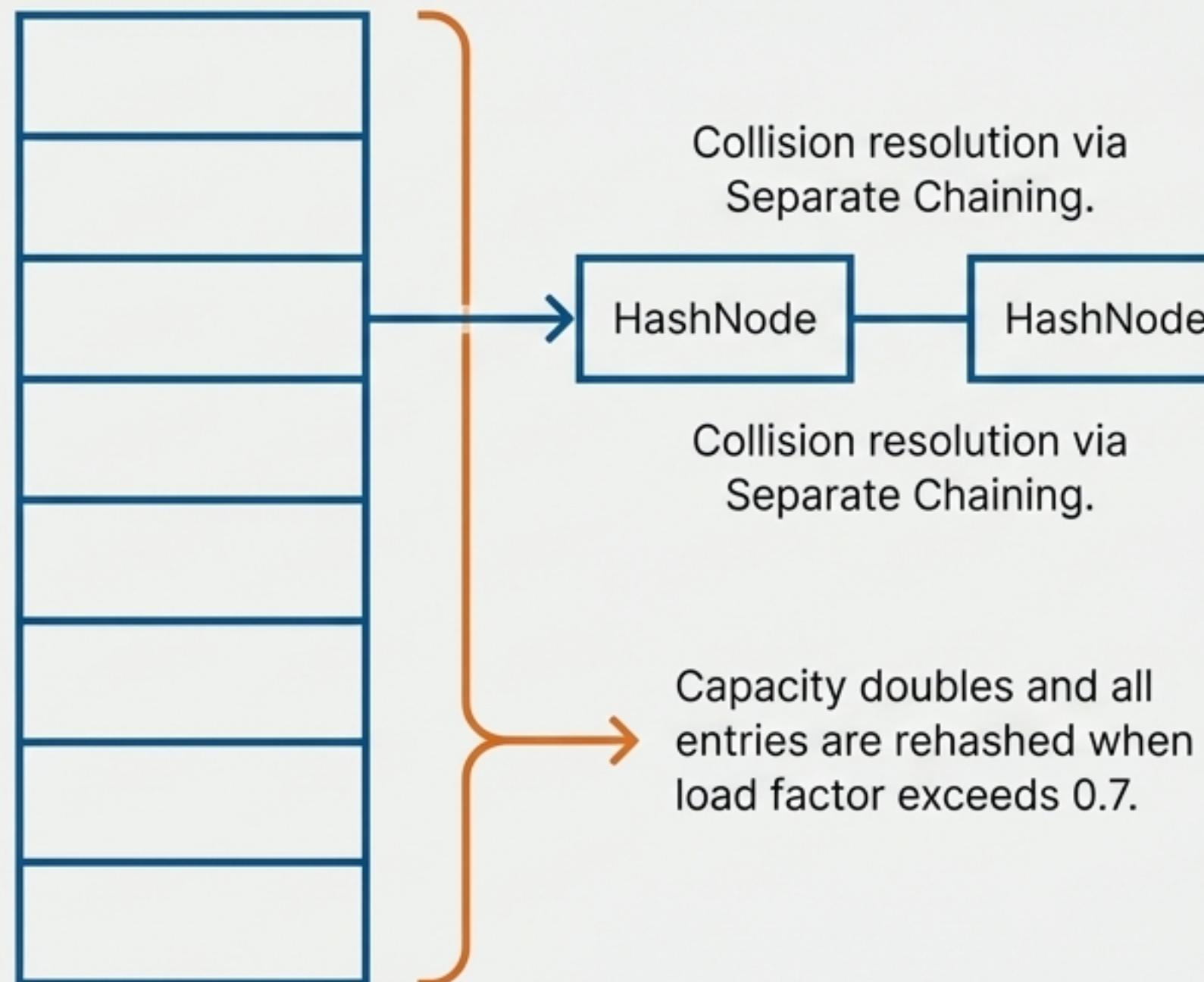


Order 5 structure minimizes disk I/O depth.

Nodes are serialized and deserialized directly to/from 'tree.bin'.

Balanced on insertion using 'splitChild' and 'insertNonFull' logic.

The Accelerator: A Custom Hash Map for O(1) User Queries.



- Custom implementation, replacing `std::unordered_map` for granular control.
- Manually manages all memory allocation and deallocation for nodes.
- Maps `username` to a list of associated `logId`s.

The Conductor: `CinelogDB` Synchronizes Writes, Rebuilds, and Queries

