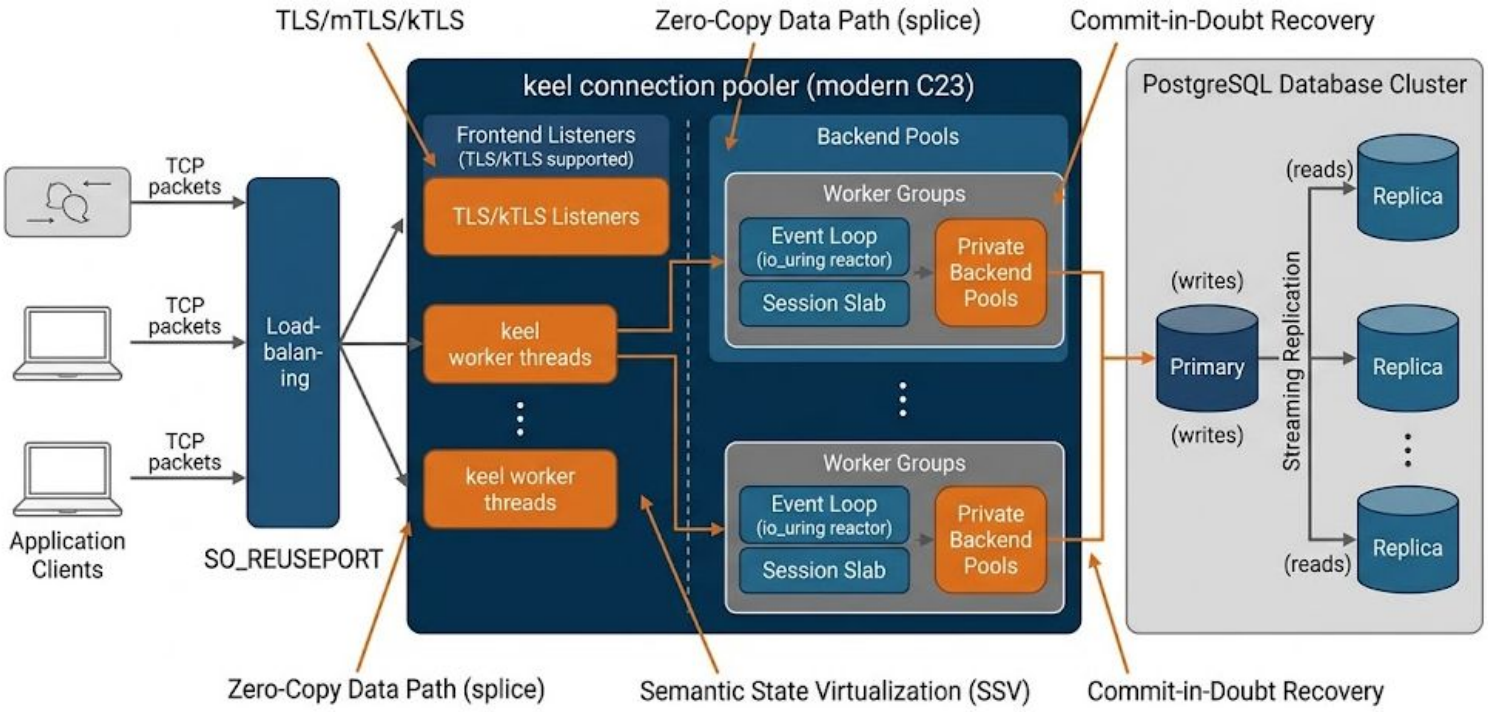
More than a Database Connection Pooler

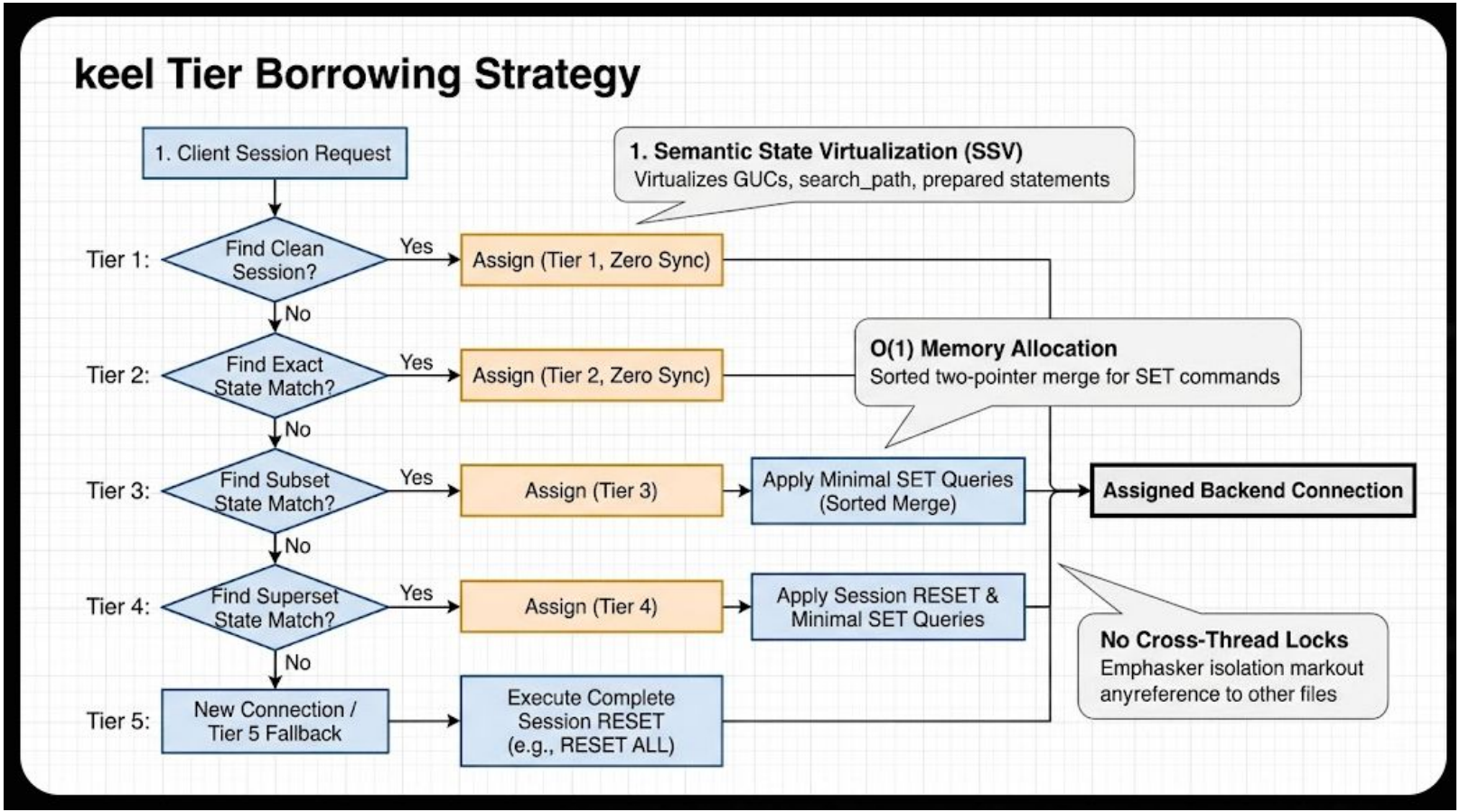
Key Design Principles

- Share nothing, scales almost linearly — each worker is fully isolated
- Zero-copy I/O with Linux splice(2) for client↔backend data transfer
- Async everything — backend connect + SCRAM-SHA-256 auth runs entirely on the reactor
- Multi-protocol by design: Plugins for PostgreSQL and MySQL implemented
- Transaction pooling with SSV (Semantic State Virtualization)
- Connection Migration — idle sessions transferred between workers
- XID Probe + Commit-in-Doubt Recovery
- Read-After-Write Consistency
- Hook/Trigger System

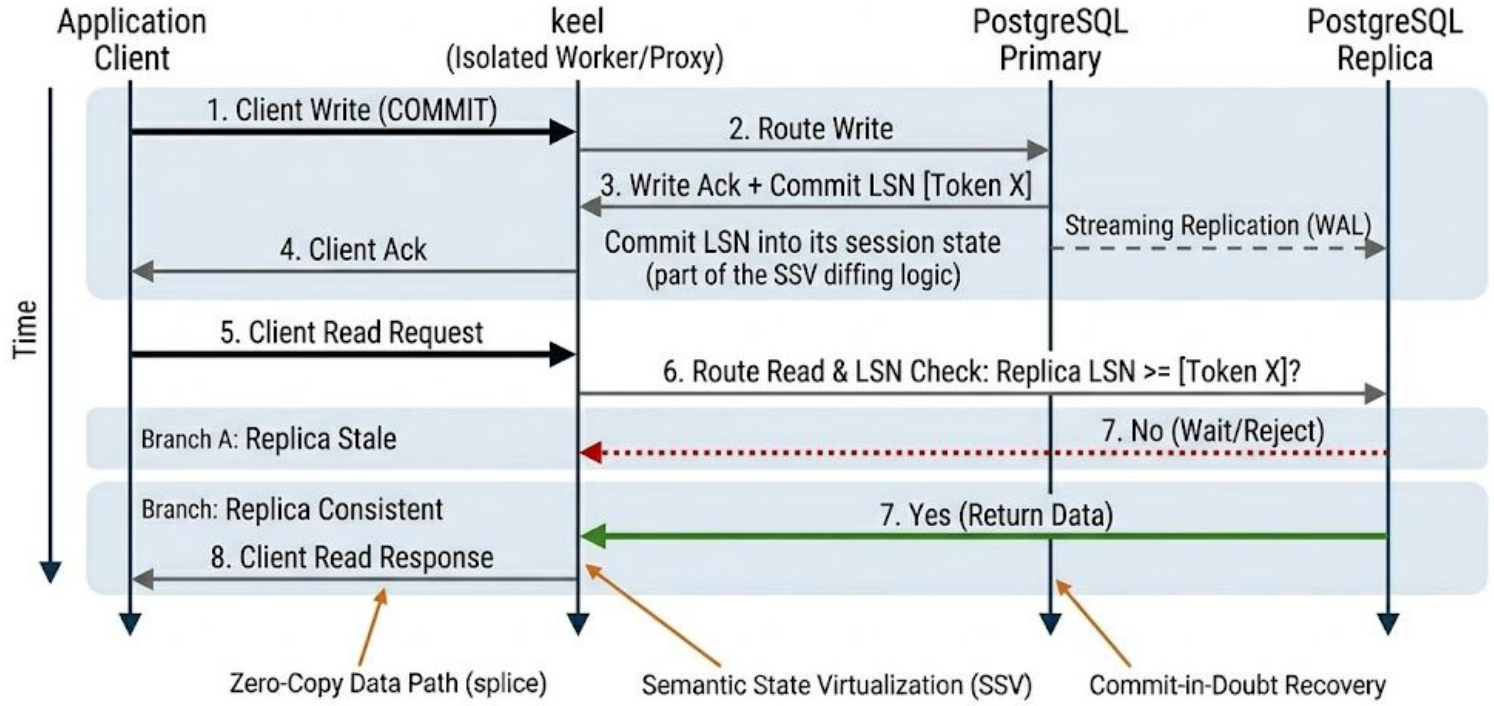
keel Overall Architecture



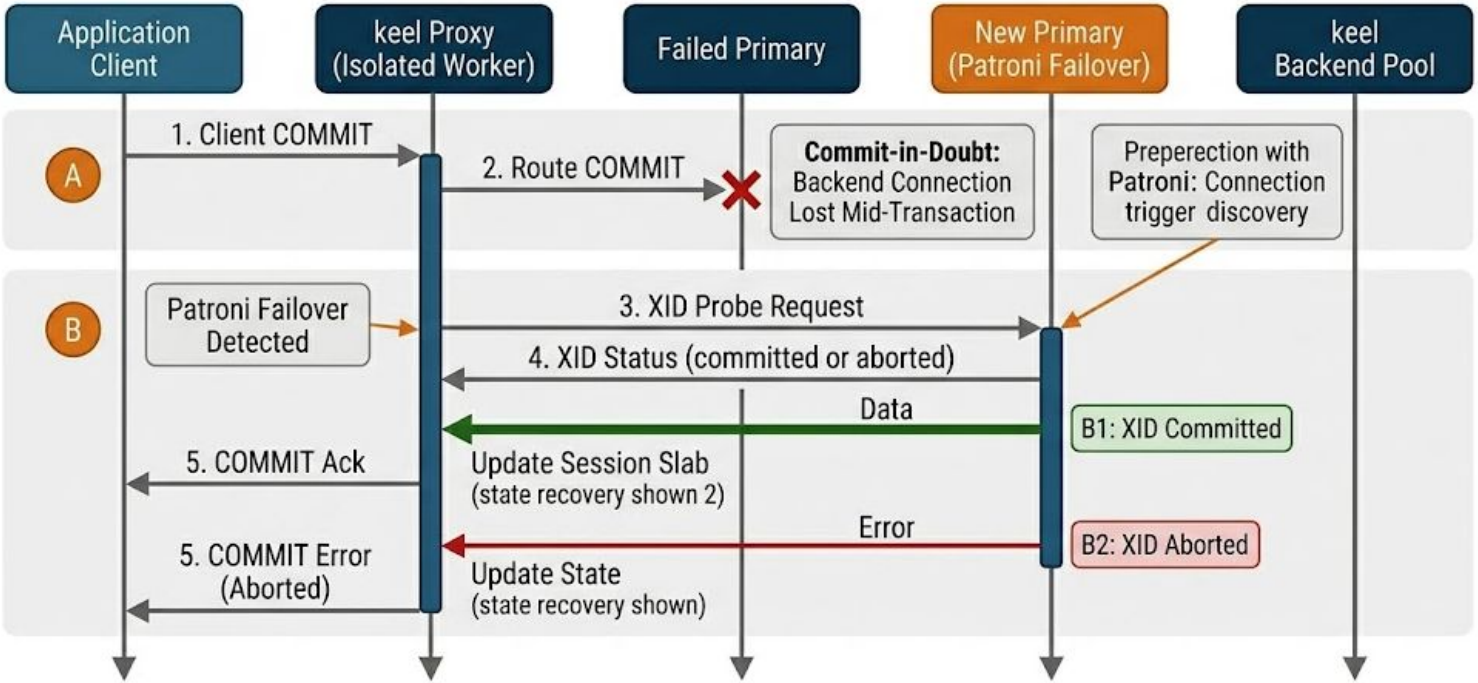
keel Tier Borrowing Strategy



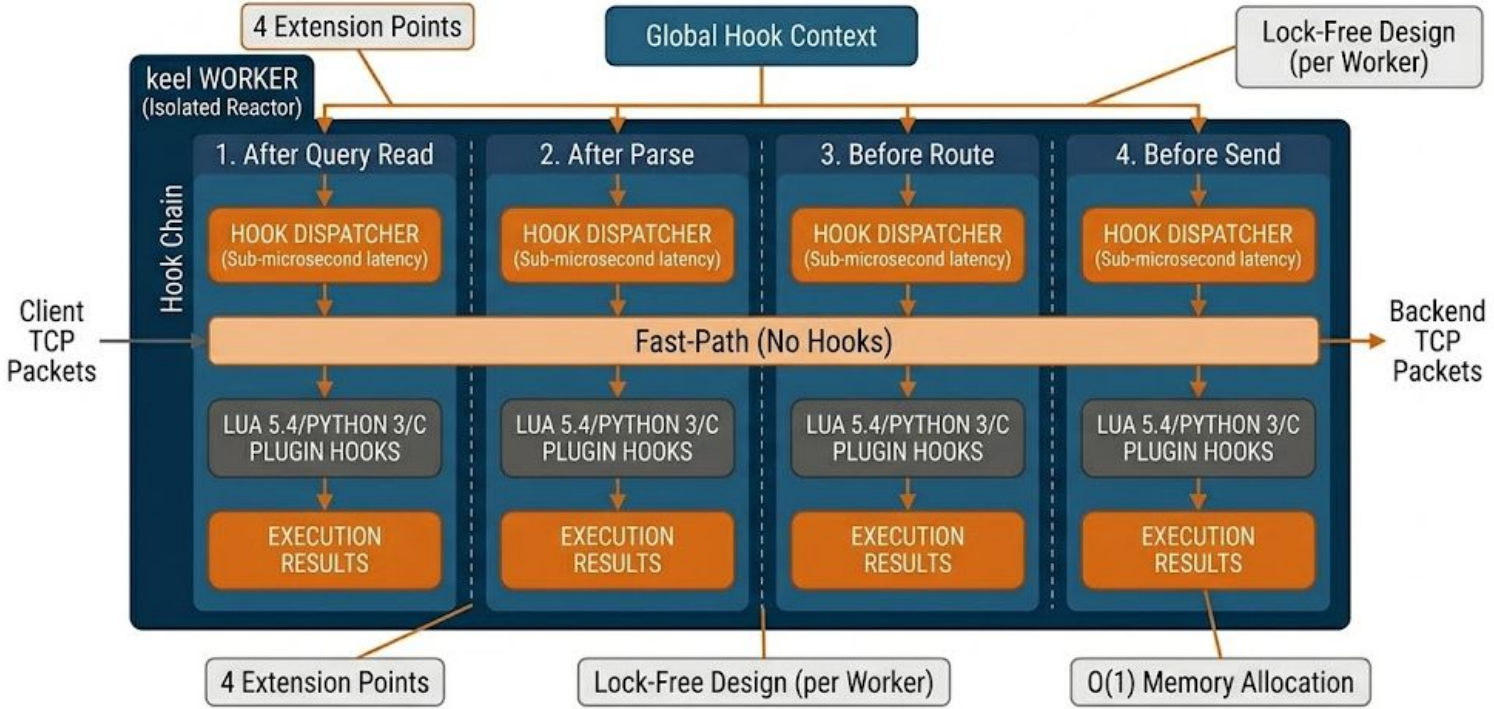
keel Read-After-Write Consistency Flow



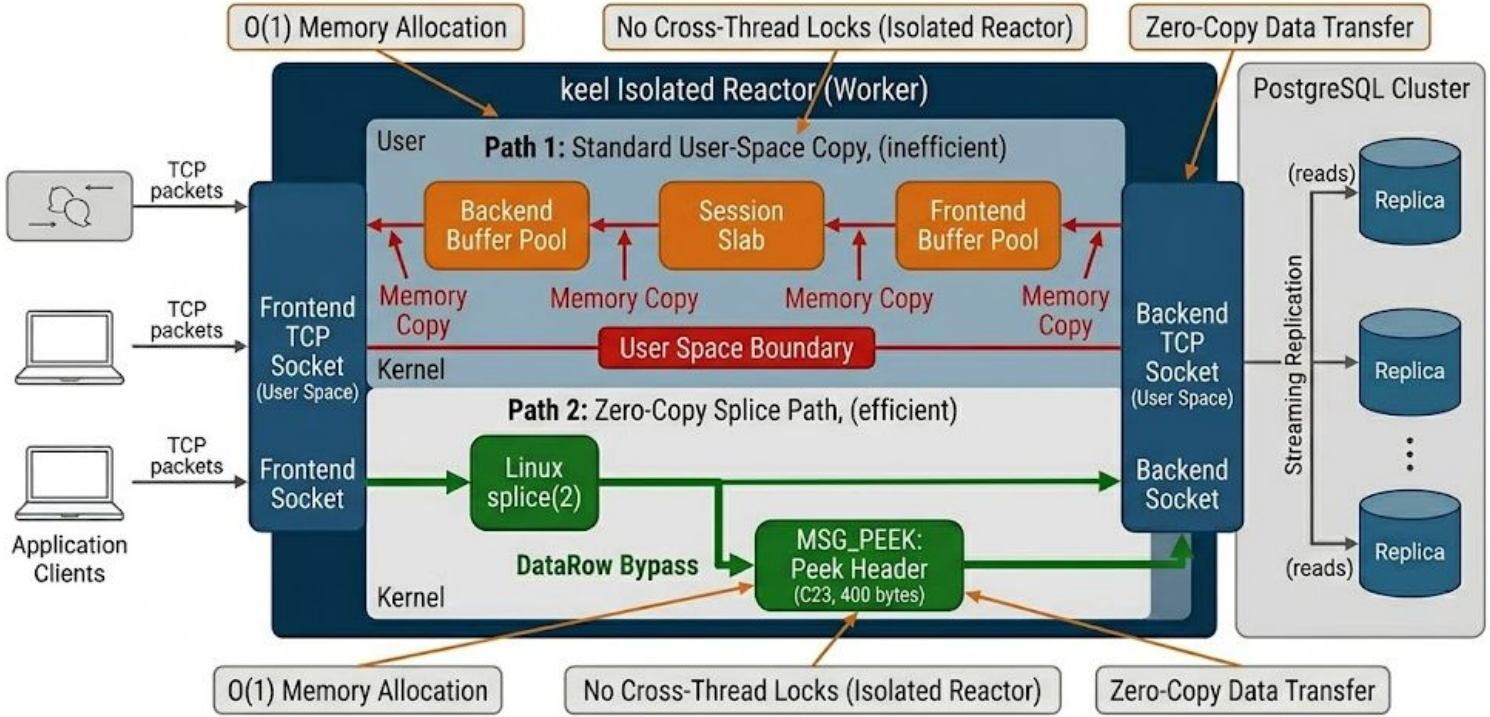
keel Commit-in-Doubt Recovery via XID Probing



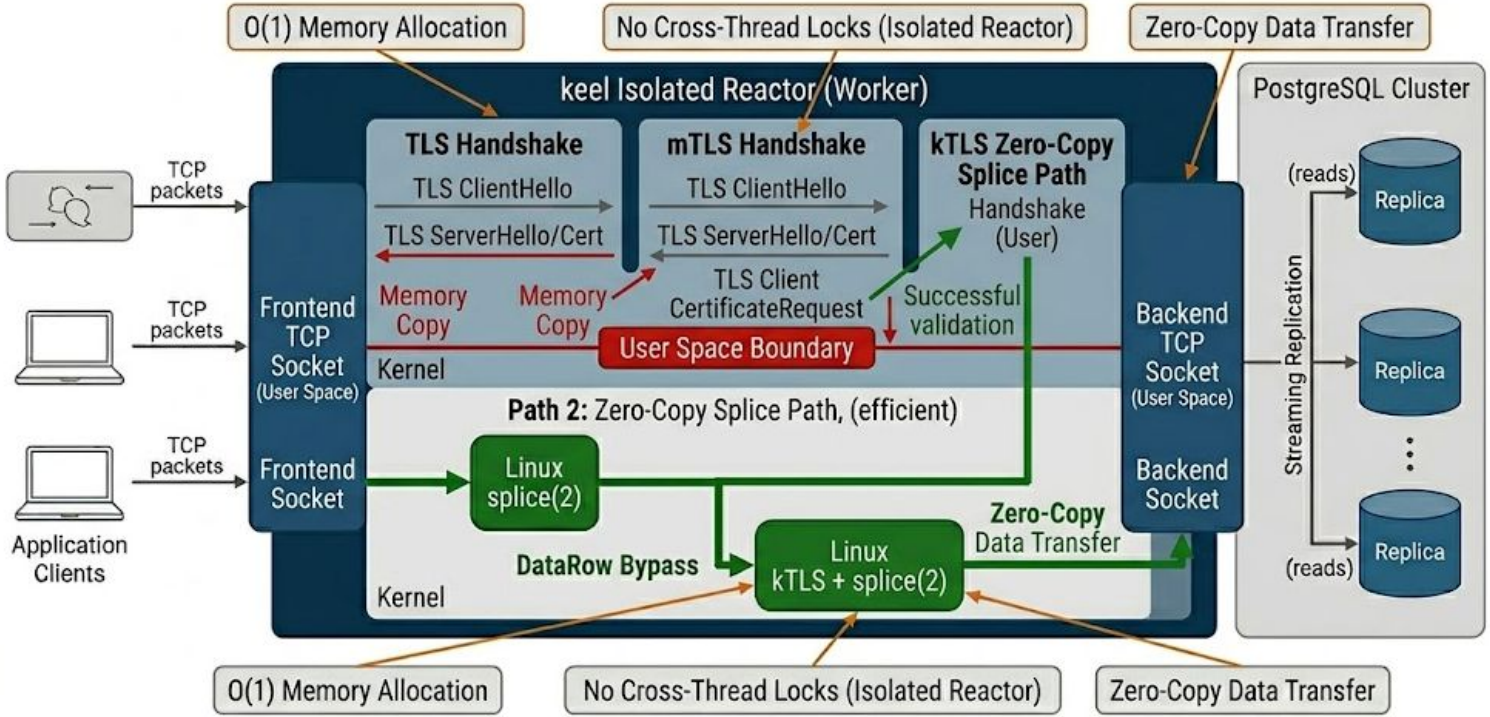
keel Hook/Trigger System and Chain Fast-Path



keel Zero-Copy Splice and MSG_PEEK + Splice DataRow Bypass

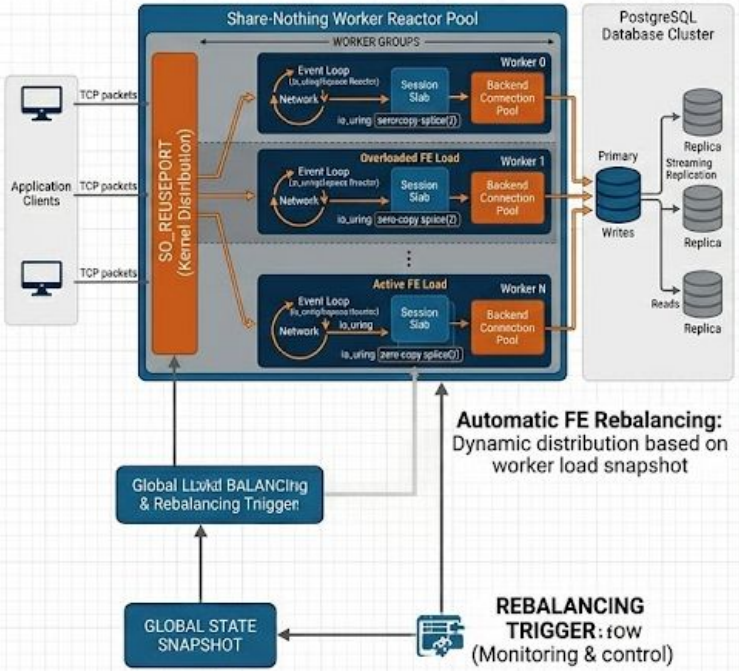


keel keel TLS + mTLS + kTLS

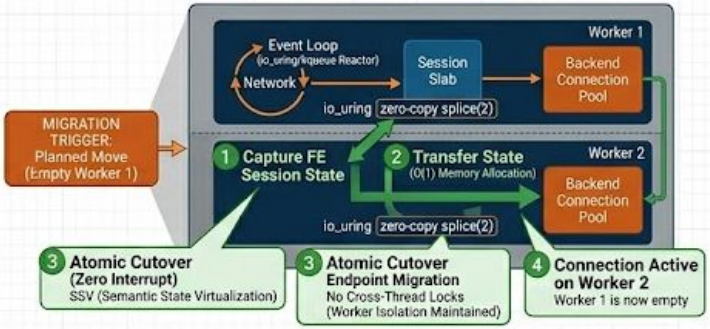


keel Dynamic Connection Migration and Rebalancing Pipeline

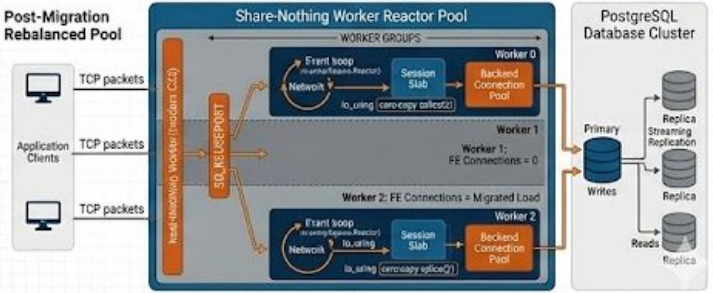
1. Global Load Balancing & Rebalancing Trigger



2. Dynamic Frontend Connection Migration Sequence



3. Rebalanced Pool & Migrated State



Aspect Ratio: 16:9

Work in progress,
first alpha release and full repo soon!

Thank you!

Drop a message: https://www.linkedin.com/in/charlybatista