

04 — Event-sourcing: šta jeste, šta nije

Event-sourcing — razbijanje zablude

Šta JESTE (i zrelo je)

Pravi **event-driven messaging**: `events.db` sa `correlation_id` / `causation_id` / `idempotency_key` UNIQUE / retry state-machine, i **pravi transactional outbox** (`outboxWrite` unutar MC transakcije + `relayOutbox`). Svjesno rješenje dual-write problema koje 90% timova ne uradi kako treba. **Pohvala stoji.**

Šta NIJE

Ovo **NIJE event sourcing**:

- Source-of-truth je mutable `tasks` CRUD tabela
- `replay()` re-isporučuje side-efekte, NE rekonstruiše stanje iz događaja
- `task_history` (67k redova) je audit changelog, ne event log
- `causation_id` je **0% popunjen** — uzročni lanac postoji kao kolona ali je prazan
- 4 "event surface" (`evidence_ledger` / `mc_history` / `hivemind` / `checkpoints`) su nezavisni audit logovi VAN busa

Zabluda koju treba razbiti

"**Trebamo li preći na pravi event-sourcing?**" → **NE**. Task je obična state-mašina (`open`→`started`→`ready`→`done`). Full event-sourcing nad tim = čisti over-engineering — isti metafora-driven nagon koji je proizveo **HexaDB** (bee/6×19 metafora ugurana u DB bez geo-potrebe). Ne pravite drugi HexaDB od event-store-a.

Gdje JEDINO vrijedi formalizovati

Orkestracijske **odluke** — zašto je agent rutirao ovamo, šta je MEHANIČ presudio, koji model, koji verдикт. Te odluke provući kroz **postojeći** bus sa popunjenim correlation_id + causation_id → deterministički replay "zašto je sistem ovo odlučio". Zlato za debug halucinacija i audit. **Ne diraj task-state, ulanči odluke.**

Najveći data-rizik

Nekontrolisani dual-write na ~10 izvora istine bez reconciliation sloja. Konkretno **fire-and-forget MC→Planka** (greška se tiho guta) + ne-idempotentni append logovi. Lijek: provući SVE side-efekte (uklj. Planka) kroz postojeći outbox + periodični reconciliation job. **Imate napola — dovršite, ne gradite novo.**

Revision #1

Created 2026-06-22 09:38:21 UTC by John

Updated 2026-06-22 09:38:21 UTC by John