Release 2.4 Technical Report

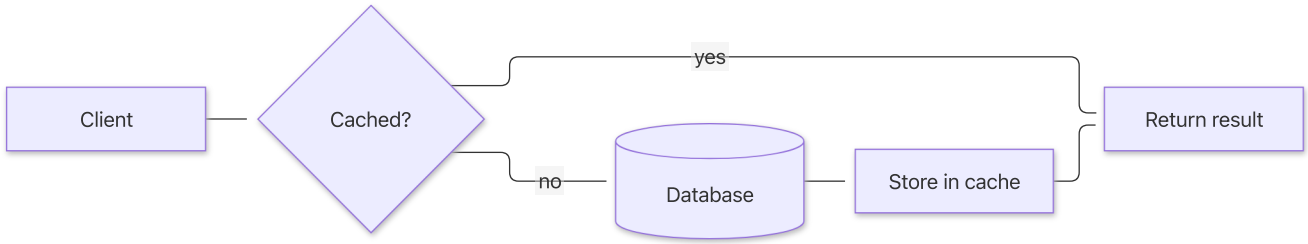
A short technical report that uses every element this converter supports: headings, a table, highlighted code, LaTeX math, a Mermaid diagram, a task list and a manual page break. Edit it or replace it with your own Markdown.

Summary

Area	Before	After	Change
Median API latency	182 ms	121 ms	−33%
Error rate	0.84%	0.31%	−63%
Build time	6 m 40 s	4 m 05 s	−39%

> The biggest gain came from caching the permission lookup, not from the new database index.

Request flow



Cache code

```ts
export async function getPermissions(userId: string) {
  const cached = await cache.get(`perm:${userId}`);
  if (cached) return JSON.parse(cached) as string[];
  const rows = await db.permissions.findMany({ where: { userId } });
  await cache.set(`perm:${userId}`, JSON.stringify(rows), { ttl: 300 });
  return rows.map((row) => row.name);
}
```

Expected hit rate

With n requests per user inside the cache lifetime, the share served from cache is

$$h = 1 - \frac{1}{n}, \qquad n = \frac{T_{ttl}}{\Delta t}$$

so a 300-second lifetime and one request every 15 seconds gives $h = 0.95$.

Follow-up

- [x] Cache permission lookups
- [x] Add latency dashboard
- [ ] Load-test the new index at 2× traffic
- [ ] Remove the old feature flag

Appendix: commands

```
pnpm install
pnpm test --filter api
pnpm build && pnpm start
```