AMAZING CTO

Articles   Get in touch ↗   ✉️ **Newsletter for CTOs**

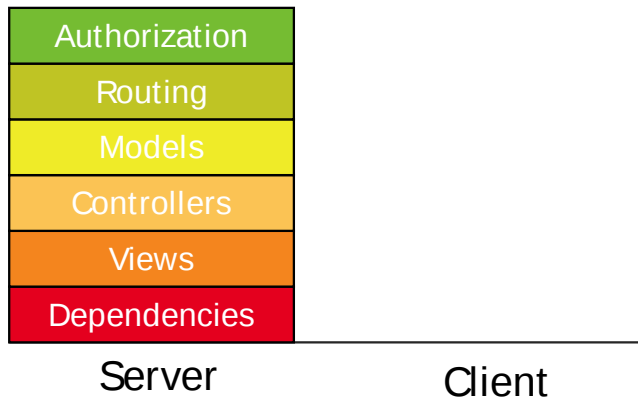*Stephan Schmidt*

Just Use Postgres for Everything

How to reduce complexity and move faster

Welcome HN. Technology is about tradeoffs. Using Postgres for everything is a tradeoff. Of course you use the right tool for the job. Often this is Postgres. Helping dozens of startups I have seen many more people overcomplicate setups than companies that use tools that are too simple for the job. If you have 1M+ customers, and 50+ developers, and you need Kafka and Spark amd Kubernetes, go ahead. If you have more systems than developers, just use Postgres. Thanks to Hugo and BunnyCDN for keeping the page fast. PS: Postgres for everything doesn't mean one server for everything ;-)
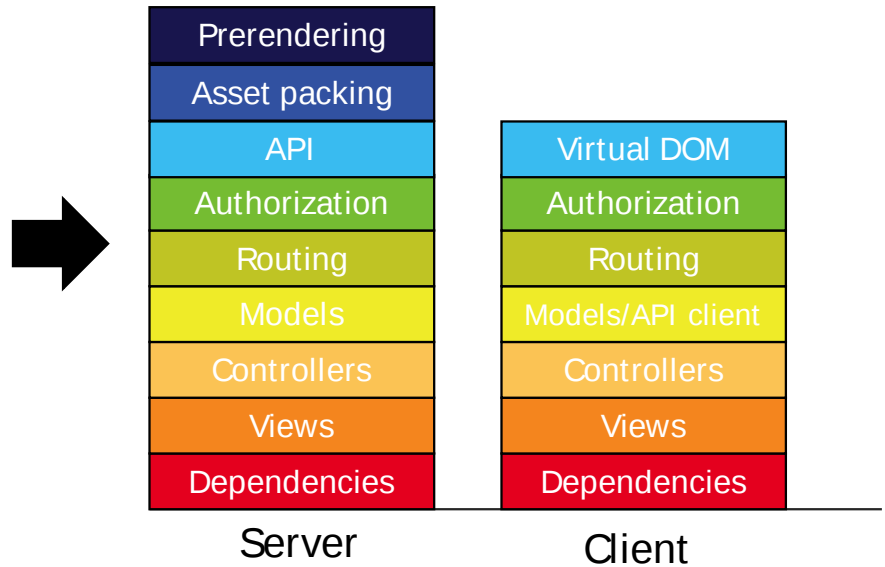
TLDR; just use Postgres for everything.

We have invited complexity through the door. But it will not leave as easily.

Complexity 2005

Complexity 2021

| Prerendering |
| Asset packing |

API	Virtual DOM
Authorization	Authorization
Routing	Routing
Models	Models/API client
Controllers	Controllers
Views	Views
Dependencies	Dependencies

Server — Client

Server — Client

Based on charts by @ryanstout and @triksweline

There is **Radical Simplicity** though.

One way to simplify your stack and reduce the moving parts, speed up development, lower the risk and deliver more features in your startup is **"Use Postgres for everything"**. Postgres can replace - up to millions of users - many backend technologies, Kafka, RabbitMQ, Mongo and Redis among them.

Use Postgres for caching instead of Redis with **UNLOGGED tables** and TEXT as a JSON data type.
Use stored procedures to add and enforce an expiry date for the data just like in Redis.

Use Postgres as a message queue with **SKIP LOCKED** instead of Kafka (if you only need a message queue).

Use Postgres with **TimescaleDB** as a data warehouse.

Use Postgres with **JSONB** to store Json documents in a database, search and index them - instead of Mongo.

Use Postgres as a cron demon to take actions at certain times, like sending mails, with **pg_cron** adding events to a message queue.

Use Postgres for **Geospacial queries**.

Use Postgres for **Fulltext Search** instead of Elastic.

Use Postgres to **generate JSON in the database**, write no server side code and directly give it to the API.

Use Postgres with a **GraphQL adapter** to deliver GraphQL if needed.

There I've said it, **just use Postgres for everything**.

About Stephan

As a CTO, Interim CTO, CTO Coach - and developer - Stephan has seen many technology departments in fast-growing startups. As a kid he taught himself coding in a department store around 1981 because he wanted to write video games. Stephan studied computer science with distributed systems and artificial intelligence at the University of Ulm. He also studied Philosophy. When the internet came to Germany in the 90 he worked as the first coder in several startups. He has

founded a VC funded startup, worked in VC funded, fast growing startups with architecture, processes and growth challenges, worked as a manager for ImmoScout and as a CTO of an eBay Inc. company. After his wife successfully sold her startup they moved to the sea and Stephan took up CTO coaching. You can find him on *LinkedIn*, on *Mastodon* or on Twitter *@KingOfCoders*

Join the AMAZING CTO newsletter like many others!

- Always free.

your email

I agree to receive your newsletters and accept the data privacy statement.

📧 **Signup**

Stephan is using rapidmail to send the newsletter because the email provider is in Germany with strong data protection laws. By registering, you agree that the data you enter may be transmitted to rapidmail. Please note their terms and conditions and data protection .

Other interesting articles for CTOs

• **Best books for CTO** • **CTO versus CTO** • **Experienced CTO Coach** • **CTO Coaching** • **CTO Jobs** • **CTO Mentor** •
CTO Mentoring • **CTO Newsletter** • **CTO Role** • **How many developers do you need?** • **Outsourcing Guide**
Technology and Roadmaps • **How to become a CTO in a company - a career path** •
Work from home and remote with Scrum • **Waterfall and Scrum**