I struggled with system design—until I learned these concepts (it might help you too):

1. Idempotent API:

↳ https://lnkd.in/erMkqwq4

2. Saga Design Pattern:

↳ https://lnkd.in/eFXC4-aJ

3. Protocol Buffers vs JSON:

↳ https://lnkd.in/egcFxe2t

4. Consistent Hashing:

↳ https://lnkd.in/eUP9DbCg

5. Capacity Planning:

↳ https://lnkd.in/ev358pj3

6. Service Discovery:

↳ https://lnkd.in/eCYYwQfU

7. Monolith vs Microservices:

↳ https://lnkd.in/e3EBtg_v

8. Microservices Lessons From Netflix:

↳https://lnkd.in/eZSM3CRB

9. What Happens When You Type a URL Into Your Browser?:

↳ https://lnkd.in/eusuDn5z

10. Caching Patterns:

↳ **https://lnkd.in/gJ8kWMxZ**

11. Modular Monolith Architecture:

↳ **https://lnkd.in/edUqkCKR**

12. How Databases Keep Passwords Securely:

↳ **https://lnkd.in/edd8-N5k**

13. Redis Use Cases:

↳ **https://lnkd.in/ekJMjMG3**

14. How to Scale an App to 10 Million Users on AWS:

↳ **https://lnkd.in/eU736g9Q**

What else should make this list?

─────

👋 PS - Join 100,000 and get the powerful system design template (it's free):

→ **https://lnkd.in/eWAp5ZwE**

─────

♻️ Repost to help others learn system design.

➕ Follow **Neo Kim** for more.