EP 38: Where do we cache data?

ALEX XU
DEC 24, 2022

♡ 167 6 Share ···

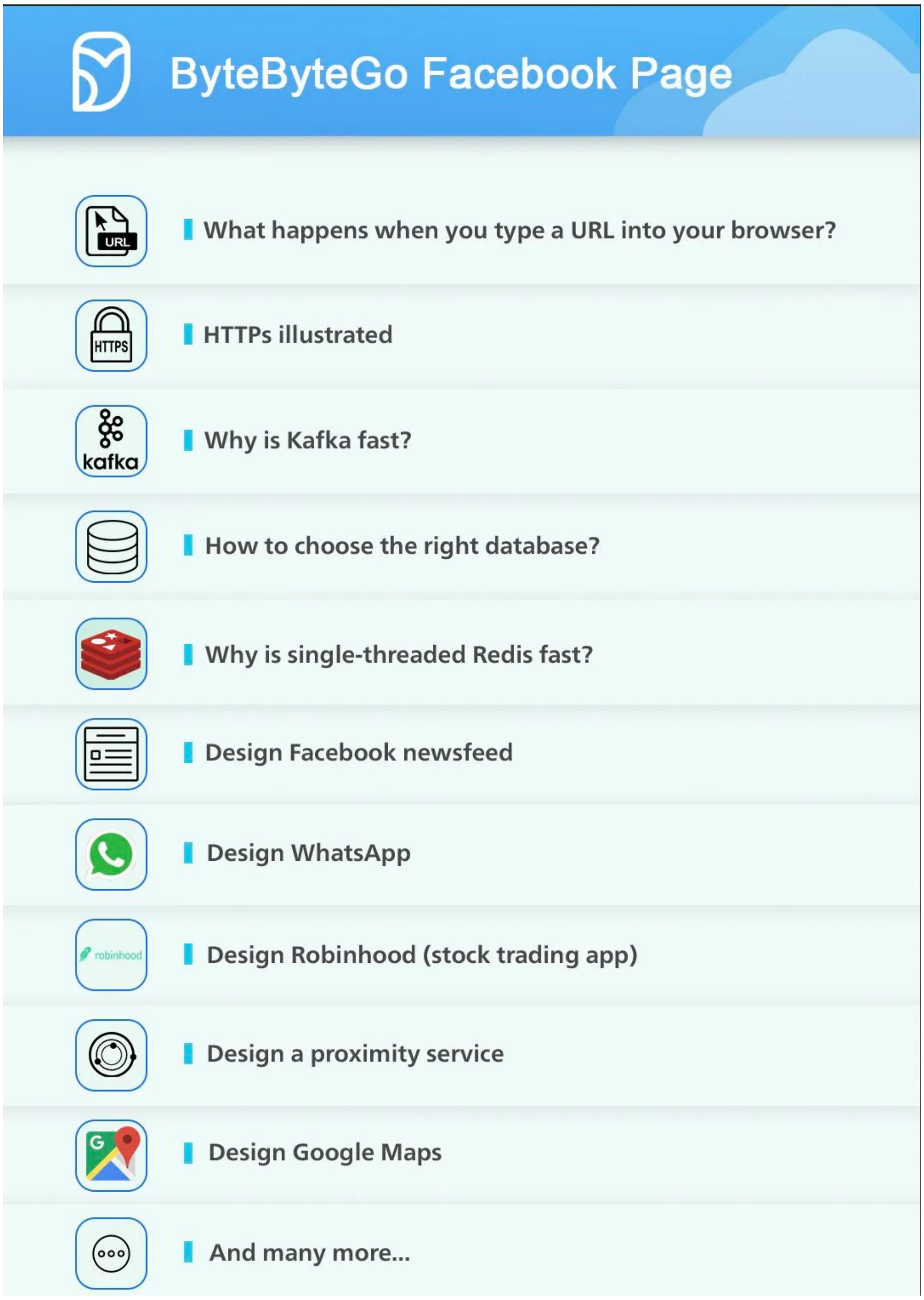
This week's system design refresher:

- ByteByteGo Facebook page
- What is OSI Model? (Youtube video)
- Where do we cache data?
- CI/CD Pipeline
- What tech stack is commonly used for microservices?

ByteByteGo Facebook page

We've launched a Facebook page and want our content to be more accessible.

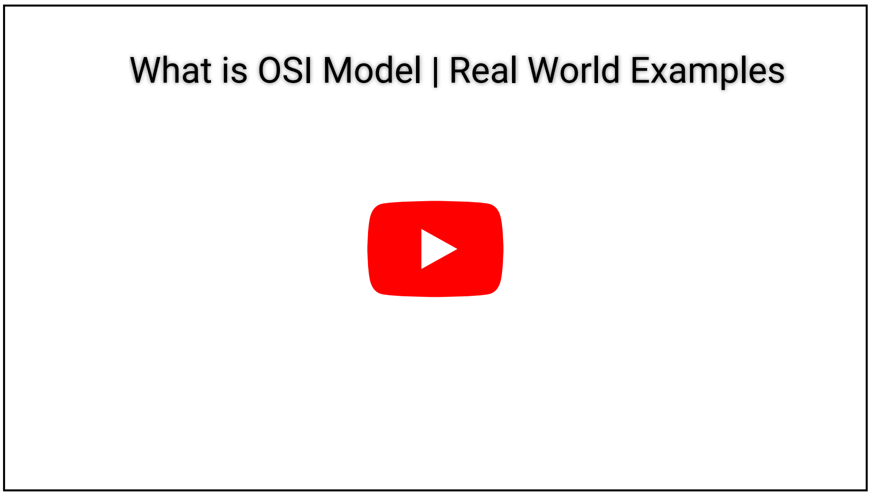
Follow us on FB: https://lnkd.in/eKnvWMx2



ByteByteGo Facebook Page

- What happens when you type a URL into your browser?
- HTTPs illustrated
- Why is Kafka fast?
- How to choose the right database?
- Why is single-threaded Redis fast?
- Design Facebook newsfeed
- Design WhatsApp
- Design Robinhood (stock trading app)
- Design a proximity service
- Design Google Maps
- And many more...

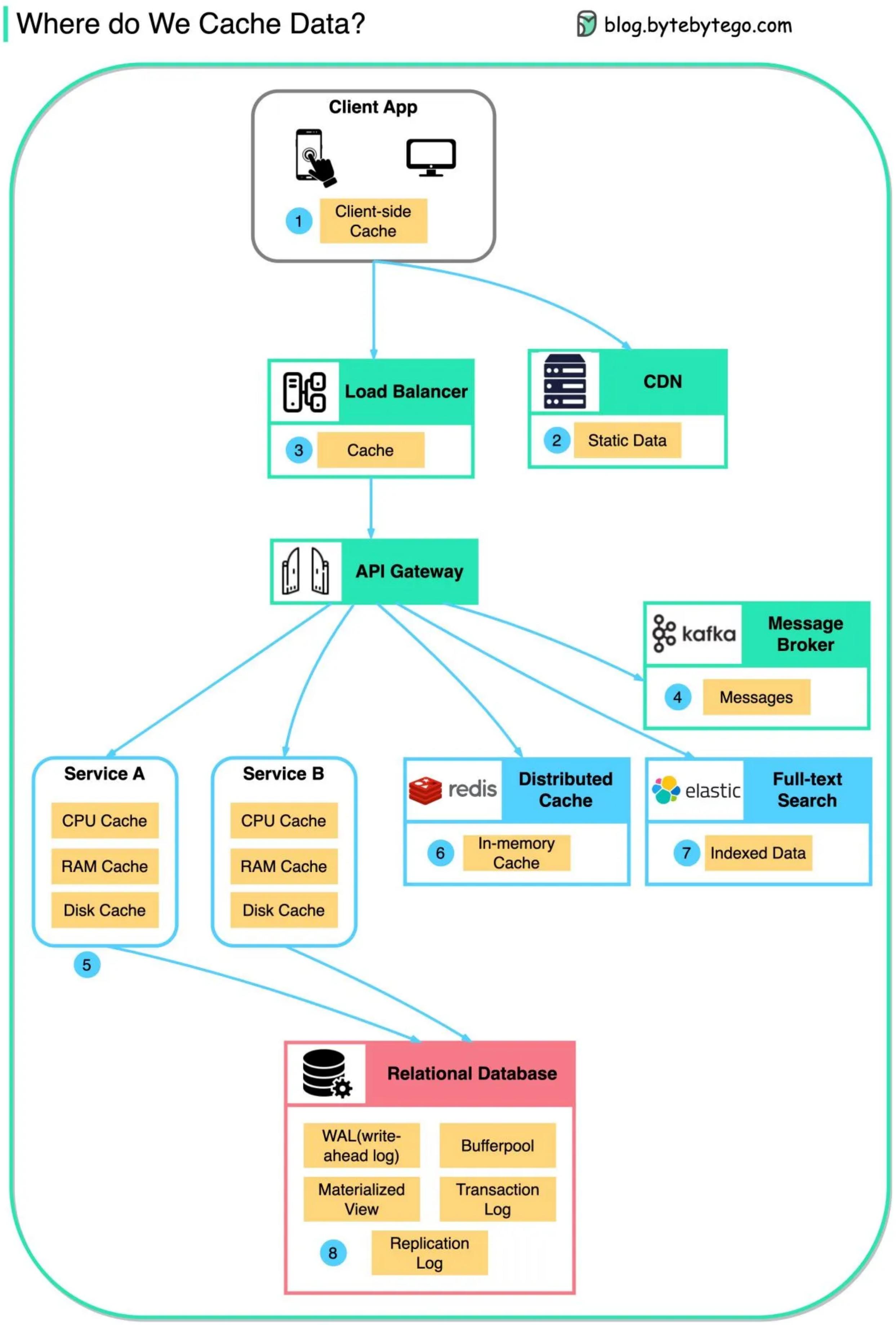
What is OSI Model?



Where do we cache data?

Data is cached everywhere, from the front end to the back end!

This diagram illustrates where we cache data in a typical architecture.



There are **multiple layers** along the flow.

1. **Client apps**: HTTP responses can be cached by the browser. We request data over HTTP for the first time, and it is returned with an expiry policy in the HTTP header; we request data again, and the client app tries to retrieve the data from the browser cache first.
2. **CDN**: CDN caches static web resources. The clients can retrieve data from a CDN node nearby.
3. **Load Balancer**: The load Balancer can cache resources as well.
4. **Messaging infra**: Message brokers store messages on disk first, and then consumers retrieve them at their own pace. Depending on the retention policy, the data is cached in Kafka clusters for a period of time.
5. **Services**: There are multiple layers of cache in a service. If the data is not cached in the CPU cache, the service will try to retrieve the data from memory. Sometimes the service has a second-level cache to store data on disk.
6. **Distributed Cache**: Distributed cache like Redis hold key-value pairs for multiple services in memory. It provides much better read/write performance than the database.
7. **Full-text Search**: we sometimes need to use full-text searches like Elastic Search for document search or log search. A copy of data is indexed in the search engine as well.
8. **Database**: Even in the database, we have different levels of caches:

- WAL(Write-ahead Log): data is written to WAL first before building the B tree index
- Bufferpool: A memory area allocated to cache query results
- Materialized View: Pre-compute query results and store them in the database tables for better query performance
- Transaction log: record all the transactions and database updates
- Replication Log: used to record the replication state in a database cluster

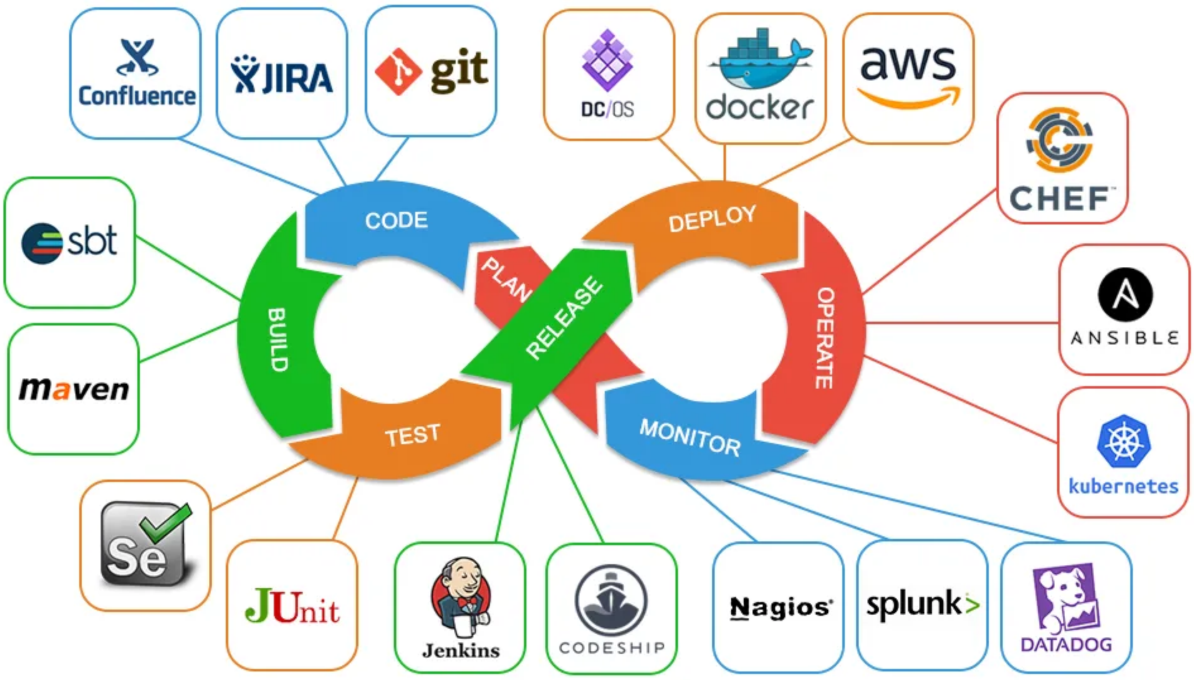
Over to you: With the data cached at so many levels, how can we guarantee the **sensitive user data** is completely erased from the systems?

CI/CD Pipeline

A CI/CD pipeline is a tool that automates the process of building, testing, and deploying software.

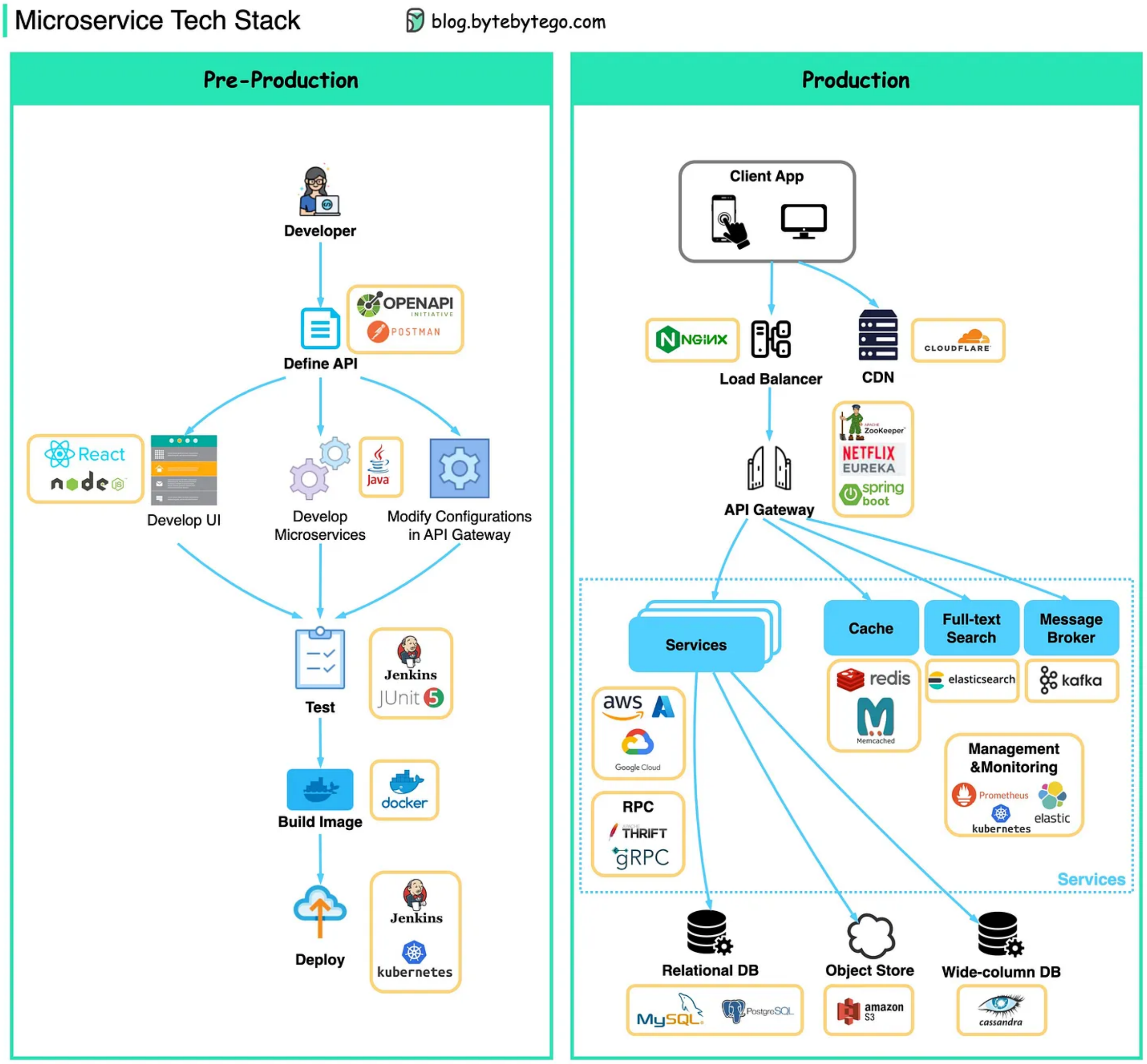
It integrates the different stages of the software development lifecycle, including code creation and revision, testing, and deployment, into a single, cohesive workflow.

The diagram below illustrates some of the tools that are commonly used.



What tech stack is commonly used for microservices?

Below you will find a diagram showing the microservice tech stack, both for the development phase and for production.



Pre-production

- Define API - This establishes a contract between frontend and backend. We can use Postman or OpenAPI for this.
- Development - Node.js or react is popular for frontend development, and java/python/go for backend development. Also, we need to change the configurations in the API gateway according to API definitions.
- Continuous Integration - JUnit and Jenkins for automated testing. The code is packaged into a Docker image and deployed as microservices.

Production

- NGinx is a common choice for load balancers. Cloudflare provides CDN (Content Delivery Network).

- API Gateway - We can use spring boot for the gateway, and use Eureka/Zookeeper for service discovery.
- The microservices are deployed on clouds. We have options among AWS, Microsoft Azure, and Google GCP.
- Cache and Full-text Search - Redis is a common choice for caching key-value pairs. ElasticSearch is used for full-text search.
- Communications - For services to talk to each other, we can use messaging infra Kafka or RPC.
- Persistence - We can use MySQL or PostgreSQL for a relational database, and Amazon S3 for object store. We can also use Cassandra for the wide-column store if necessary.
- Management & Monitoring - To manage so many microservices, the common Ops tools include Prometheus, Elastic Stack, and Kubernetes.

Featured job openings

HEIR: Senior Software Engineer, Full Stack (United States)

167 Likes

6 Comments

Write a comment...

Haris Pasha Dipto Dec 25, 2022

I think "Node.js or react is popular for frontend development" is not absolutely correct as nodejs is mostly used for backend development.

LIKE (5) REPLY SHARE

3 replies

Paul Brobst Writes Paul’s Substack Dec 25, 2022

“With the data cached at so many levels, how can we guarantee the sensitive user data is completely erased from the systems?”

Every cache entry need a time-to-live so it will time out and eventually a be discarded. Ground truth data storage and backups need a different strategy, likely annotations/metadata around which data is deemed sensitive so it can be deleted directly.

♡ LIKE (3) REPLY SHARE ...

4 more comments...