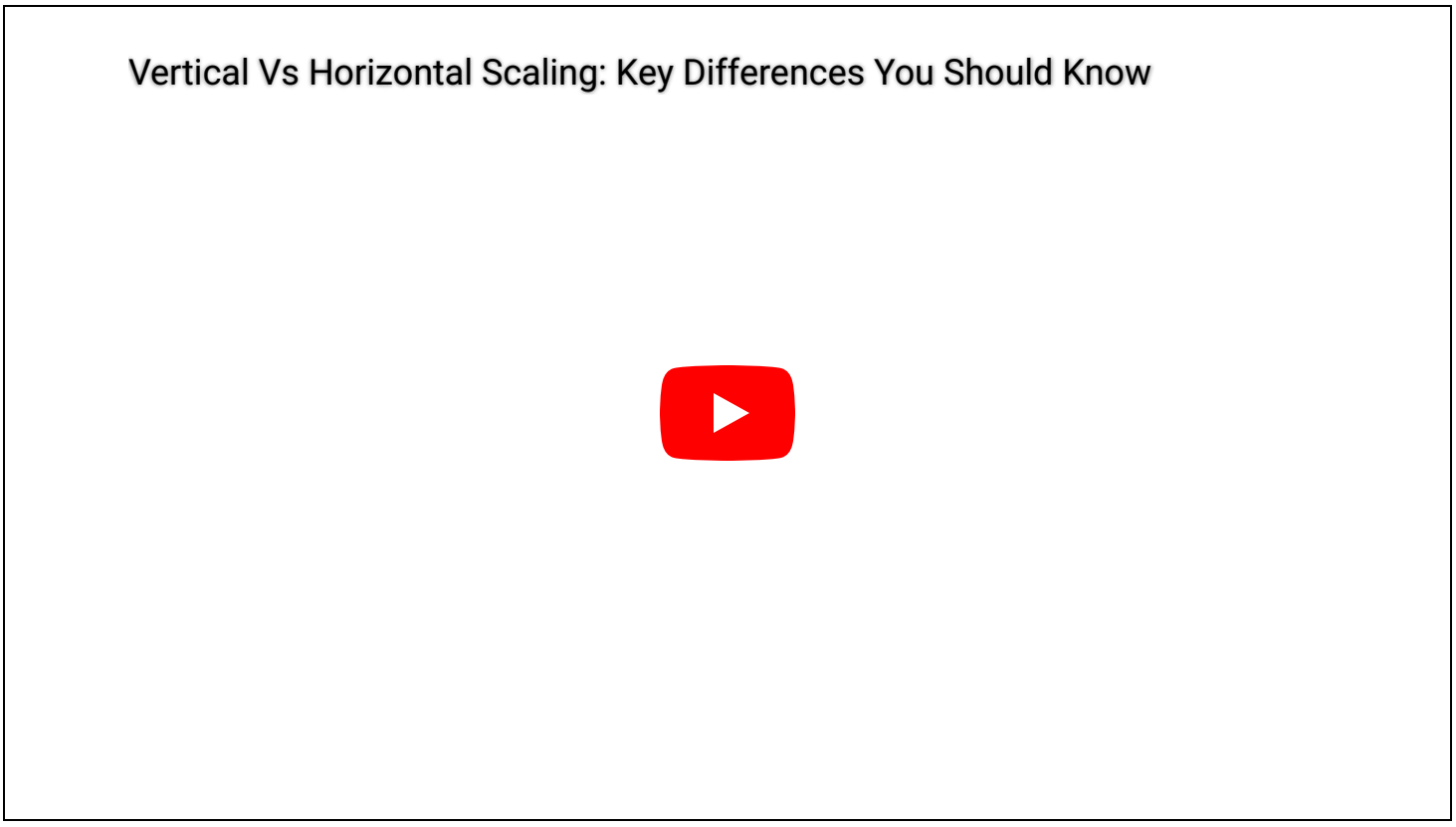
EP91: REST API Authentication Methods

BYTEBYTEGO
DEC 23, 2023

This week's system design refresher:

- Vertical vs Horizontal Scaling (Youtube video)
- 9 of my favorite engineering blogs
- REST API Authentication Methods
- Symmetric encryption vs asymmetric encryption
- How does Redis persist data?

Vertical Vs Horizontal Scaling: Key Differences You Should Know



Top 9 Engineering blog favorites

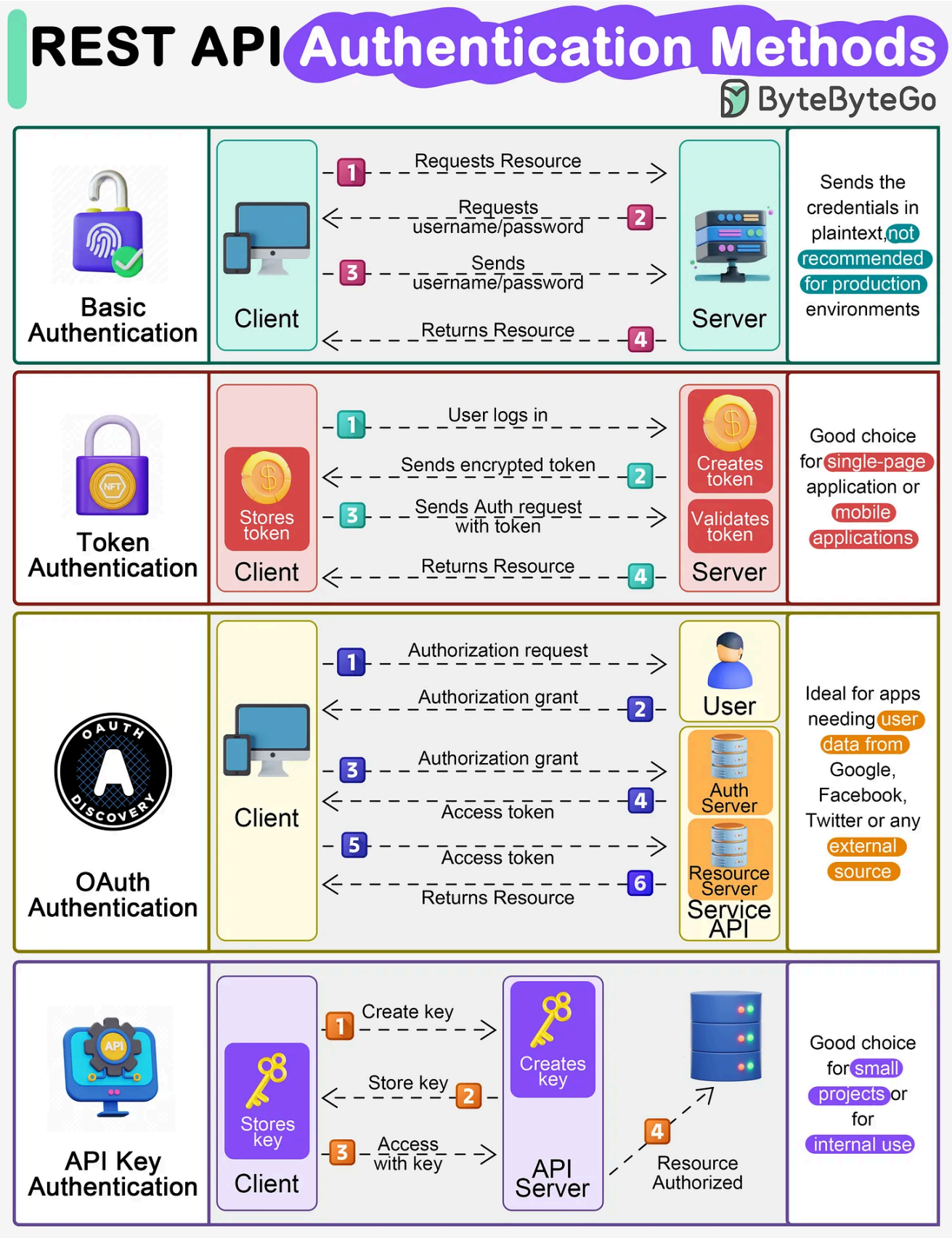
There are over 1,000 engineering blogs. Here are my top 9 favorites:

- Netflix TechBlog
- Uber Blog
- Cloudflare Blog
- Engineering at Meta
- LinkedIn Engineering
- Discord Blog
- AWS Architecture
- Slack Engineering
- Stripe Blog

Over to you - What are some of your favorite engineering blogs?

REST API Authentication Methods

Authentication in REST APIs acts as the crucial gateway, ensuring that solely authorized users or applications gain access to the API's resources.



Some popular authentication methods for REST APIs include:

1. Basic Authentication:
   Involves sending a username and password with each request, but can be less secure without encryption.

When to use:
Suitable for simple applications where security and encryption aren't the primary concern or when used over secured connections.

2. Token Authentication:
   Uses generated tokens, like JSON Web Tokens (JWT), exchanged between client and server, offering enhanced security without sending login credentials with each request.

   When to use:
   Ideal for more secure and scalable systems, especially when avoiding sending login credentials with each request is a priority.

3. OAuth Authentication:
   Enables third-party limited access to user resources without revealing credentials by issuing access tokens after user authentication.

   When to use:
   Ideal for scenarios requiring controlled access to user resources by third-party applications or services.

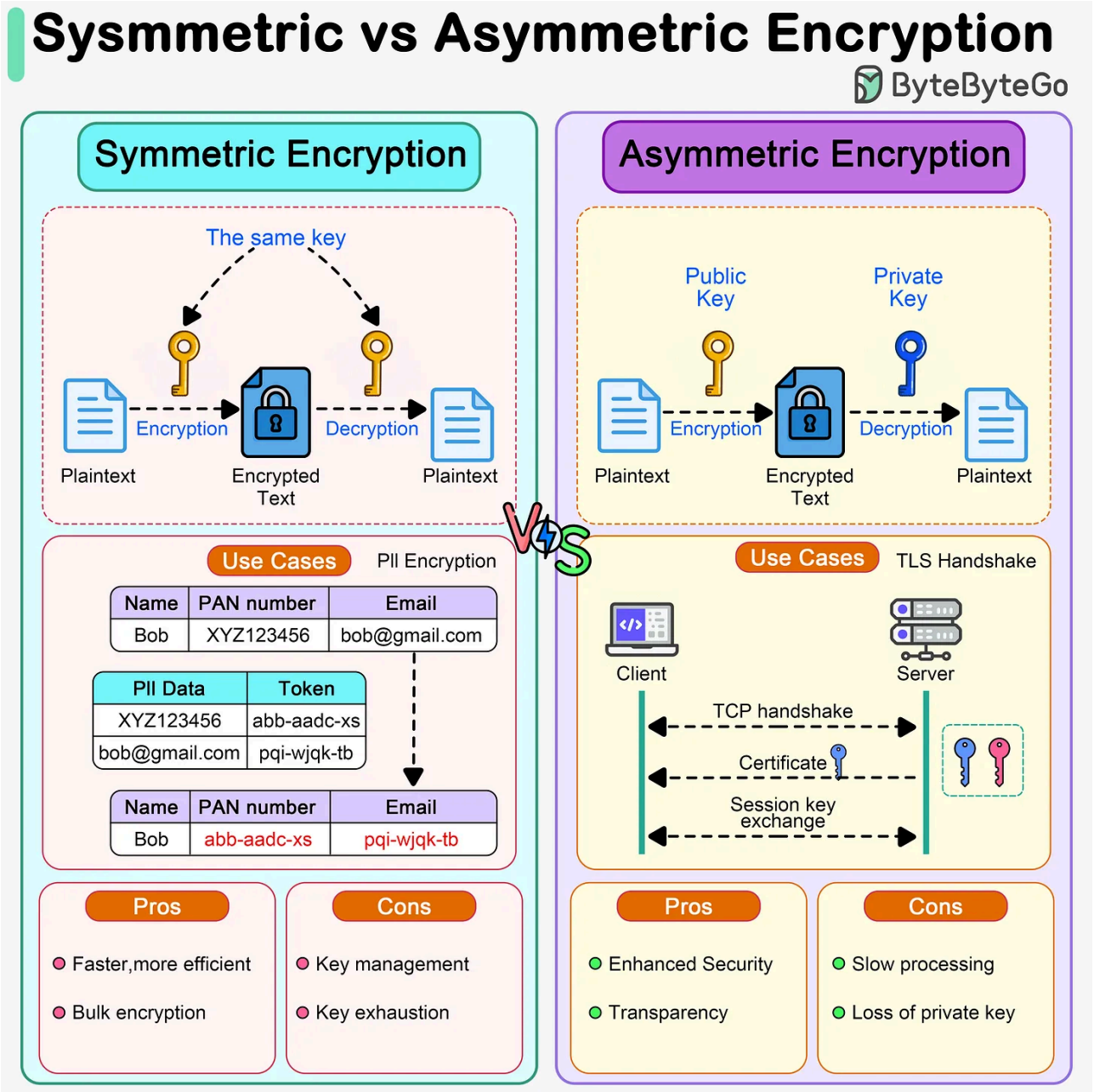
4. API Key Authentication:
   Assigns unique keys to users or applications, sent in headers or parameters; while simple, it might lack the security features of token-based or OAuth methods.

   When to use:
   Convenient for straightforward access control in less sensitive environments or for granting access to certain functionalities without the need for user-specific permissions.

Over to you: Which REST API authentication method do you find most effective in ensuring both security and usability for your applications?

Symmetric encryption vs asymmetric encryption

Symmetric encryption and asymmetric encryption are two types of cryptographic techniques used to secure data and communications, but they differ in their methods of encryption and decryption.



- In symmetric encryption, a single key is used for both encryption and decryption of data. It is faster and can be applied to bulk data encryption/decryption. For example, we can use it to encrypt massive amounts of PII (Personally Identifiable Information) data. It poses challenges in key management because the sender and receiver share the same key.
- Asymmetric encryption uses a pair of keys: a public key and a private key. The public key is freely distributed and used to encrypt data, while the private key is kept secret and used to decrypt the data. It is more secure than symmetric encryption because the private key is never shared. However, asymmetric encryption is slower because of the complexity of key generation and maths computations. For example, HTTPS uses asymmetric encryption to exchange session keys during TLS handshake, and after that, HTTPS uses symmetric encryption for subsequent communications.

How does Redis persist data?

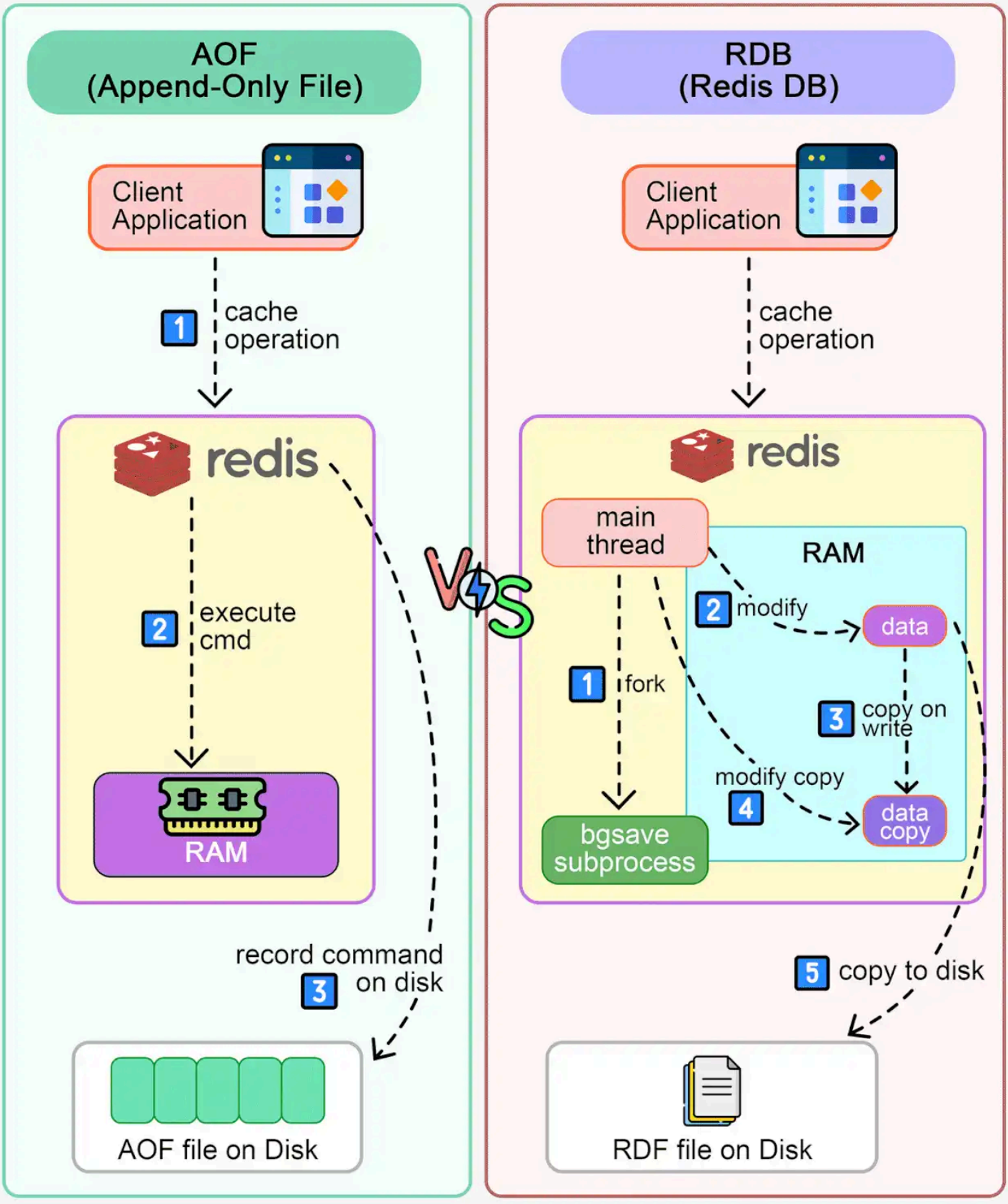
Redis is an in-memory database. If the server goes down, the data will be lost.

The diagram below shows two ways to persist Redis data on disk:

1. AOF (Append-Only File)
2. RDB (Redis Database)

How Redis Persists Data?

ByteByteGo



Note that data persistence is not performed on the critical path and doesn't block the write process in Redis.

- AOF
  Unlike a write-ahead log, the Redis AOF log is a write-after log. Redis executes commands to modify the data in memory first and then writes it to the log file. AOF log records the commands instead of the data. The event-based design simplifies data recovery. Additionally, AOF records commands after the command has been executed in memory, so it does not block the current write operation.
- RDB
  The restriction of AOF is that it persists commands instead of data. When we use the AOF log for recovery, the whole log must be scanned. When the size of the log is large, Redis takes a long time to recover. So Redis provides another way to persist data - RDB.

  RDB records snapshots of data at specific points in time. When the server needs to be recovered, the data snapshot can be directly loaded into memory for fast recovery.

  Step 1: The main thread forks the‘ bgsave’ sub-process, which shares all the in-memory data of the main thread. ‘bgsave’ reads the data from the main thread and writes it to the RDB file.

  Steps 2 and 3: If the main thread modifies data, a copy of the data is created.

  Steps 4 and 5: The main thread then operates on the data copy. Meanwhile ‘bgsave’ sub-process continues to write data to the RDB file.
- Mixed
  Usually in production systems, we can choose a mixed approach, where we use RDB to record data snapshots from time to time and use AOF to record the commands since the last snapshot.

8 Comments

Write a comment...

Name Dec 23, 2023

I love your diagrams? Do you use specific tool or?

LIKE (8) REPLY SHARE

Big Tech Digest Big Tech Digest Dec 24, 2023 · *edited Dec 24, 2023*

Thanks for sharing your favourite engineering blogs! Shameless plug: I'm building a Substack newsletter called "Big Tech Digest" where I aggregate links to the latest articles from over 300 Big Tech and startup engineering blogs like Meta, Google, Uber, Airbnb, Doordash, and more. I'm sending them out every two weeks for free with a short summary.

LIKE (3) REPLY SHARE

6 more comments...