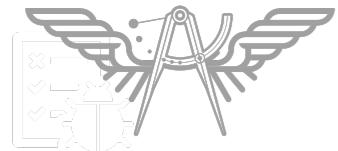


Redis Workshop





Page

Messages

Notifications 3

Insights

Publishing Tools

Settings

Help ▾



somkiat.cc

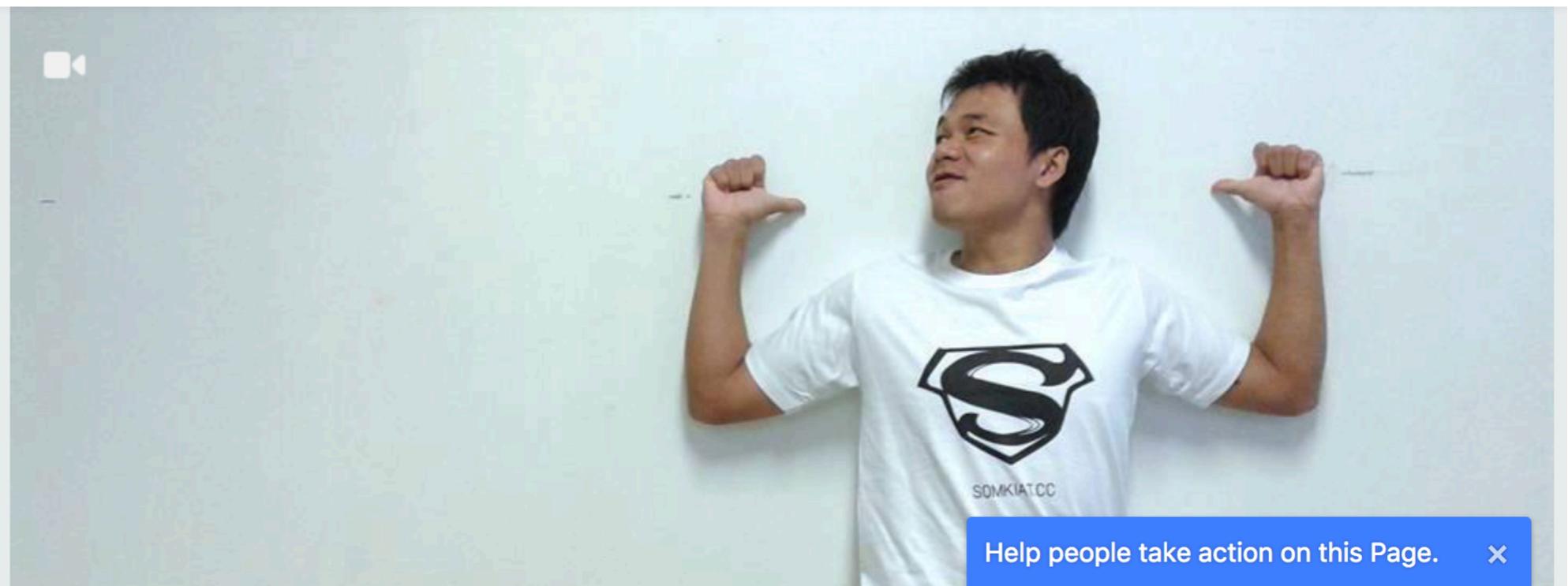
@somkiat.cc

Home

Posts

Videos

Photos



Help people take action on this Page. 

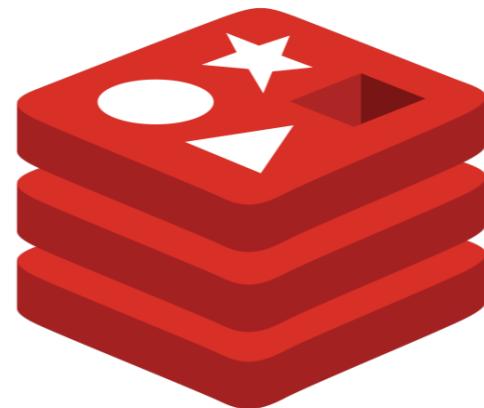
+ Add a Button



Redis

3

Redis Workshop



<https://github.com/up1/course-redis>



Topics (1)

Database models

Why and what Redis ?

Think about Redis

NoSQL database

Working with Redis

Use cases



Topics (2)

Data types

Data modeling

Redis deployment topologies

Monitoring

Performance tuning and benchmark



Redis



Products ▾ Resources ▾ Docs Pricing



Login Book a meeting

Try Redis

YOUR APP IS ABOUT
TO GET FASTER

Deliver faster **experiences** with the fastest cache.



<https://redis.io/>



Redis

© 2020 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Redis

REmote DIctionary Server

Key-value database

Keep data in memory (save to disk)

Expiration data

Data structure server



Database Ranking

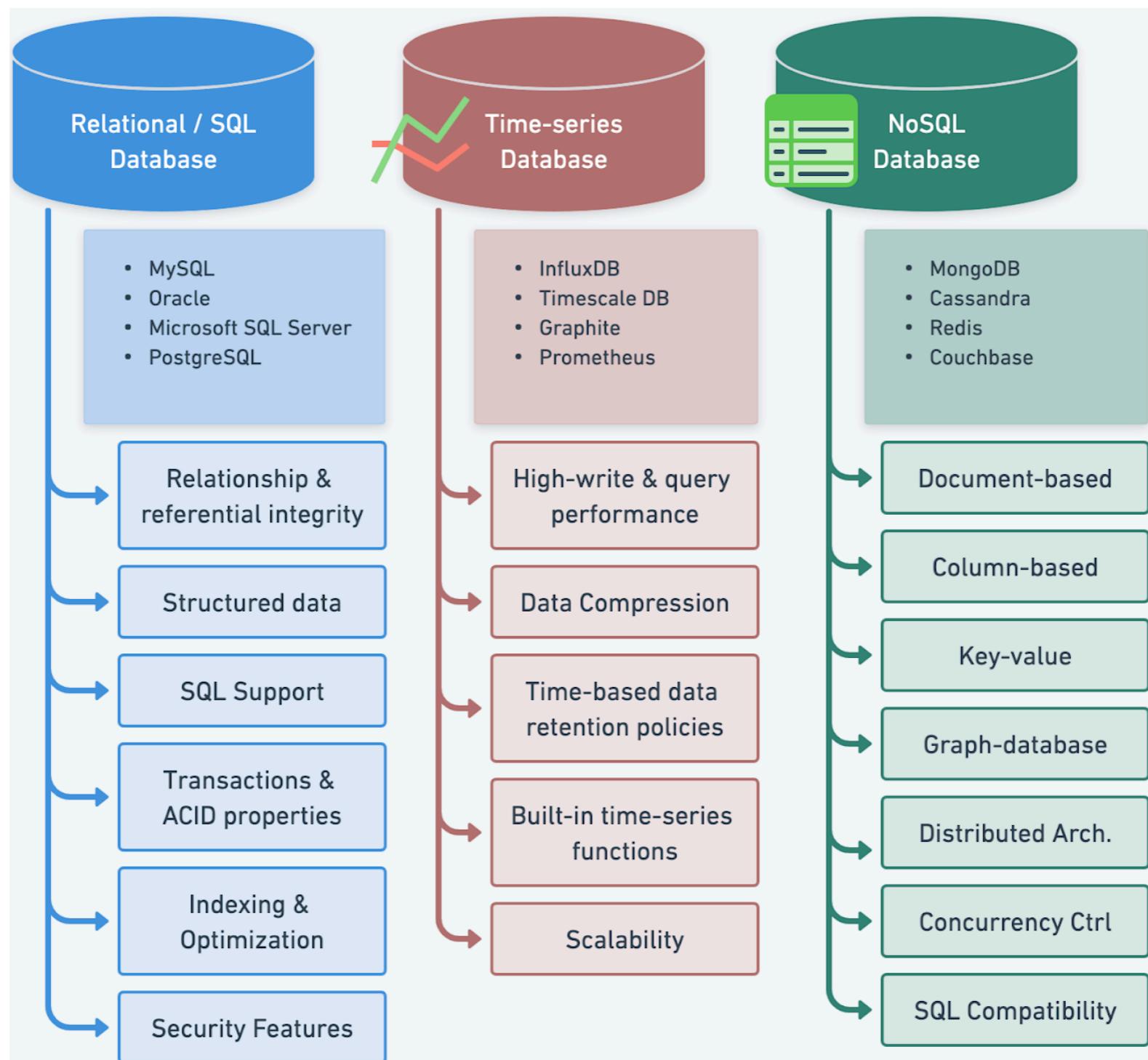
426 systems in ranking, December 2025

Rank			DBMS	Database Model	Score		
Dec 2025	Nov 2025	Dec 2024			Dec 2025	Nov 2025	Dec 2024
1.	1.	1.	Oracle	Relational, Multi-model i	1234.40	-5.38	-29.39
2.	2.	2.	MySQL	Relational, Multi-model i	868.49	+2.67	-135.27
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model i	722.52	+3.65	-83.18
4.	4.	4.	PostgreSQL	Relational, Multi-model i	659.42	+8.06	-6.95
5.	5.	5.	MongoDB +	Multi-model i	372.27	+0.60	-28.11
6.	6.	↑7.	Snowflake	Relational	202.45	+4.61	+55.09
7.	7.	↓6.	Redis	Key-value, Multi-model i	142.48	-2.61	-7.79
8.	8.	↑13.	Databricks	Multi-model i	138.02	+6.51	+50.33
9.	9.	9.	IBM Db2	Relational, Multi-model i	115.75	-3.52	-7.03
10.	10.	↓8.	Elasticsearch	Multi-model i	109.83	-4.14	-22.49

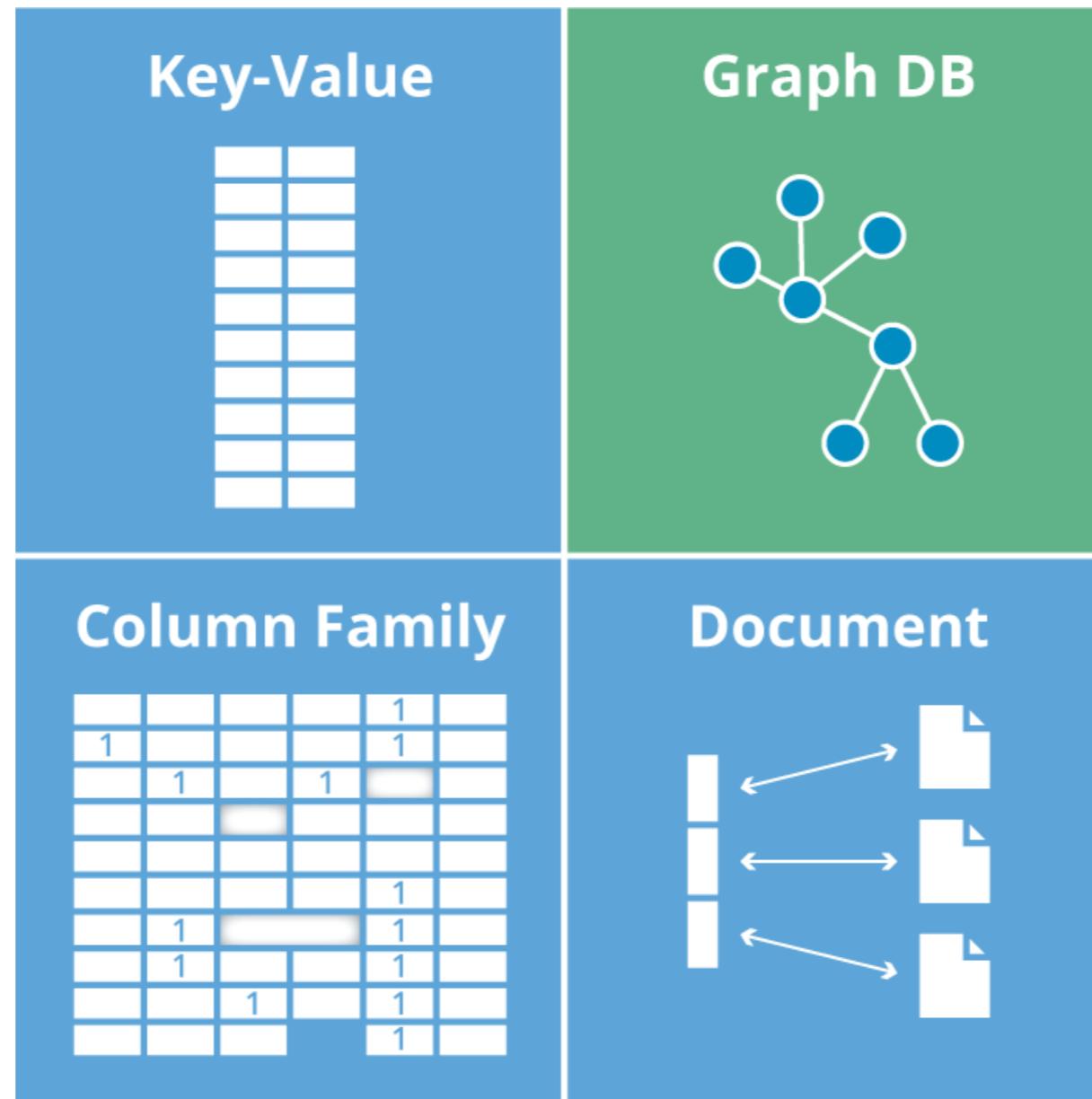
<https://db-engines.com/en/ranking>



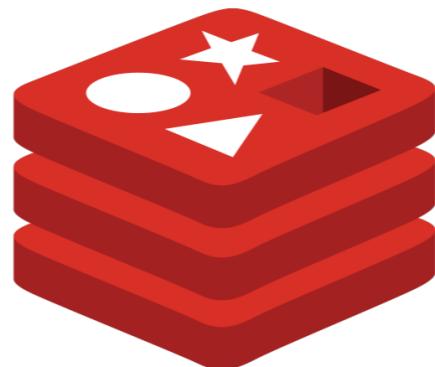
Type of databases



NoSQL database



Key-value databases

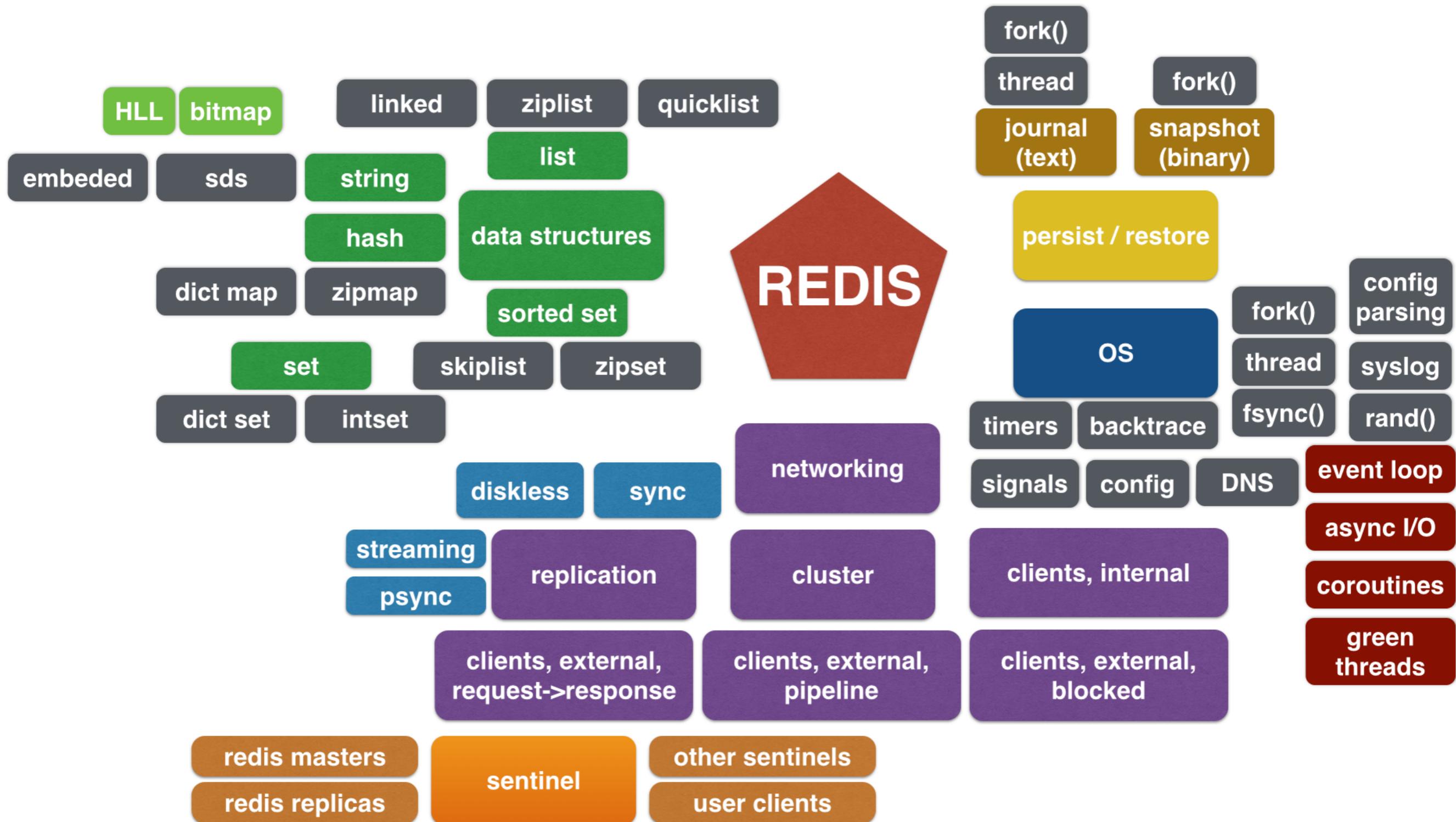


Valkey



Learn Redis ?

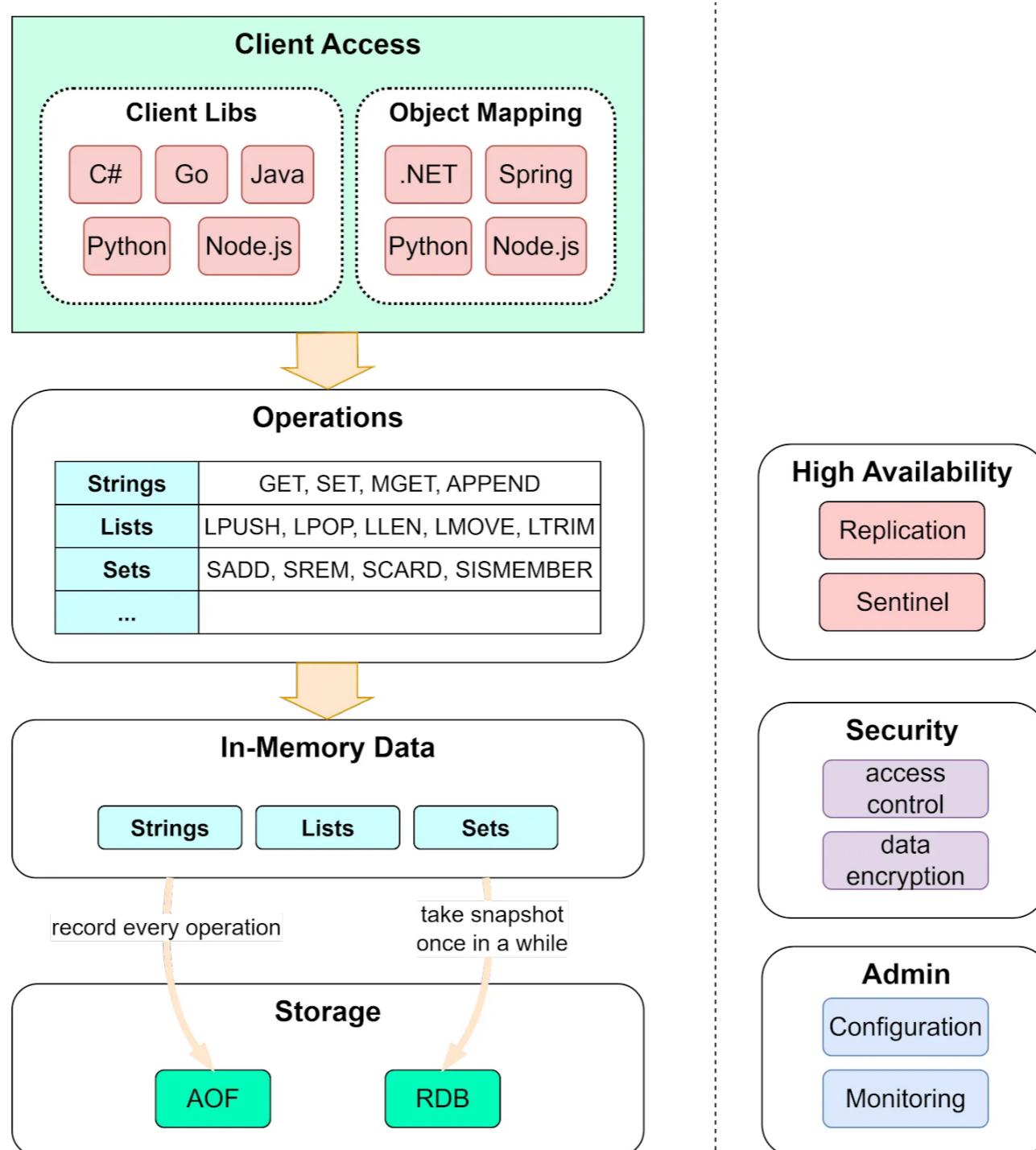




High level architecture



High Level View of Redis



<https://blog.bytebytogo.com/p/a-crash-course-in-redis>



Why Redis Fast ?



Why Redis Fast ?

RAM-based database
I/O multiplexing and single-thread read/write
Efficient low-level data structure

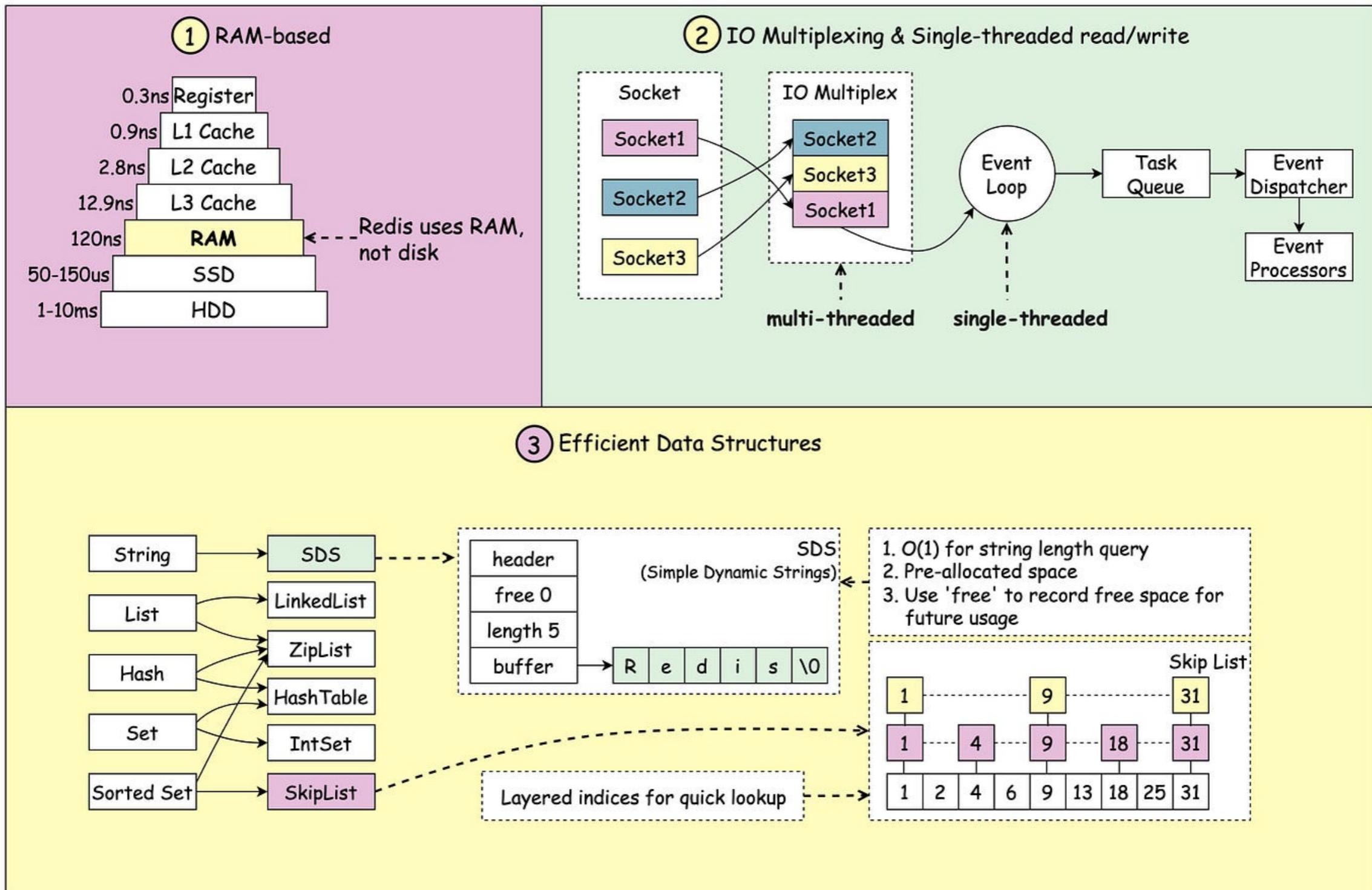
Storage	Access time
CPU cache	1 nanosecond
RAM	100 nanoseconds
SSD	100 microseconds
HDD	10 microseconds



Why is Redis so fast?

100k QPS

 blog.bytebytego.com



<https://blog.bytebytego.com/p/why-is-redis-so-fast>



Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Advantage of Single thread

No context switching

No synchronization required

Minimal overhead



Disadvantage of Single thread

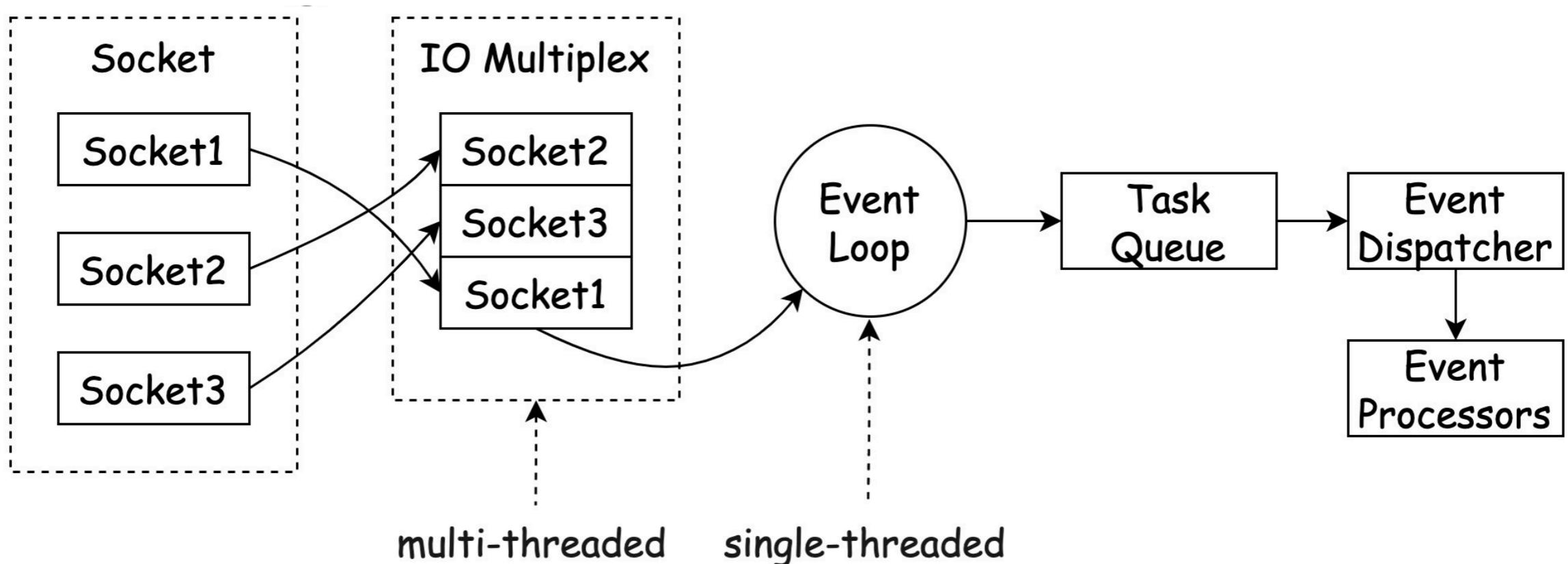
Poor CPU utilization

Scalability limitation

Unresponsiveness for blocking operation



I/O multiplexing and single-thread read/write



Atomic command and Data consistency

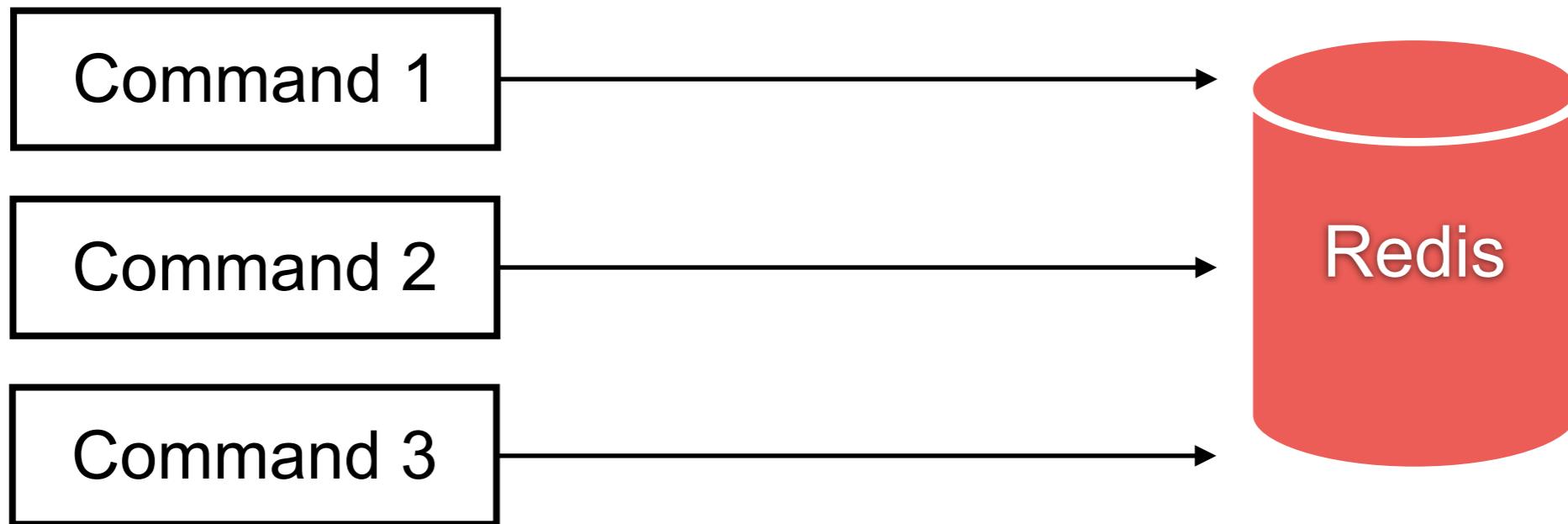
Each command is **atomic**

Eliminate the risk of race condition



Reduce network latency

Working with pipeline in Redis

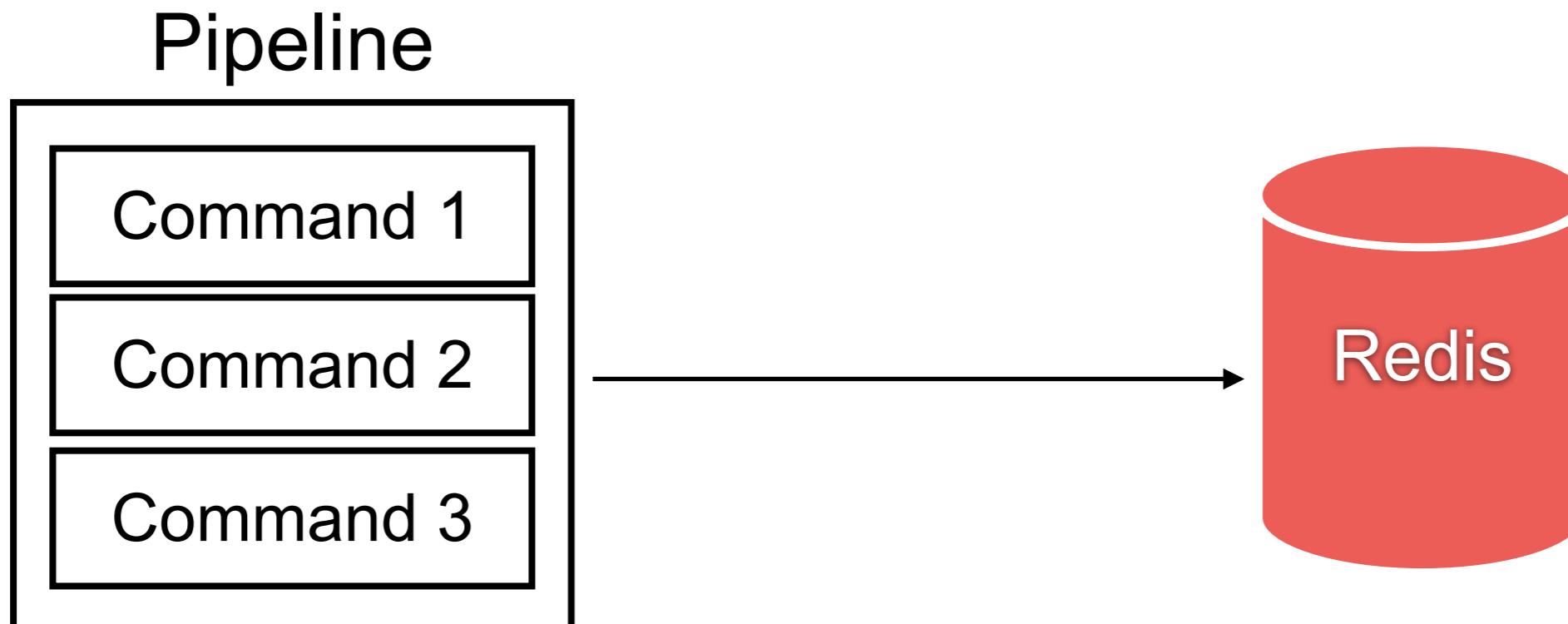


<https://redis.io/docs/latest/develop/use/pipelining/>



Reduce network latency

Working with pipeline in Redis
Reduce network overhead

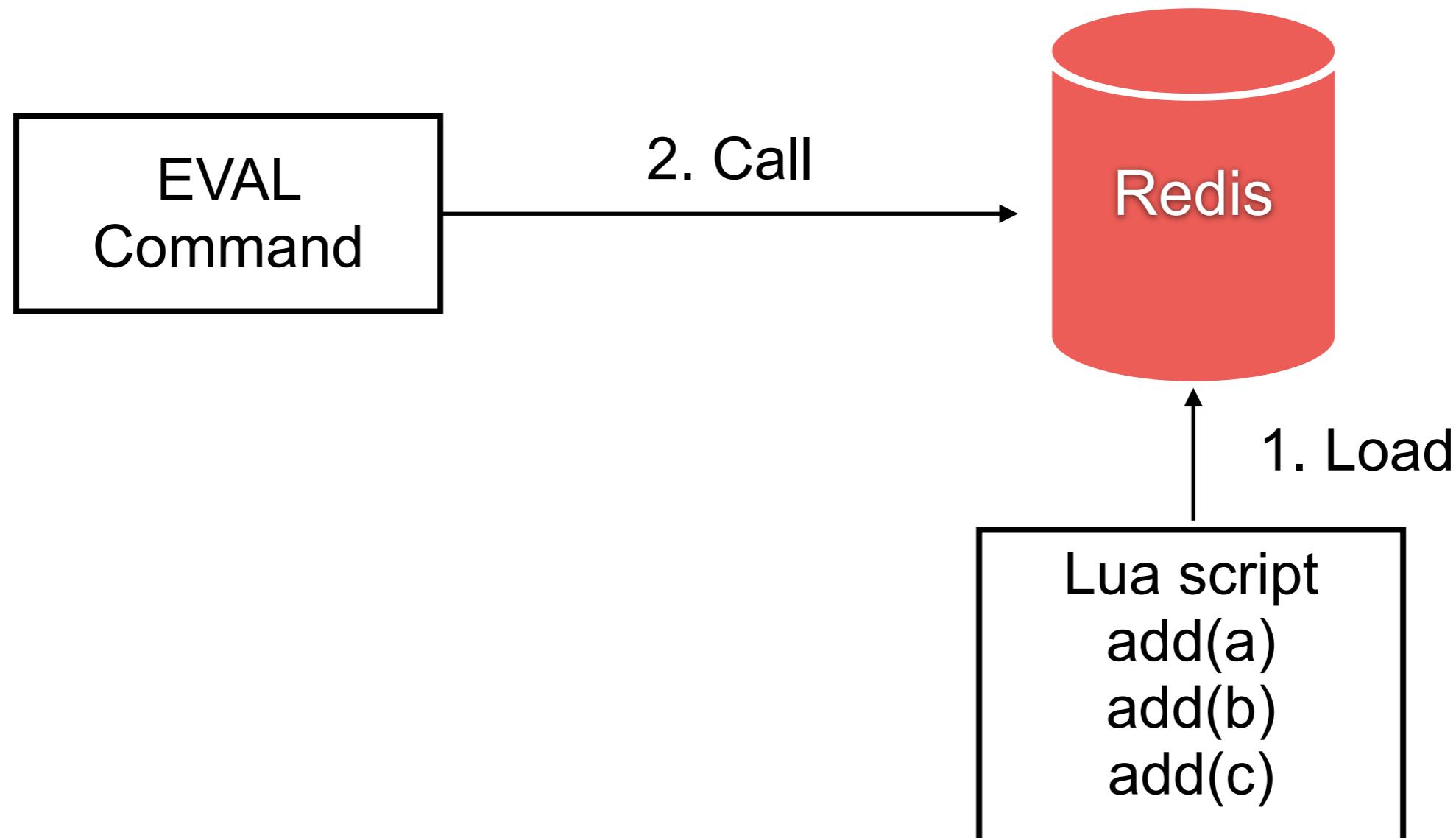


<https://redis.io/docs/latest/develop/use/pipelining/>



Reduce network latency

Write Lua script in Redis (like store procedure)

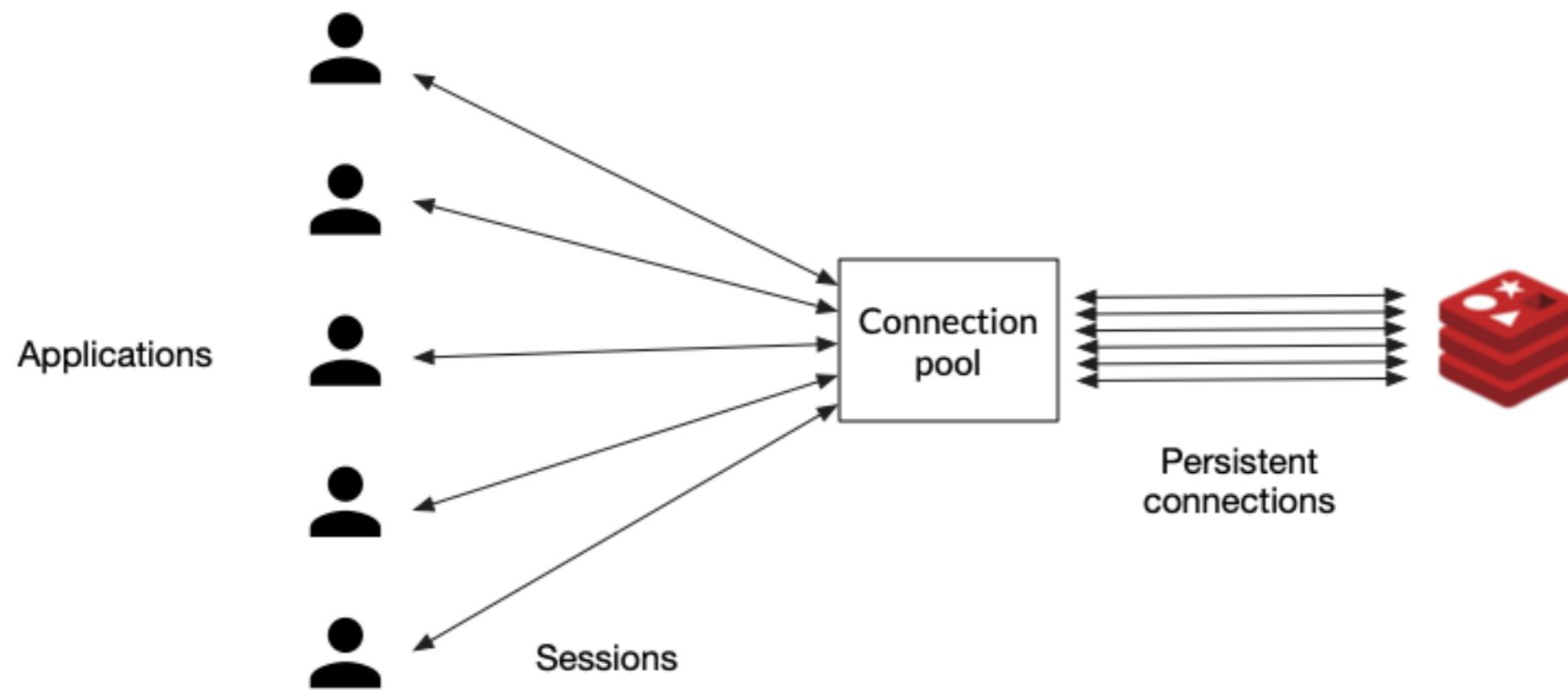


<https://redis.io/docs/latest/develop/interact/programmability/eval-intro/>



Use connection pool in client

Open and close connections cause overhead !!



Redis connection management

Unmanaged

Connection Pooling

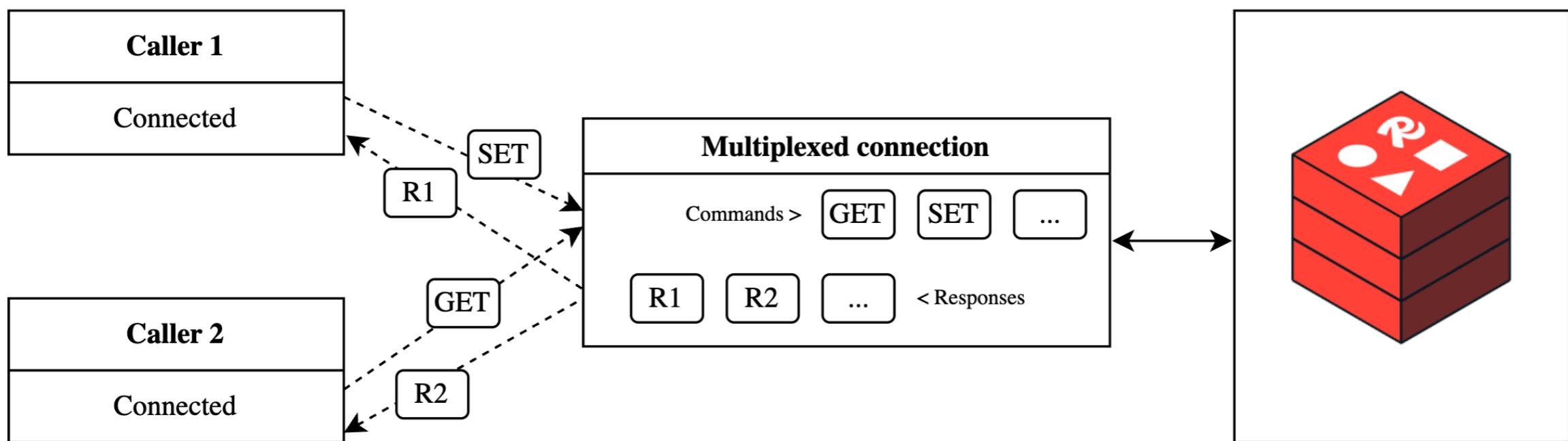
Multiplexing

<https://redis.io/blog/multiplexing-explained/>



Multiplexing

Keep a single connection open and use for all traffic



<https://redis.io/docs/latest/develop/clients/pools-and-muxing/>

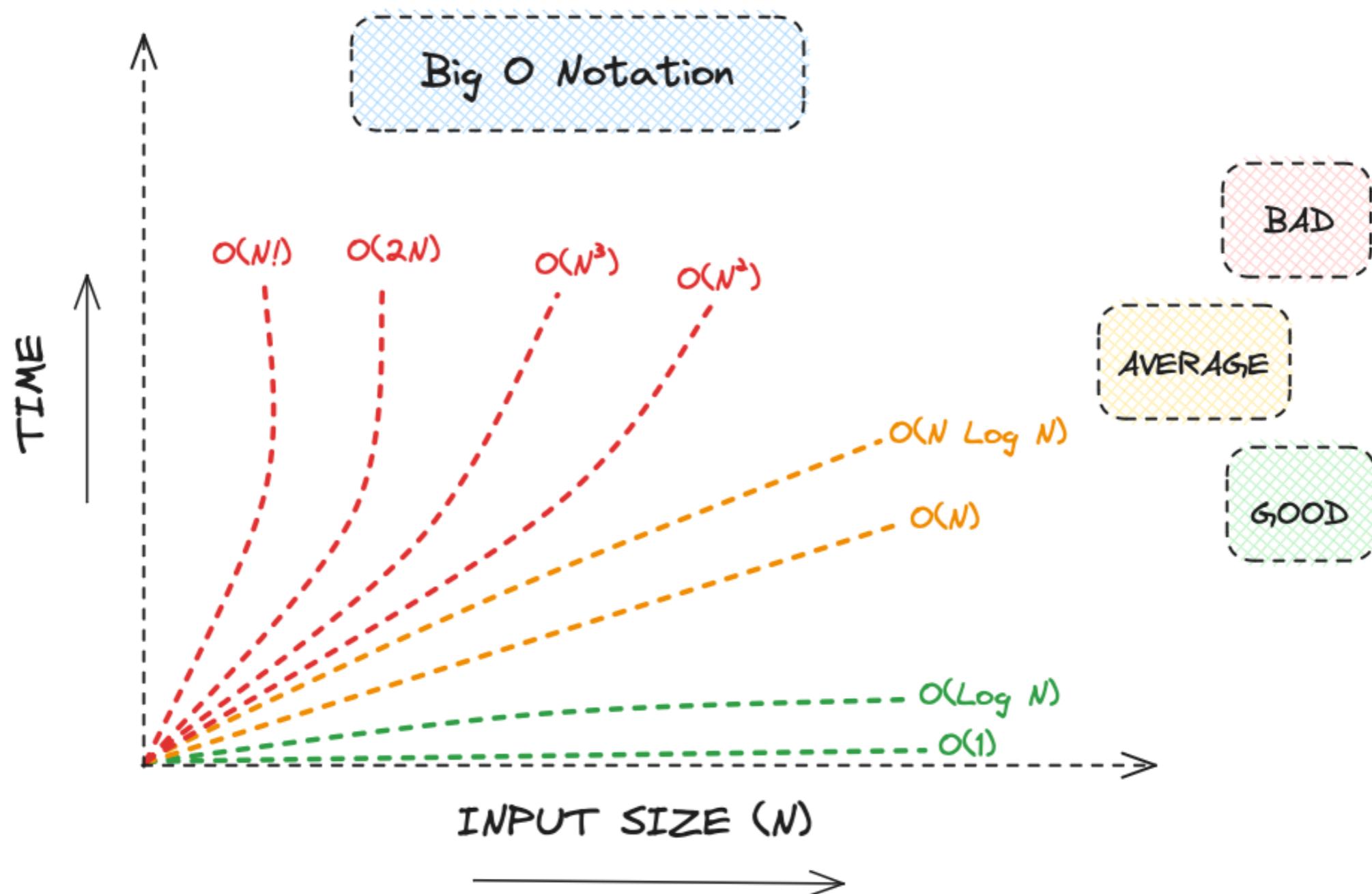


Efficient low-level data structure

<https://redis.io/technology/data-structures/>



Big O Notation



<https://www.linkedin.com/pulse/big-o-notation-critical-tool-modern-software-michael-santana-8uaxf/>

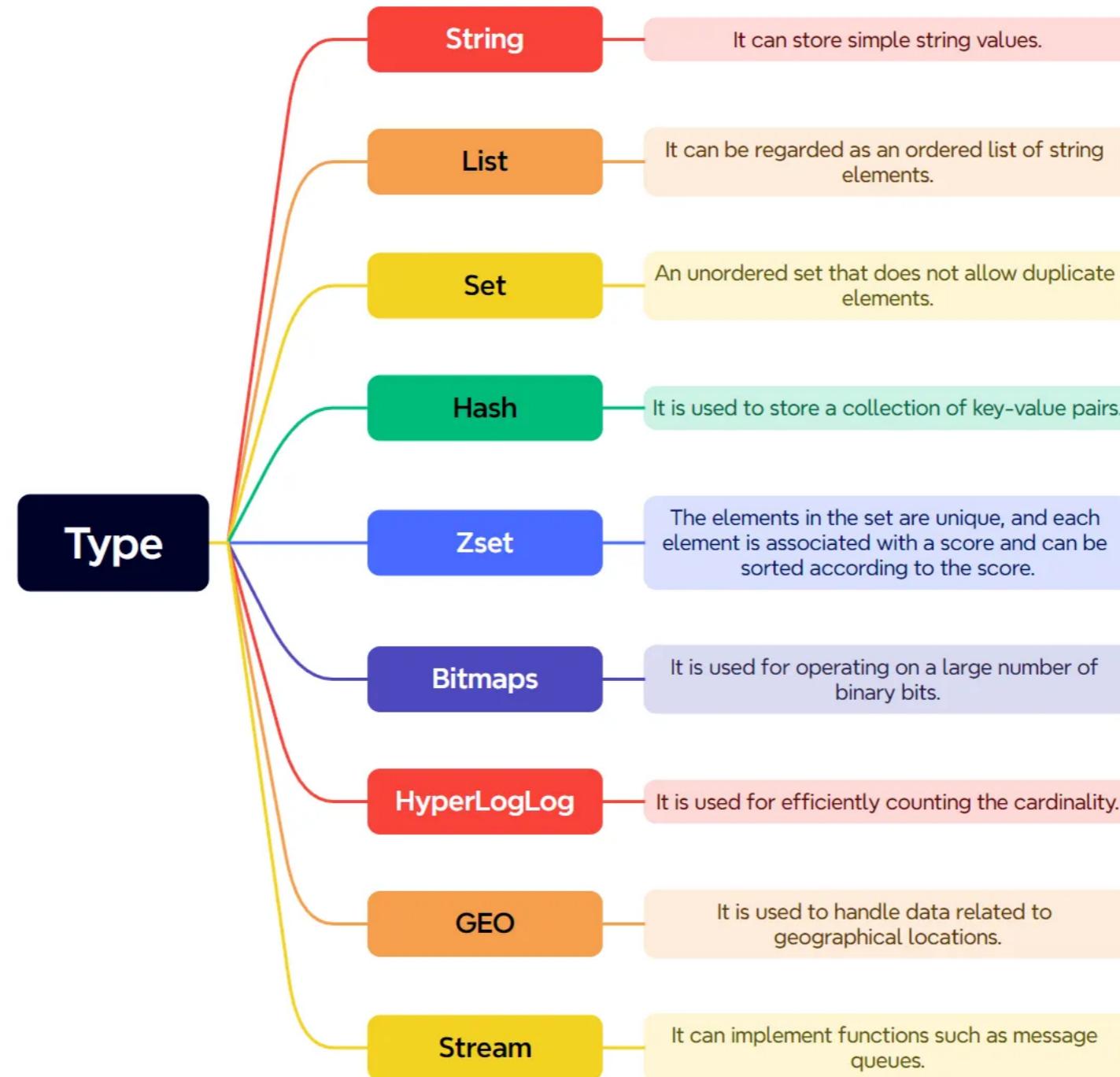


Data Structure in Redis

Basic Data Structures	Strings	Used for cache, counter, distributed locks, sessions	GET, SET, MGET, APPEND, SUBSTR, STRLEN
	Lists	Used for message queues	LPUSH, LPOP, LLEN, LMOVE, LTRIM
	Sets	Used for intersections, unions etc	SADD, SREM, SCARD, SISMEMBER, SINTER
	Hashes	Used for caches	HGET, HSET, HMGET, HINCRBY
	Sorted Sets (ZSet)	Used for ranking	ZADD, ZRANGE, ZREVRANGE, ZRANGEBYSCORE, ZREMRANGEBYSCORE, ZRANK
Added After Redis 2.2	Streams	Append only log for event sourcing, sensor monitoring etc	
	Geospatial	Store and query coordinates	
	Bitmaps	User daily login status	
Added Later - Probabilistic Data Structures for Big Data	HyperLogLog	Cardinality for big data	
	Bloom Filter	Check presence of an element in a set	
	Cuckoo Filter	Check presence of an element in a set	
	t-digest	Estimate the percentile of a data stream	
	Top-k	Find the most frequent items in a data stream	
	Count-Min Sketch	Estimate the frequency of an element in a data stream	



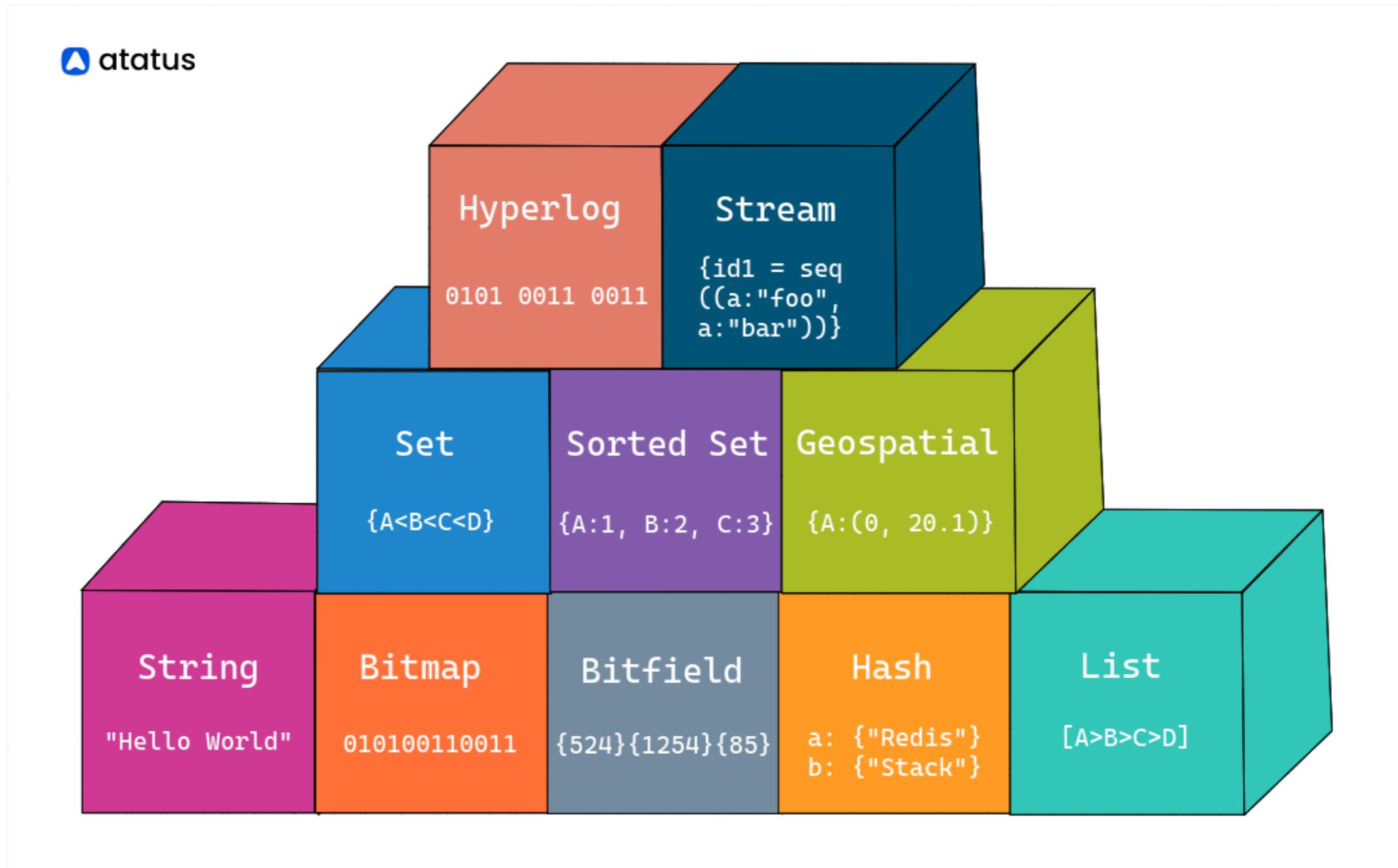
Data Structure in Redis



<https://redis.io/docs/latest/develop/data-types/>



Data Structure in Redis



<https://www.atatus.com/blog/redis-tutorial-exploring-data-types-architecture-and-key-features/>



Redis

© 2020 - 2025 Siam Chamnankit Company Limited. All rights reserved.

More ..

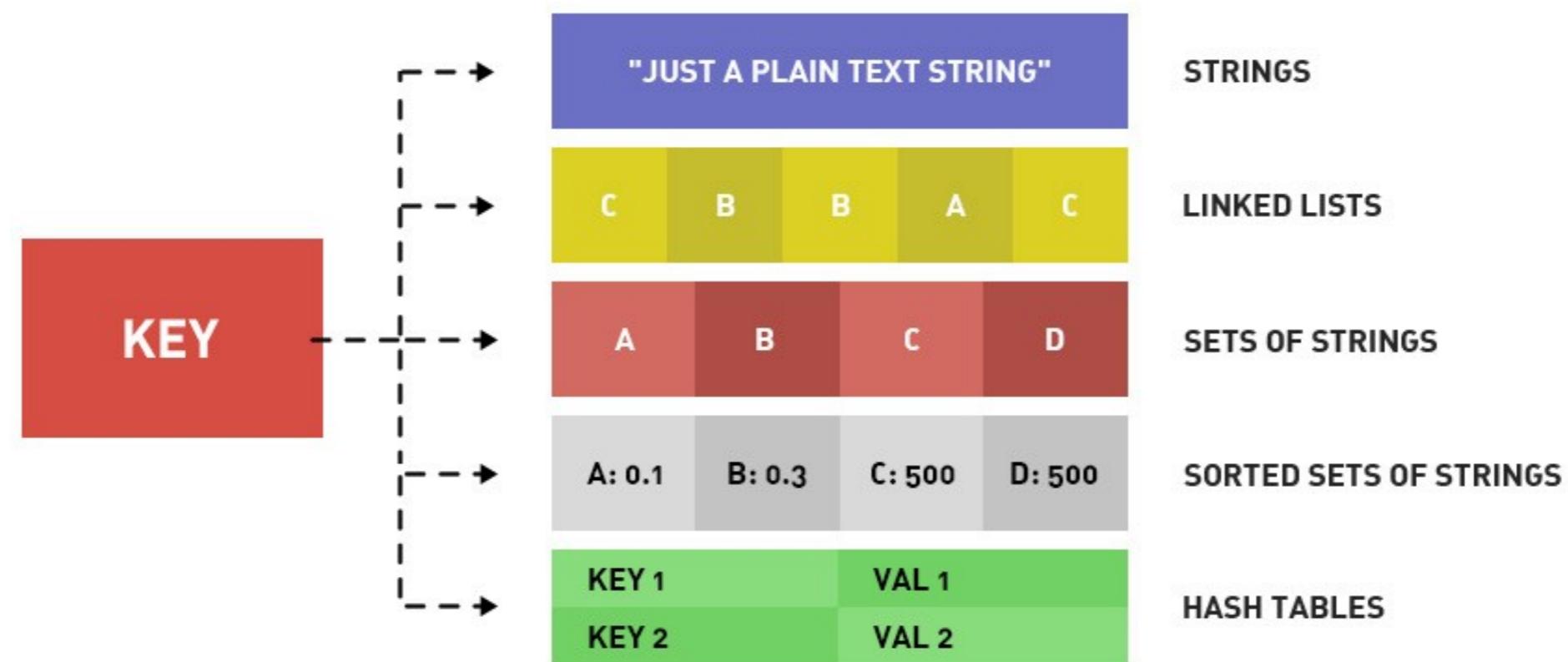
RedisJSON

Vector Database

RedisSearch

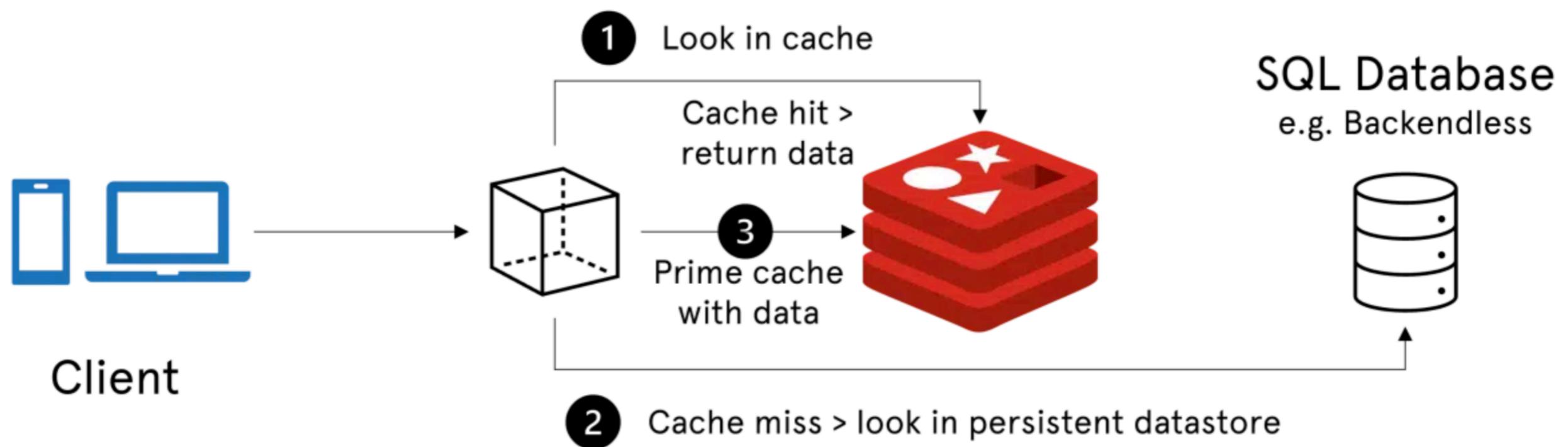


Caching data ?



Simple Use Case

Caching Data for applications
Reduce workload of database



Caching Strategies ?

Read

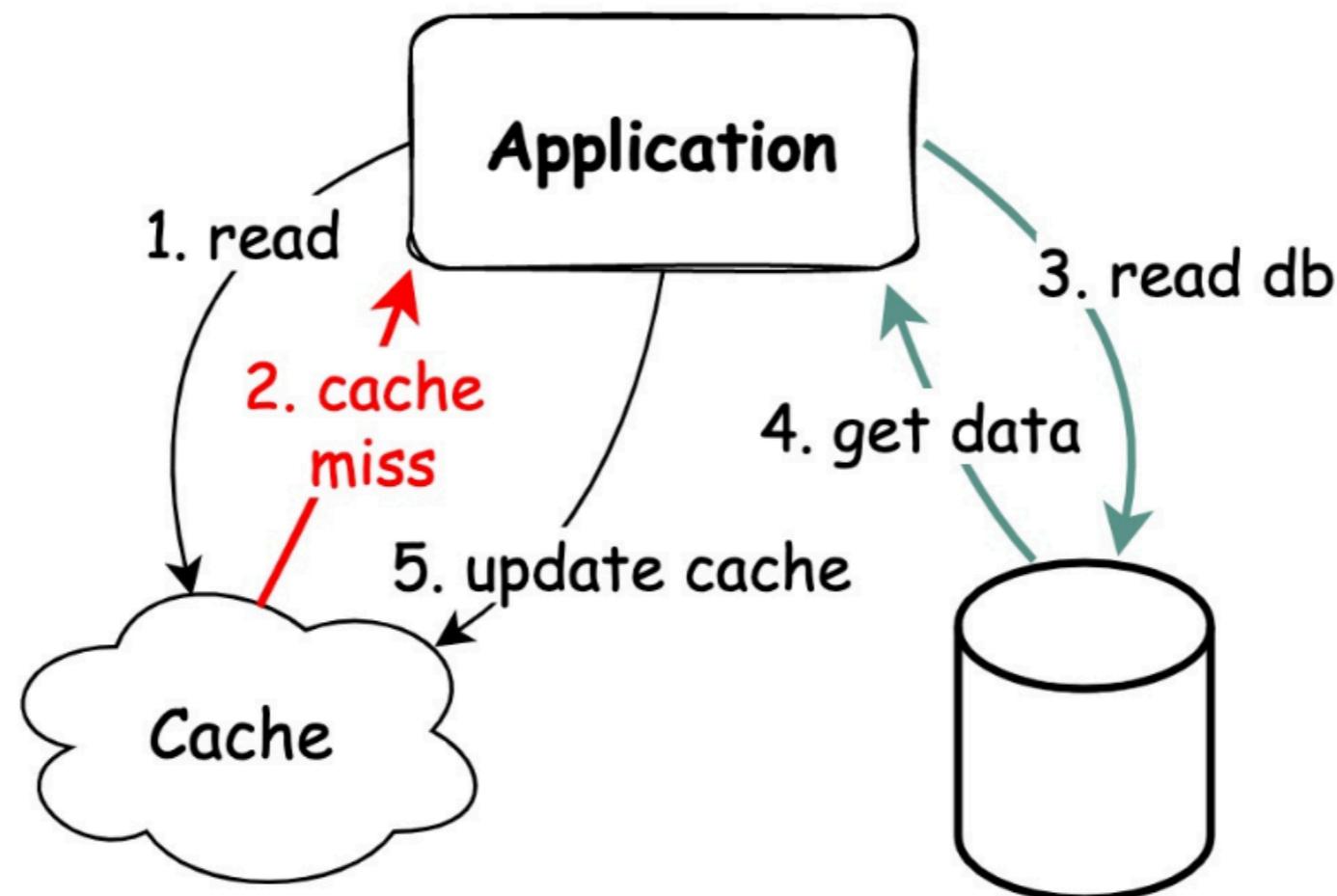
Write

<https://blog.bytebytego.com/p/top-caching-strategies>

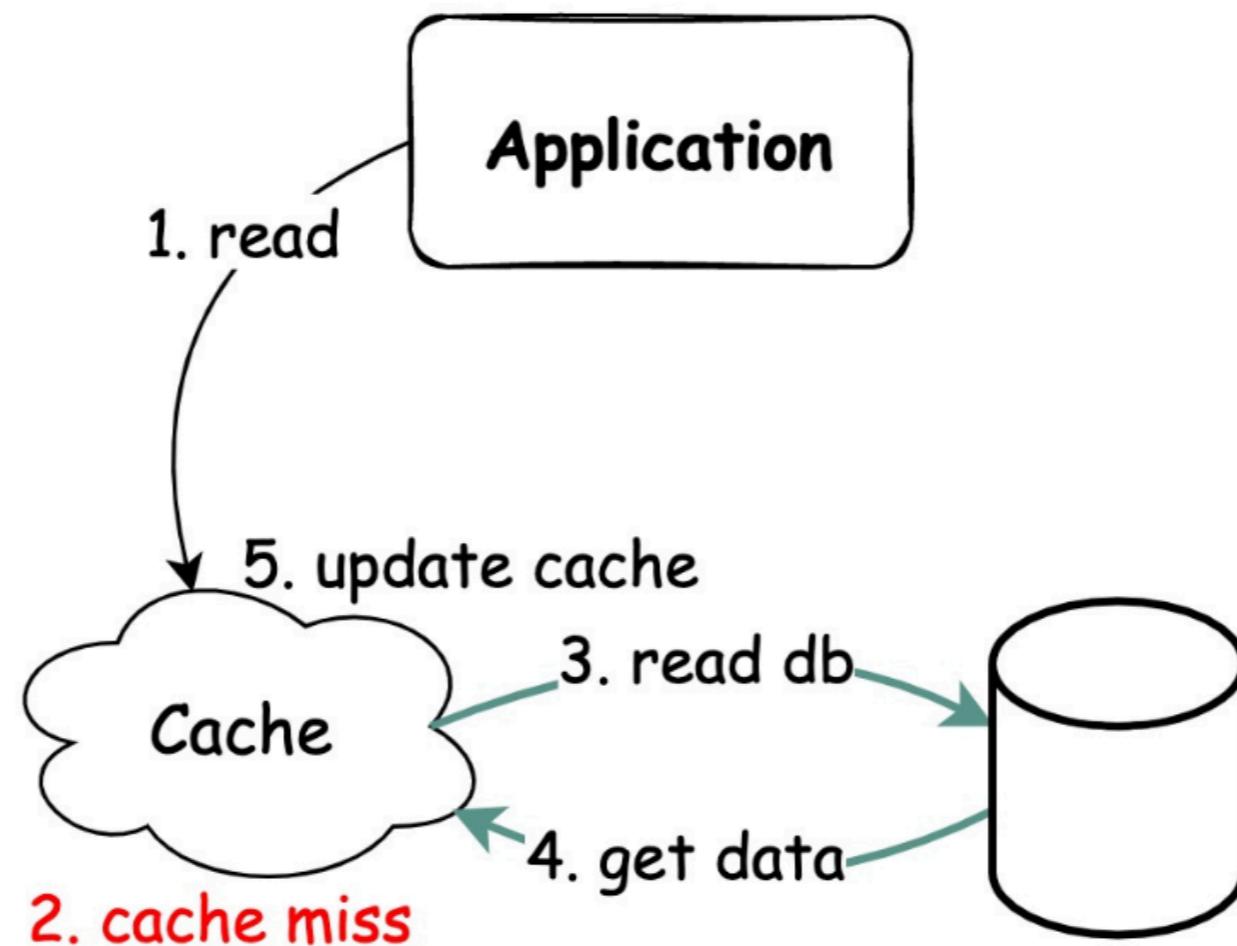
<https://blog.algomaster.io/p/top-5-caching-strategies-explained>



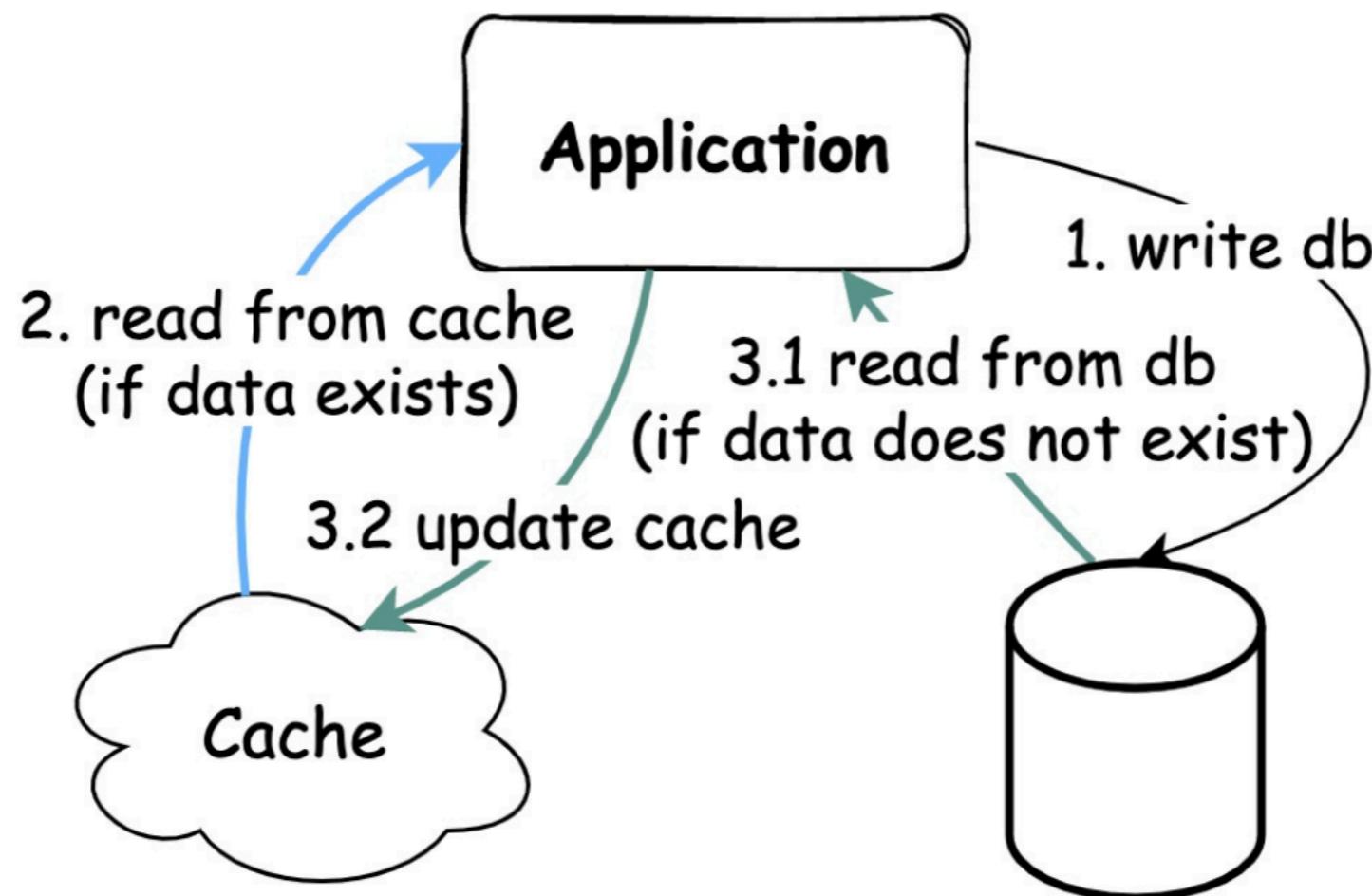
Read strategy - Cache Aside



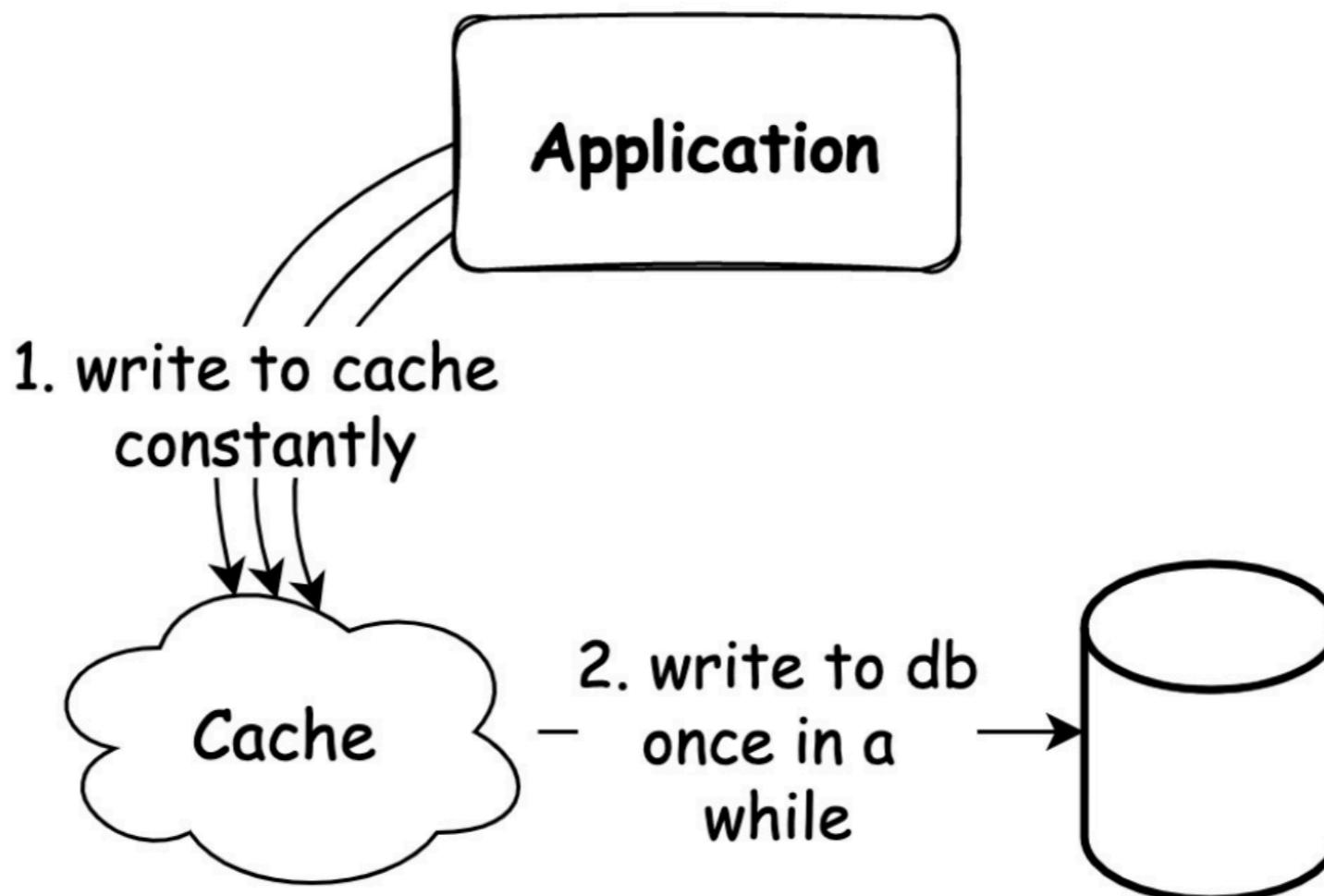
Read strategy - Read through



Write strategy - Write around

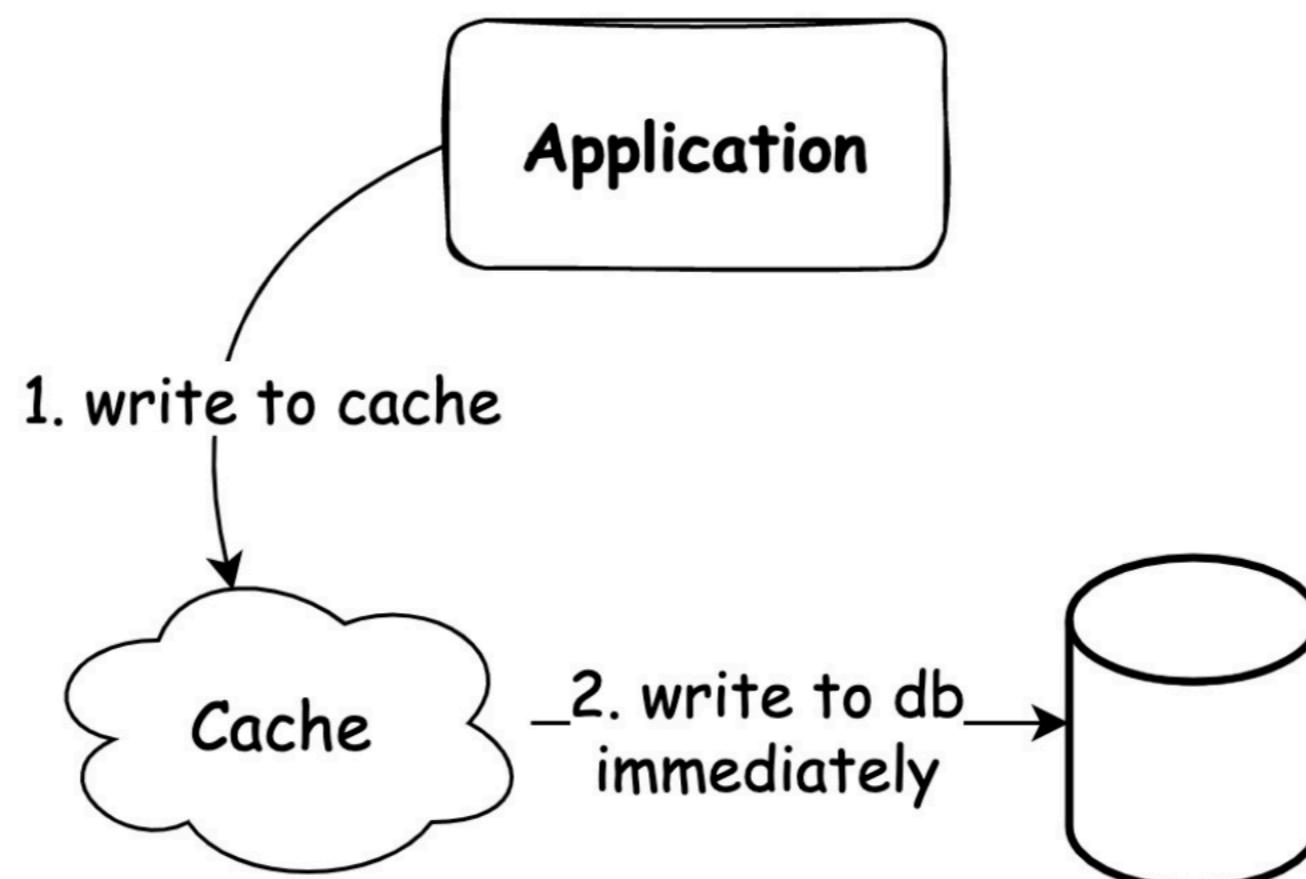


Write strategy - Write back



Write strategy - Write through

Write Strategy - Write Through



<https://redis.io/learn/howtos/solutions/caching-architecture/write-through>



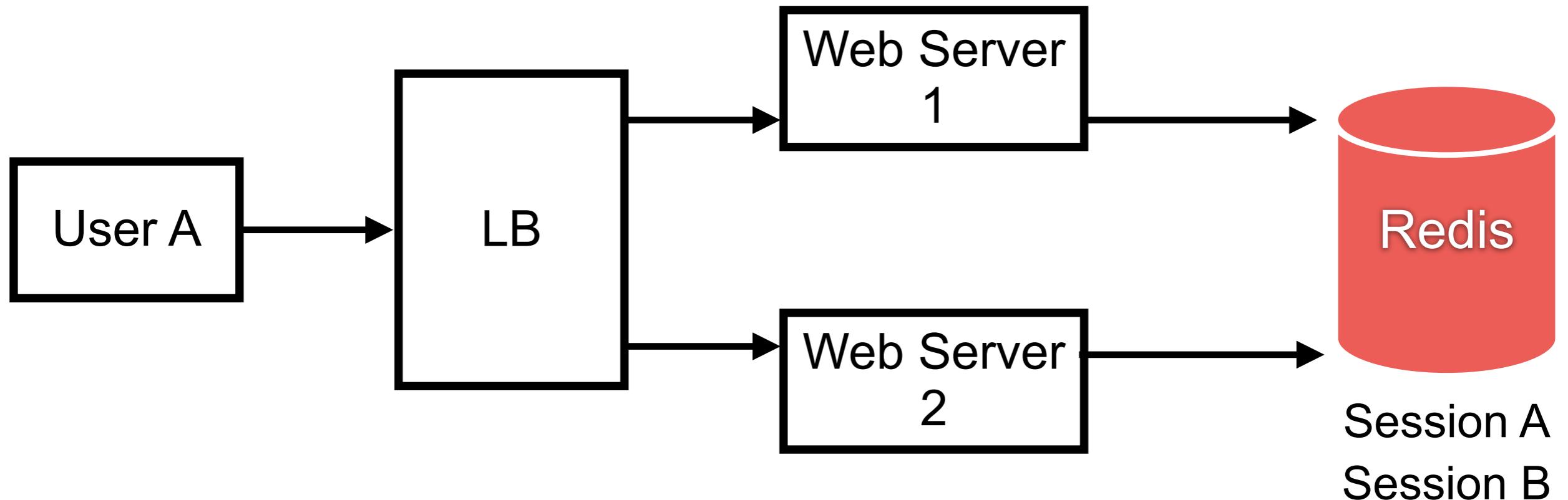
Use Cases

Session or Token data
Message broker (pub/sub)
Job queue
Realtime analytic
Fraud detection
Gaming

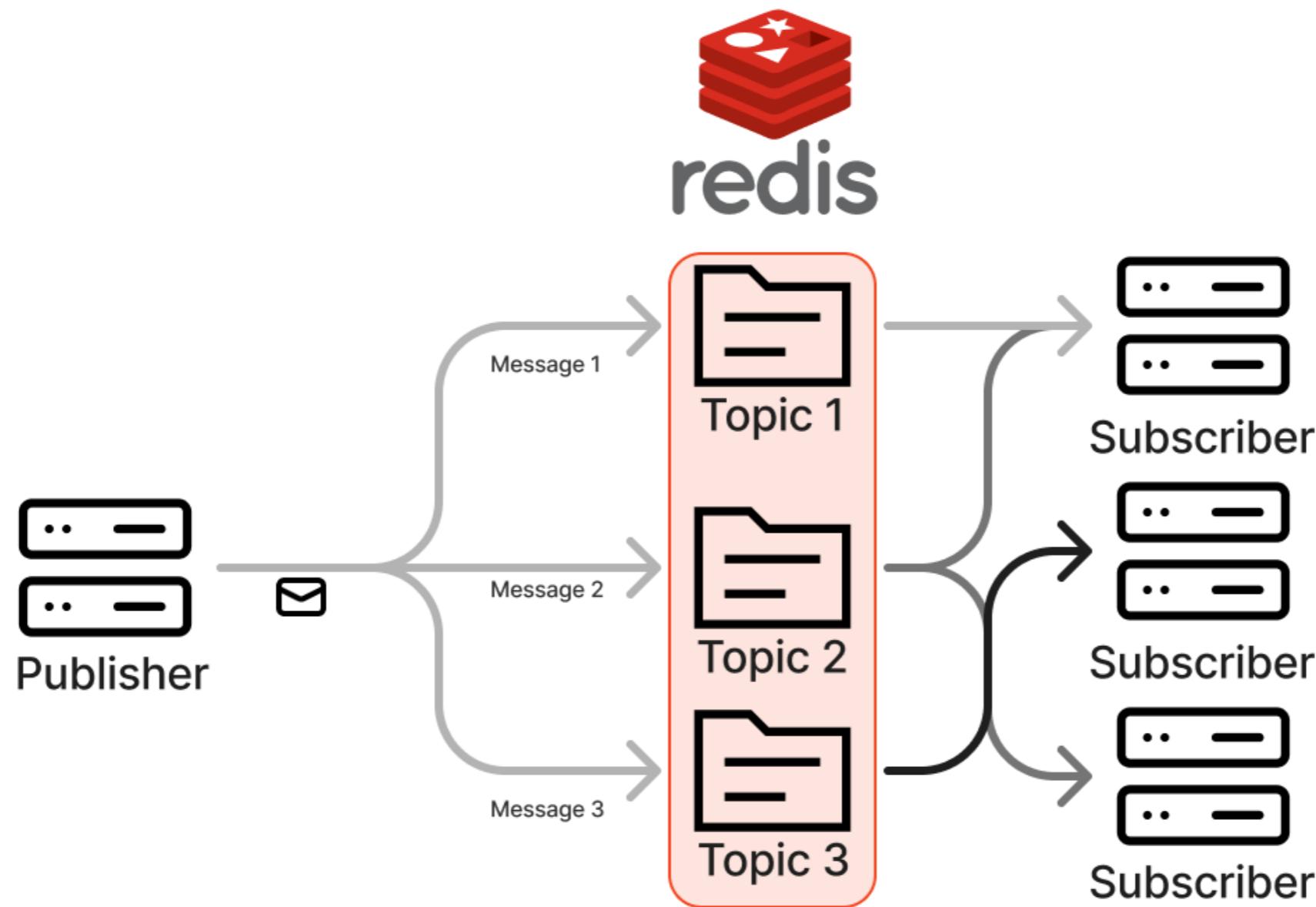


Session or Token data

Data with expire time
key=value(String)



Pub/Sub



<https://redis.io/docs/latest/develop/interact/pubsub/>



Sales leaderboard

BY WIN RATE

BY LEADS WON

BY VALUE WON

BY ACTIVITIES

Sales Reps



Enrique Romero



46



Clayton Clark



45



Andrew McGary



21



Michael Francis



19



Kurt Deruiter



8



Tiffany Falls



5



Brian Harvey



5



Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

48

Redis key naming !!



objectType:objectId:field

```
SET user:1000:name "Somkiat"
```



Data modeling with Redis

<https://redis.io/blog/nosql-data-modeling/>



How do I structure my data ?



Start with questions !!

Requirement

Read vs Write

User experience of your application

Data that user need (schema of data)

Size of data

Performance

Scalability

Schemaless != Not design schema



Data modeling with Redis ?

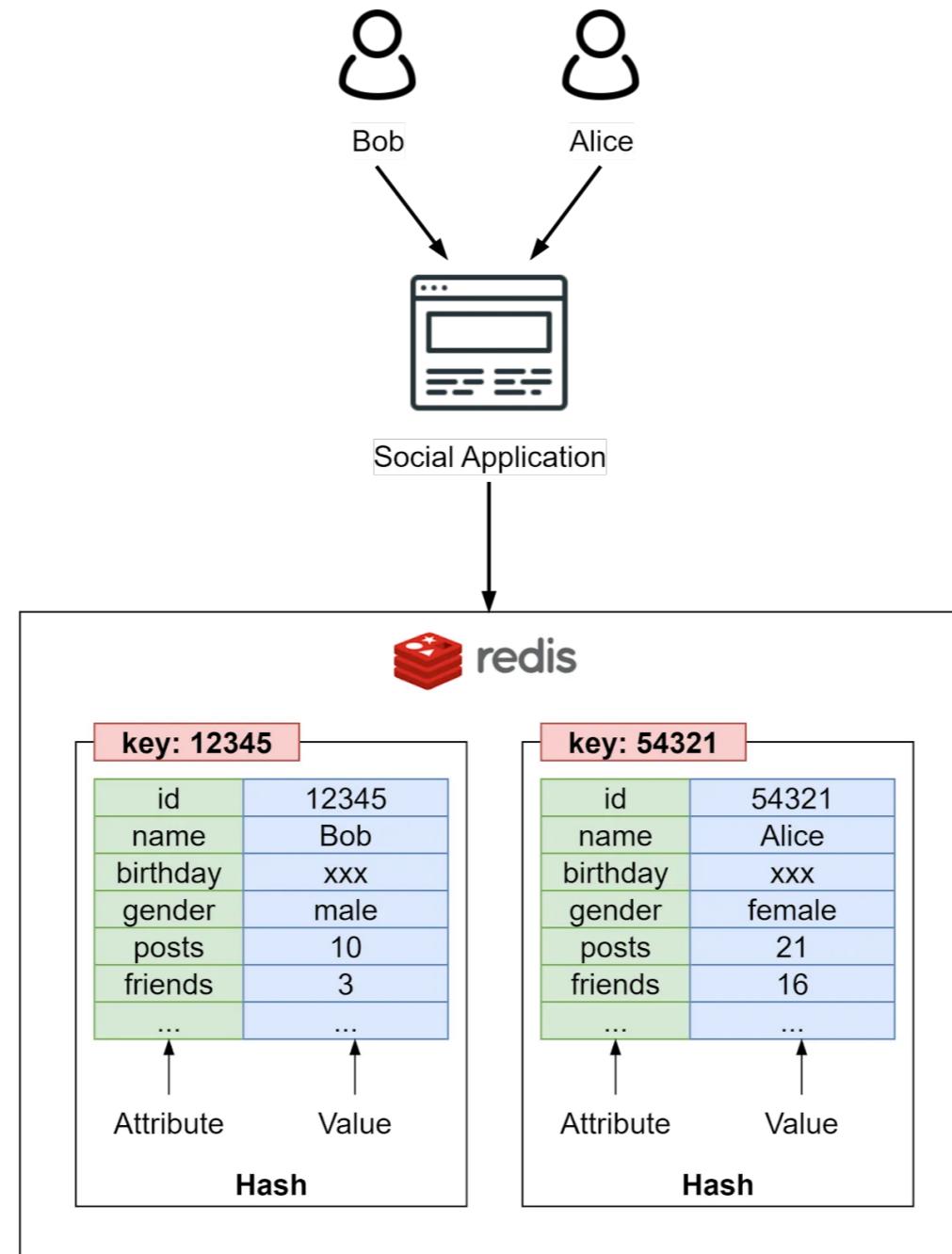
Data structure
Relationships



Data Structure



User profile



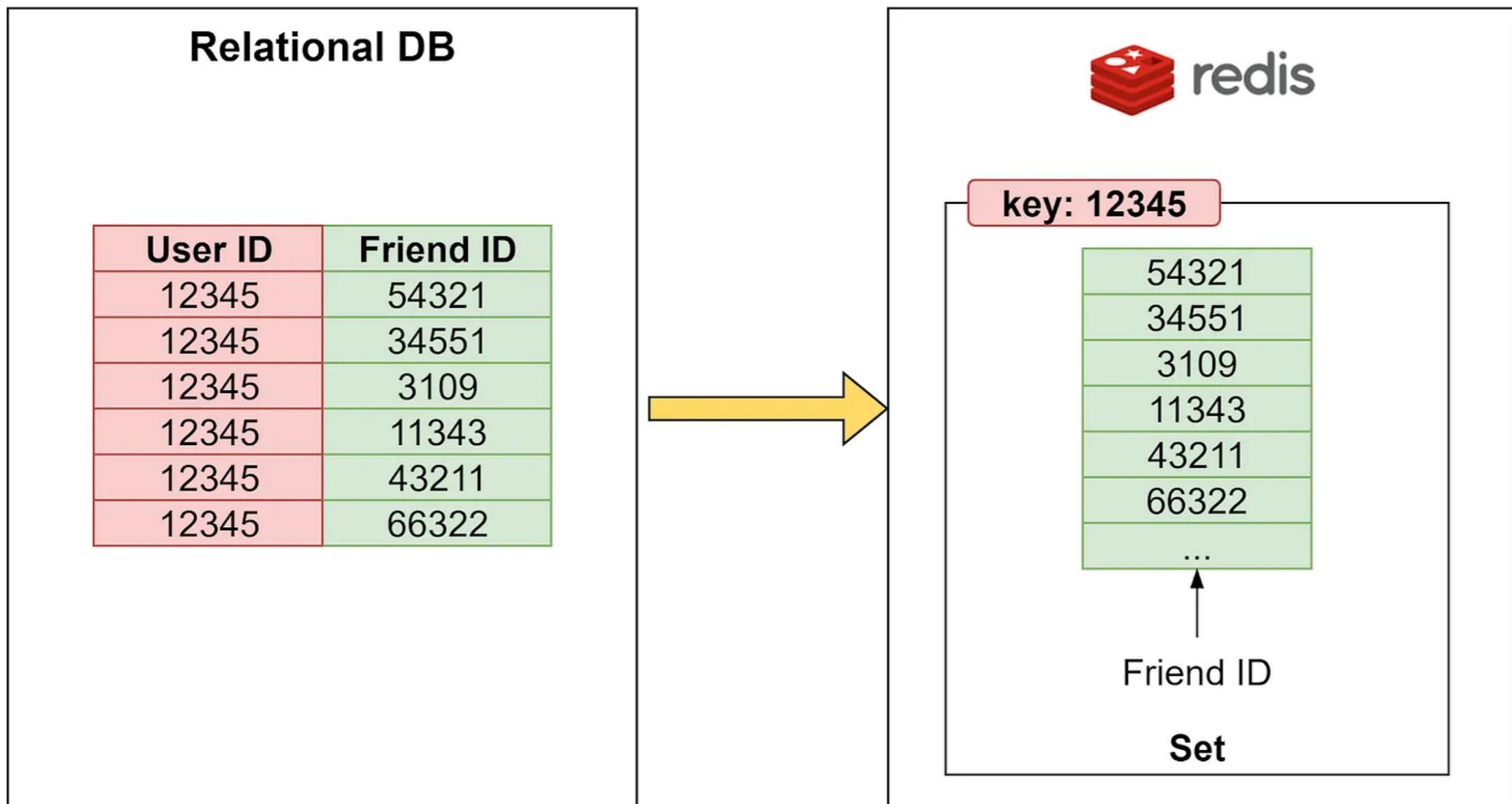
<https://blog.bytebytogo.com/p/redis-can-do-more-than-caching>



Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

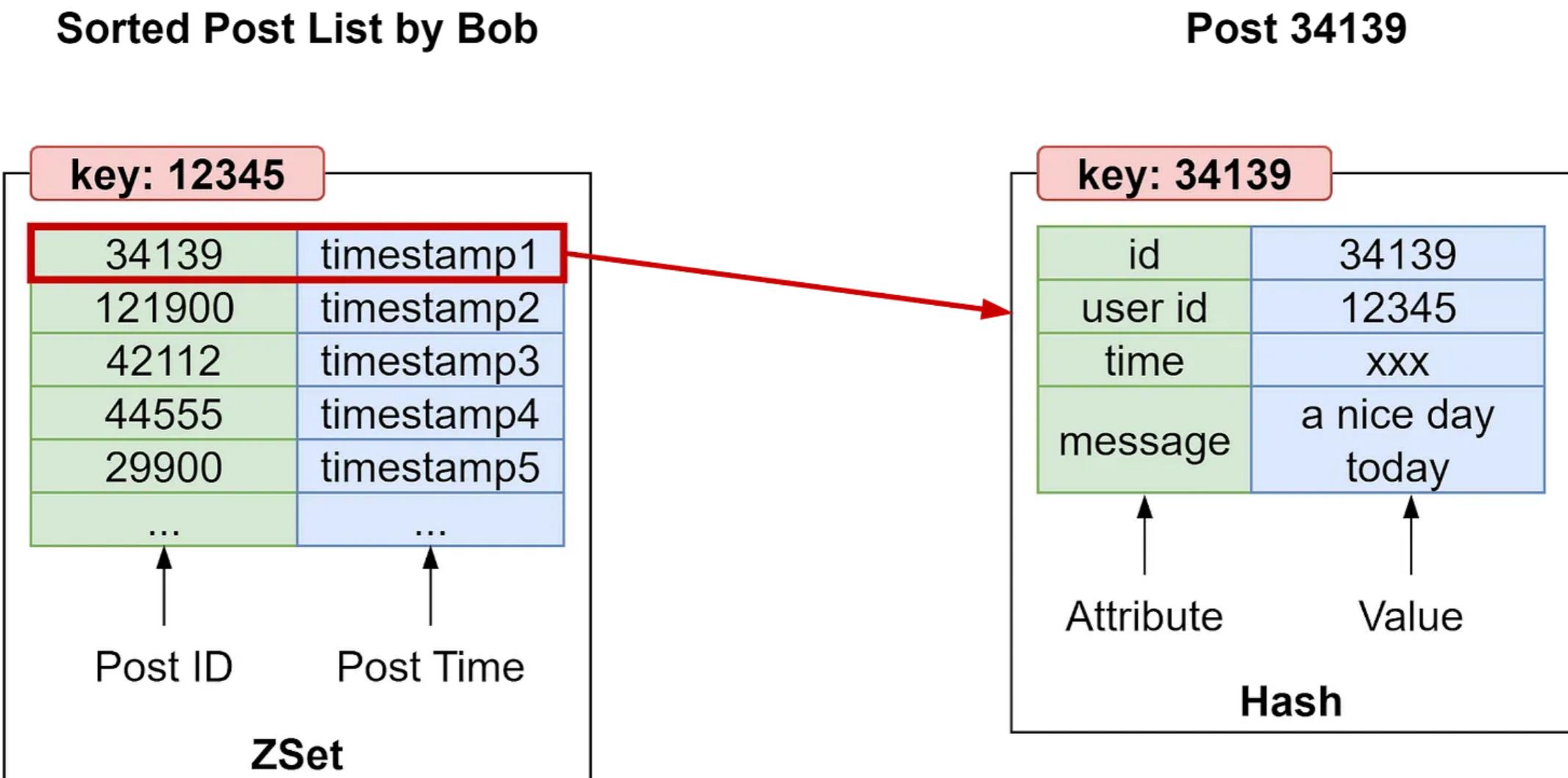
User's friend



<https://blog.bytebytogo.com/p/redis-can-do-more-than-caching>

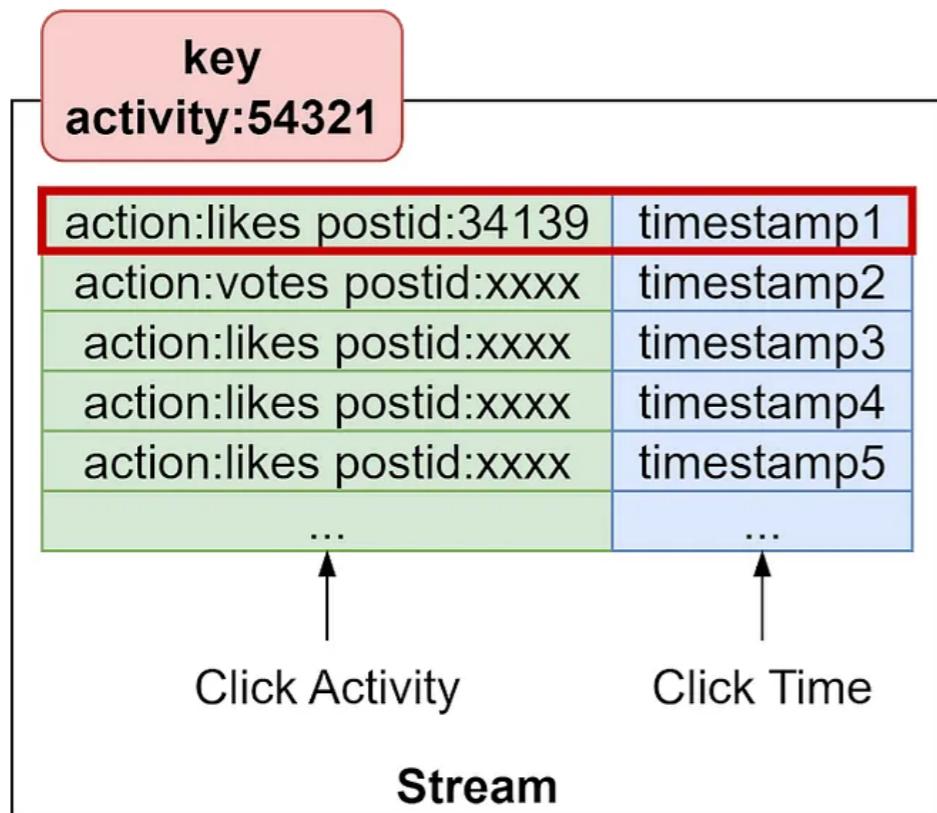


Post or Topic in Social media

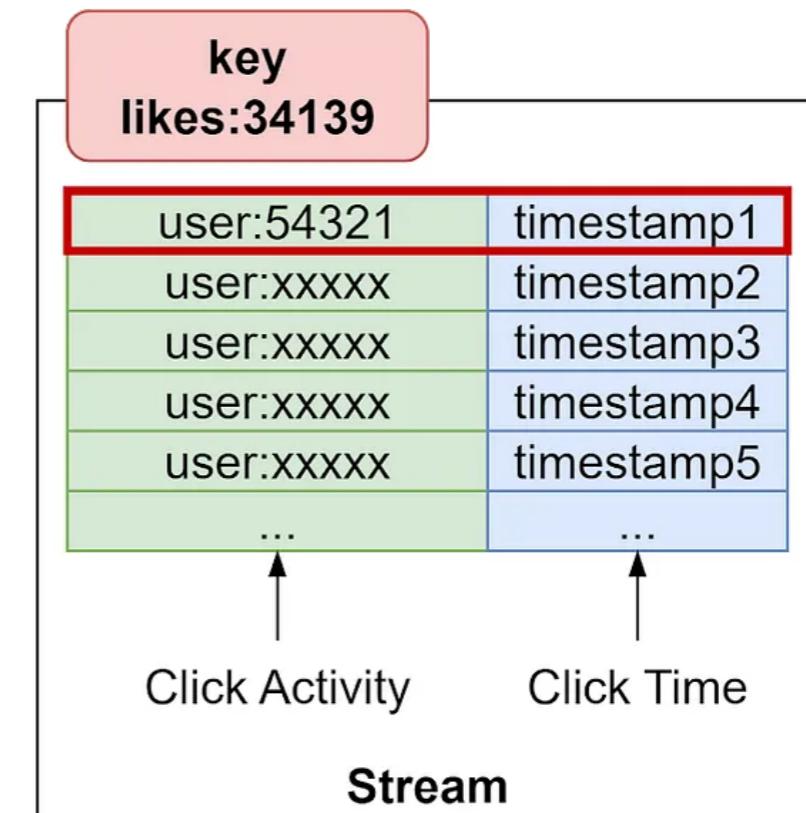


Activity in Social media

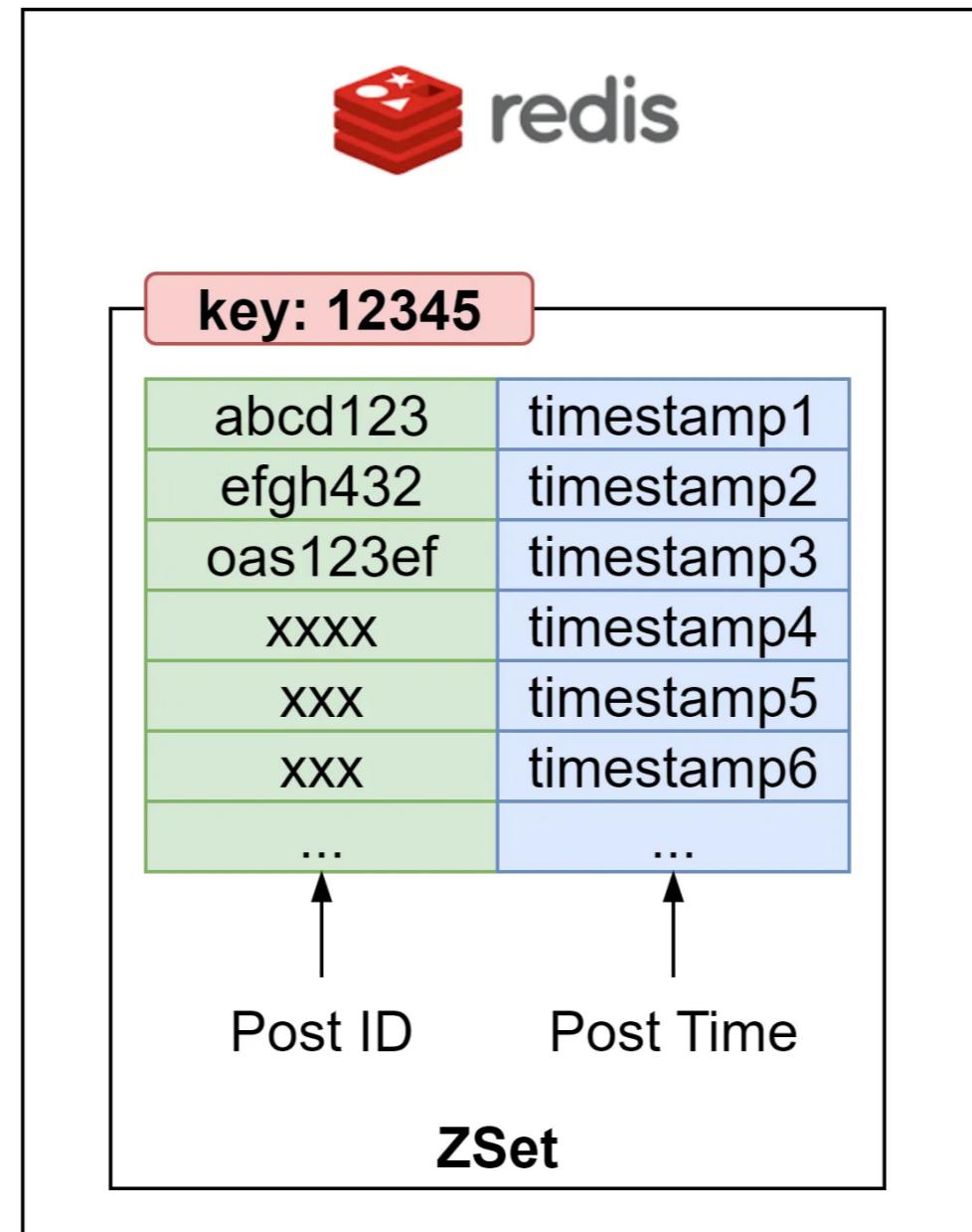
Alice's Activities



"Likes" Action on Post 34139



My feed



Leaderboard for gamification

Step Counts		Time Remaining			
	1st		4th	3	16
Amanda Smith	7,535	Jason Alexander	6,314	Days	Hours
	2nd		5th	37	33
Kyler Jones	7,091	Carson Tyler	5,512	Min	Sec
	3rd		6th	Prizes	
Brie Almater	6,383	Lisa Artz	4,589	 1st Place: \$1000	
		 2nd Place: \$600			
		 3rd Place: \$250			

<https://redis.io/solutions/leaderboards/>



Redis

© 2020 - 2025 Siam Chamnankit Company Limited. All rights reserved.

More data structure

Location-based service

Tracking user state with bitmap
Probabilistic



Probabilistic

Bloom filter

Hyperloglog

Top-K

<https://redis.io/docs/latest/develop/data-types/probabilistic/>



Redis

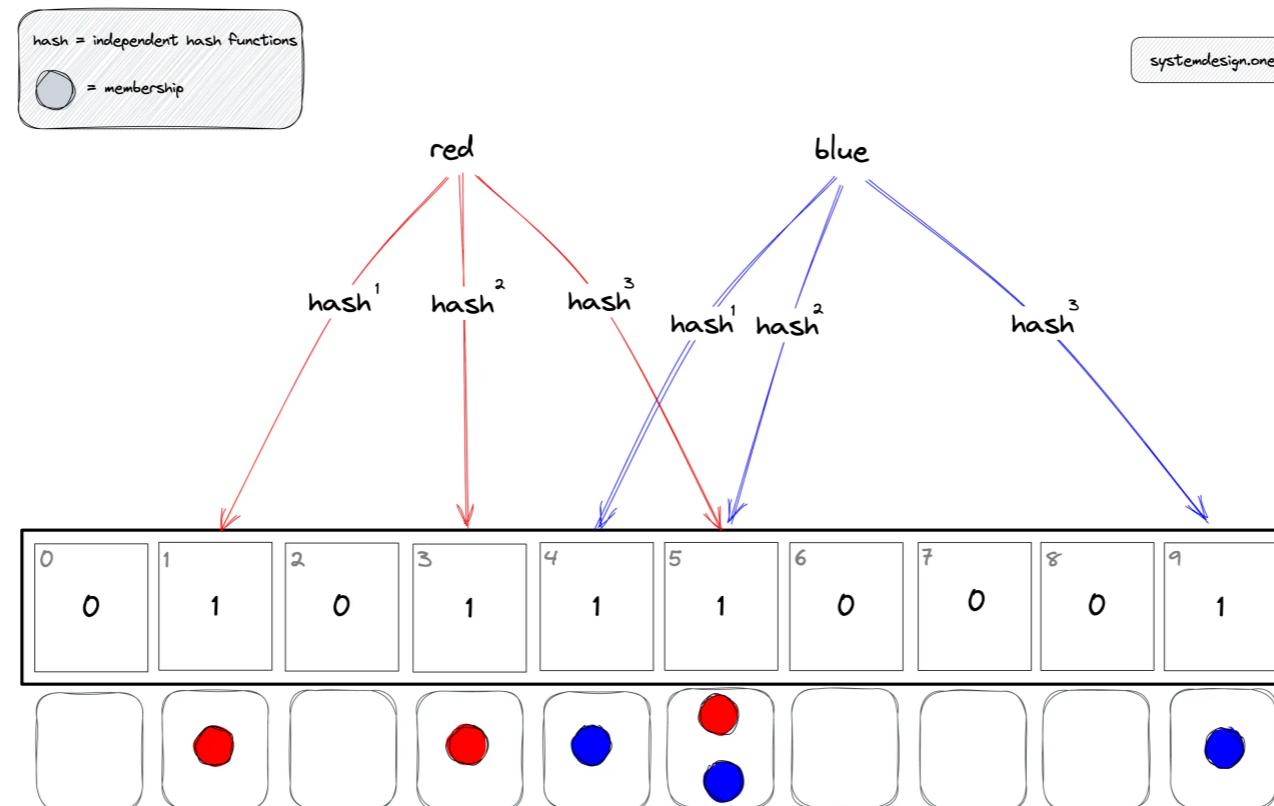
© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Bloom filter

Check data in redis

Reduce size of data in memory (but have error rate)

Balance between memory and error rate



<https://redis.io/docs/latest/develop/data-types/probabilistic/bloom-filter/>



Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Relationships of Data



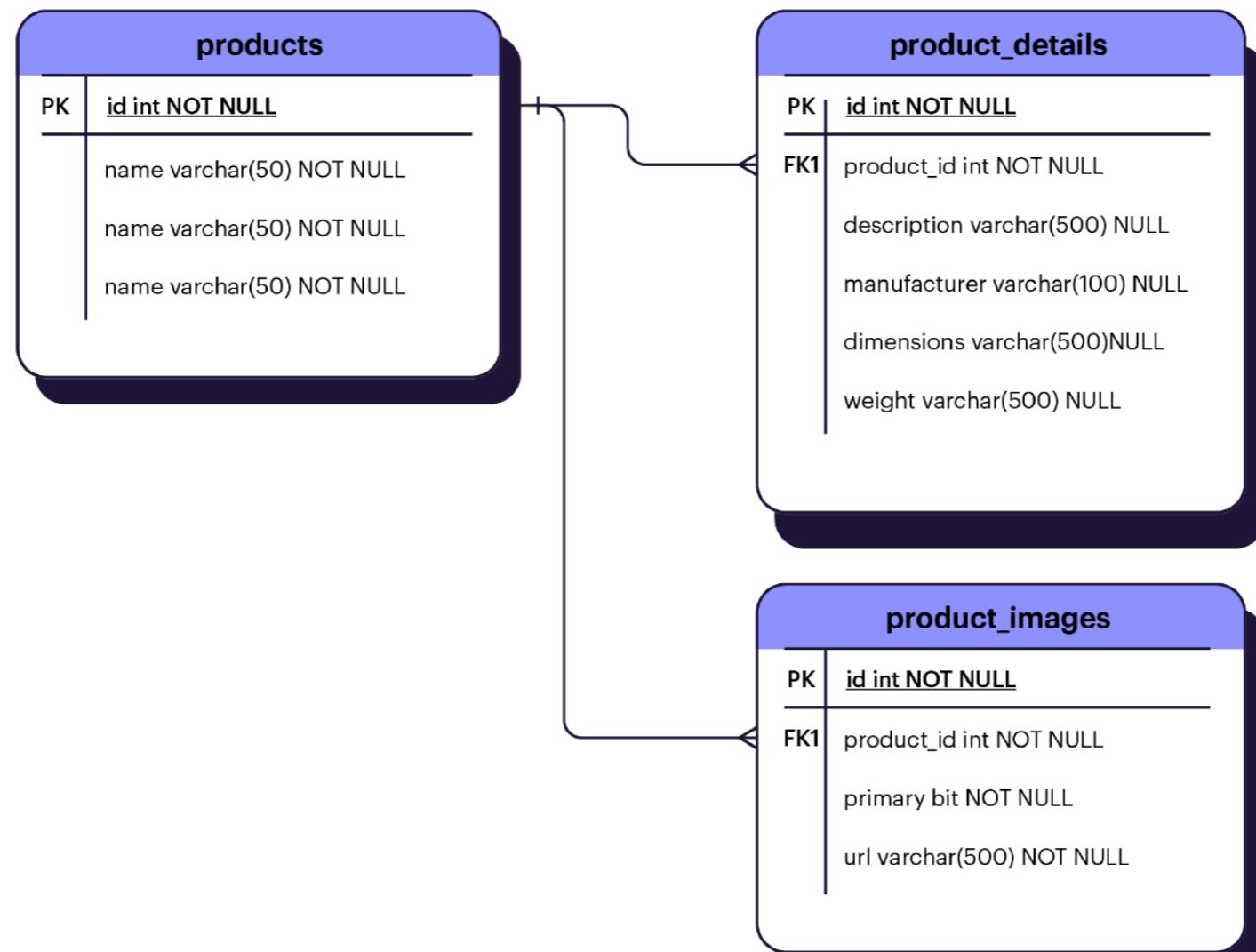
Relationships

One-to-one
One-to-many
Many-to-many



One-to-one

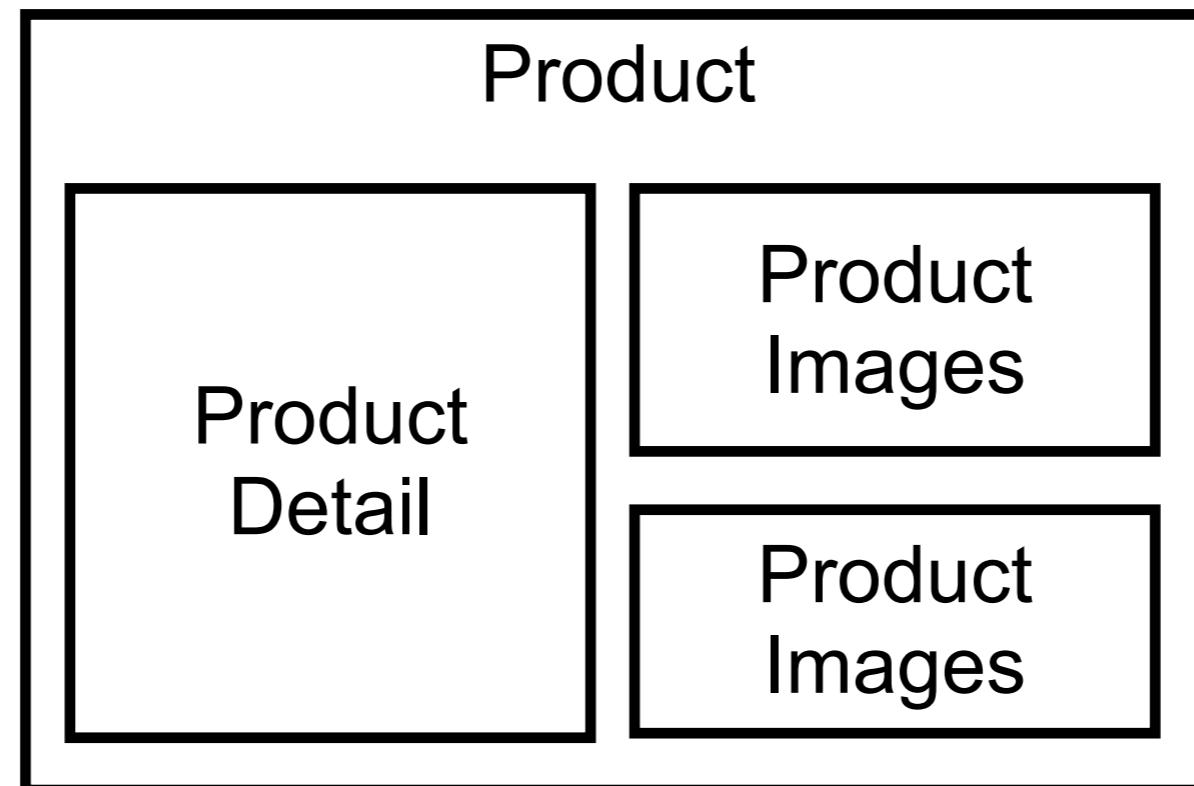
Product => Product detail



One-to-one in Redis

Embedded pattern for get product details by id

key=products:id



Search products ?



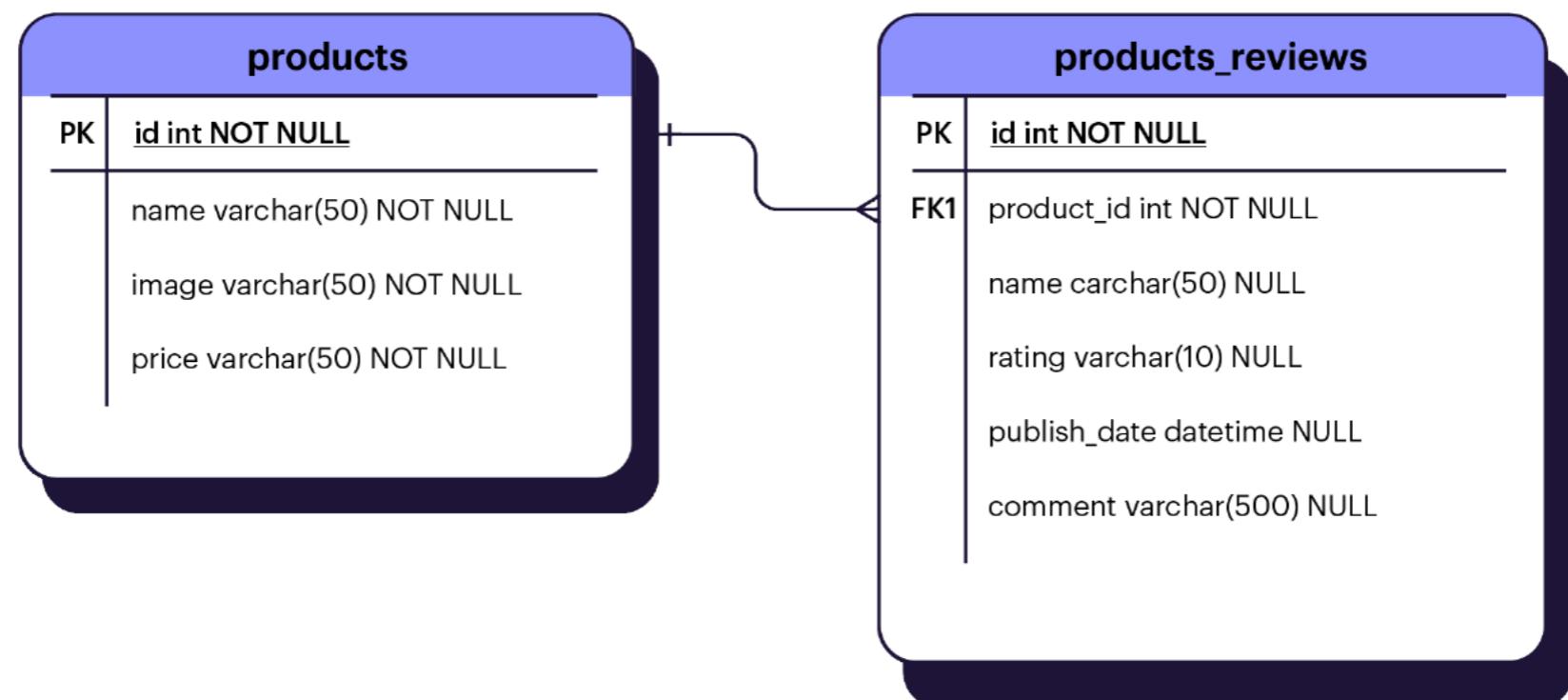
Redisearch

https://redis.io/docs/latest/operate/oss_and_stack/stack-with-enterprise/search/



One-to-many

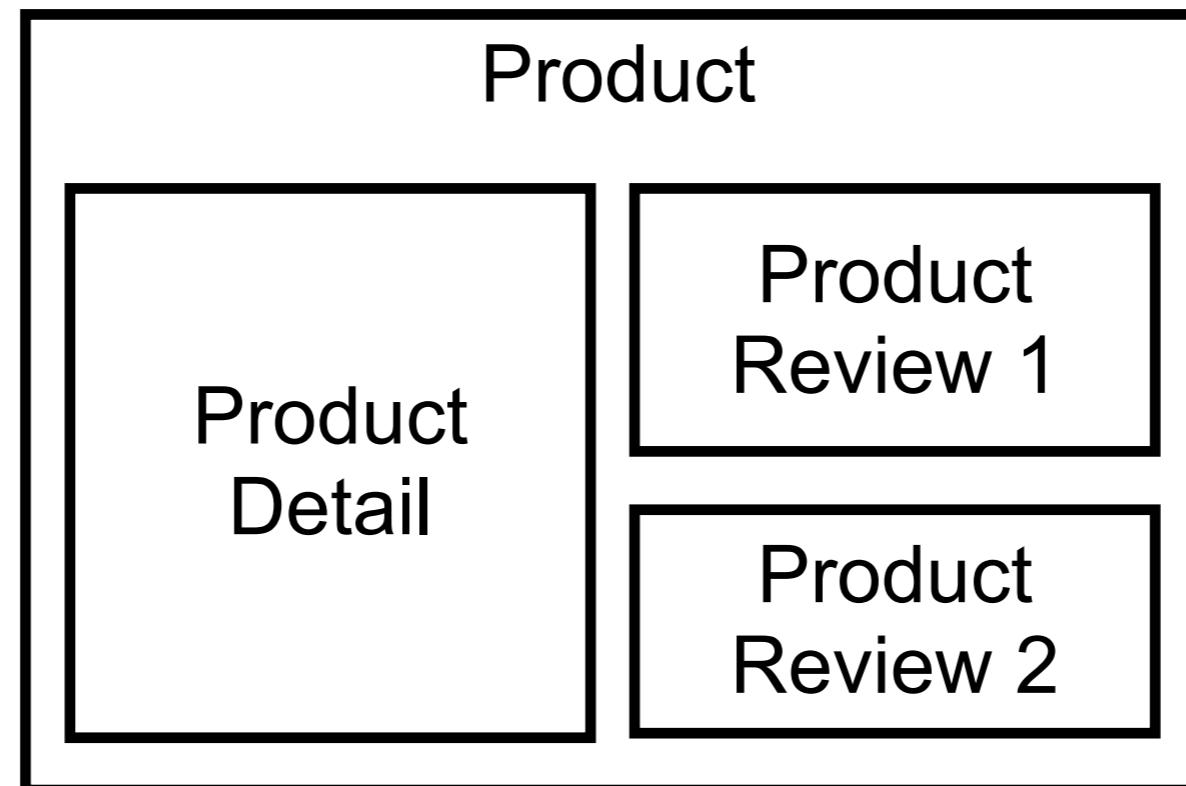
Product => Product reviews



One-to-many in Redis

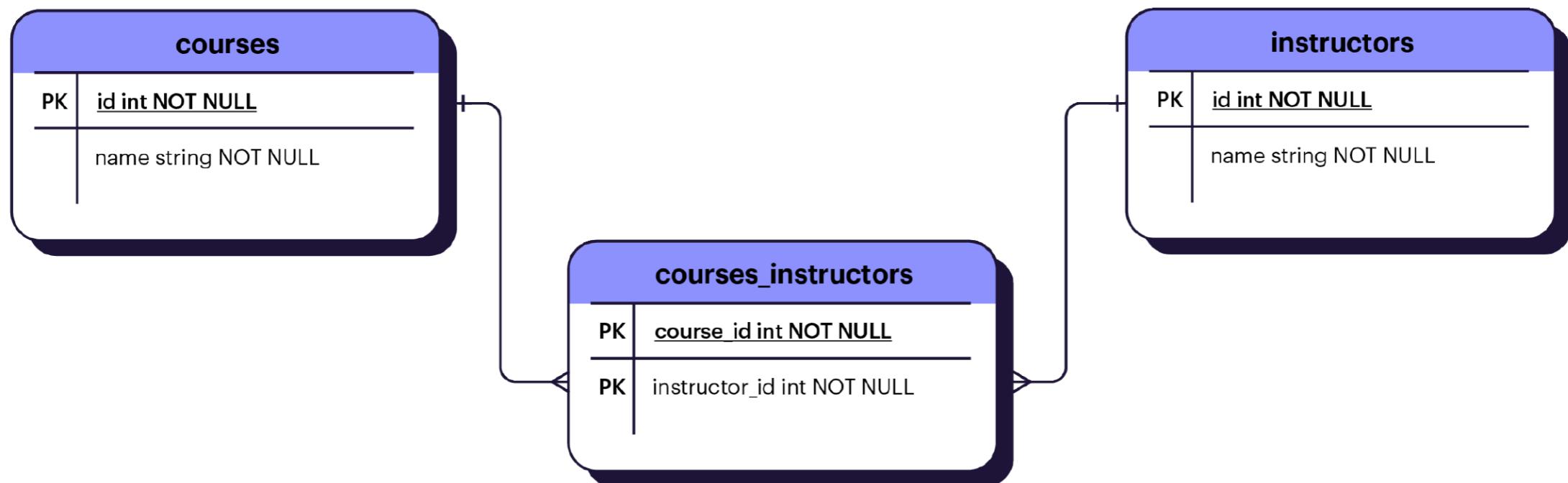
Partial Embedded pattern for recently review (newest)

key=products:id



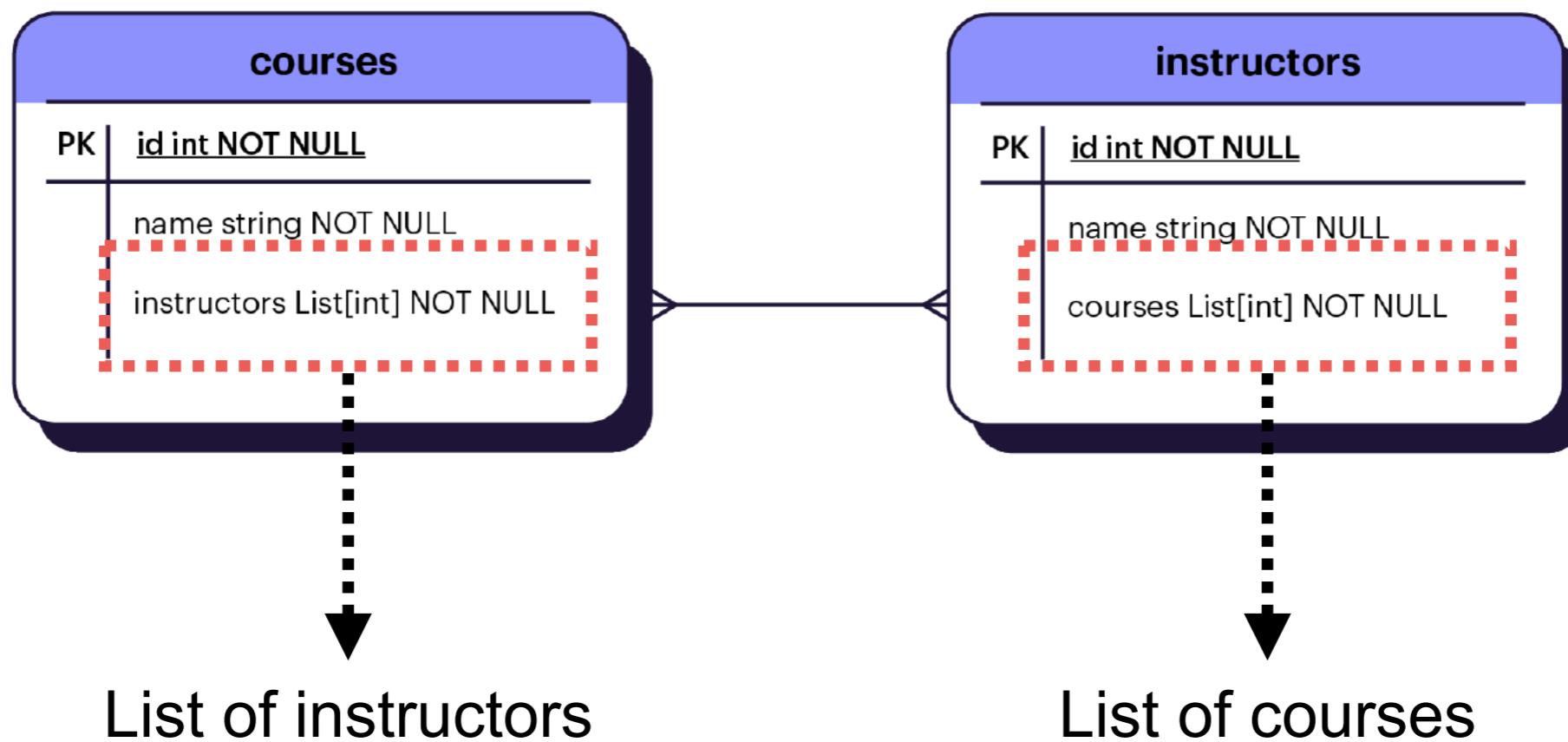
Many-to-many

Courses and Instructors



Many-to-many in Redis

Courses and Instructors (many but small !!)



More patterns



Aggregate pattern

Product rating !!



ร้านแนะนำ ยาง สไตล์ Watch สายนาฬิกา Watch UI Smart Watch ปรับง่าย สาย 41/42/44/45/46/47mm

4.8 ★★★★★ 1.4 พัน Ratings

FLASH SALE

฿23 ฿50 -54%

โค้ดส่วนลดจาก
ร้านค้า ลด ฿1

การจัดส่ง จะได้รับ
รับโค้ดส่วนลด

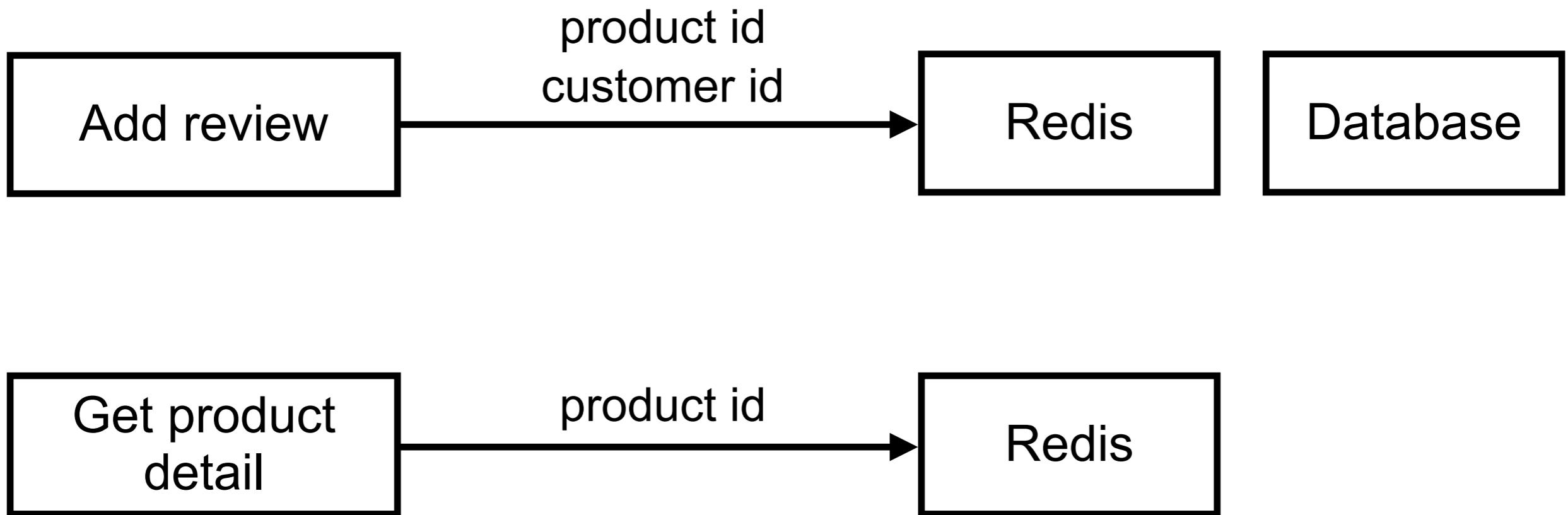
ช้อปปิ้งรันดี ✓ ช้อปปิ้งรันดี

products	
PK	<code>id int NOT NULL</code>
	<code>name string NOT NULL</code>
	<code>numReviews int NOT NULL</code>
	<code>sumRatings int NOT NULL</code>



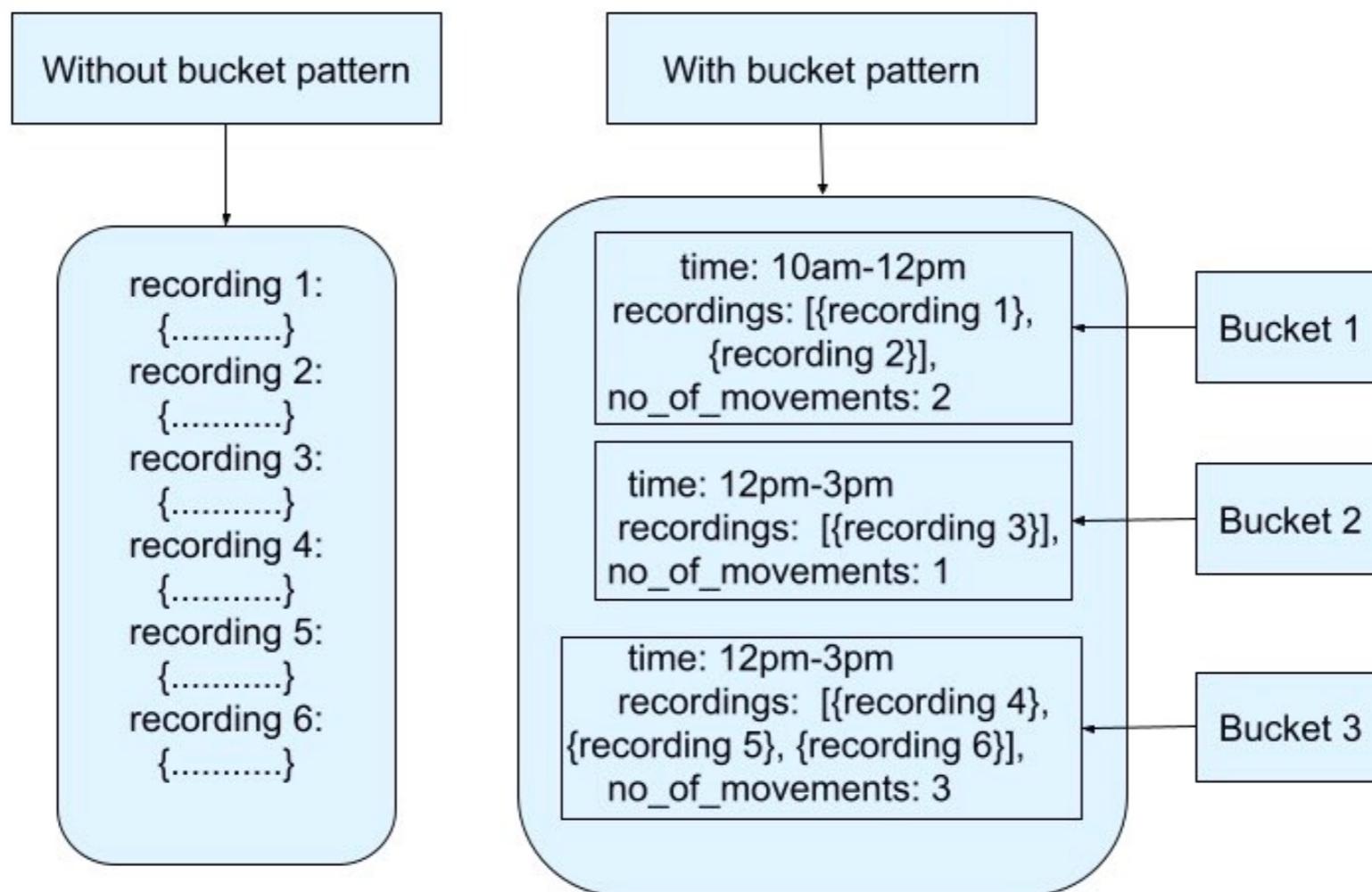
Aggregate pattern in Redis

Product rating !!



Bucket pattern

Working with time series database



Configurations !!

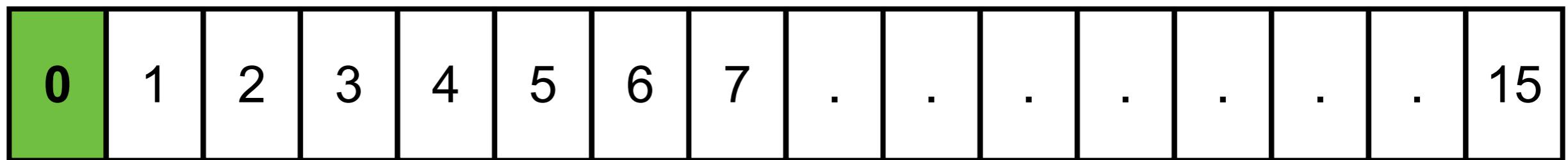
redis.conf

<https://www.dragonflydb.io/guides/redis-best-practices>



Redis databases

Default = 16



Redis databases

Isolation the data
Easy data separation
Flexible in configuration
Selective data management

Memory
eviction

FLUSHALL



Max memory

Default = none

Allow size of memory to Redis

Working with memory eviction policies



Redis Persistence

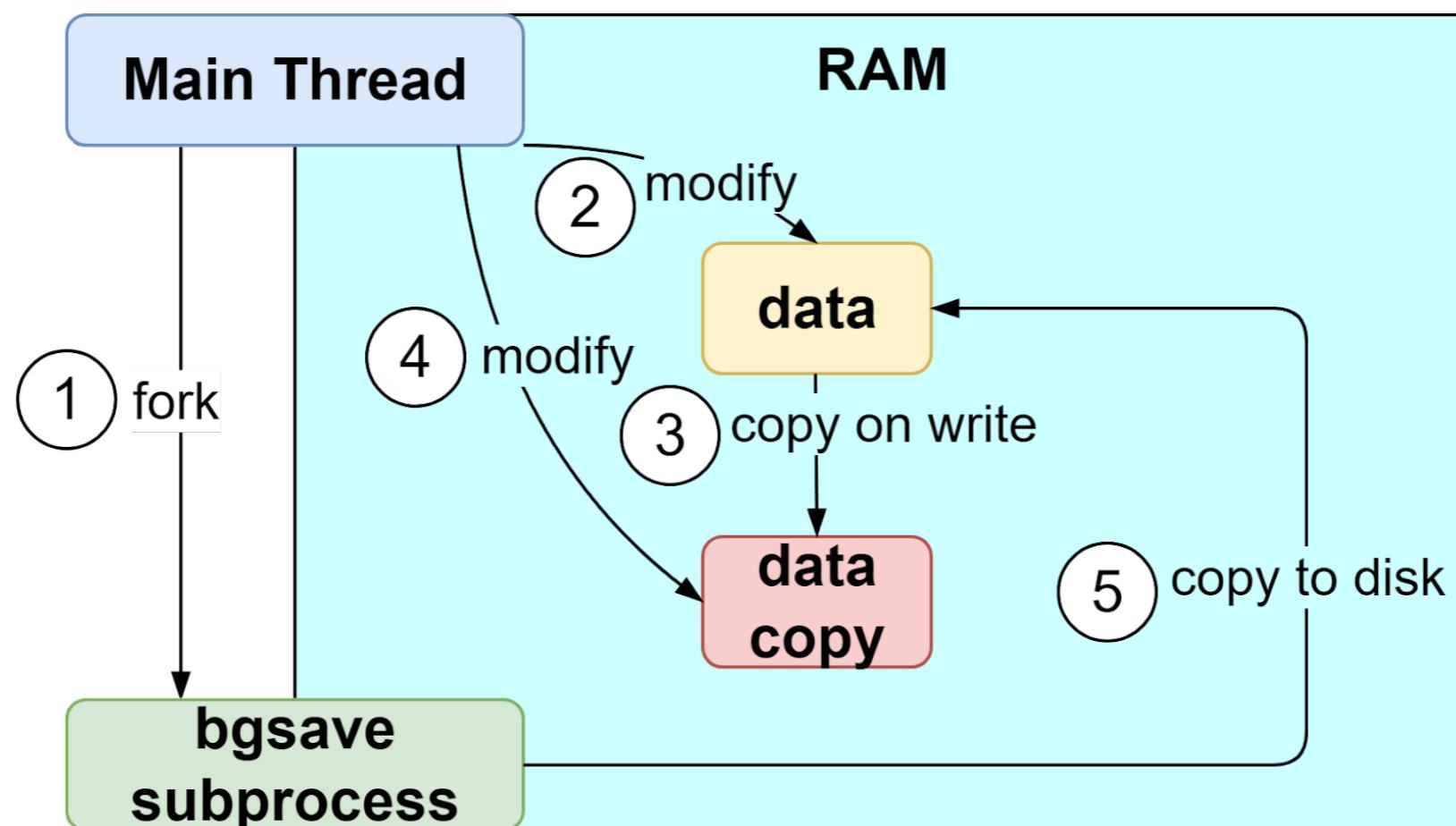
RDB (Redis Database)
AOF (Append Only File)
No persistence
RDB + AOF

https://redis.io/docs/latest/operate/oss_and_stack/management/persistence/



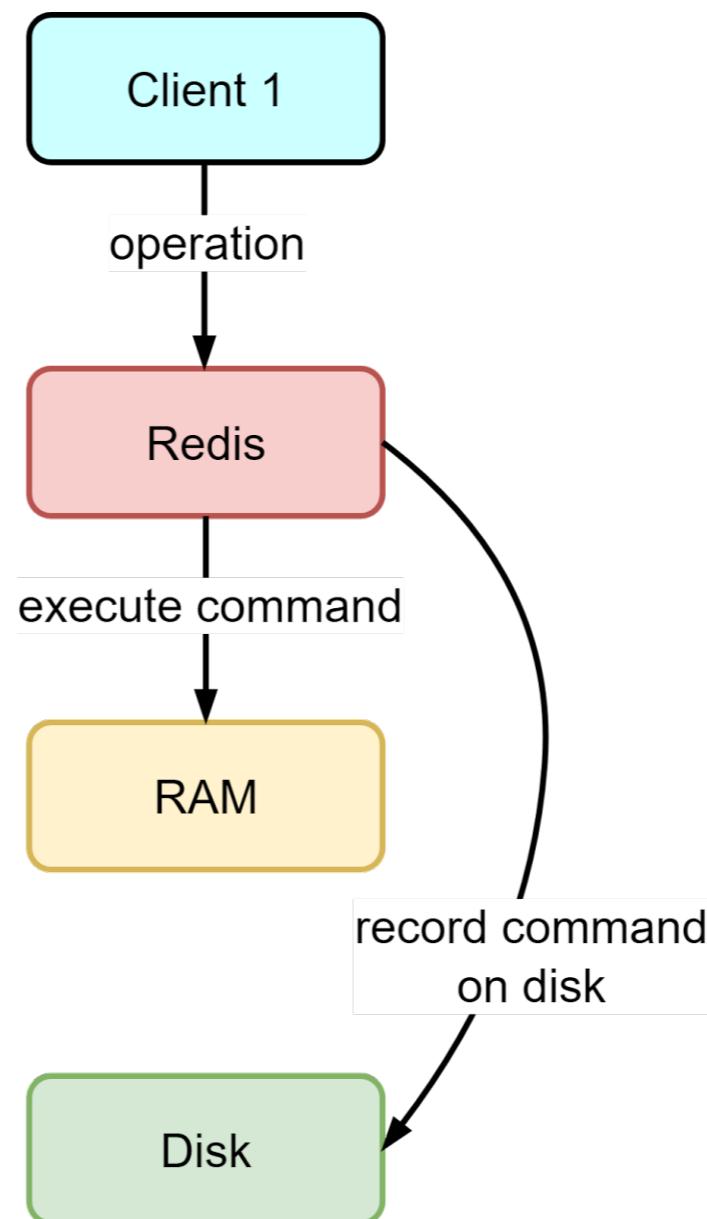
RDB (Redis Database)

Snapshot data from point-in-time
Copy-on-write solution, don't block on write



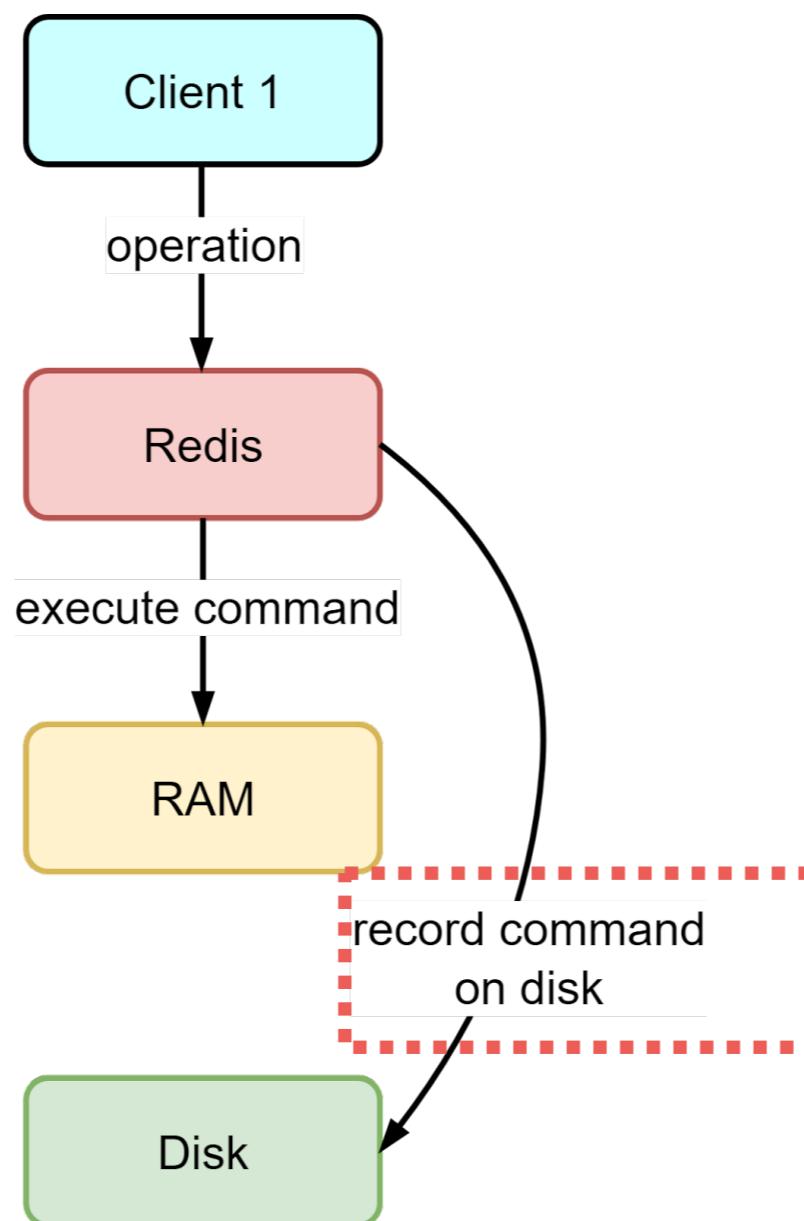
AOF (Append Only File)

Write-after log



AOF (Append Only File)

Every commands write data on disk !!



Bad performance !!
High I/O on server



AOF (Append Only File)

Try to configuration in Redis

Solution	Description
Always	Write all commands to disk
Everysec	Write all commands in memory buffer first, write to disk every second
No	Write all commands in memory buffer first, decision to flush by OS !!



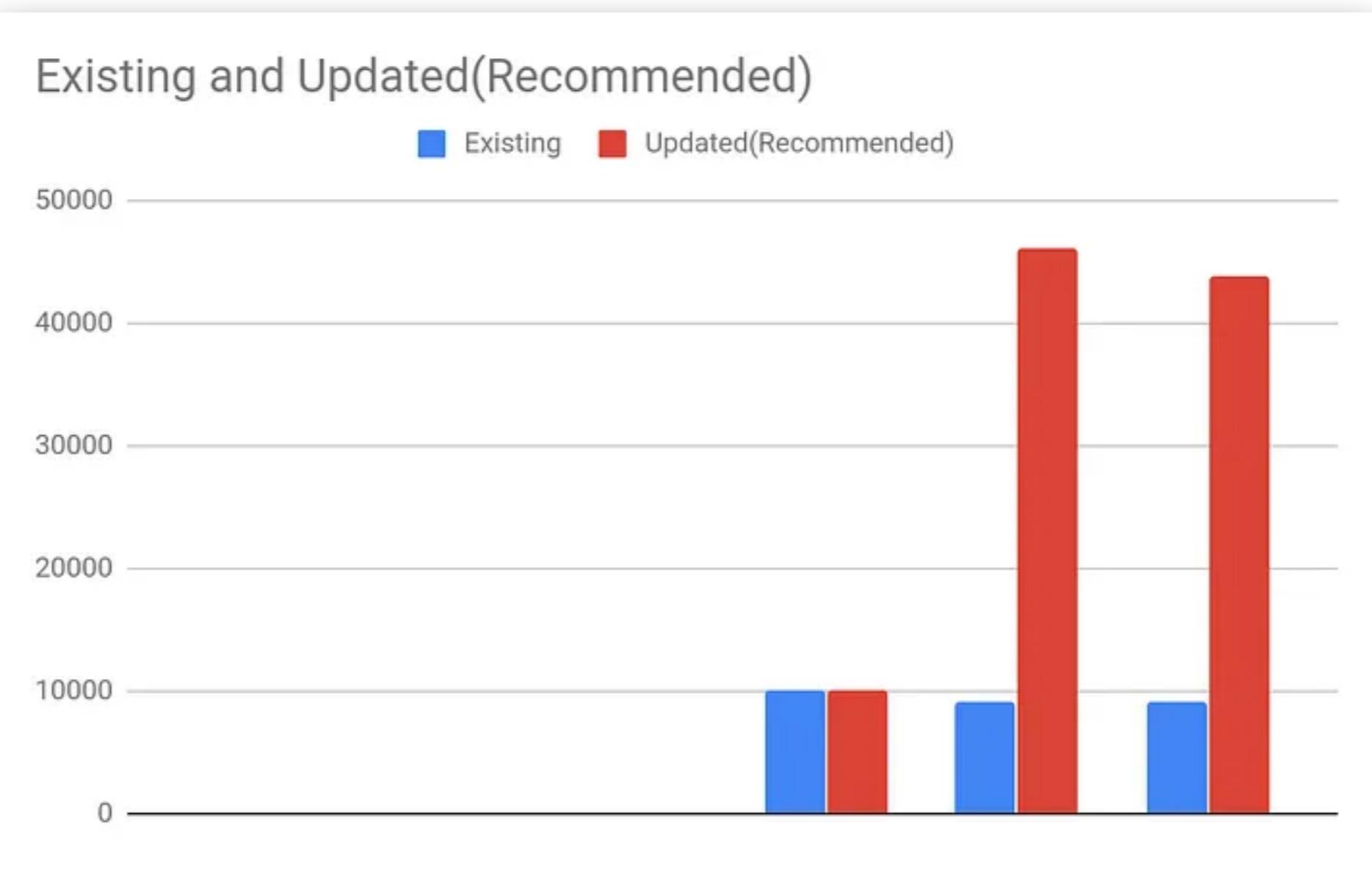
Keep-Alive

tcp-keepalive 0

Allow the same TCP connection
for HTTP conversation instead of
opening a new one with each new request



Keep-Alive



<https://iamabhishek-dubey.medium.com/redis-best-practices-and-performance-tuning-c48611388bbc>



Security

Add password to authorized
requirepass in redis.conf

<https://www.dragonflydb.io/guides/redis-best-practices>



Limitation of Redis !!



Limitation of Redis !!

Memory size with large data
OOM (Out of Memory)

Redis provides eviction policy

<https://redis.io/docs/latest/operate/rs/databases/memory-performance/eviction-policy/>



Redis eviction policy

Eviction Policy	Description
noeviction	New values aren't saved when memory limit is reached When a database uses replication, this applies to the primary database
allkeys-lru	Keeps most recently used keys; removes least recently used (LRU) keys
allkeys-lfu	Keeps frequently used keys; removes least frequently used (LFU) keys
allkeys-random	Randomly removes keys
volatile-lru	Removes least recently used keys with expire field set to true
volatile-lfu	Removes least frequently used keys with expire field set to true
volatile-random	Randomly removes keys with expire field set to true
volatile-ttl	Removes least frequently used keys with expire field set to true and the shortest remaining time-to-live (TTL) value

<https://redis.io/docs/latest/operate/rs/databases/memory-performance/eviction-policy/>



Limitation of Redis

Single thread limitation with high throughput

Scaling Redis with Clustering

Partition workload by data type in each instance

Redis 6+ supports Multi-thread I/O

<https://redis.io/docs/latest/operate/rs/databases/memory-performance/eviction-policy/>



Redis Topologies

Scale up

Scale out

Read/Write

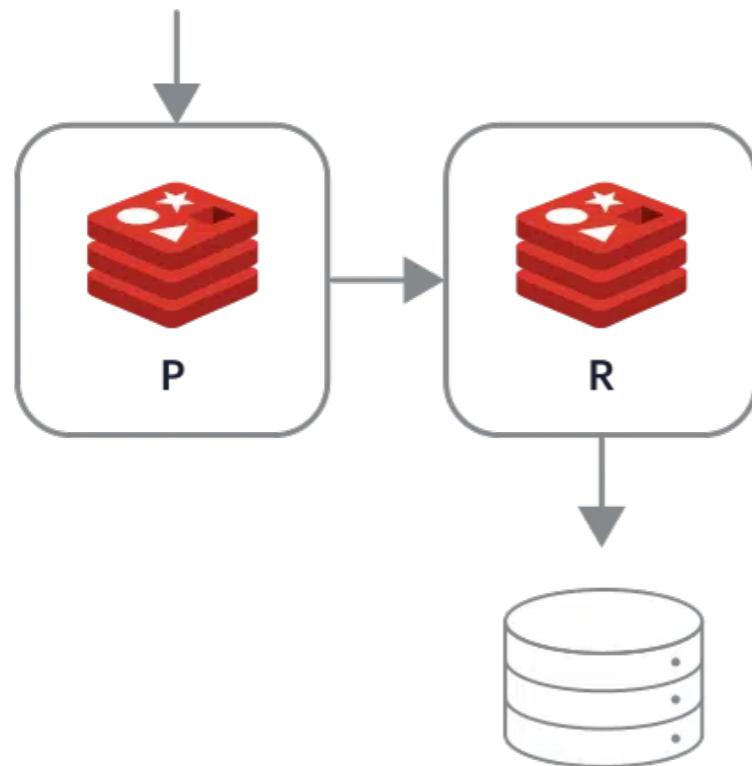
High availability



Speed vs Reliability

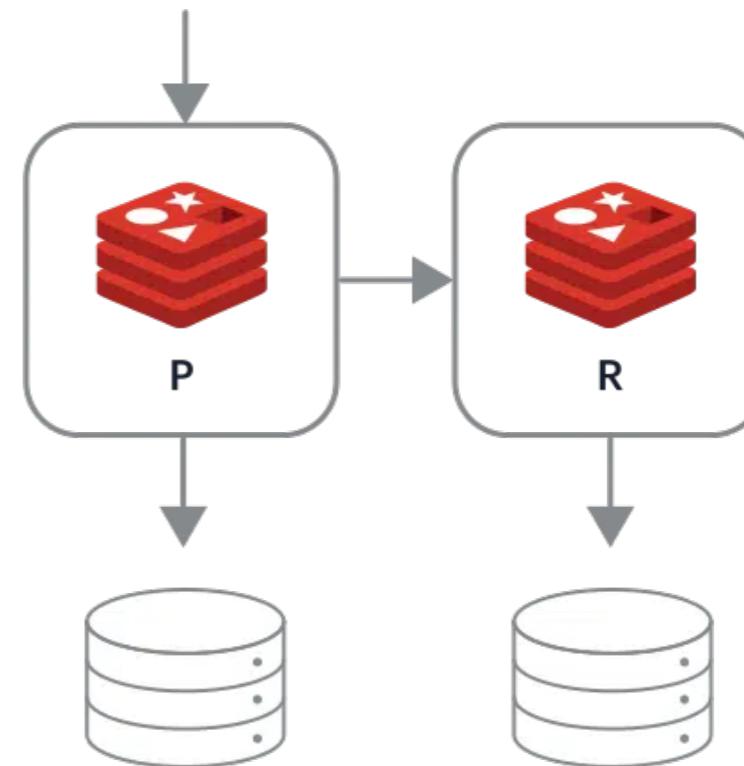
Tuned for Speed

Data-Persistence at the replica



Tuned for Reliability

Data-Persistence at the primary & replica



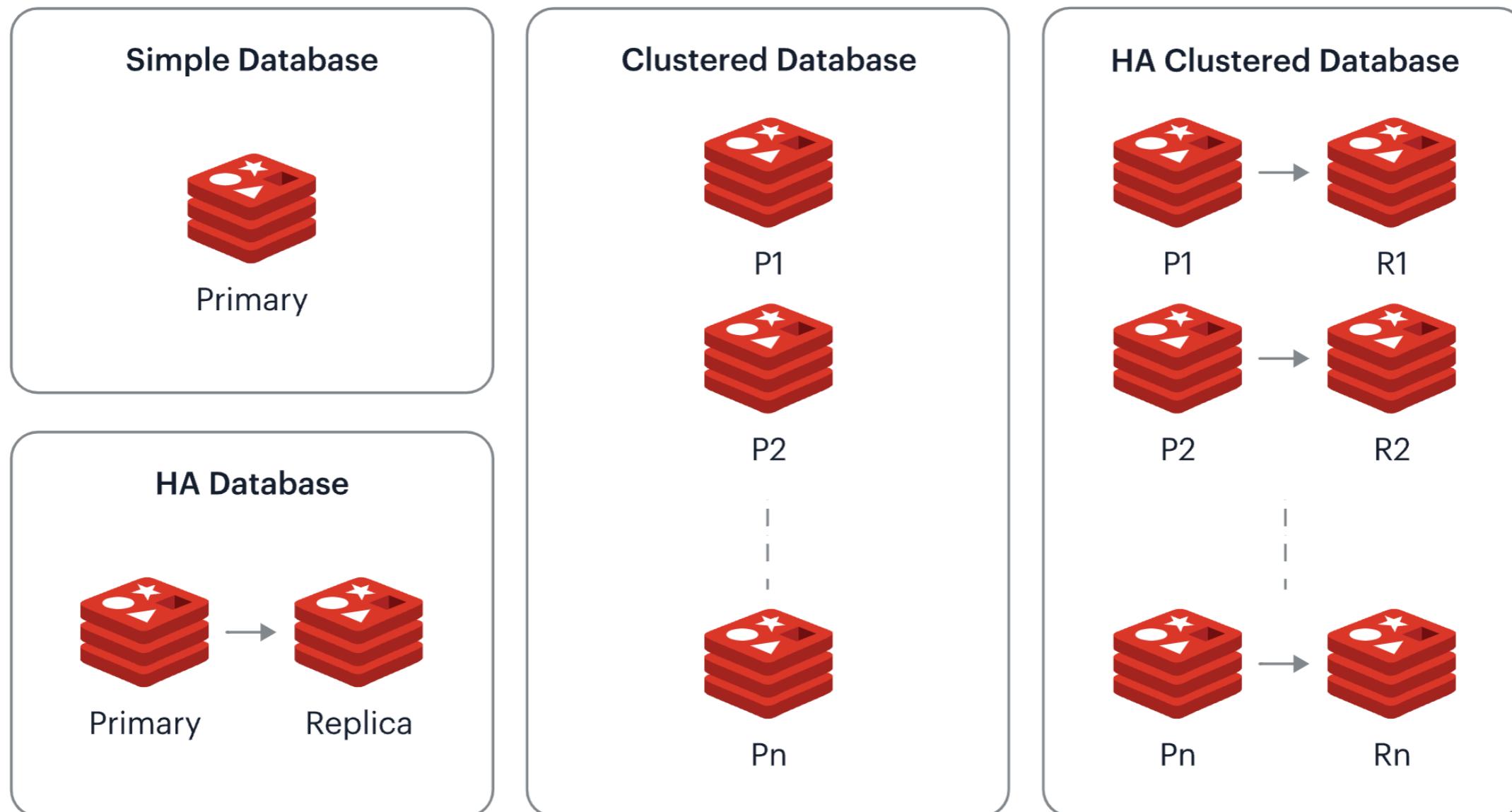
Redis Topologies

Single node
Multi-nodes (replication)
Master-slaves
Sentinel
Cluster + Sharing

<https://architecturenotes.co/p/redis>



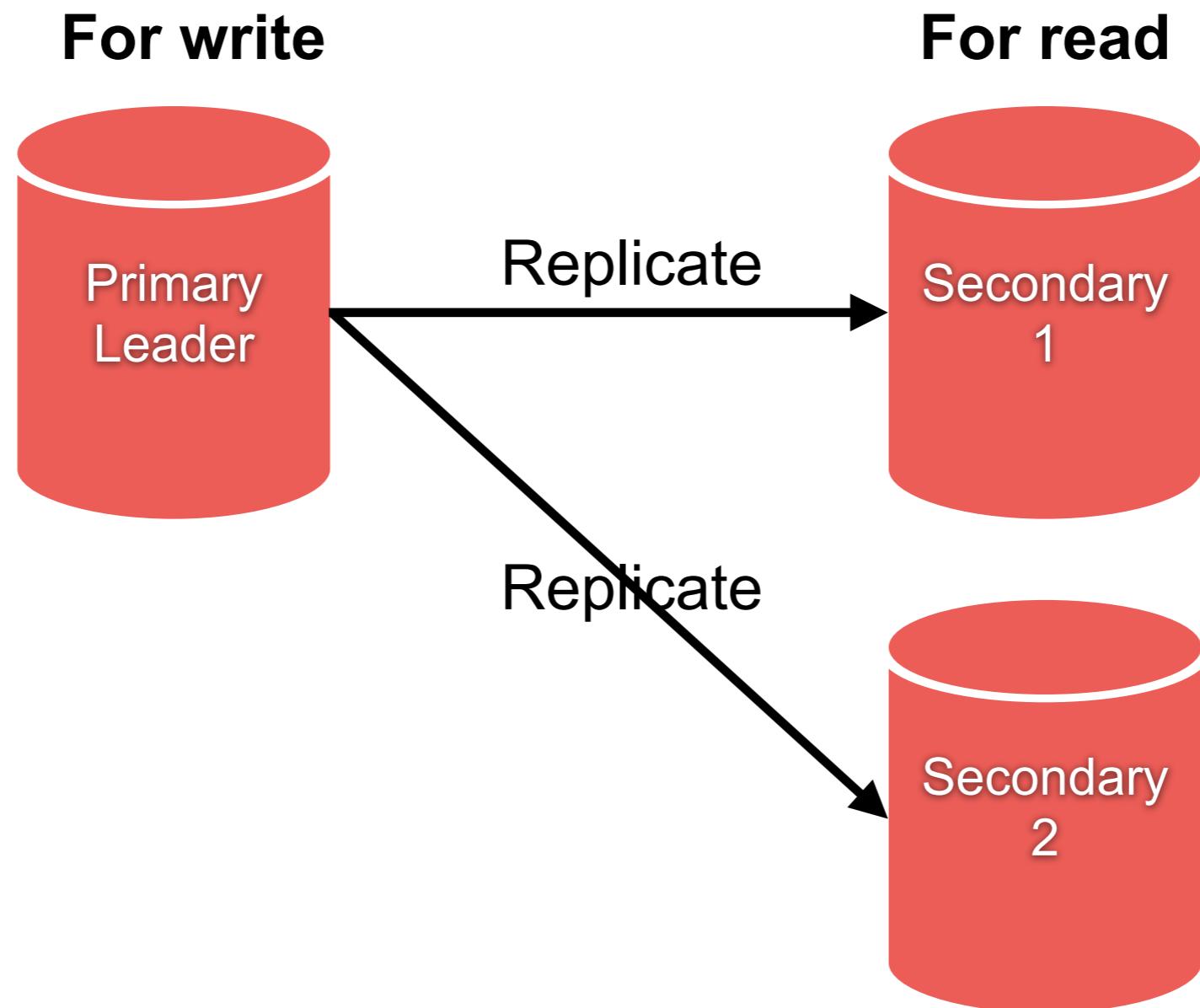
Redis Topologies



<https://redis.io/technology/redis-enterprise-cluster-architecture/>

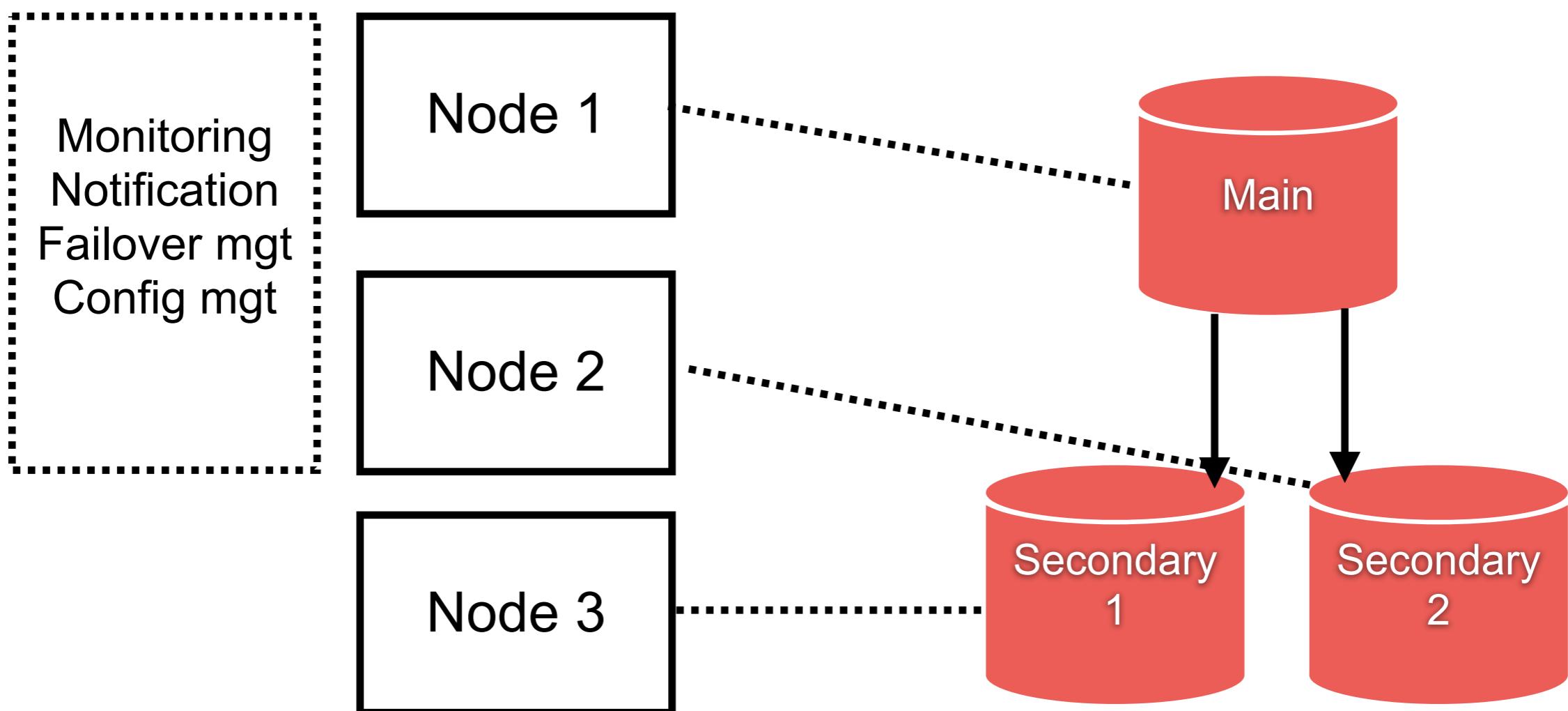


Replication



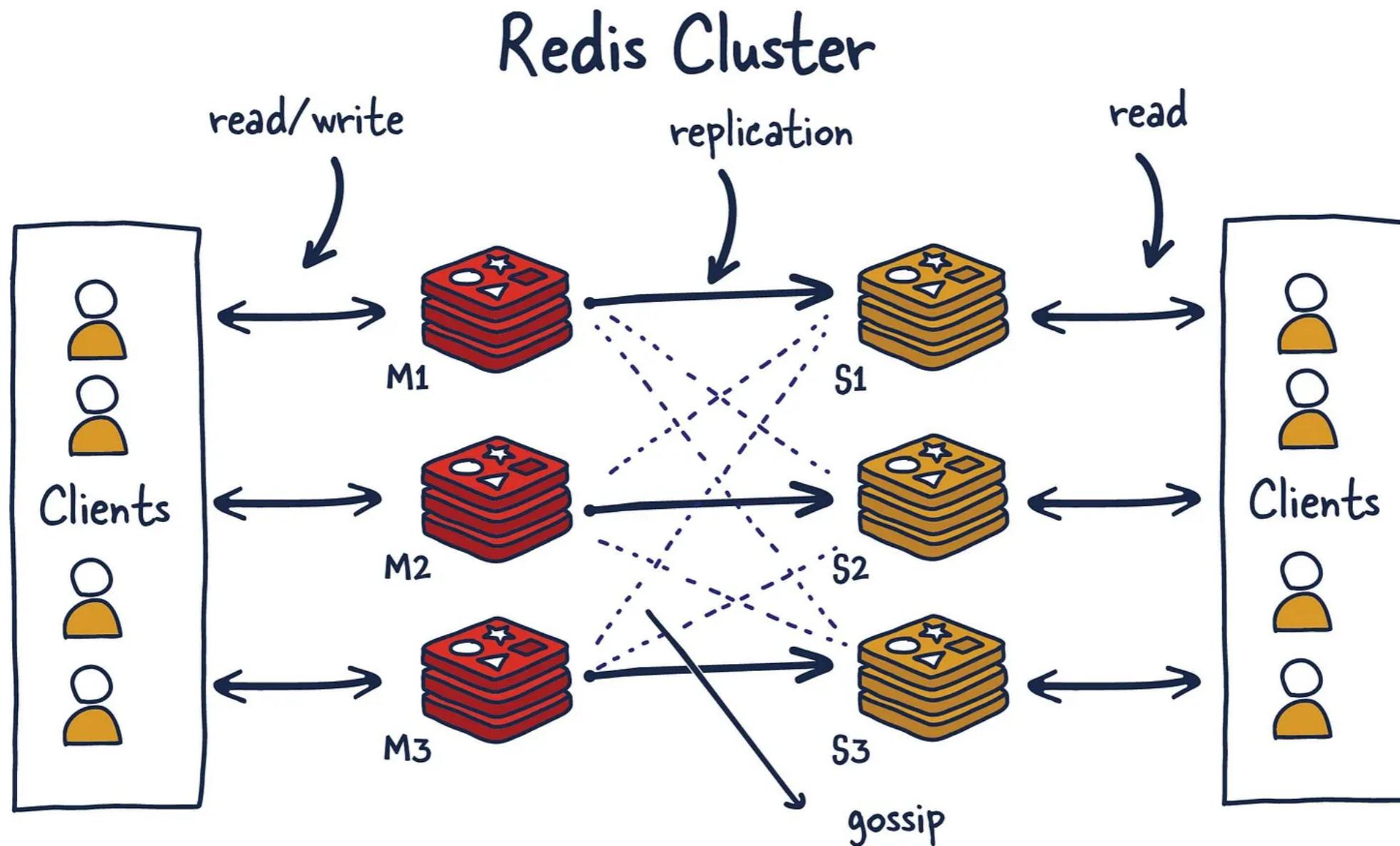
Redis sentinel (HA)

Sentinel nodes (odd number)



Redis cluster

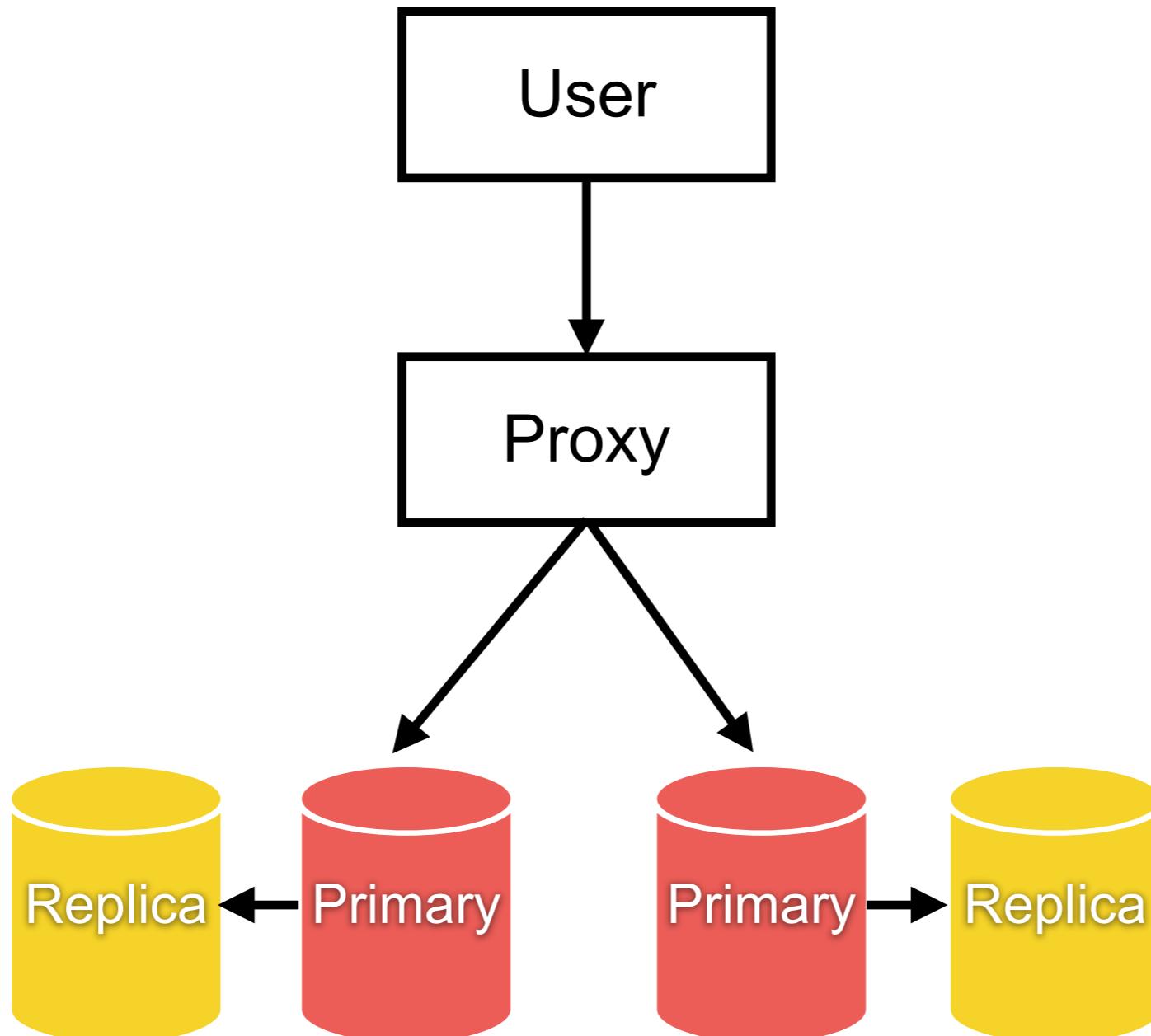
Minimum 6 nodes !!



https://redis.io/docs/latest/operate/oss_and_stack/management/scaling/



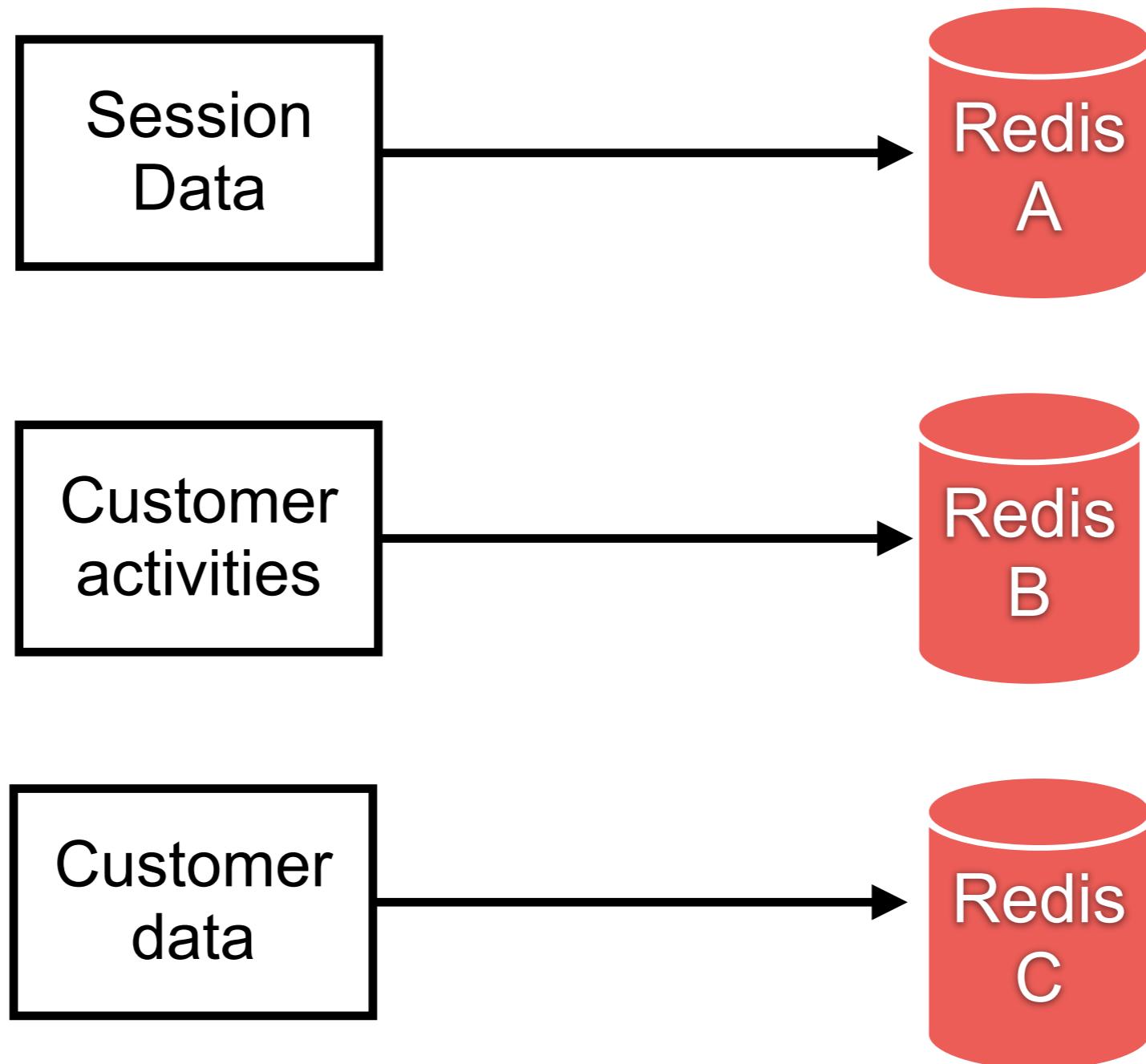
Proxy-based with HA



<https://semaphoreci.com/blog/redis-architectures>



Partition Data into nodes



Redis Capacity Planning

Understand your data

Estimate memory requirements

Understand read/write loads

Consider data persistence

Plan for redundancy and failover

Consider network bandwidth

Monitor and adjust !!

<https://www.dragonflydb.io/guides/redis-best-practices>



Monitoring



Redis Monitoring ?

Latency

Memory usage

CPU utilization

Throughput

Network

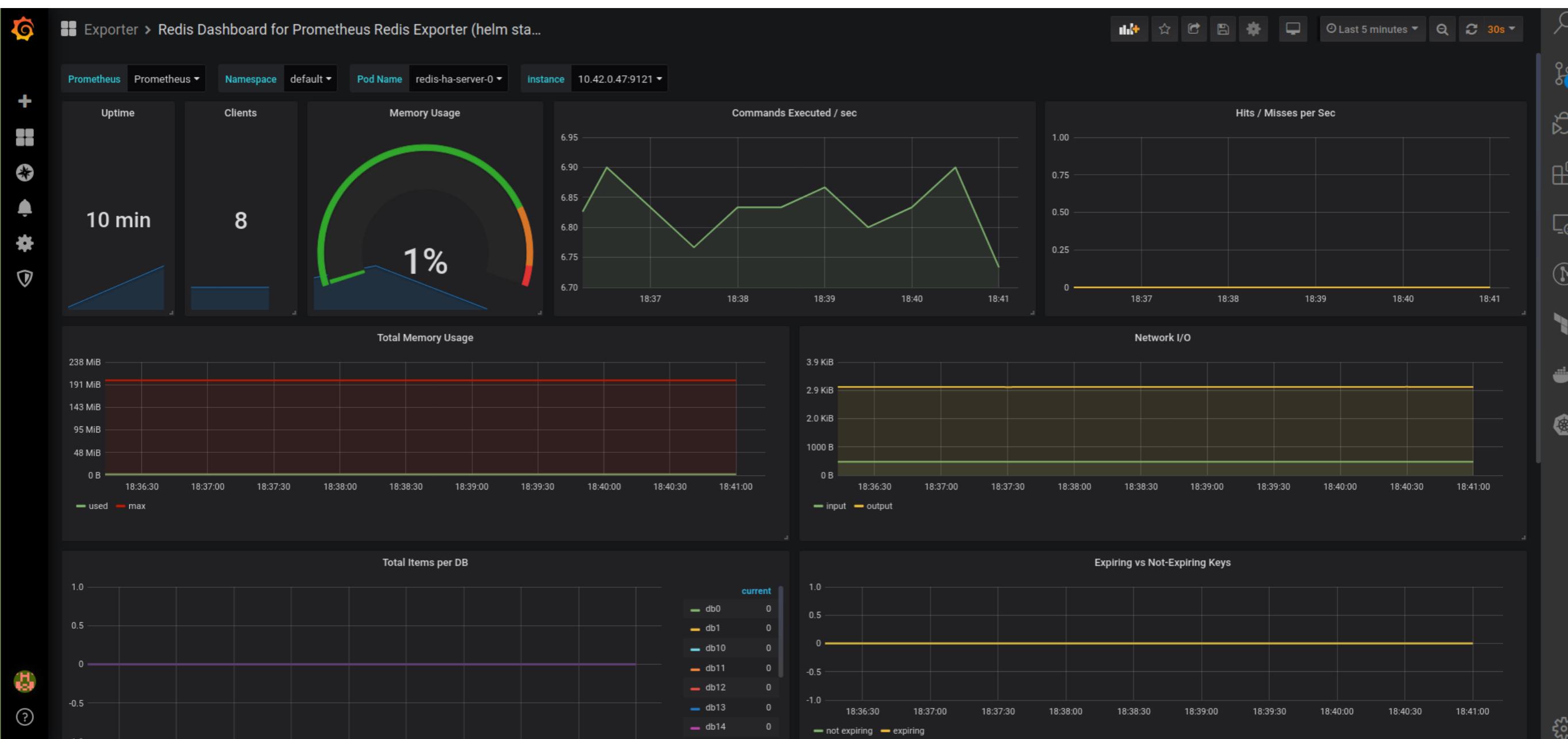
Keyspace

Logging

<https://uptrace.dev/blog/redis-monitoring>



Grafana Stack



<https://redis.io/docs/latest/integrate/prometheus-with-redis-cloud/>

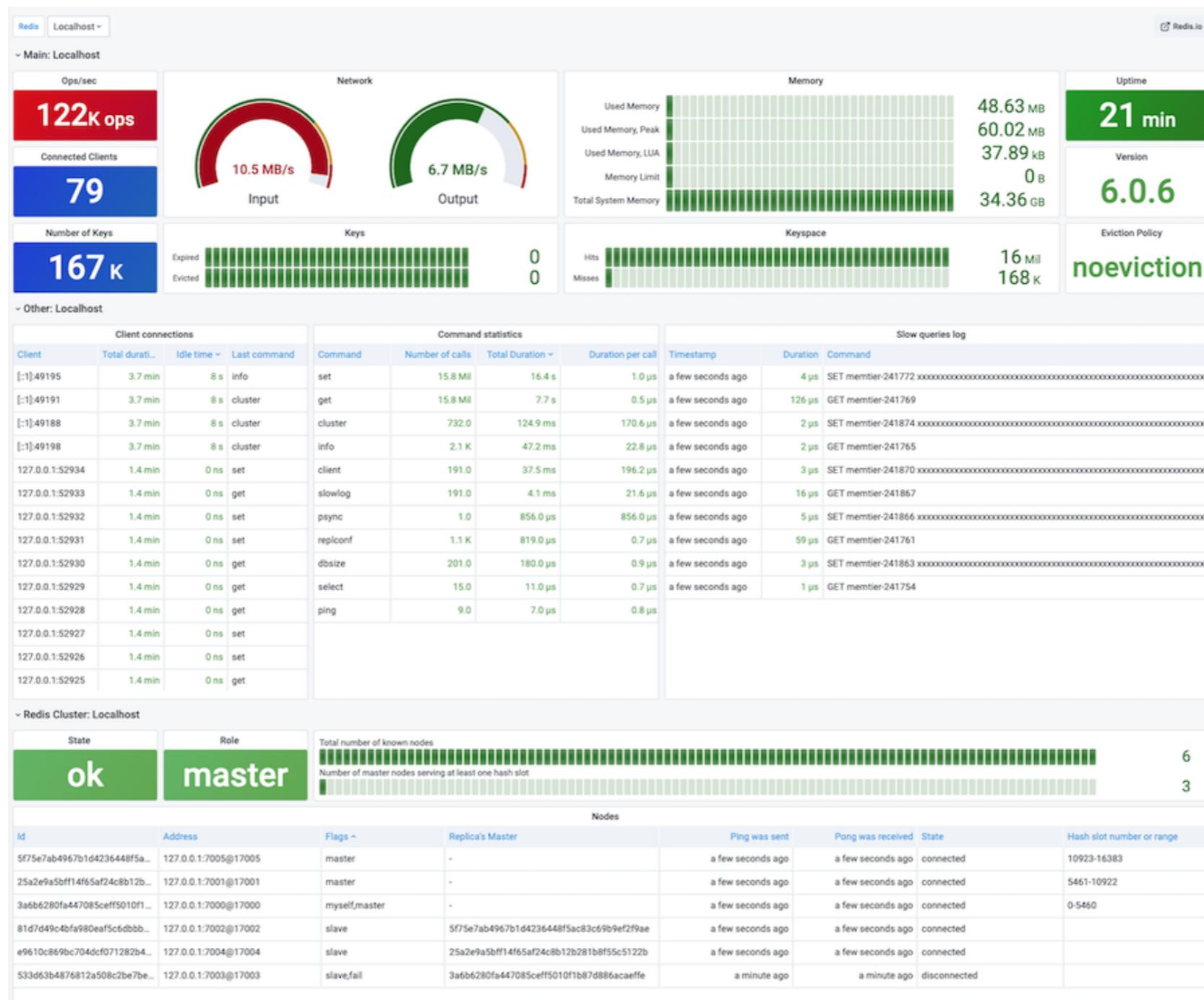


Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

106

Grafana Stack



<https://grafana.com/grafana/dashboards/12776-redis/>

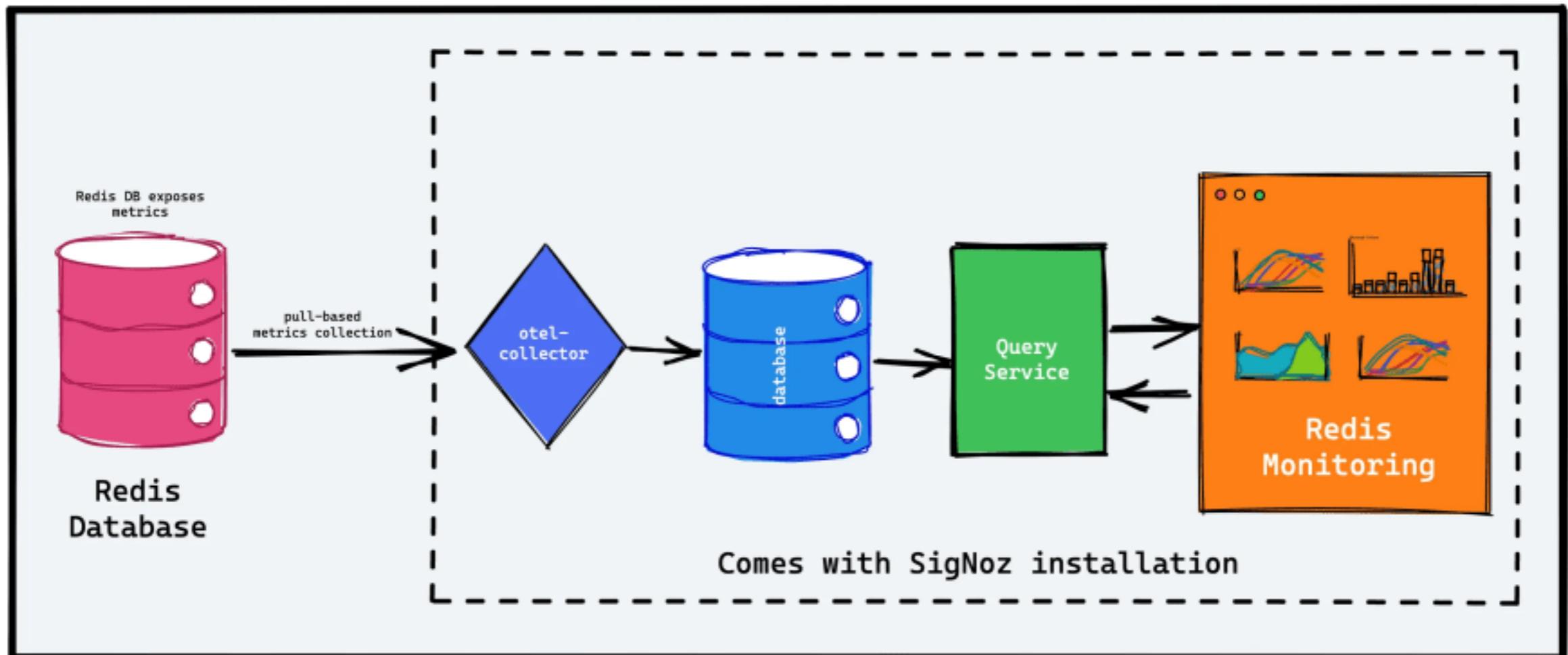


Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

107

OpenTelemetry



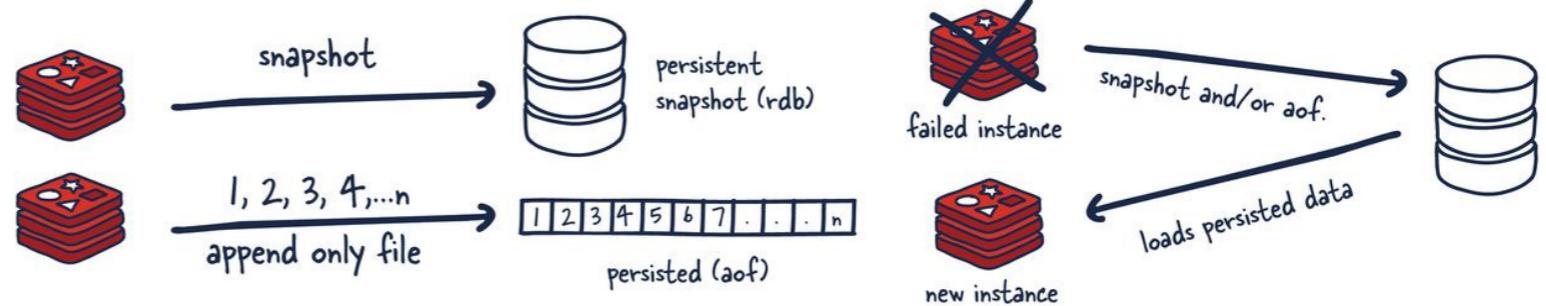
<https://signoz.io/blog/redis-opentelemetry/>



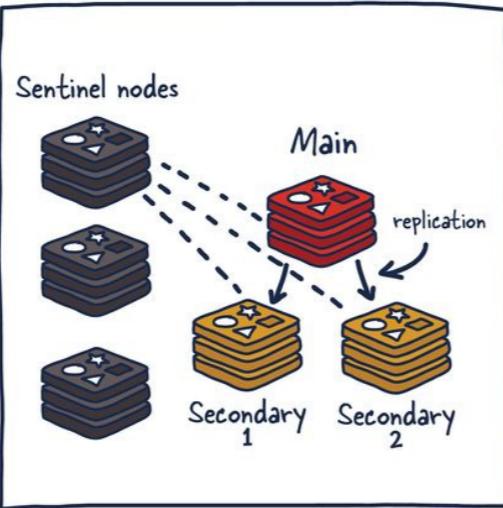
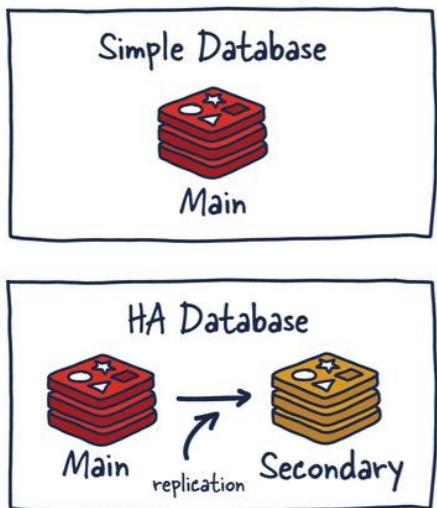
Summary



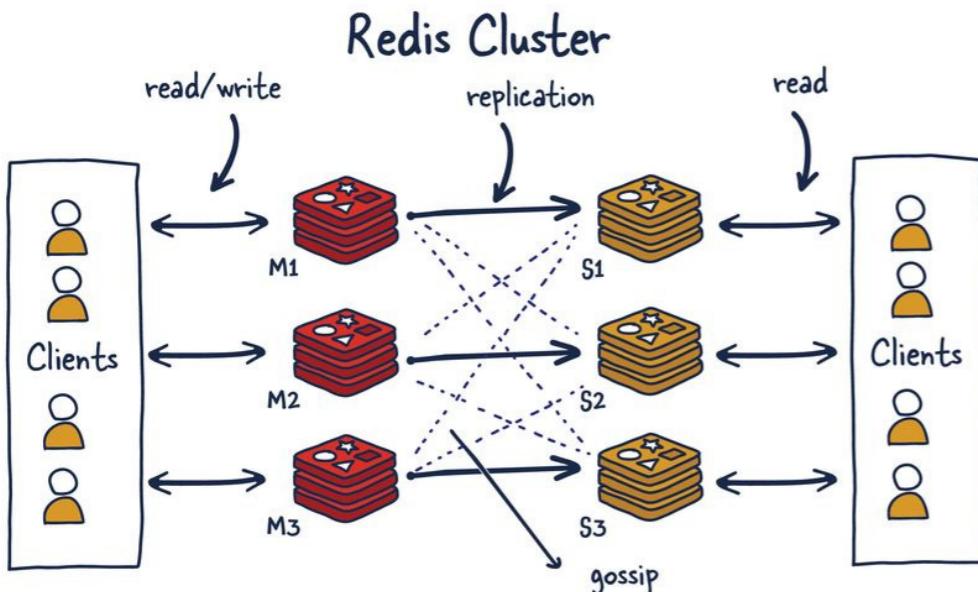
redis EXPLAINED



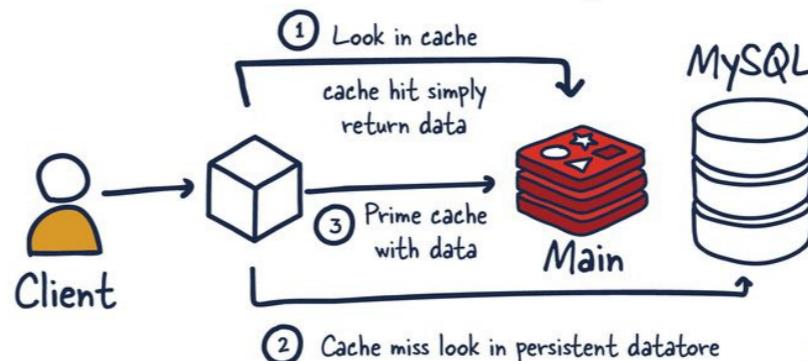
Redis setup



Redis sentinel



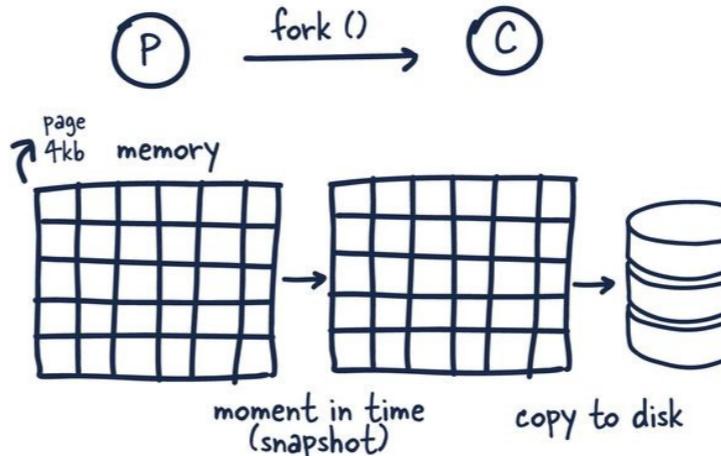
How is redis traditionally used



KEY
foo
VALUE
bar

hello world	String
011011010110111101101101	Bitmap
{23334}{6634728}{916}	Bitfield
{a: "hello", b: "world"}	Hash
[A>B>C>C]	List
{A<B<C}	Set
{A:1, B:2, C:3}	Sorted set
{A: (50.1, 0.5)}	Geospatial
01101101 01101111 01101101	Hyperlog
{id1=time1.seq({ a: "foo", a: "bar" }) }	Stream

Forking



architecture notes

<https://architeturenotes.co/p/redis>



Redis

© 2020 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Q/A

