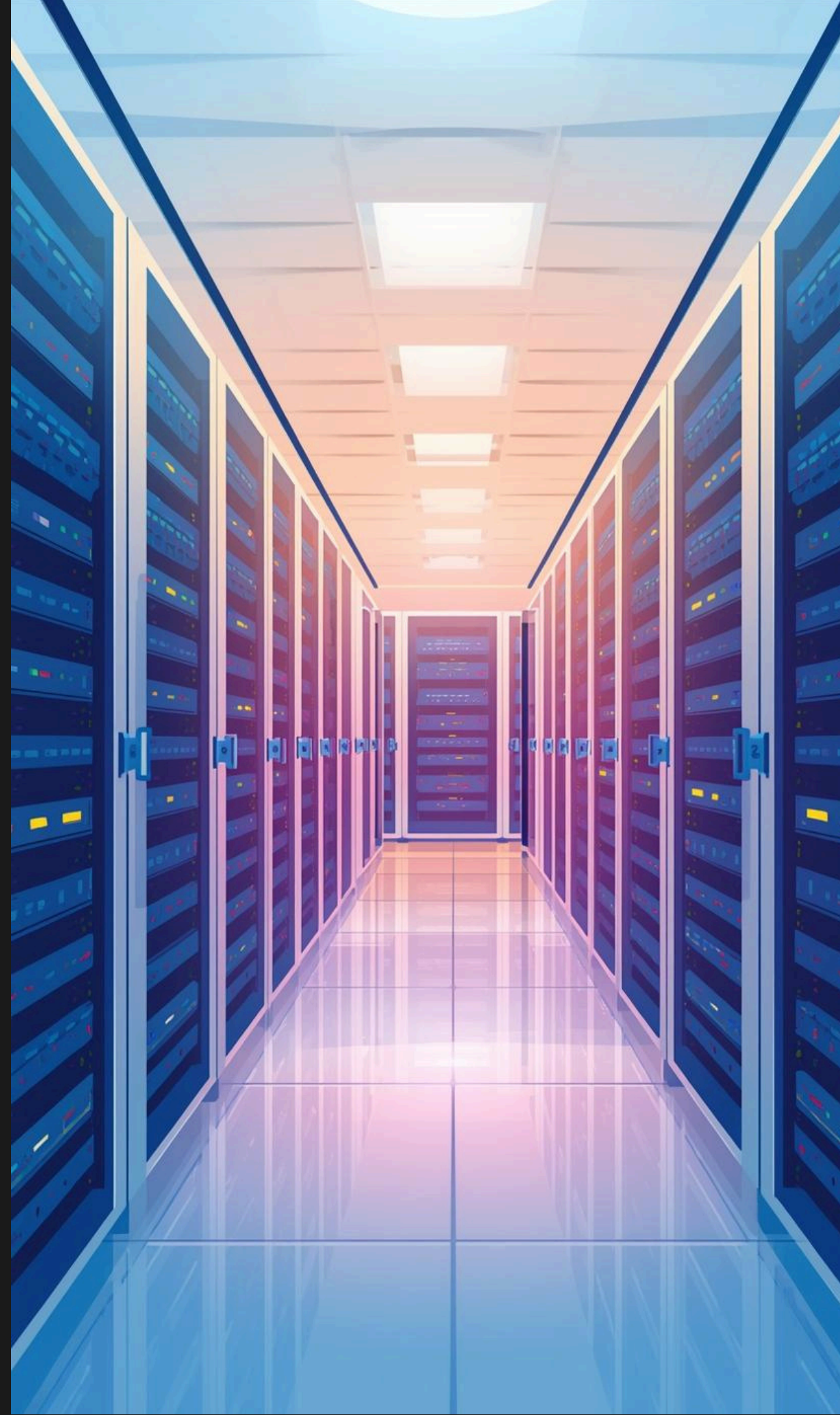


Caches distribués: démonstration avec Redis

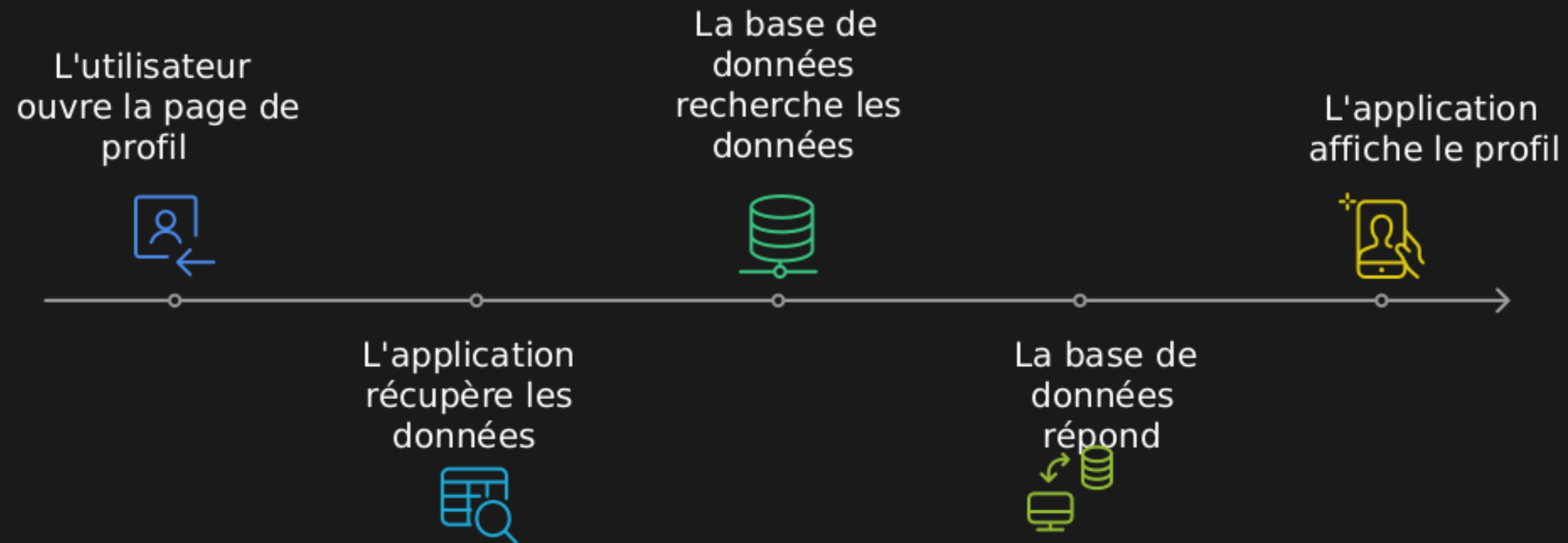
NIZAR GUERRABI
MOHAMED AMINE SALMANE



Plan

- ⦿1. Introduction
- ⦿2. types de cache
- ⦿3. pattern de caches
- ⦿4. Redis
- ⦿5. démo

INTRODUCTION

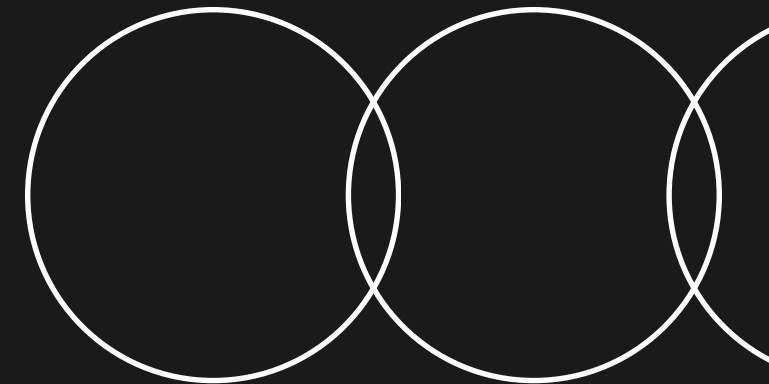


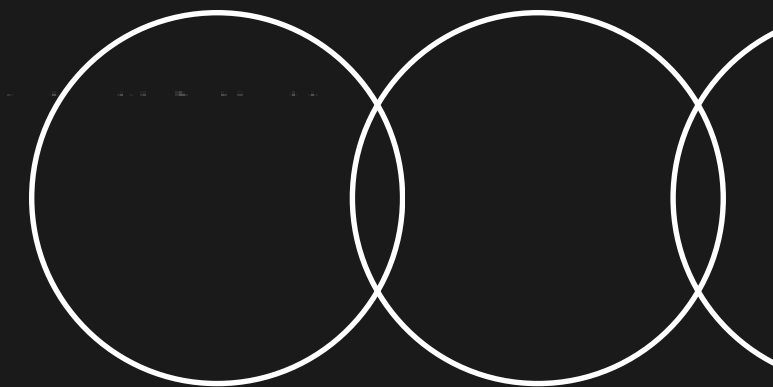
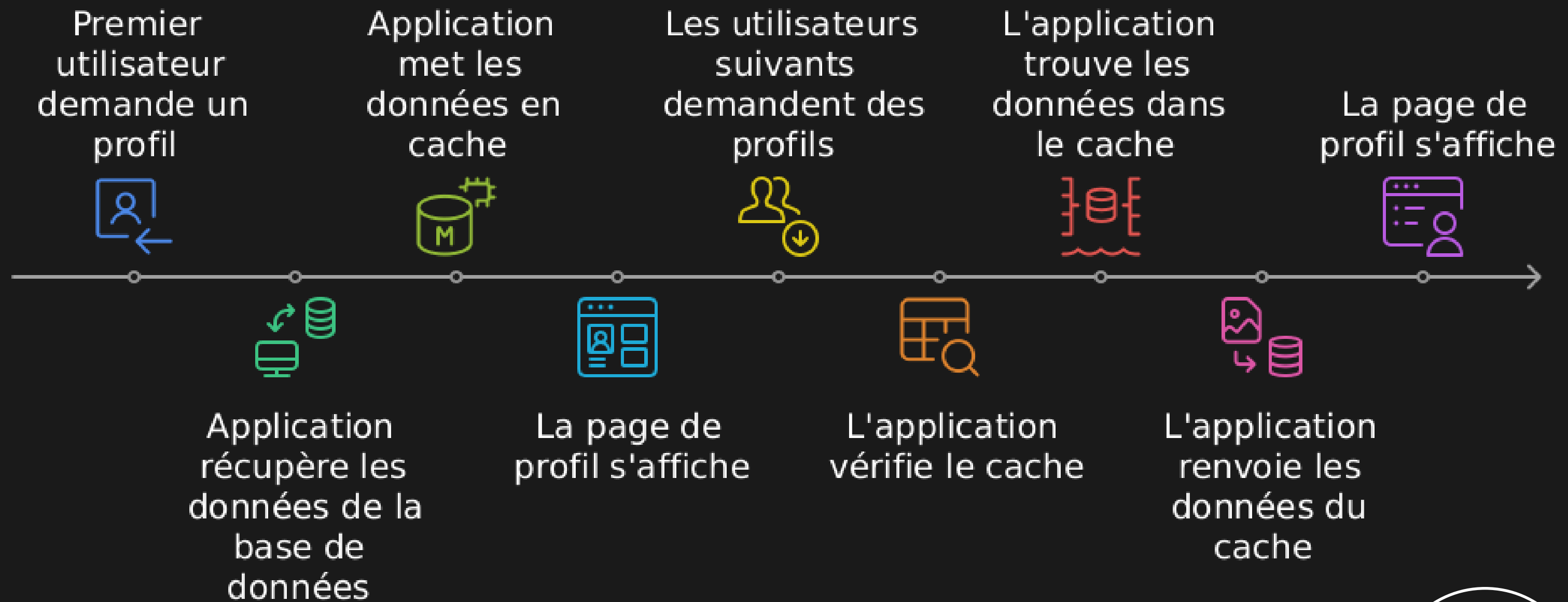
⚠
500
10 000
utilisateurs
demandent des
profils

10 000
requêtes SQL
sont envoyées
⚠
500

⚠
La base de
données
devient lente

L'application
devient lente
⚠





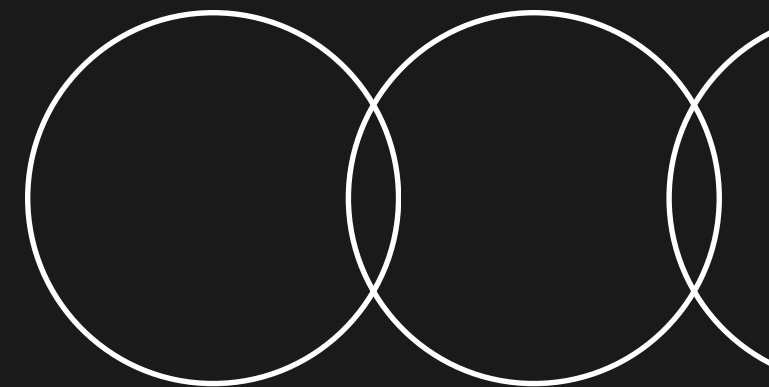
Le cache

RAM

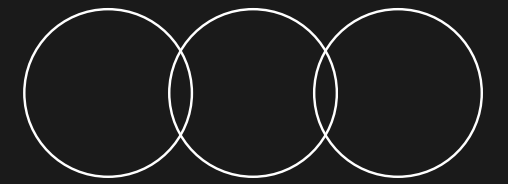
VS

Une base de
données

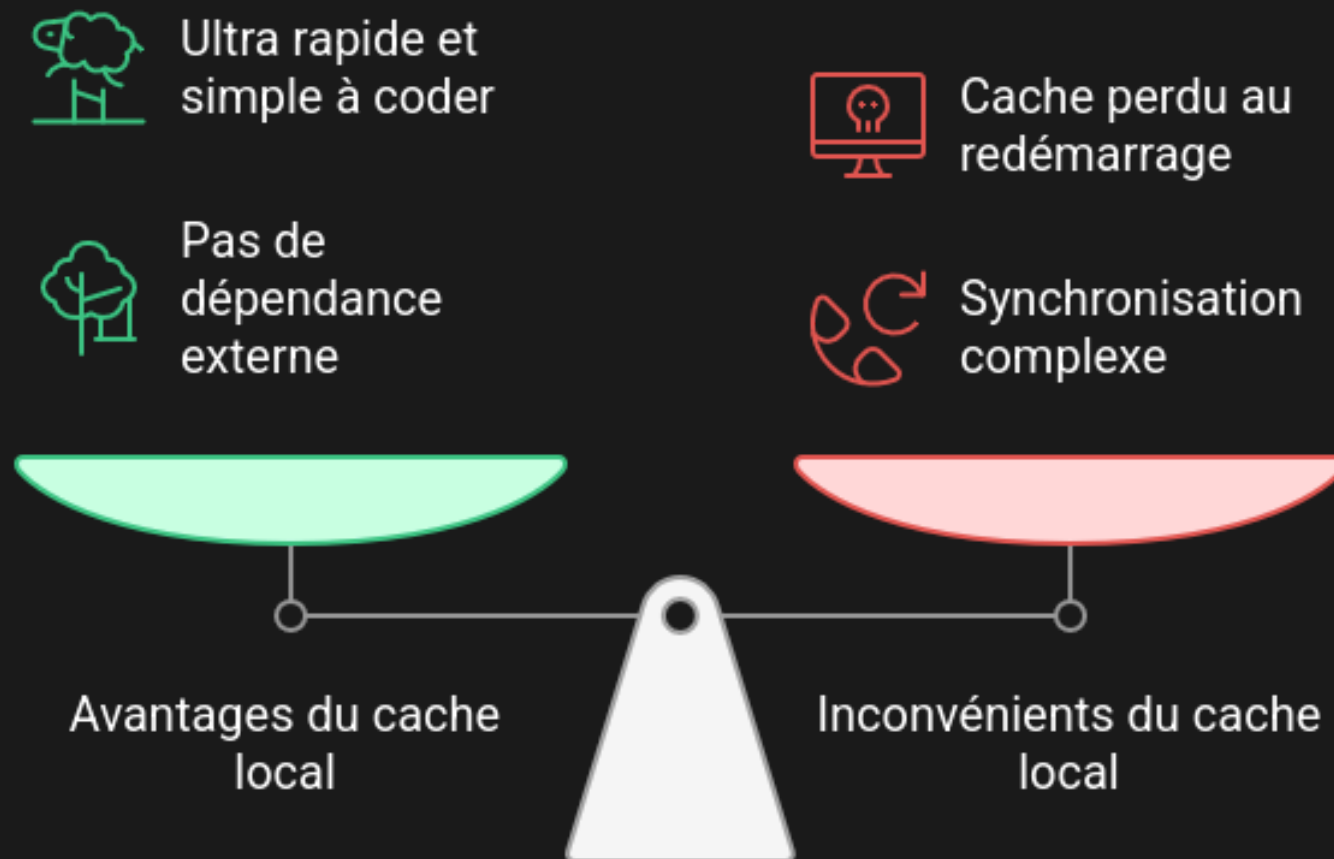
le disque dur



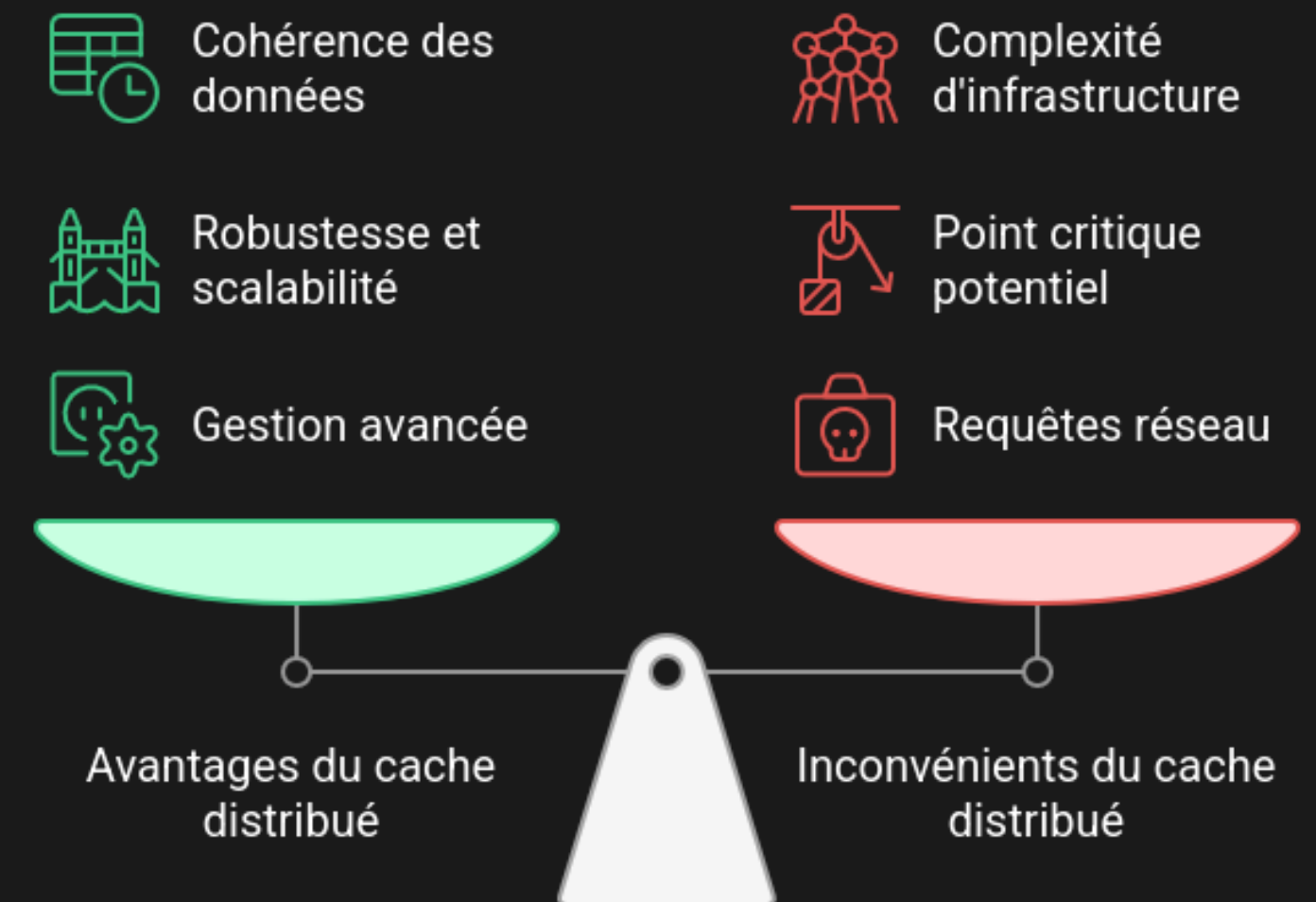
Les types de cache



Cache niveau code applicatif

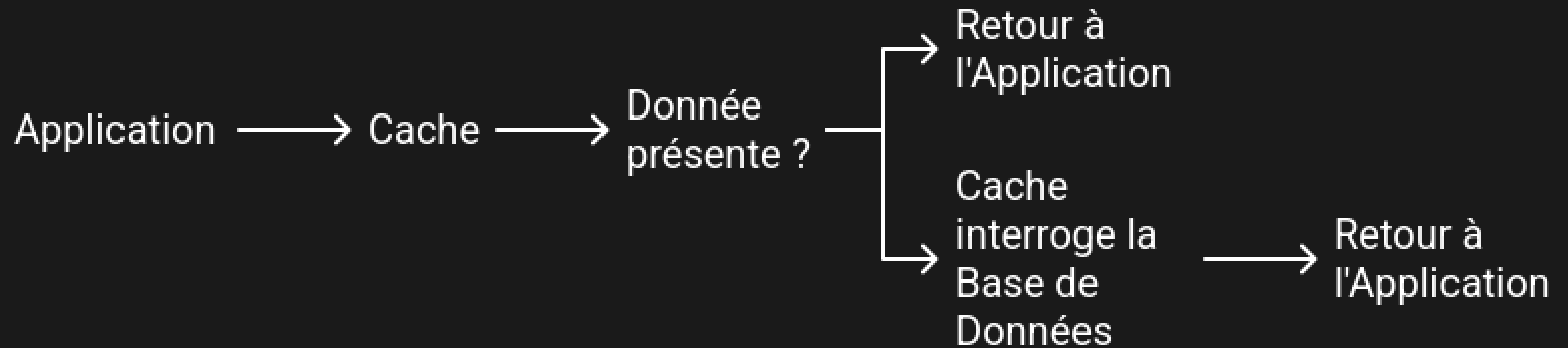


Cache distribué



Pattern cache

Read Through Pattern



Pattern cache

Write Through Pattern

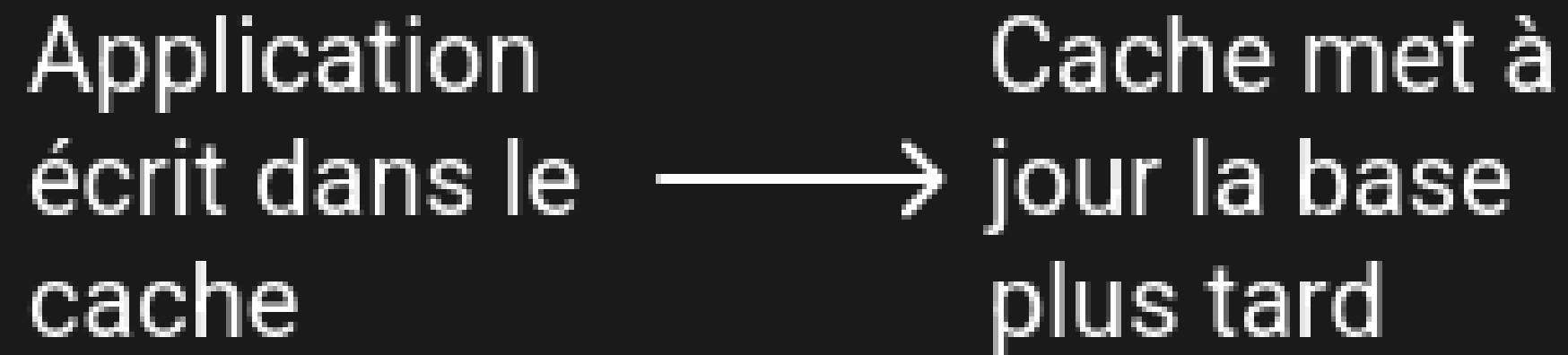
Processus d'écriture de données



Pattern cache

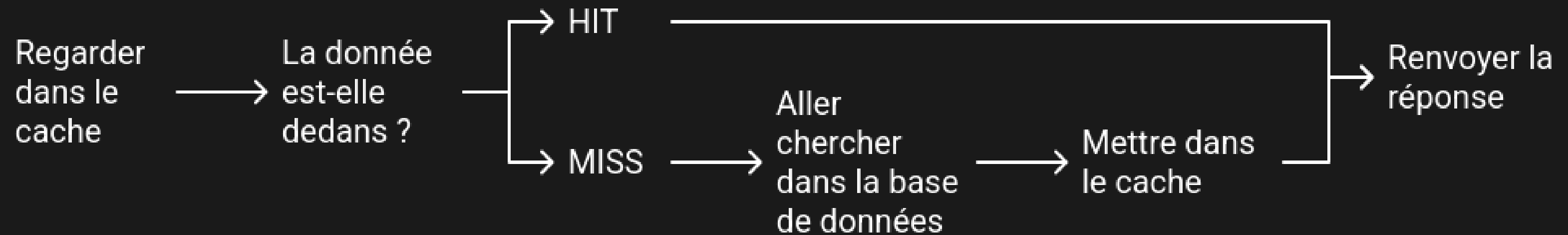
Write Behind Pattern

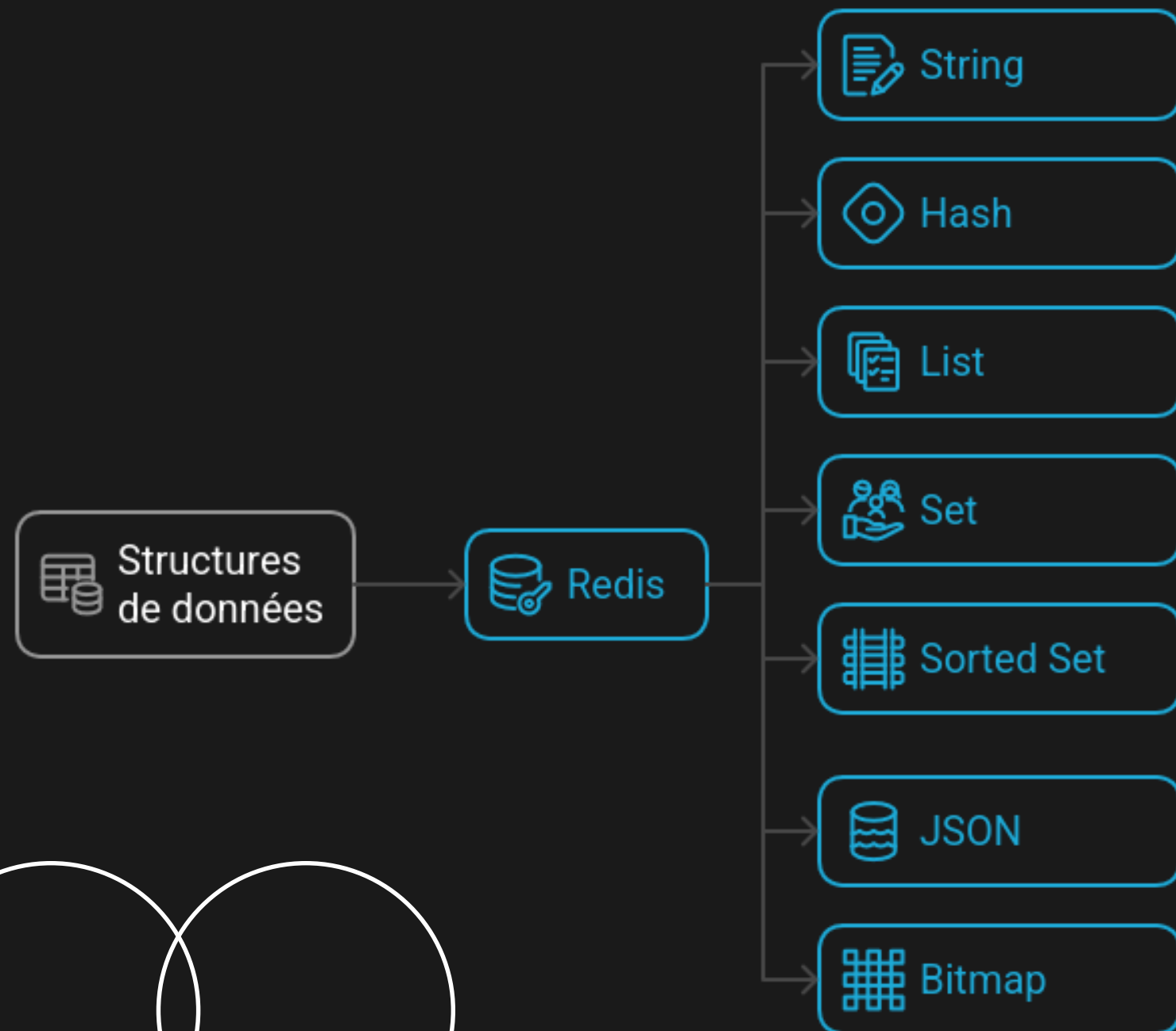
Processus de Write Back



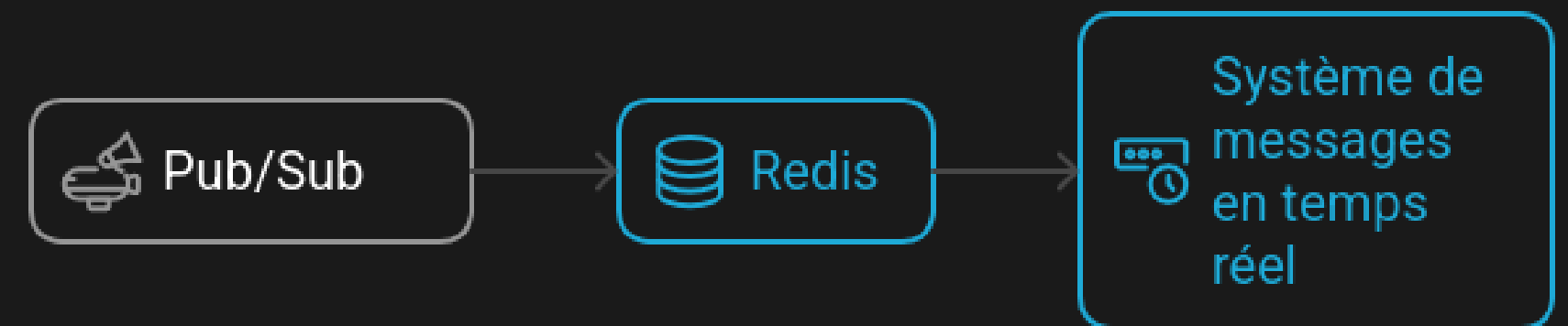
Pattern cache

Cache Aside Pattern

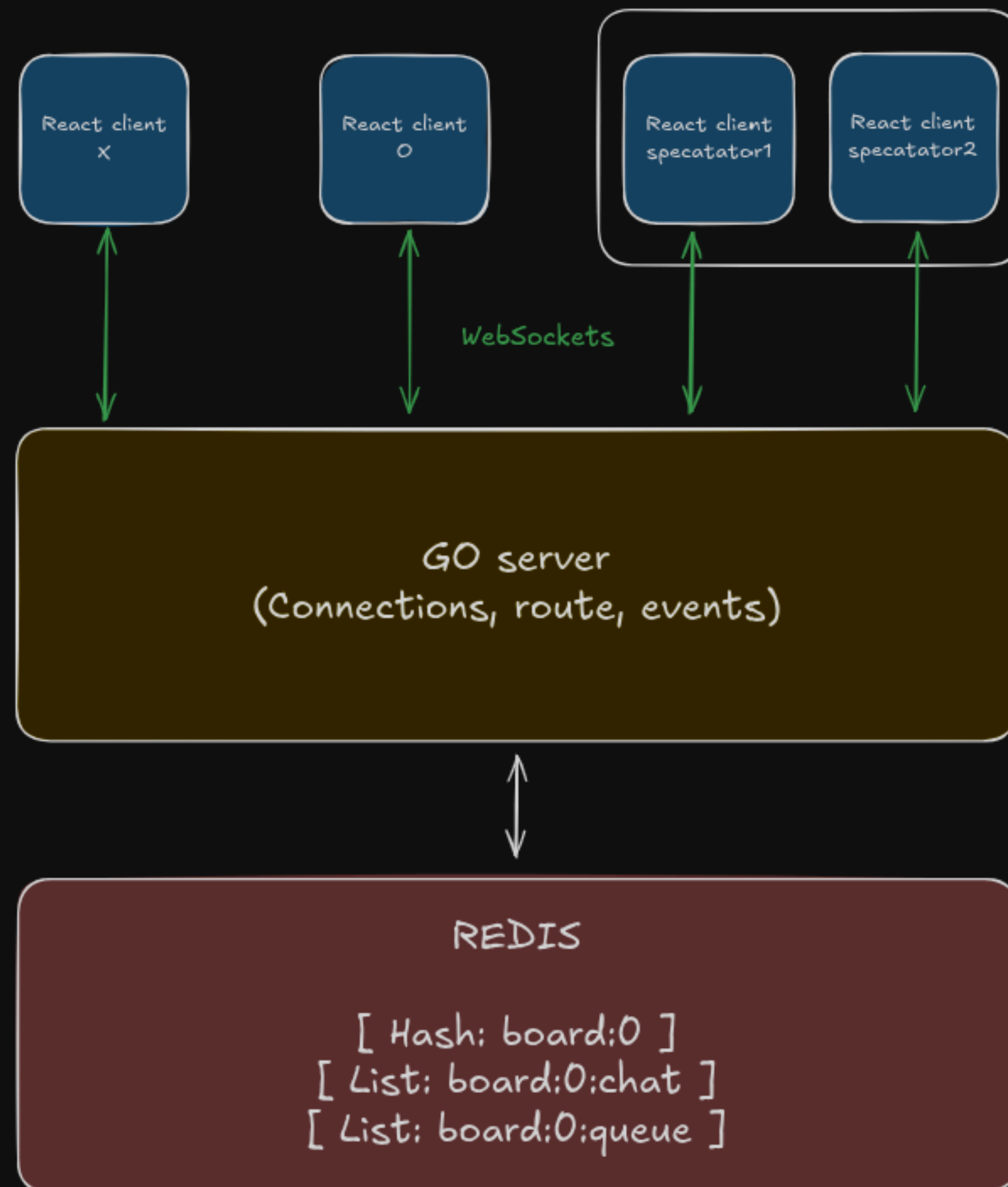




Publish / Subscribe



Demo : application real time
Tic-Tac-Toe
avec REDIS





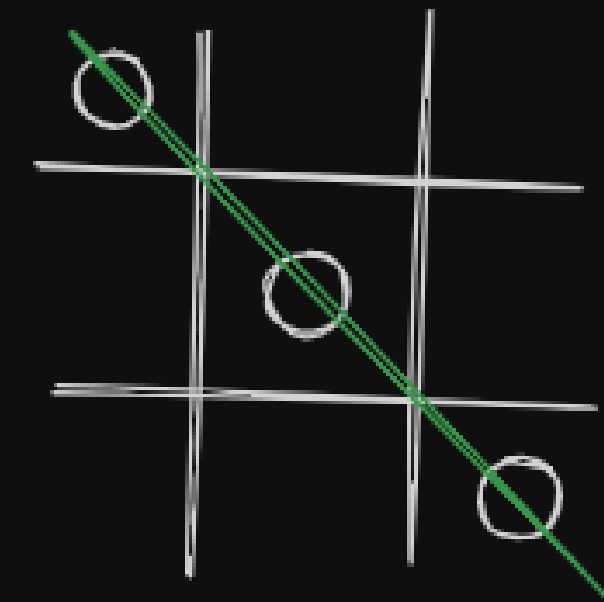
REDIS

Base de Données, Cache et Broker

1 Bdd => Stockage de l'état actuel du plateau de jeu et des informations des joueurs.

2 Broker => Diffusion instantanée des événements de jeu et de chat à tous les clients connectés.

3 Cache => Accès rapide à l'historique récent du chat et gestion de la file d'attente des spectateurs.



sous-ressource / collection

name space

type

id

[Hash: board:0 ↓]

[List : board:0:chat]

[List : board:0:queue]

```
127.0.0.1:6379> HGETALL board:0
```

1) "cells"

2) " _ ' _ ' _ ' _ ' _ ' _ ' _ ' _ "

3) "turn"

4) "X"

5) "winner"

6) " "

```
7) "playerXName"
```

8) "1"

```
9) "player0Name"
```

10) "2"

keys

vals

<http://redisx0.duckdns.org>