

YC Tech

웹 백엔드 실무 개발 프로젝트



연세대학교 미래교육원
YC (Yonsei X Codepresso) Tech Academy

6주차 recap



To DO

- ❑ Spring boot docker image 생성하여 ECR 에 push
- ❑ Github action 구성하여 CI/CD 도입
- ❑ AWS EC2 에서 Spring boot 구동
- ❑ ASG 를 EC2 에 구성
- ❑ RDS 구성
- ❑ 이미지 업로드 서비스 구현










CI/CD



Github action

- <https://docs.github.com/ko/actions>
 - <https://github.com/marketplace?type=actions>
-
- 소프트웨어 **workflow**를 자동화할 수 있도록 도와주는 도구
 - test
 - CI/CD
 - 자동화 스크립트
 - **Workflow** : 여러 **Job**으로 구성되고, **Event**에 의해 트리거될 수 있는 자동화된 프로세스 최상위 개념 **Workflow** 파일은 **YAML**으로 작성되고, **Github Repository**의 **.github/workflows** 폴더 아래에 저장됨
 - **Event** : **Workflow**를 **Trigger**(실행)하는 특정 활동이나 규칙
 - **Job** : 여러 **Step**으로 구성되고, 가상 환경의 인스턴스에서 실행됨 다른 **Job**에 의존 관계를 가질 수 있고, 독립적으로 병렬 실행도 가능함
 - **Step** : **Task**들의 집합으로, 커맨드를 날리거나 **action**을 실행할 수 있음
 - **Action** : **Workflow**의 가장 작은 블록(**smallest portable building block**) **Job**을 만들기 위해 **Step**들을 연결할 수 있음

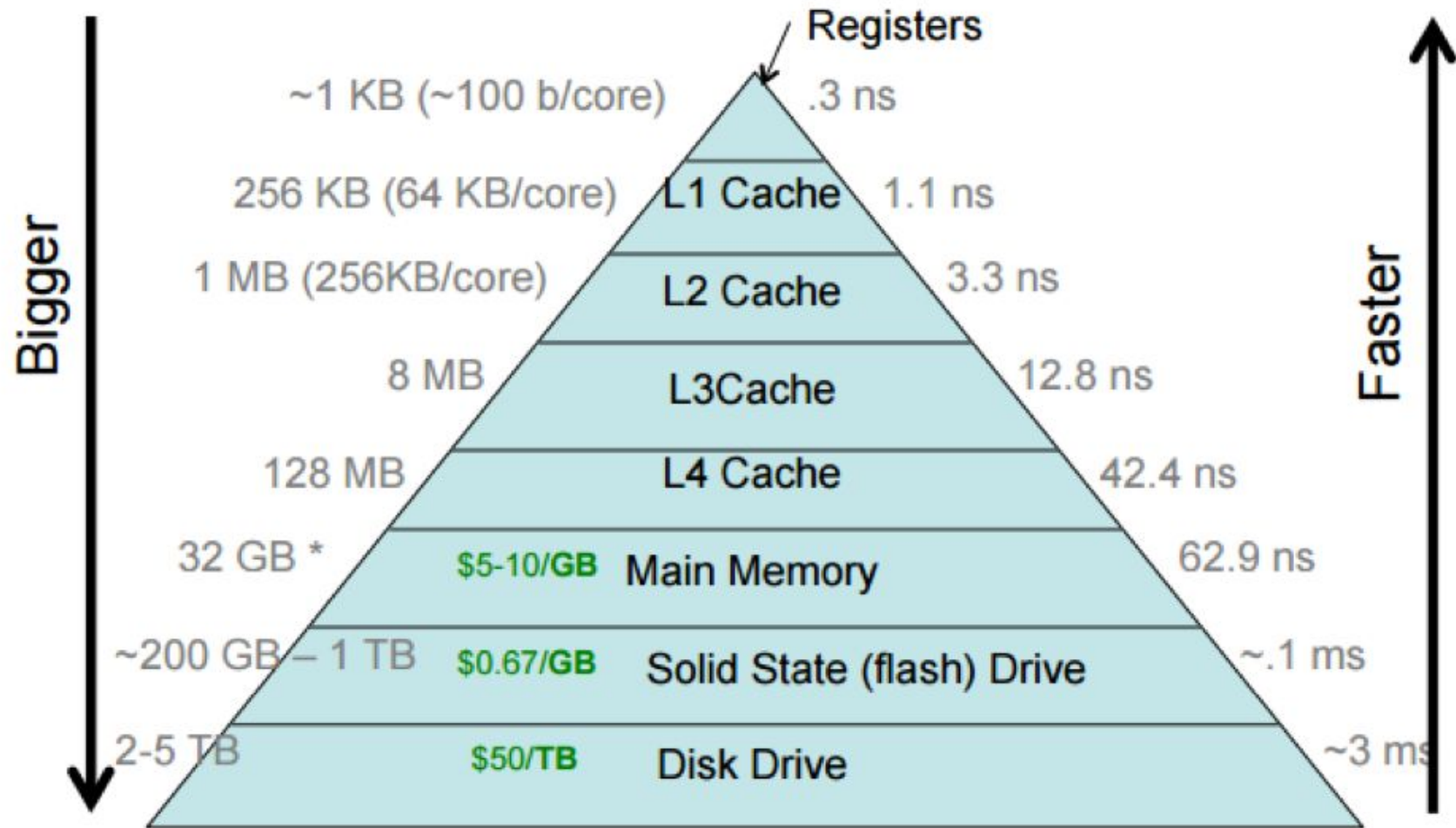
Github action

Repository secrets			
	AWS_ACCESS_KEY_ID	Updated 14 hours ago	 
	AWS_DEFAULT_REGION	Updated 14 hours ago	 
	AWS_SECRET_ACCESS_KEY	Updated 13 hours ago	 

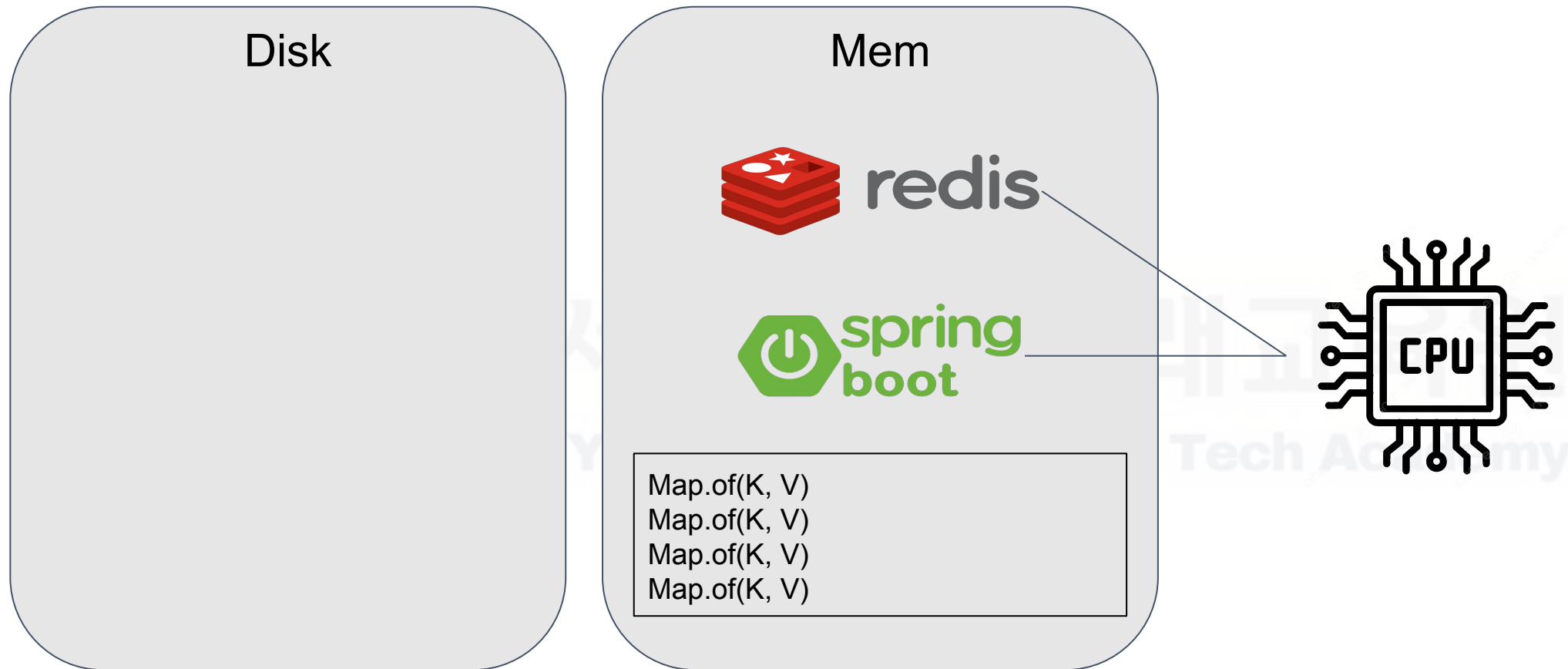
Cache



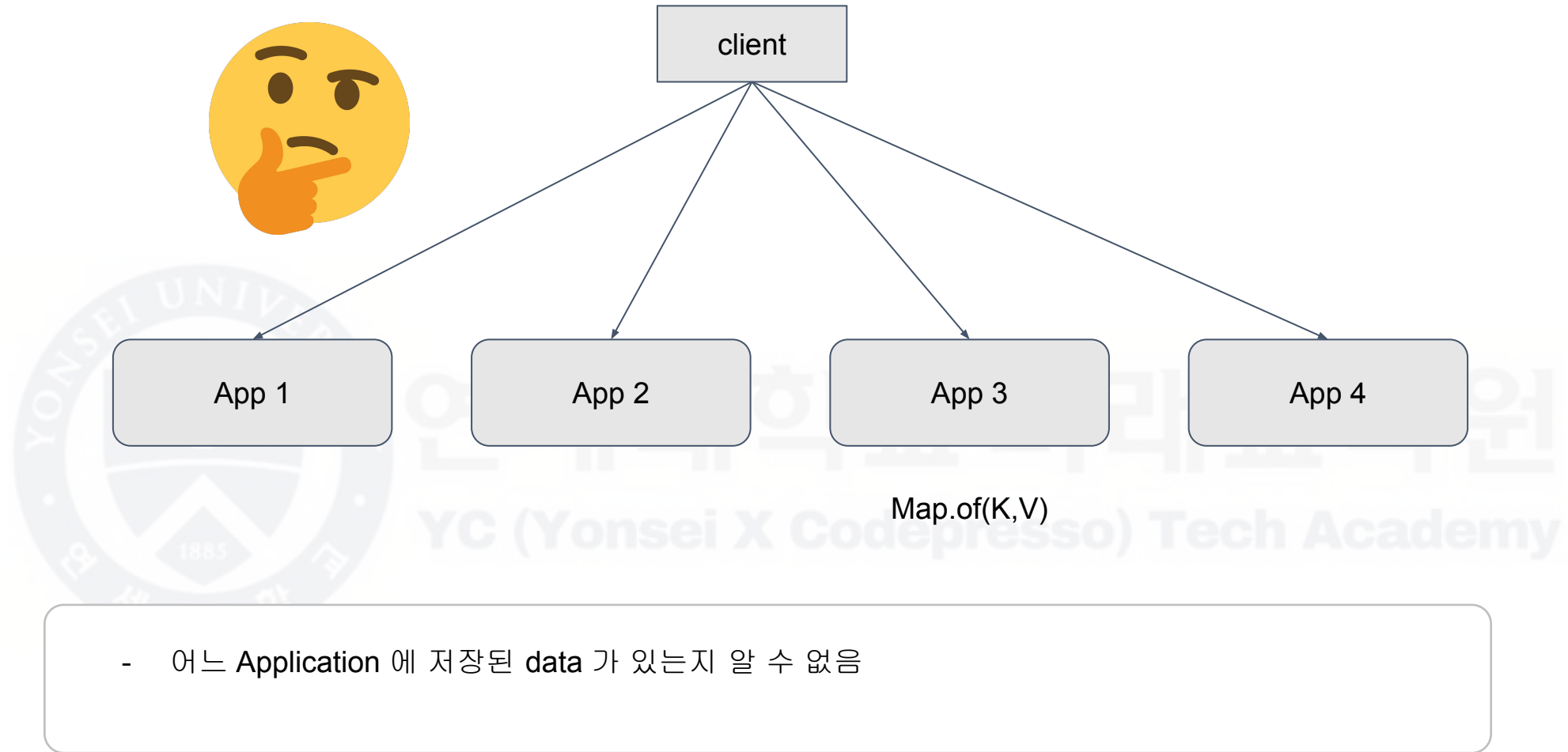
Cache



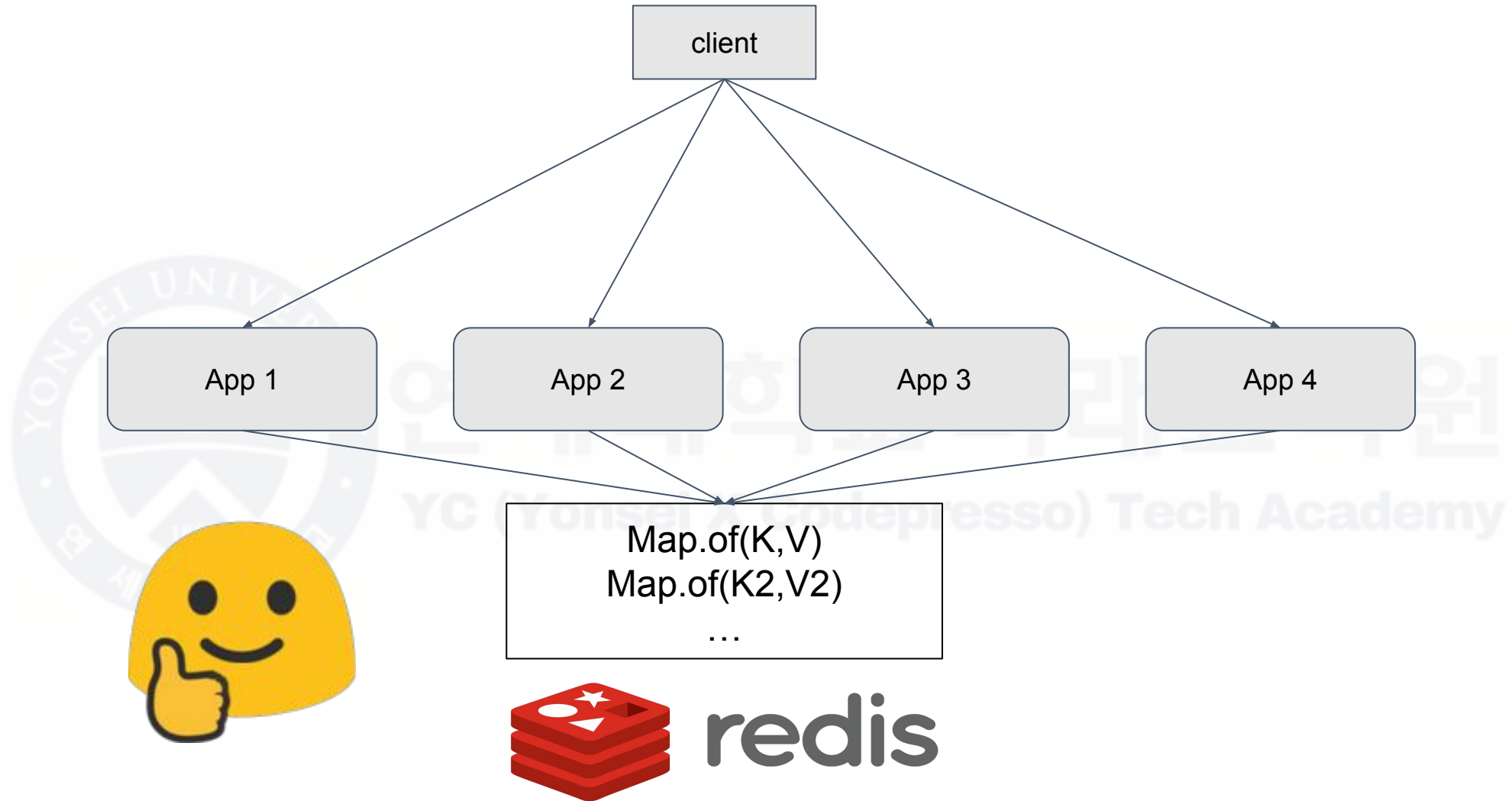
Cache



분산 환경에서의 Cache



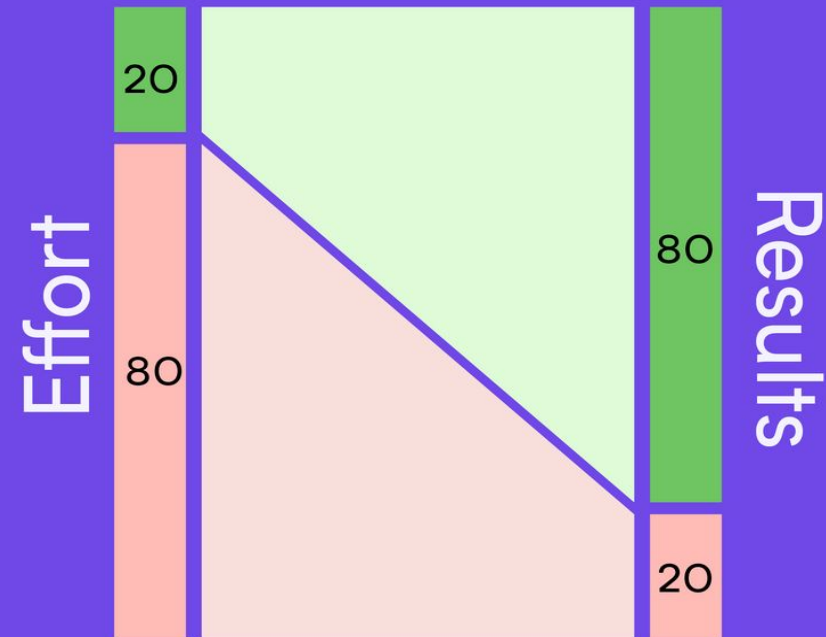
분산 환경에서의 Cache



The 80-20 Rule

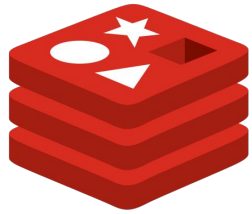
“For many events, roughly 80% of the effects come from 20% of the causes.” – Pareto

Therefore 20% of the effort produces 80% of the results but the last 20% of the results consumes 80% of the effort,



Redis



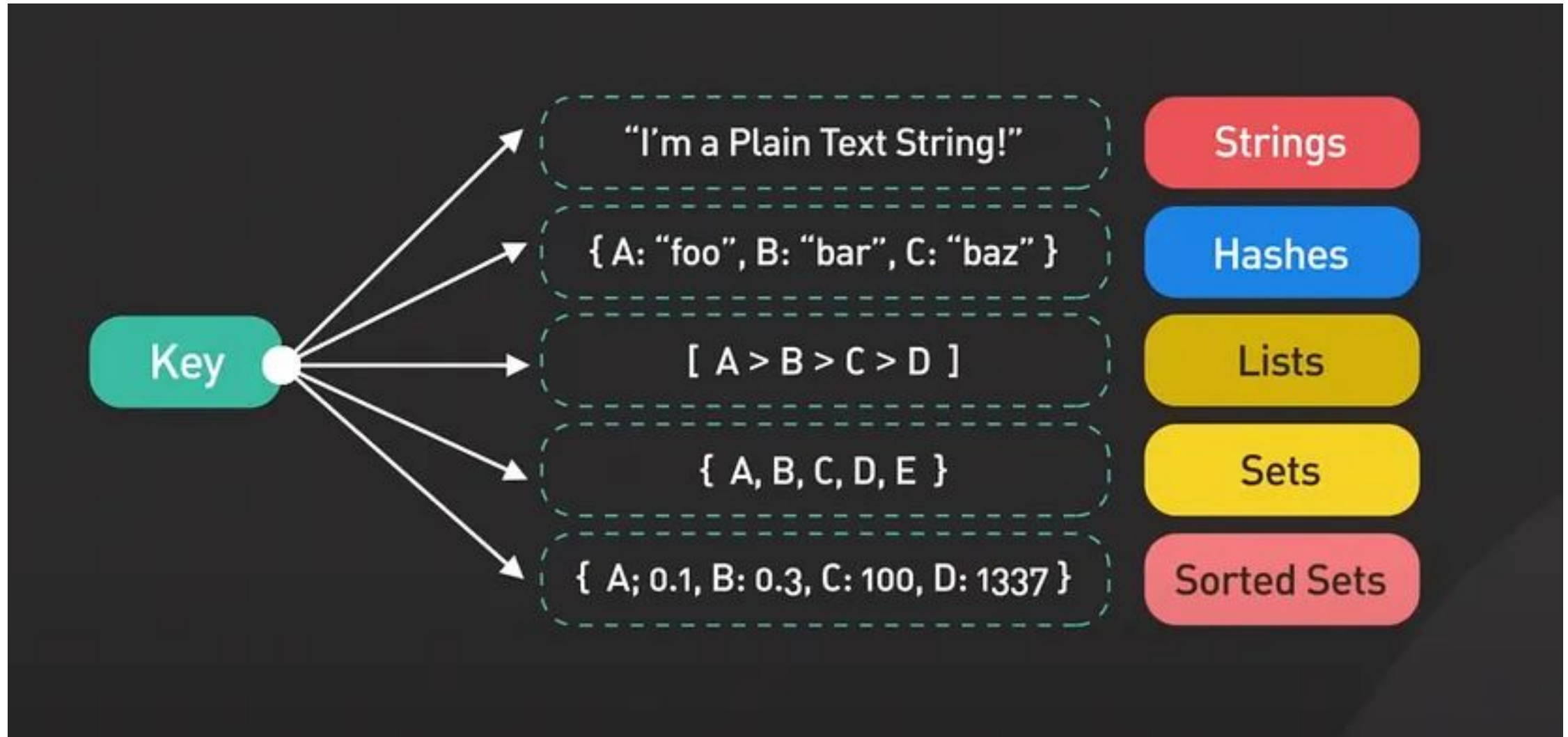


redis

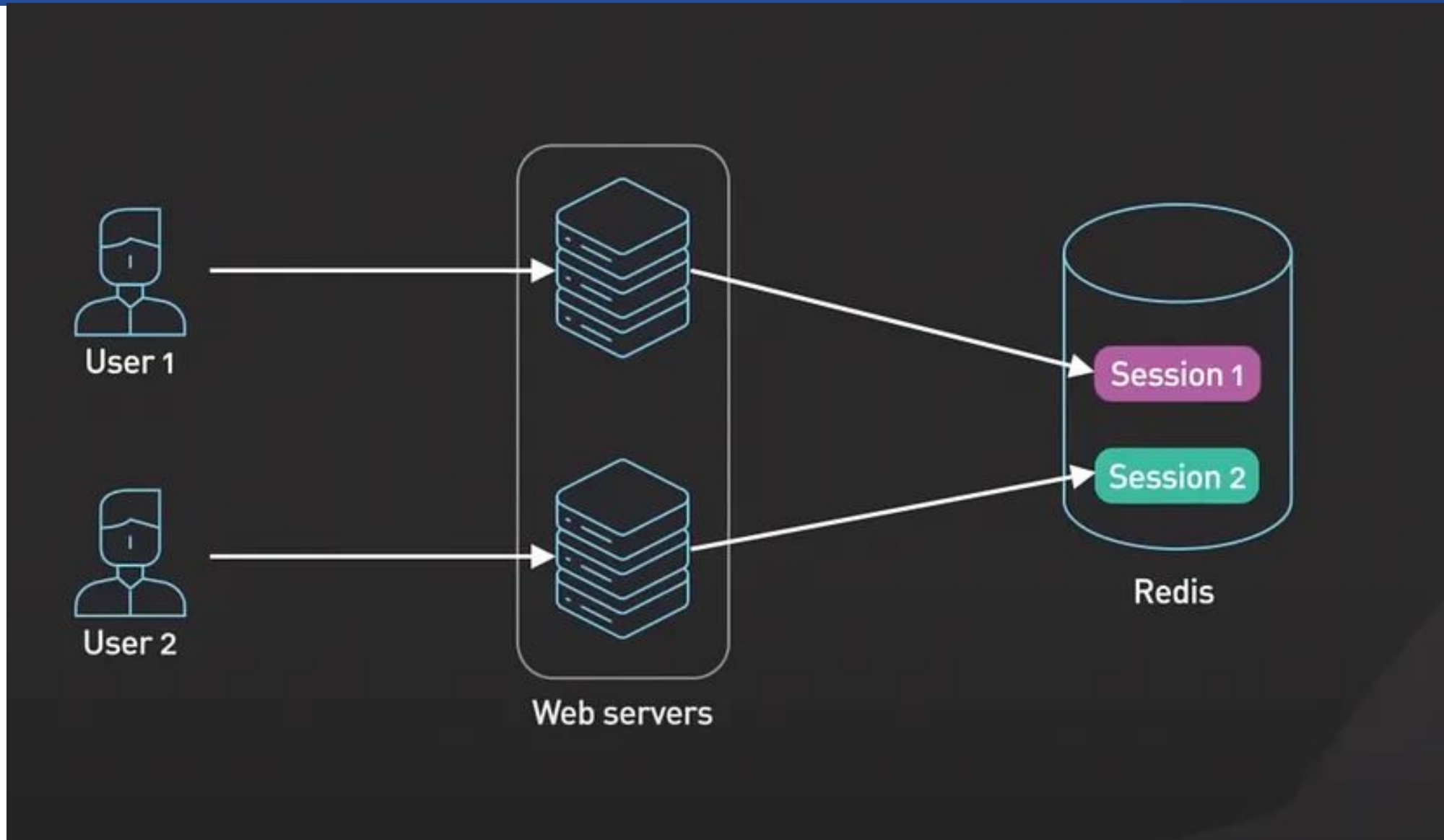
Remote Dictionary Server

1. **Redis**는 데이터베이스, 캐시, 메시지 브로커 및 스트리밍 엔진으로 사용되는 오픈 소스(BSD 라이선스), 인메모리 **Key Value** 데이터 구조 저장소 입니다.
2. **Redis**는 문자열 , 해시 , 목록 , 집합 , 범위 쿼리가 있는 정렬된 집합 , 비트맵 , 하이퍼로그 로그 , 지리 공간 인덱스 및 스트림 과 같은 컬렉션을 제공합니다 .
3. **Redis**에는 복제 , **Lua** 스크립팅 , **LRU** 축출 , 트랜잭션 및 다양한 수준의 디스크 지속성 이 내장되어 있으며 다음을 통해 고가용성을 제공합니다.
4. **Redis Sentinel** 및 **Redis** 클러스터를 통한 자동 파티셔닝을 제공합니다.
5. **TTL** 설정을 통한 휘발성 데이터 관리

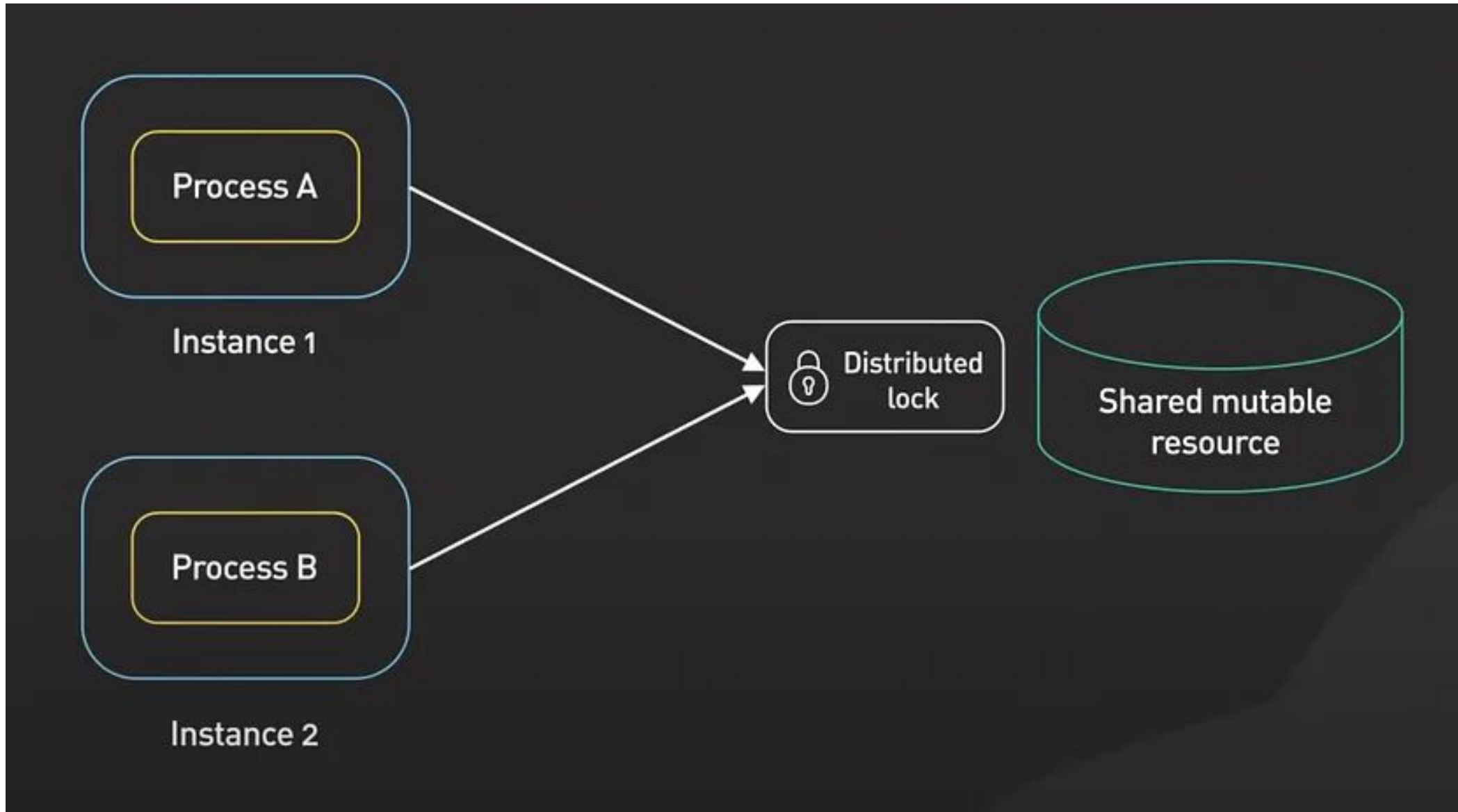
Redis Usecase



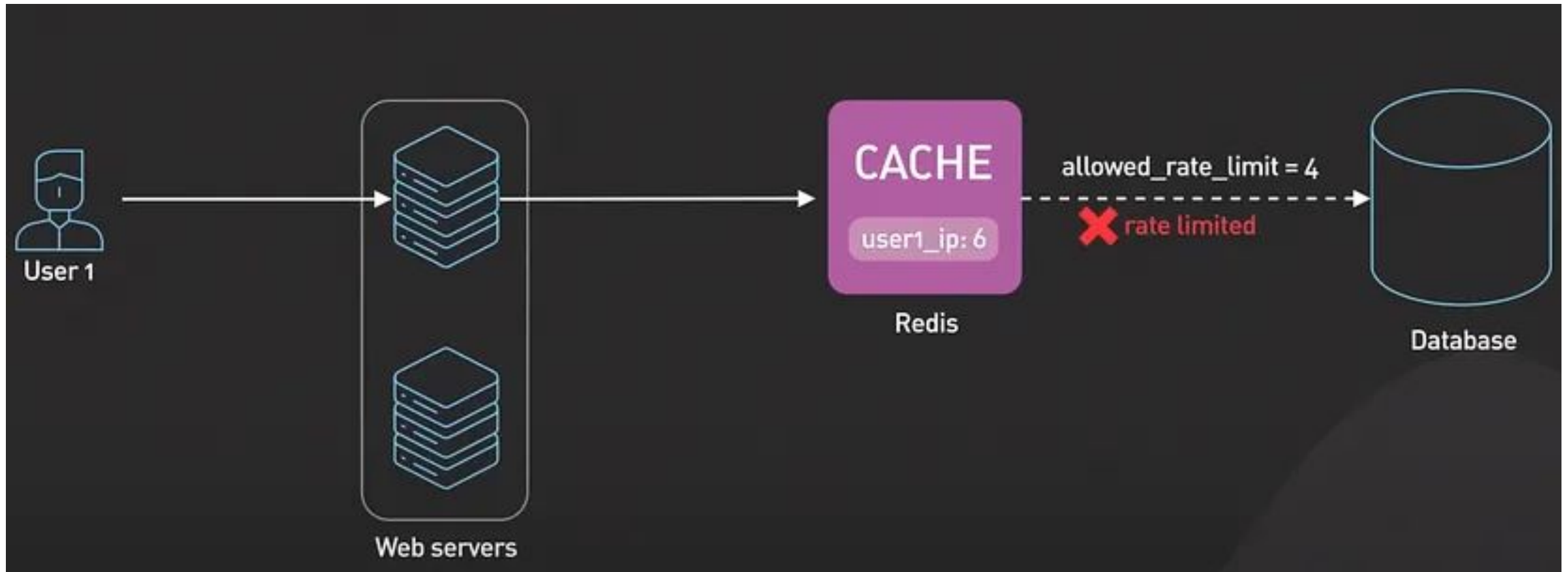
Redis Usecase



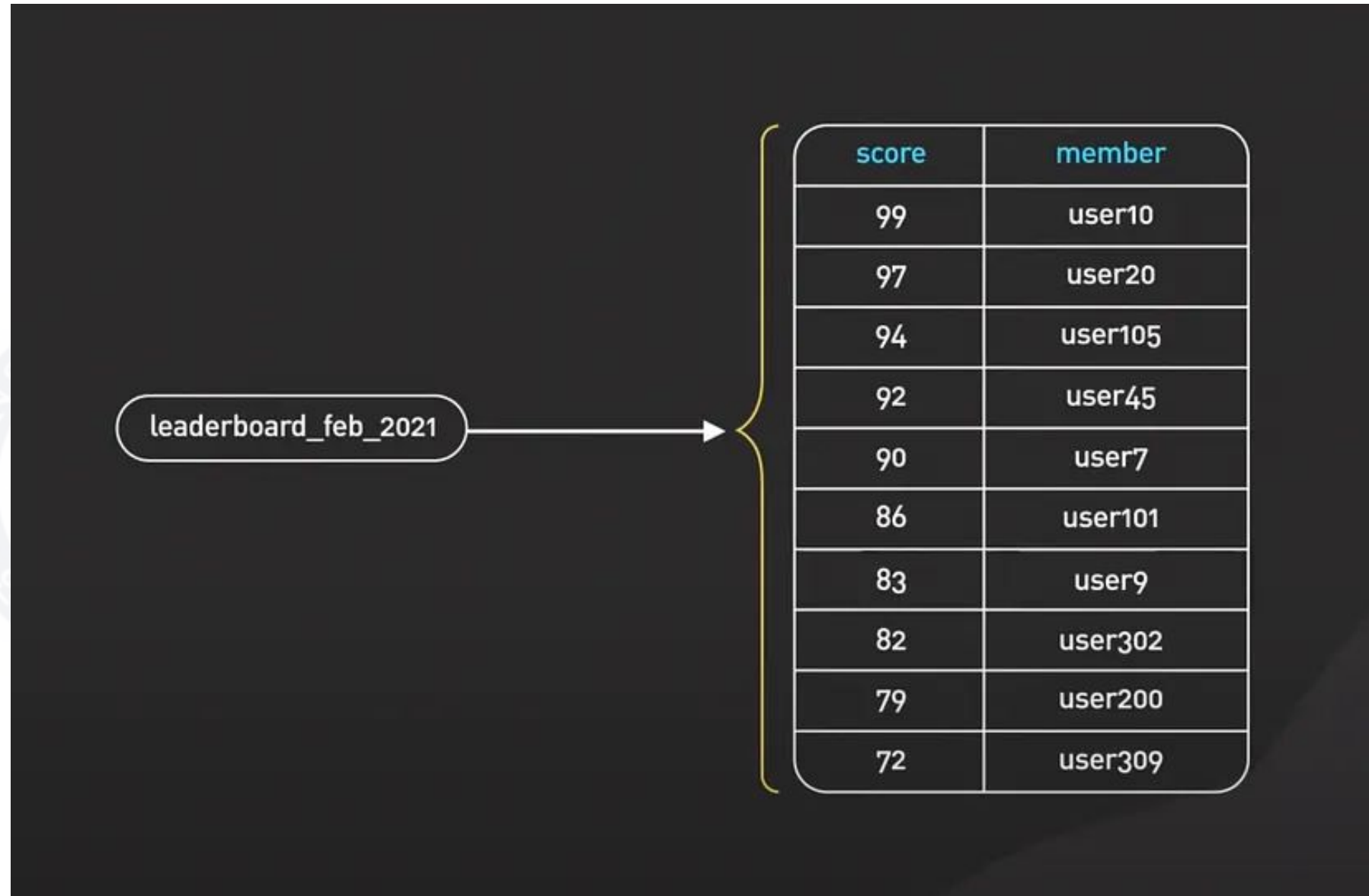
Redis Usecase



Redis Usecase



Redis Usecase



- 매우 긴 키는 좋지 않음. 예컨데 **1MB** 길이의 키는 메모리 관리 측면 뿐만 아니라 키를 조회할 때 고비용의 키 비교 로직을 실행해야 할 수 있음. 키 값이 너무 길다면 차라리 **SHA-1** 등으로 **hashing** 하는 것이 나음
- 매우 짧은 키 또한 좋지 않음. **user:1000:followers** 대신 **u1000flw** 로 사용하는 것은, 메모리상 이득은 작고 가독성은 해치는 키 (**key**가 차지하는 공간은 **value**에 비해 작기 때문에) 비록 작은 메모리상 이득은 있지만, 가독성과 메모리 효율사이의 적당한 균형을 찾아야 함
- 고정된 스키마를 활용. **object-type:id** 형식 ex) **user:1000** 만약 여러 단어를 조합해야 할 일이 있으면, **.**이나 **-** 가 주로 활용. ex) **comment:4321:reply.to** , **comment:4321:reply-to**

- **Persistence option** 은 생각보다 불안정함
- 휘발성이 강한 데이터만 저장
 - 세션, 실행기록, 랭킹 등
- 기존에 **Redis** 에 저장되어있는 객체와 호환성을 보장해야 함

Spring boot cache



@EnableCaching

```
@EnableCaching
@Configuration
public class CacheConfig {
    ...
}
```

- @Cacheable 로 cache 를 처리하기 위해서 추가
- ConcurrentMapCacheManager: Java의 ConcurrentHashMap을 사용해 구현한 캐시를 사용하는 캐시매니저
- SimpleCacheManager: 기본적으로 제공하는 캐시가 없어 사용할 캐시를 직접 등록하여 사용하기 위한 캐시매니저
- CaffeineCacheManager: Java 8로 Guava 캐시를 재작성한 Caffeine 캐시를 사용하는 캐시 매니저
- JCacheCacheManager: JSR-107 기반의 캐시를 사용하는 캐시 매니저

```
@Cacheable("bestSeller")
public Book getBestSeller(String bookNo) {
}
```

- 메소드에 **@Cacheable** 어노테이션을 붙여주면 캐시에 데이터가 없을 경우에는 기존의 로직을 실행한 후에 캐시에 데이터를 추가하고, 캐시에 데이터가 있으면 캐시의 데이터를 반환


```
@Cacheable(value = "bestSeller", key = "#bookNo")  
public Book getBestSeller(String bookNo, User user, Date dateTime) {  
}
```

- 메소드의 파라미터가 없다면 0이라는 디폴트 값을 **Key**로 사용하여 저장
- 만약 메소드의 파라미터가 여러 개라면 파라미터들의 **hashCode** 값을 조합하여 키를 생성
- 여러 개의 파라미터 중에서도 1개의 키 값으로 지정하고 싶은 경우 **Key** 값을 별도로 지정해주면 된다.

```
@Cacheable(value = "bestSeller", key = "#book.bookNo", condition = "#user.type == 'ADMIN'")  
public Book getBestSeller(Book book, User user, Date dateTime) {  
}
```

- 파라미터 값이 특정 조건인 경우에만 캐시를 적용하기를 원한다면 **condition**을 이용

The background is a solid blue color with several abstract geometric elements. In the top-left and bottom-right corners, there are circular halftone patterns of varying densities. Diagonal lines in a lighter shade of blue cross these patterns. On the right side, there are two thin, light-blue circles. The overall design is modern and minimalist.

EOD