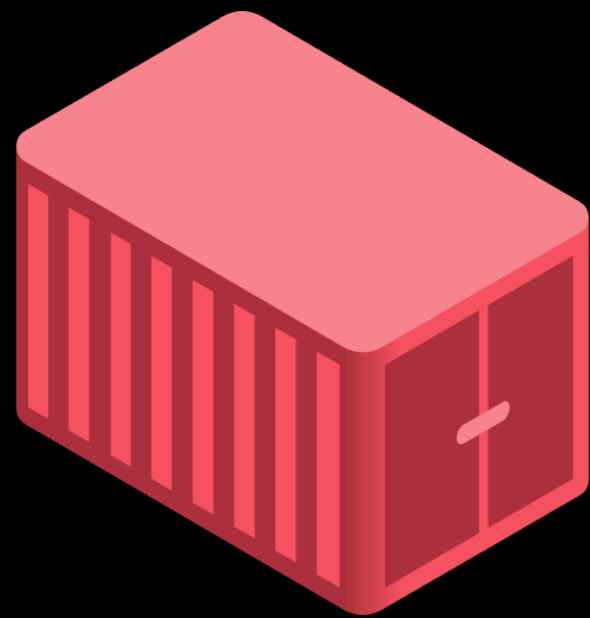


# Bean scope

부제: 사실 거의 Singleton 씁니다

# Bean scope



**IoC Container**

**Control**



**Bean**



**Bean**



**Bean**

...

# Bean scope

우리가 배운 Bean은 Singleton으로 관리된다고 했다 !

→ 무조건?

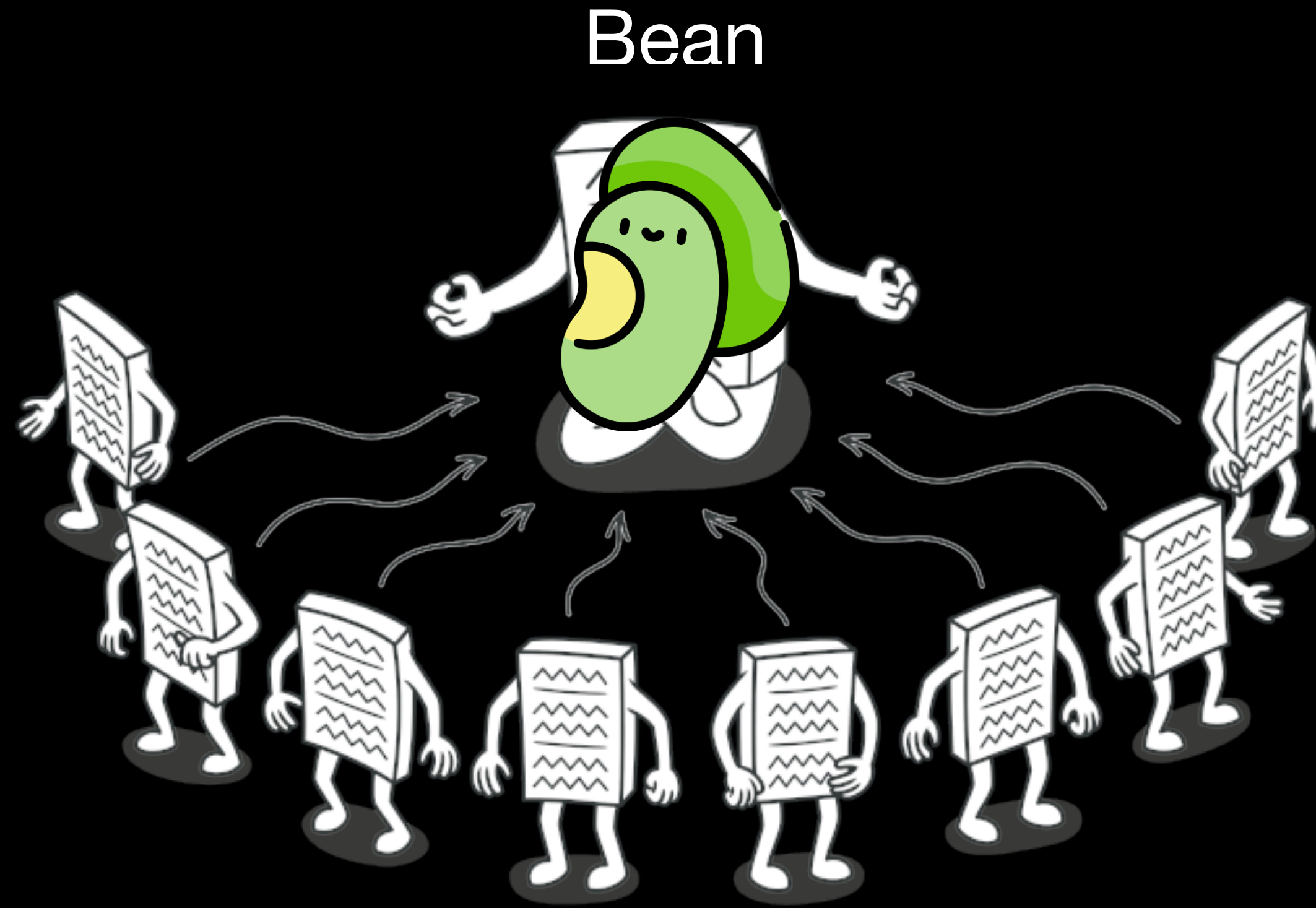
아니!

Spring은 너의 미래까지 생각한다 🙄

다양한 Bean scope을 지원 ~ !

# Bean scope

## Singleton scope

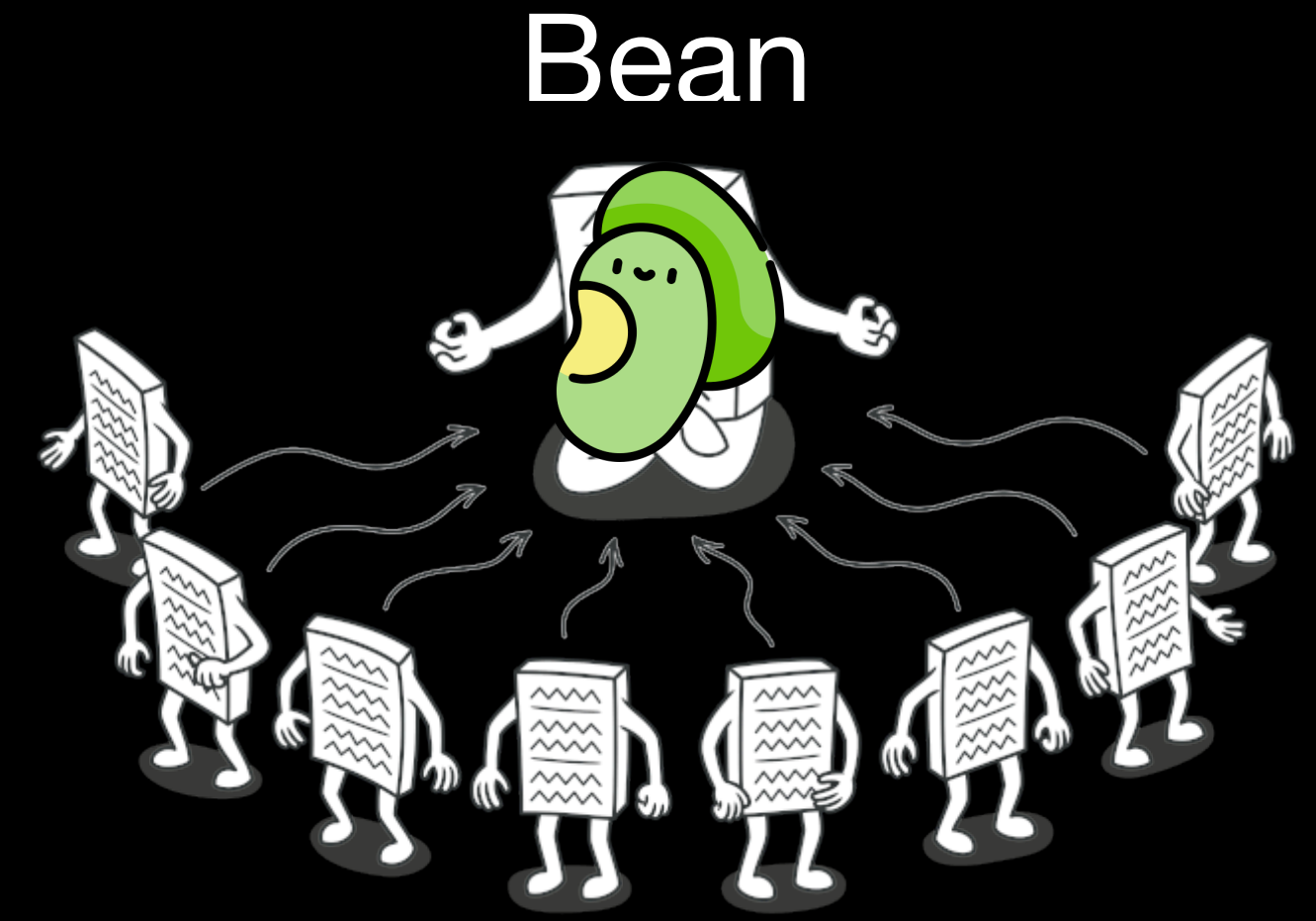


### Spring의 기본 Bean scope

- Spring container의 시작부터 끝까지 유지되는 가장 넓은 범위의 Scope
- Bean을 사용하는 모든 곳에서 같은 객체를 사용 (공유)

# Bean scope

## Singleton scope



## 그럼 언제 Singleton scope을 사용해야하는가?

### 상태가 없는 공유 객체

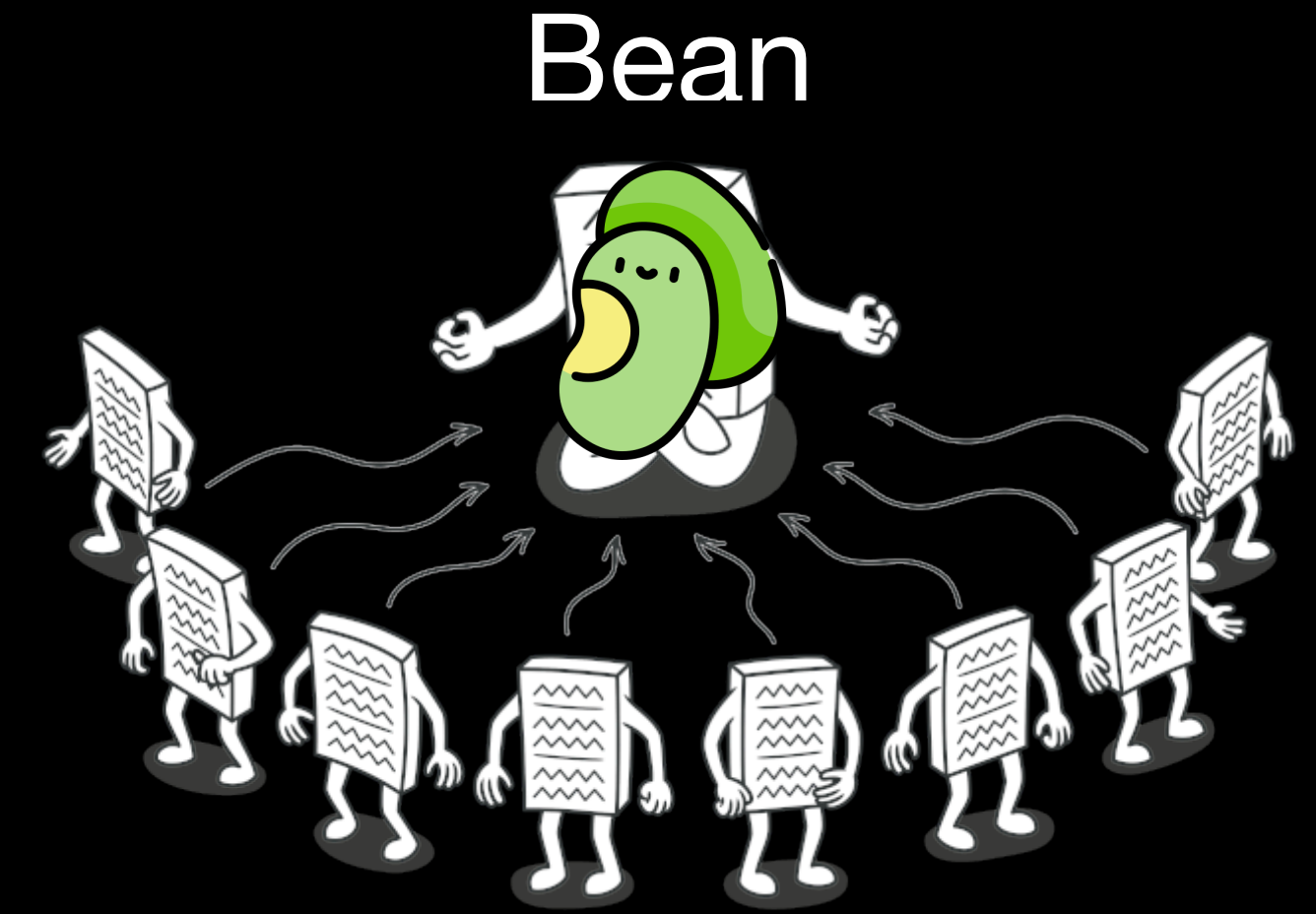
- 상태를 가지고 있지 않은 객체는 동기화 비용이 없음. 따라서 매번 이 객체를 참조하는 곳에서 새로운 객체를 생성할 이유가 없음.

### 읽기용으로만 상태를 가진 공유 객체

- 1번과 유사하게 상태를 가지고 있으나 읽기 전용이므로 여전히 동기화 비용이 들지 않음. 매 요청마다 새로운 객체 생성할 필요가 없음.

# Bean scope

## Singleton scope



## 그럼 언제 Singleton scope을 사용해야하는가?

공유가 필요한 상태를 지닌 공유 객체 [조심해서 사용해야 함]

- 객체 간의 반드시 공유해야 할 상태를 지닌 객체가 하나 있다면, 이 경우에는 해당 상태의 쓰기를 가능한 동기화 할 경우 싱글톤도 적합함.
- 하지만 쓰기가 가능하므로 잘못된 정보가 쓰여지지 않는지 유의해야 함.

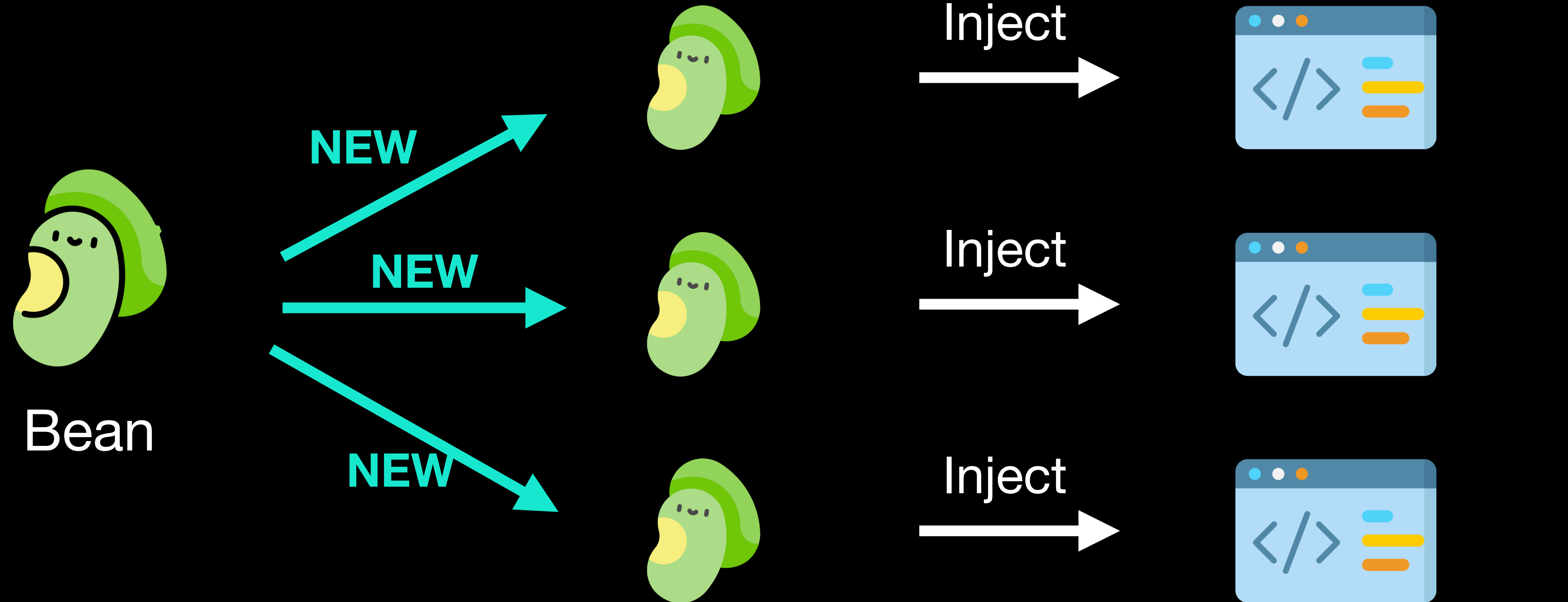
쓰기가 가능한 상태를 지니면서도 사용빈도가 매우 높은 객체

- 애플리케이션 안에서 정말로 사용빈도가 높다면, 쓰기 접근에 대한 동기화 비용을 감안하고서라도 싱글톤을 고려할만하다.
- 장시간에 걸쳐 매우 많은 객체가 생성되는 경우, 해당 객체가 매우 작은 양의 쓰기상태를 가지고 있는 경우, 객체 생성비용이 매우 큰 경우



# Bean scope

Prototype scope - @Scope("prototype")



새로운 인스턴스를 생성해 반환하는 Bean scope

- Spring container는 Prototype bean의 생성과 의존관계 주입, 초기화까지만 관여
- 결국, Bean을 주입 받으면 다 다른 객체를 주입받게 되는 것

# Bean scope

Prototype scope - @Scope("prototype")

그럼 언제 **Prototype scope**을 사용해야하는가?

쓰기가 가능한 상태를 지닌 객체

쓰기가 가능한 상태가 많아서 동기화 비용이 객체 생성 비용보다 크다면 싱글톤으로 적합하지 않는 경우, Prototype bean 사용

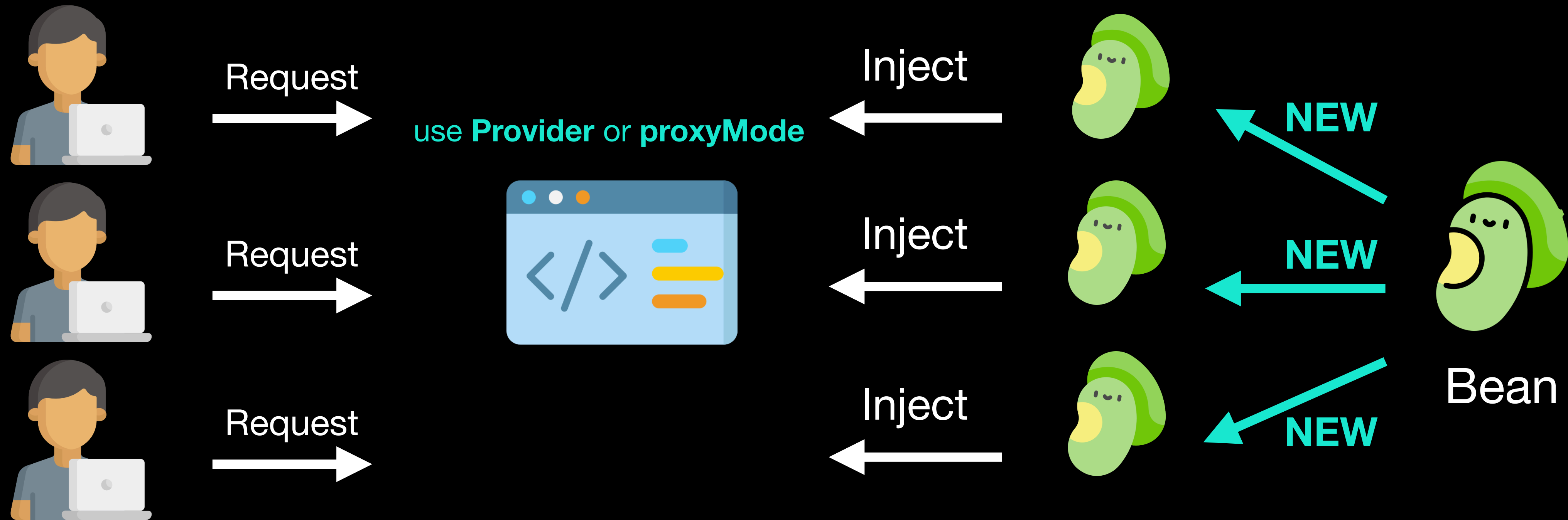
새로운 **Bean**을 주입받아야 하는 경우

여러 사유로 인해 Bean의 새로운 객체를 주입받아야 하는 경우 사용



# Bean scope

## Web scope - @Scope("request")



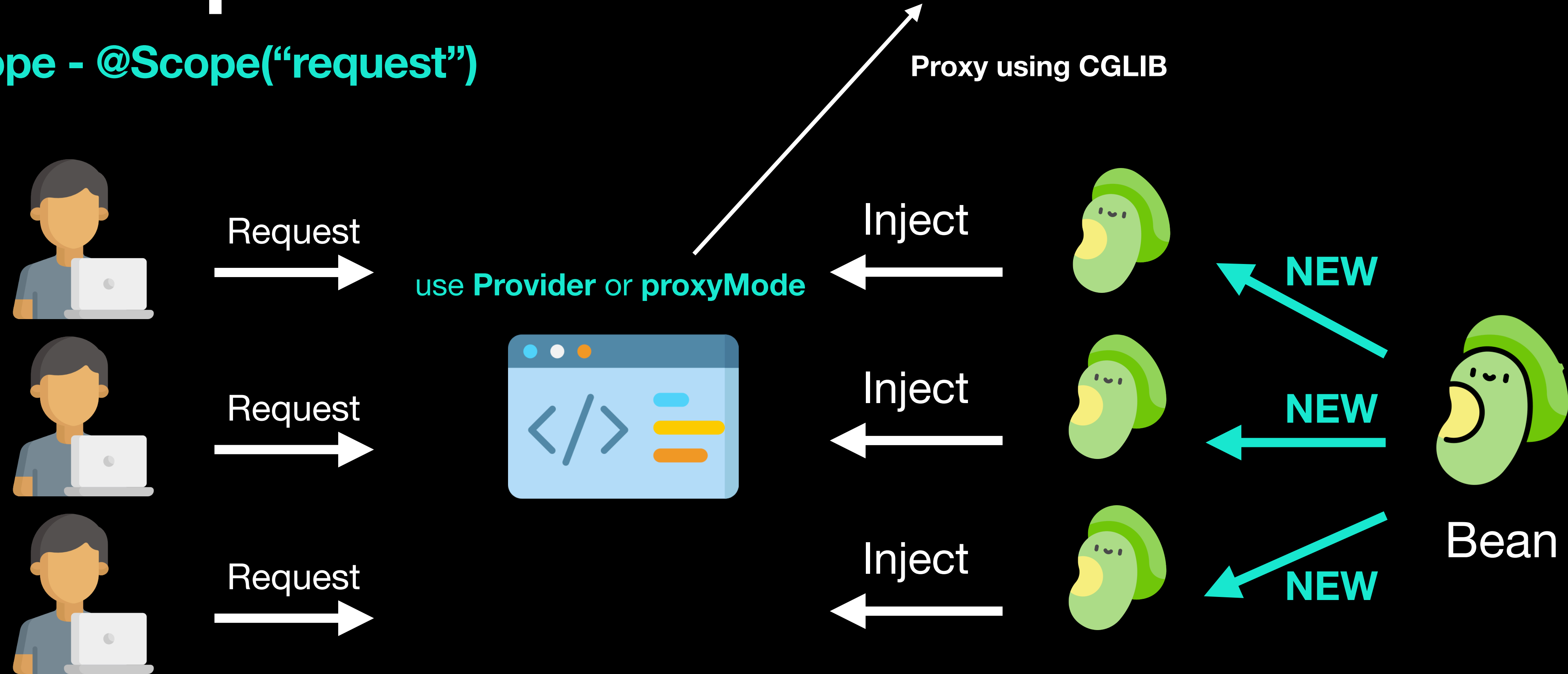
웹 HTTP 요청이 들어오고 나갈 때까지 유지되는 Scope

- 즉, 각각의 HTTP 요청마다 별도의 인스턴스가 생성됨
- Request가 들어와야만 Bean이 만들어지기 때문에 특정 설정이 필요함

# Bean scope

## Web scope - @Scope("request")

`@Scope(value="request", proxyMode=ScopedProxyMode.TARGET_CLASS)`



웹 HTTP 요청이 들어오고 나갈 때까지 유지되는 Scope

- 즉, 각각의 HTTP 요청마다 별도의 인스턴스가 생성됨
- Request가 들어와야만 Bean이 만들어지기 때문에 **특정 설정**이 필요함

# Bean scope

## Web scope

Web Scope는 request 이외에도 `session`, `application`, `websocket` scope이 존재함

**session**        Scopes a single bean definition to the lifecycle of an HTTP Session.

**application**    Scopes a single bean definition to the lifecycle of a ServletContext.

**websocket**      Scopes a single bean definition to the lifecycle of a WebSocket.