

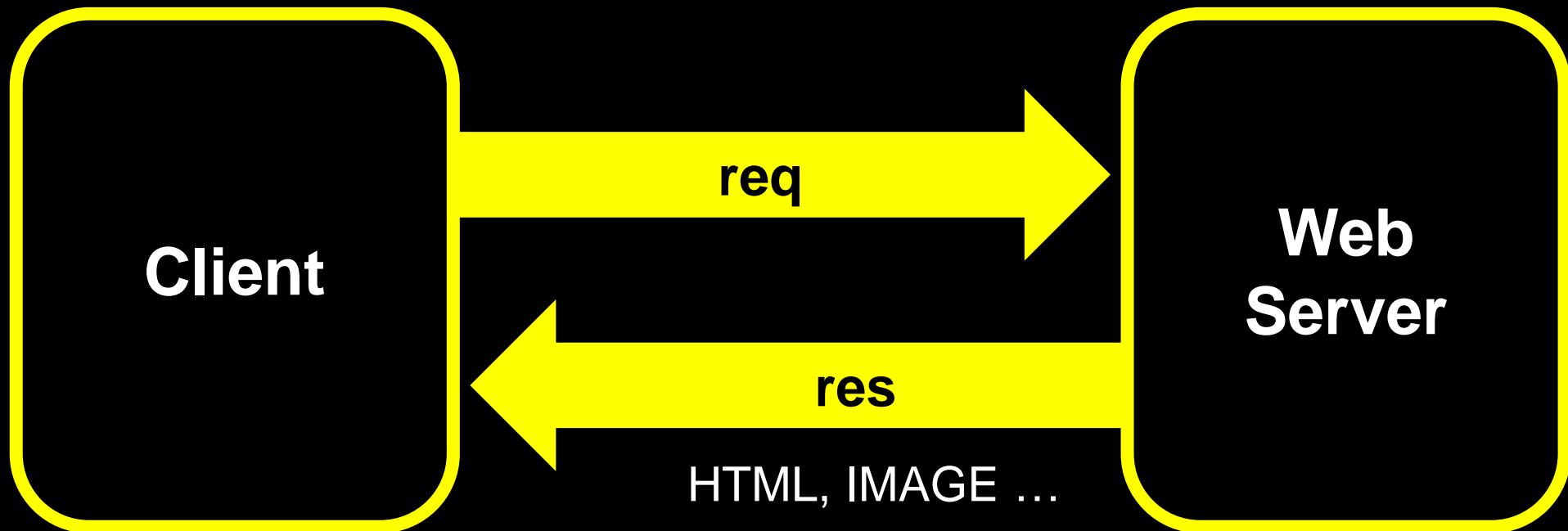
Servlet & Spring

Servlet

HTML등의 웹 콘텐츠를 생성하고 전달하기 위해
Servlet 클래스의 구현 규칙을 지켜
자바로 만들어진 **프로그램**

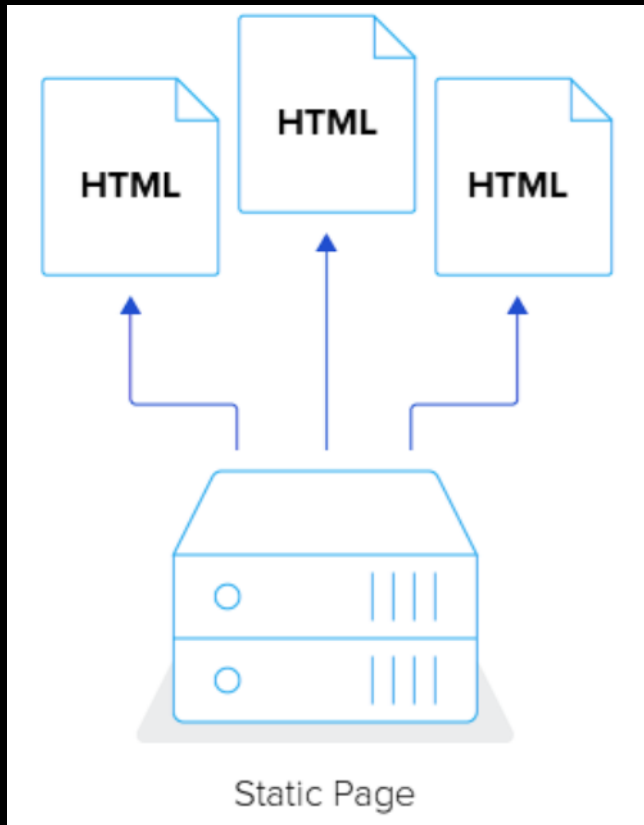
Servlet은 왜 생겼을까?

정적 데이터만 전달하는 Web Server



Servlet은 왜 생겼을까?

정적 데이터만 전달하는 Web Server

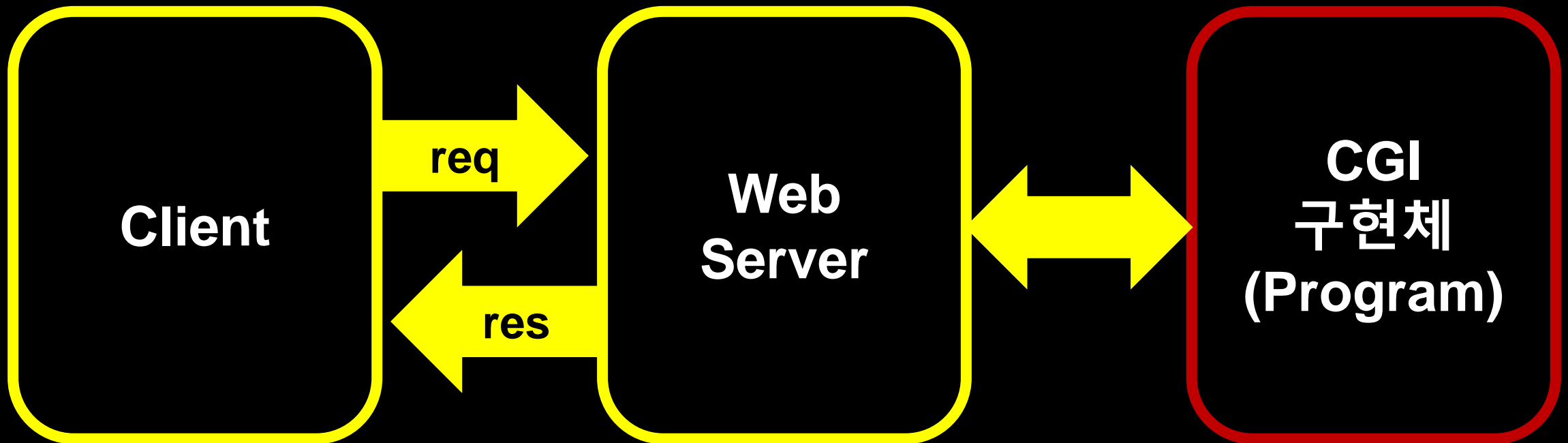


Ex) 홈페이지의 회사소개

- ⇒ 저장된 정보만 보여주기에 서비스가 한정적
- ⇒ 사용자의 요청에 따라
다른 처리를 해줄 수 있는 웹 페이지가 필요해짐

Servlet은 왜 생겼을까?

동적 데이터를 처리하는 CGI (Common Gateway Interface)



Servlet은 왜 생겼을까?

동적 데이터를 처리하는 CGI (Common Gateway Interface)

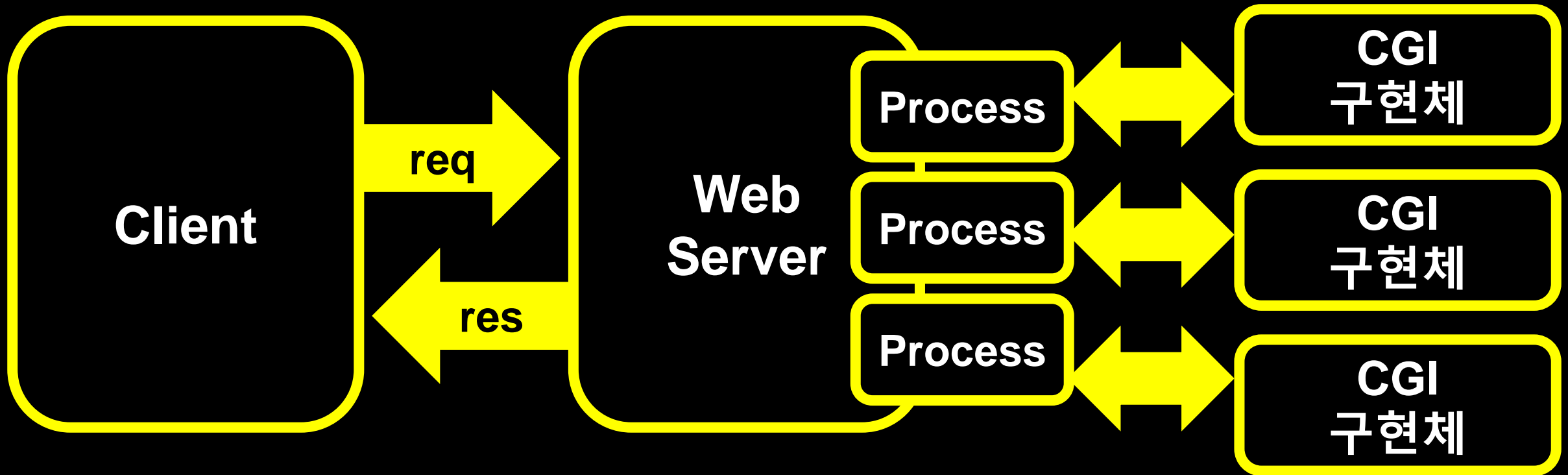


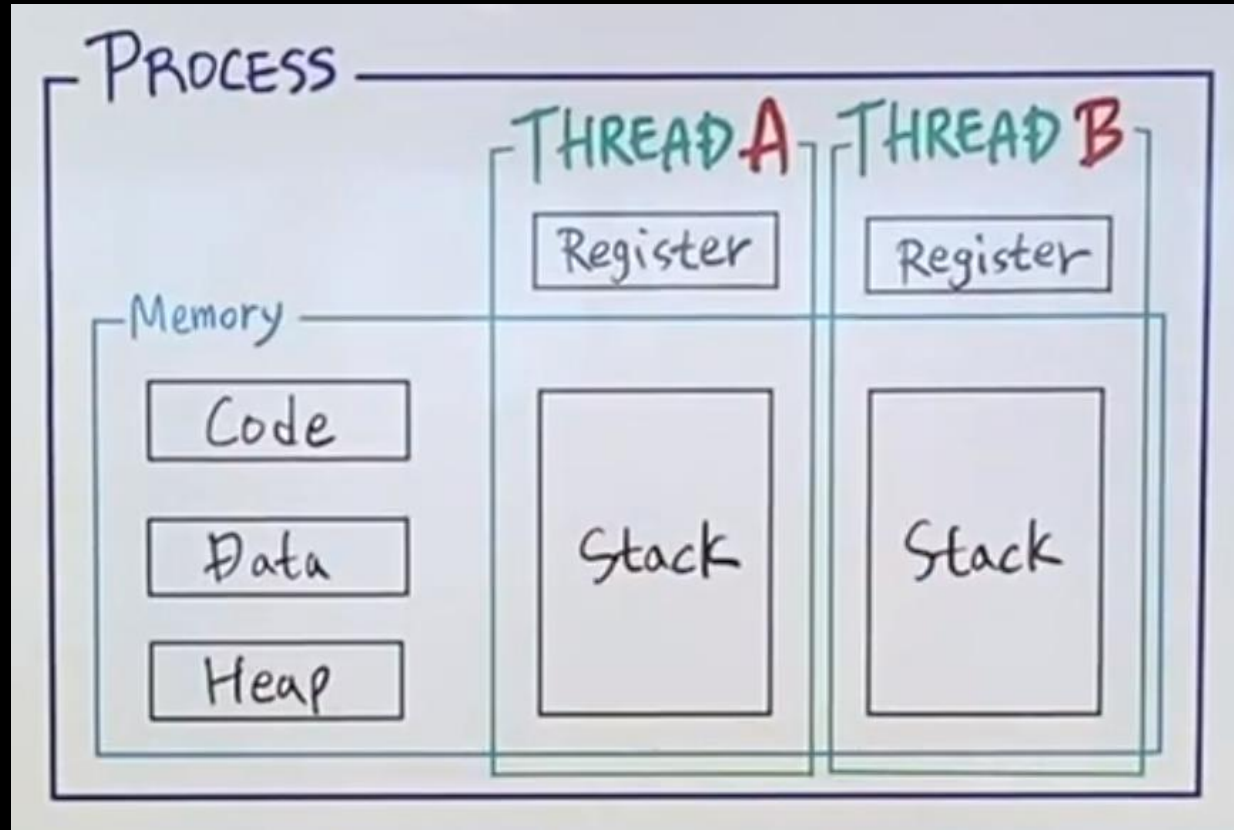
CGI

http 통신규약을 사용하는 웹서버가
프로그램(웹 어플리케이션)과
데이터를 주고 받는 처리 **규약**

Servlet은 왜 생겼을까?

동적 데이터를 처리하는 CGI (Common Gateway Interface)





Process

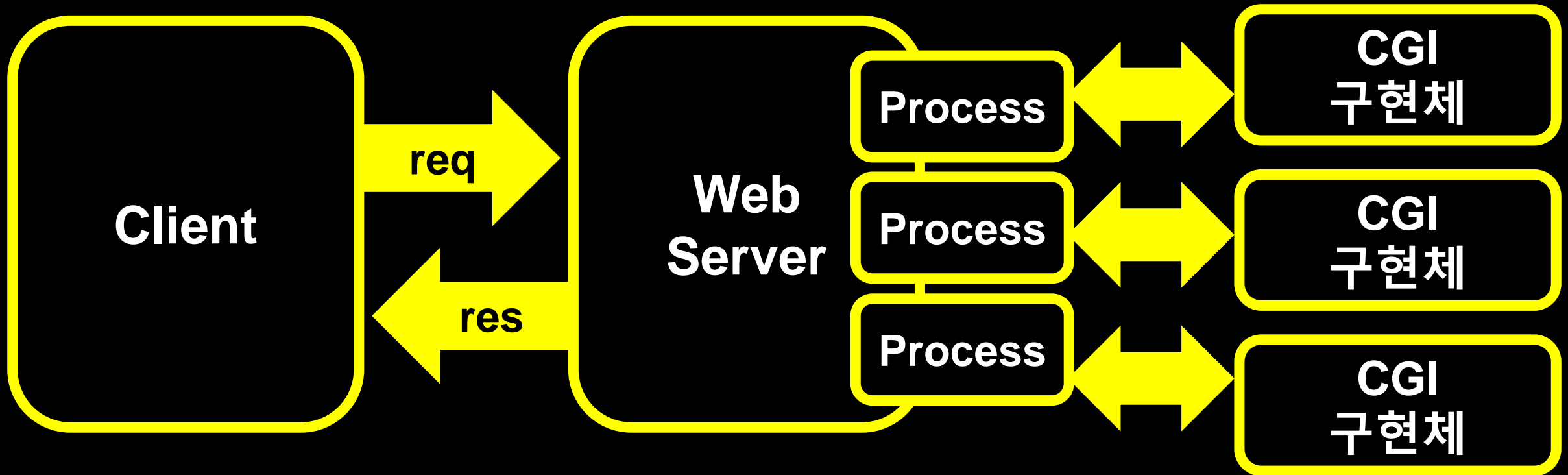
실행중인 프로그램 인스턴스
(메모리에 적재된 프로그램)

Thread

한 프로세스 내의 동작 흐름

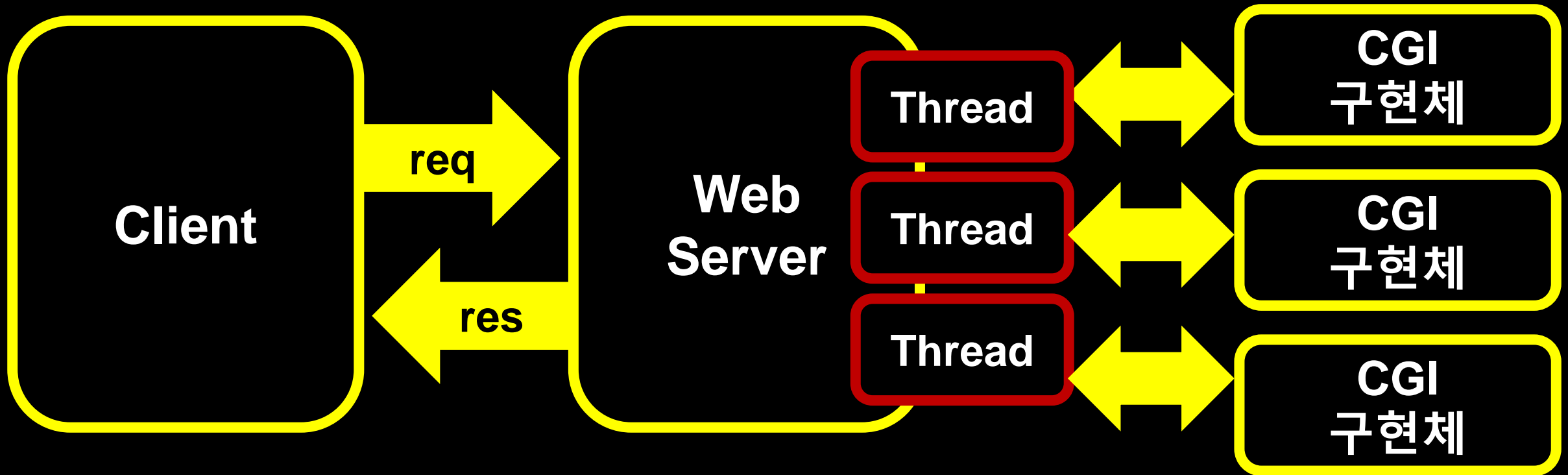
Servlet은 왜 생겼을까?

동적 데이터를 처리하는 CGI (Common Gateway Interface)



Servlet은 왜 생겼을까?

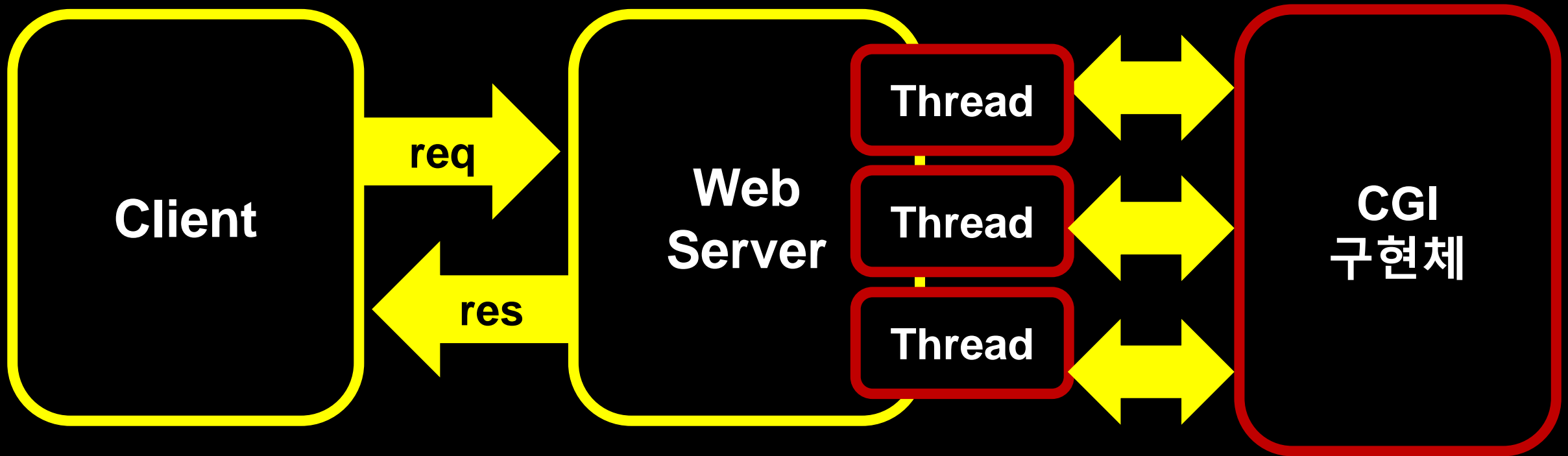
동적 데이터를 처리하는 CGI (Common Gateway Interface)



개선점 1. Process에서 Thread로

Servlet은 왜 생겼을까?

동적 데이터를 처리하는 CGI (Common Gateway Interface)

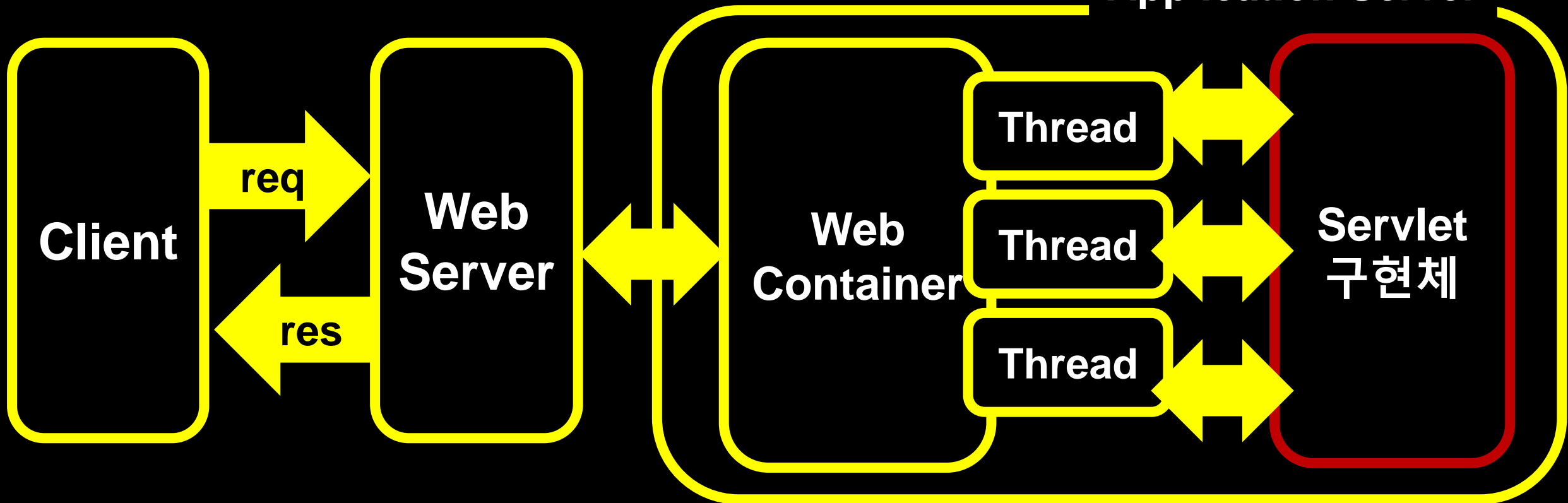


개선점 2. 여러 Instance에서 Singleton으로

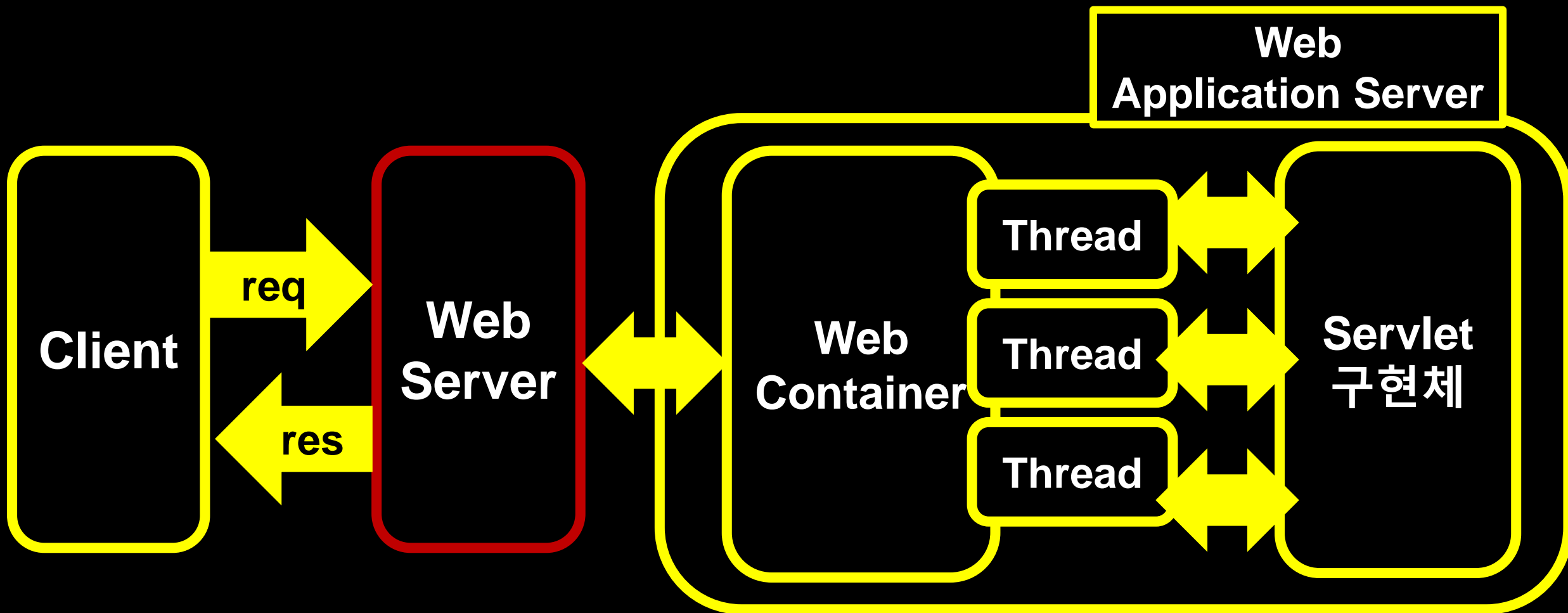
Servlet은 왜 생겼을까?

CGI를 보완한 Servlet

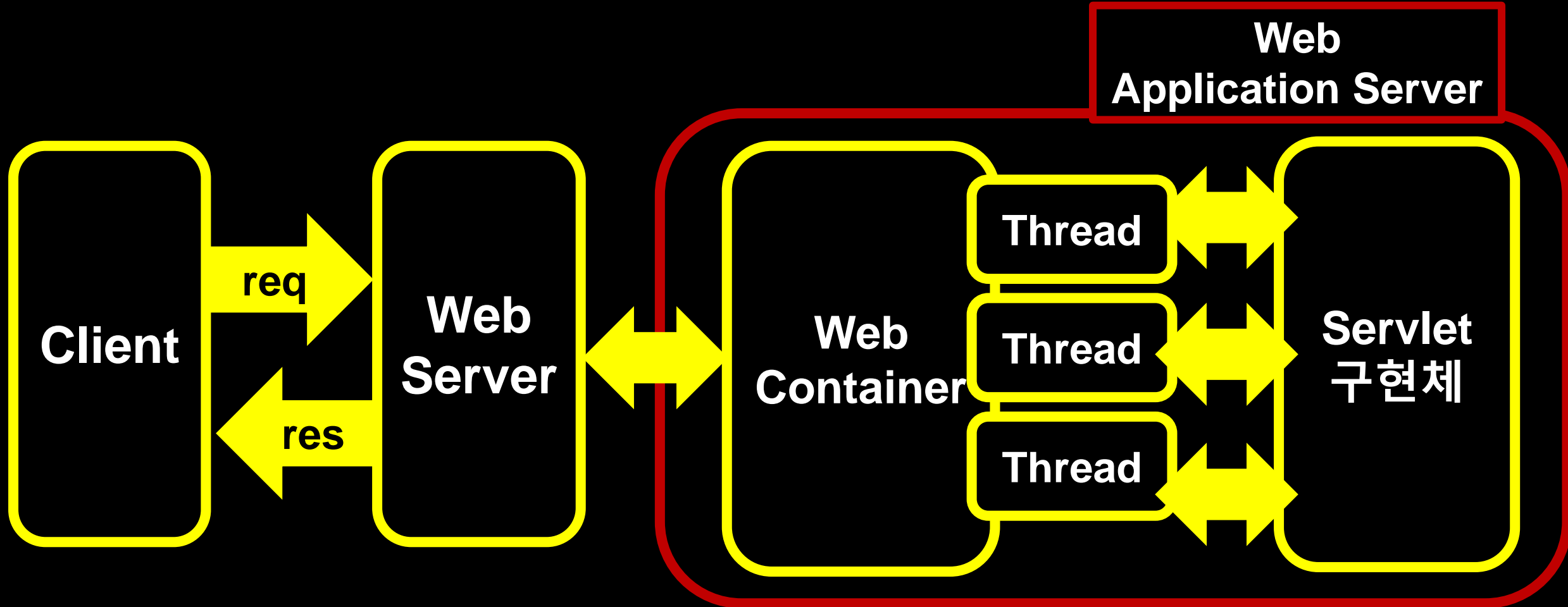
Web
Application Server



Web Server

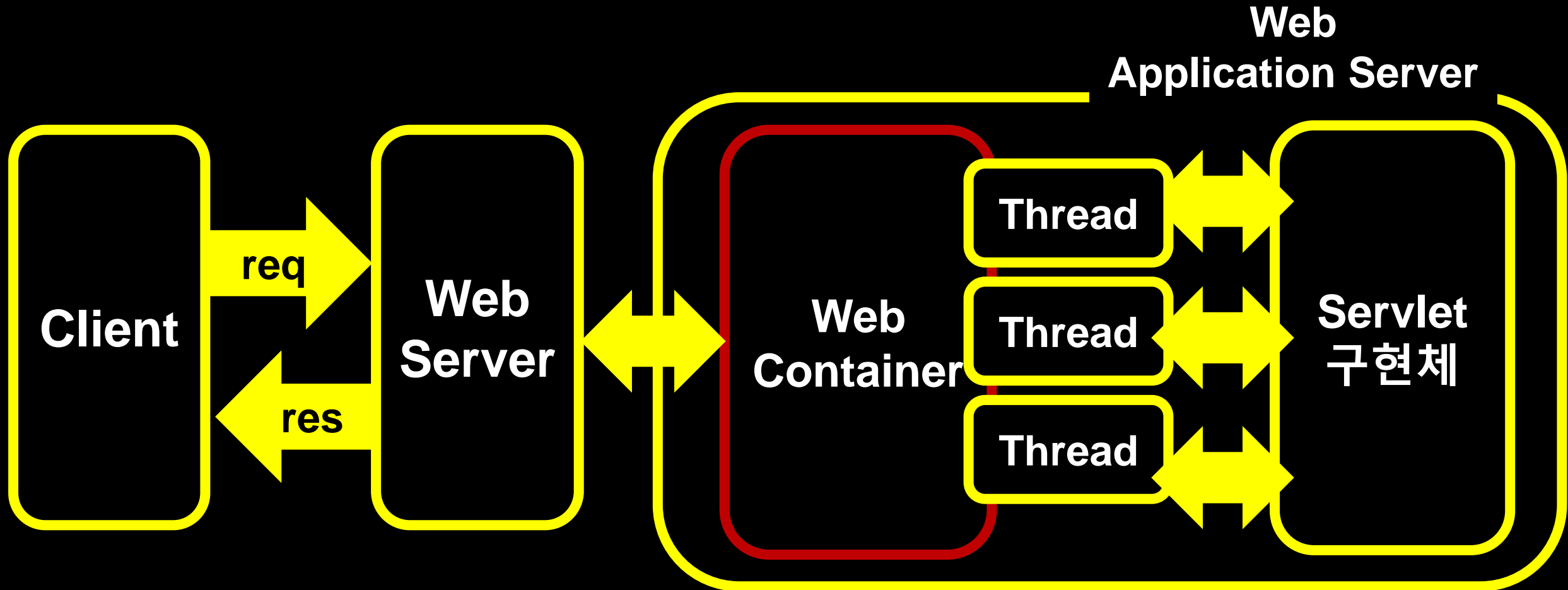


Web Application Server



Web Container :

요청이 들어오면 Thread를 생성하고, Servlet을 실행시킨다.

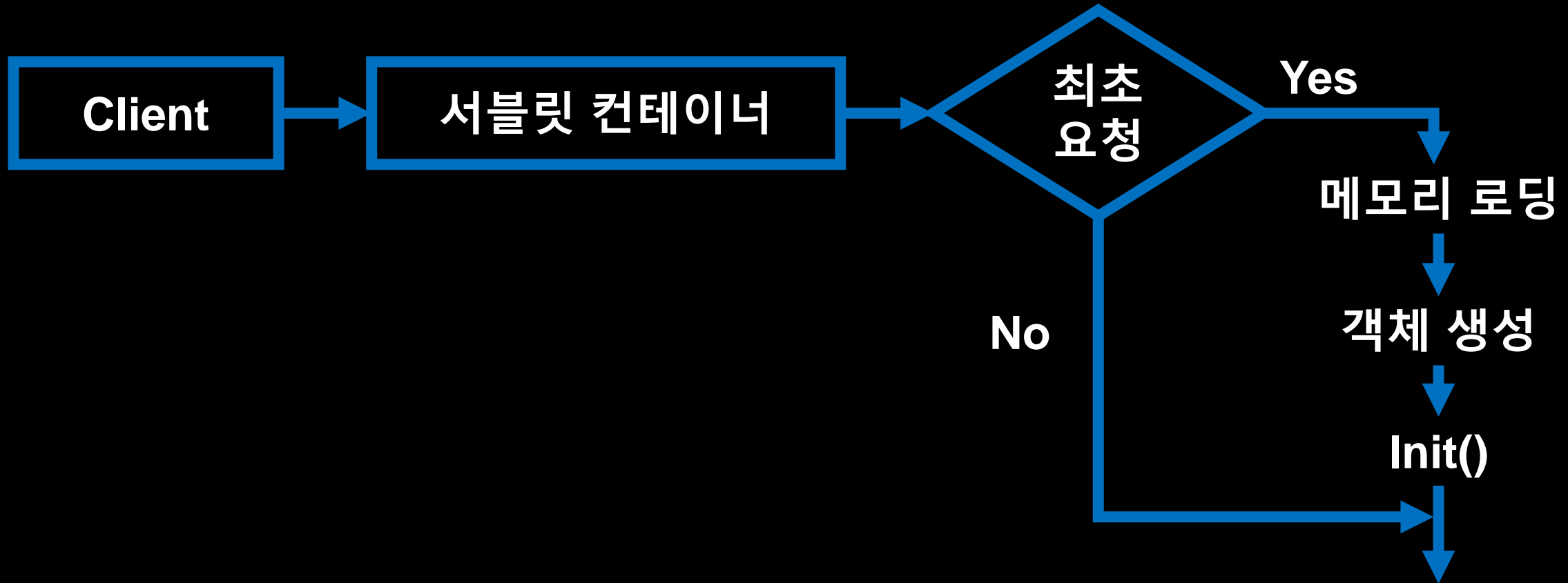


Servlet의 생명주기

1. Servlet은 생성될 수 있으며 **init()**을 호출한다.
2. Client로부터 발생한 모든 요청은 **service()**를 통해 다룬다.
3. Servlet이 서비스를 마치게 되면, **destroy()**를 통해 종료되며 GC에 의해 메모리 정리가 된다.

각 메서드는 Servlet Container(Tomcat)가 호출해준다.

Servlet의 생명주기



이미 생성된 Servlet 객체는 메모리에 남겨두어
동일한 Servlet에 대해서는 다시 init()을 실행하지 않고 재사용함

Spring에서

Servlet을 어떻게 사용할까?