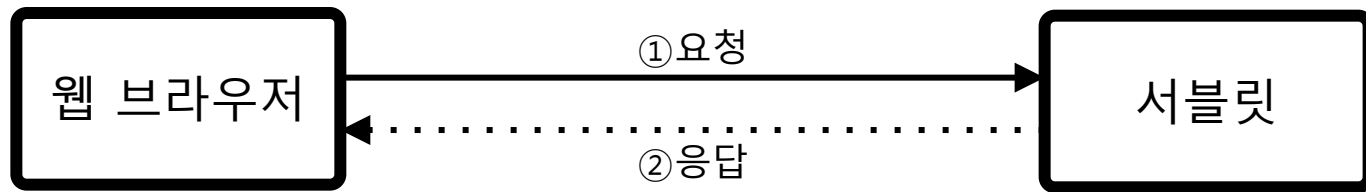
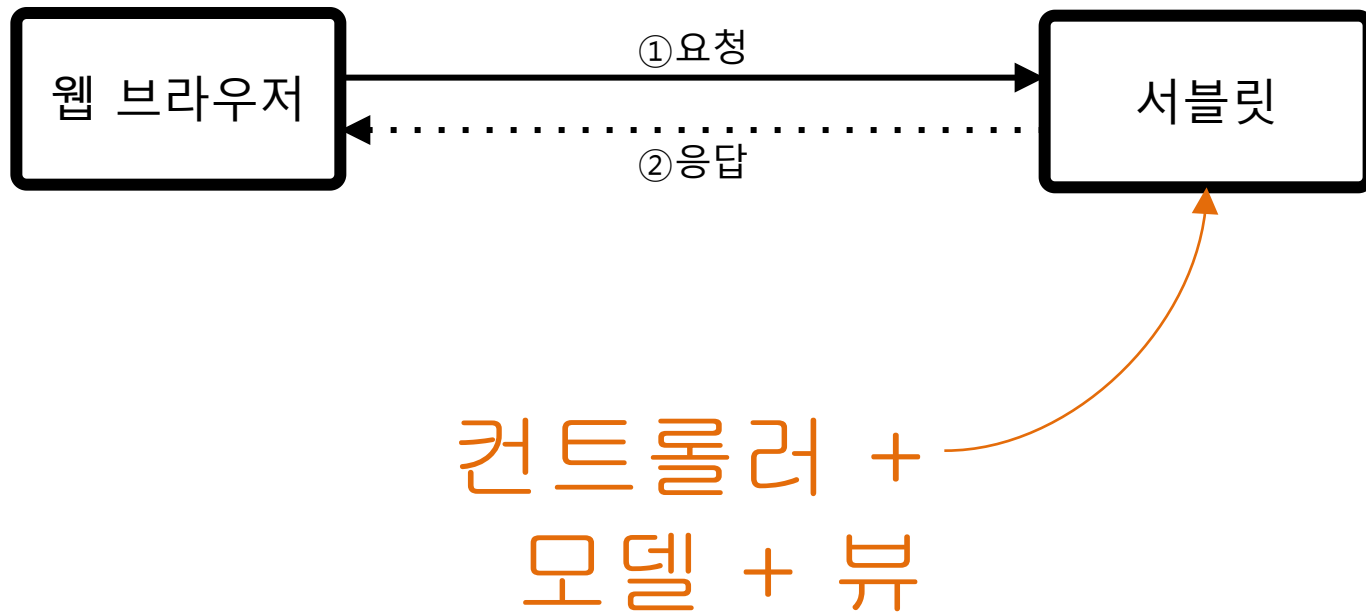


5.4 서블릿에서 뷰 분리하기

5.4 서블릿에서 뷰 분리하기



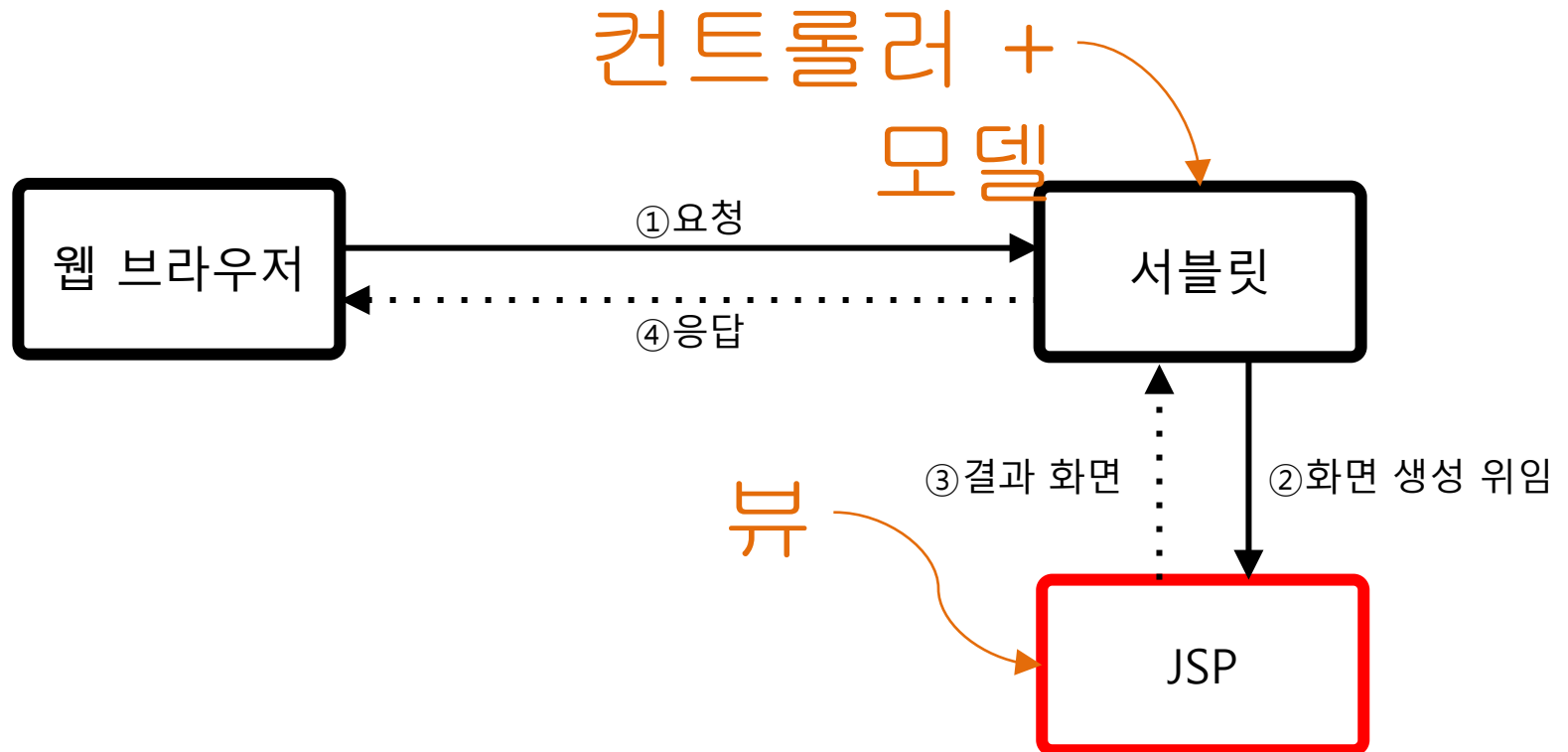
5.4 서블릿에서 뷰 분리하기



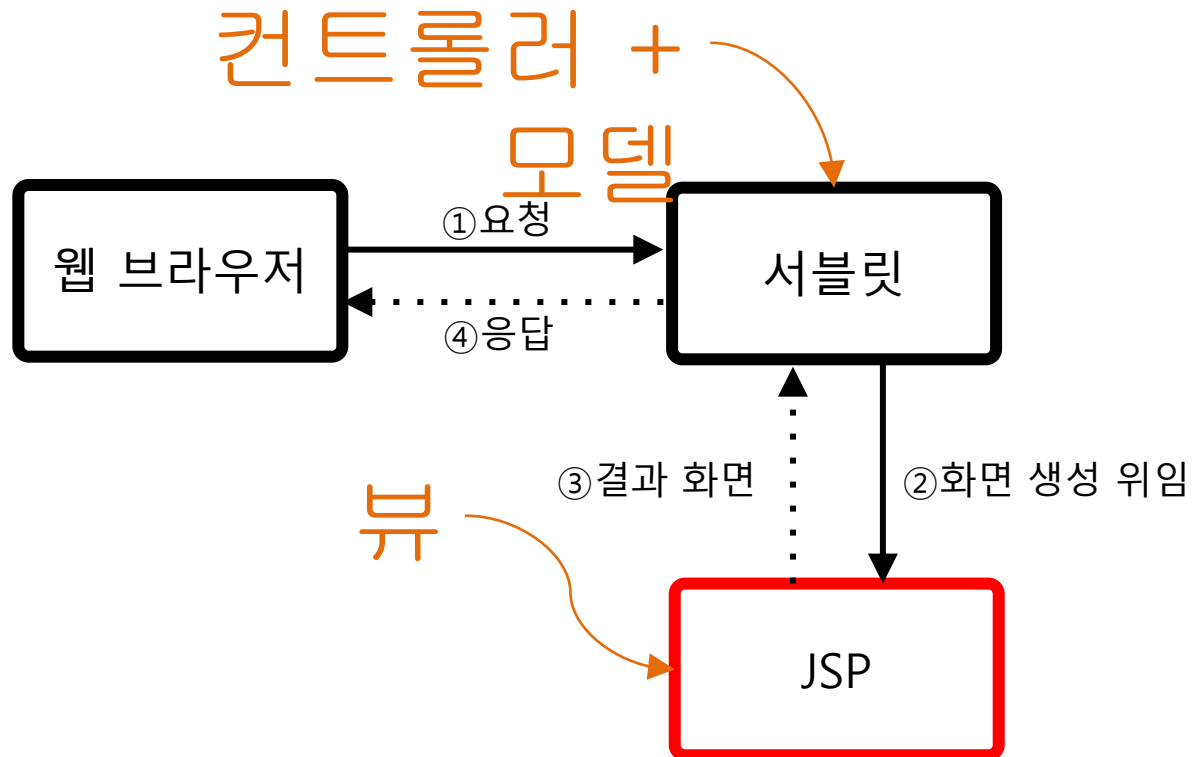
5.4 서블릿에서 뷰 분리하기

뷰 분리

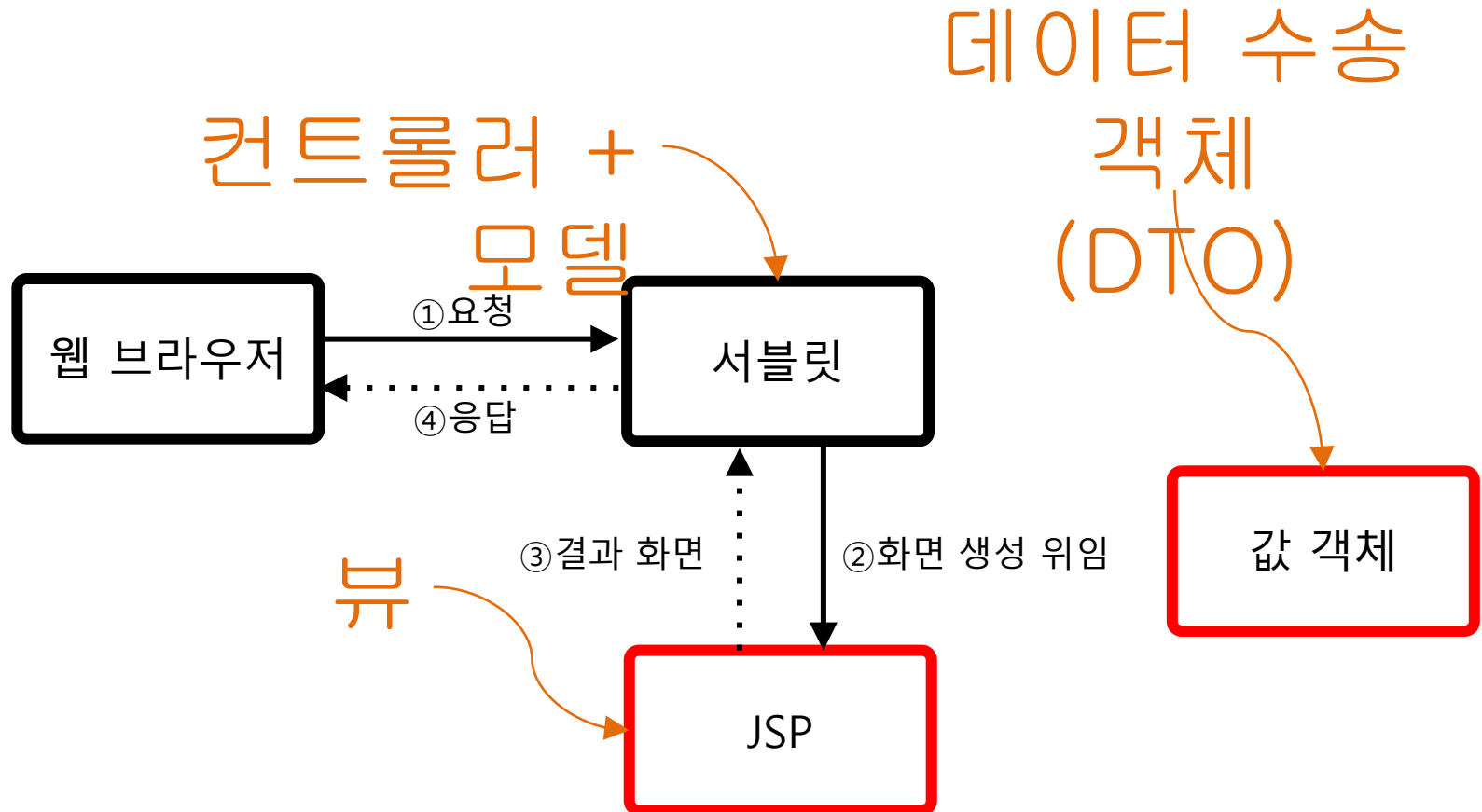
5.4 서블릿에서 뷰 분리하기



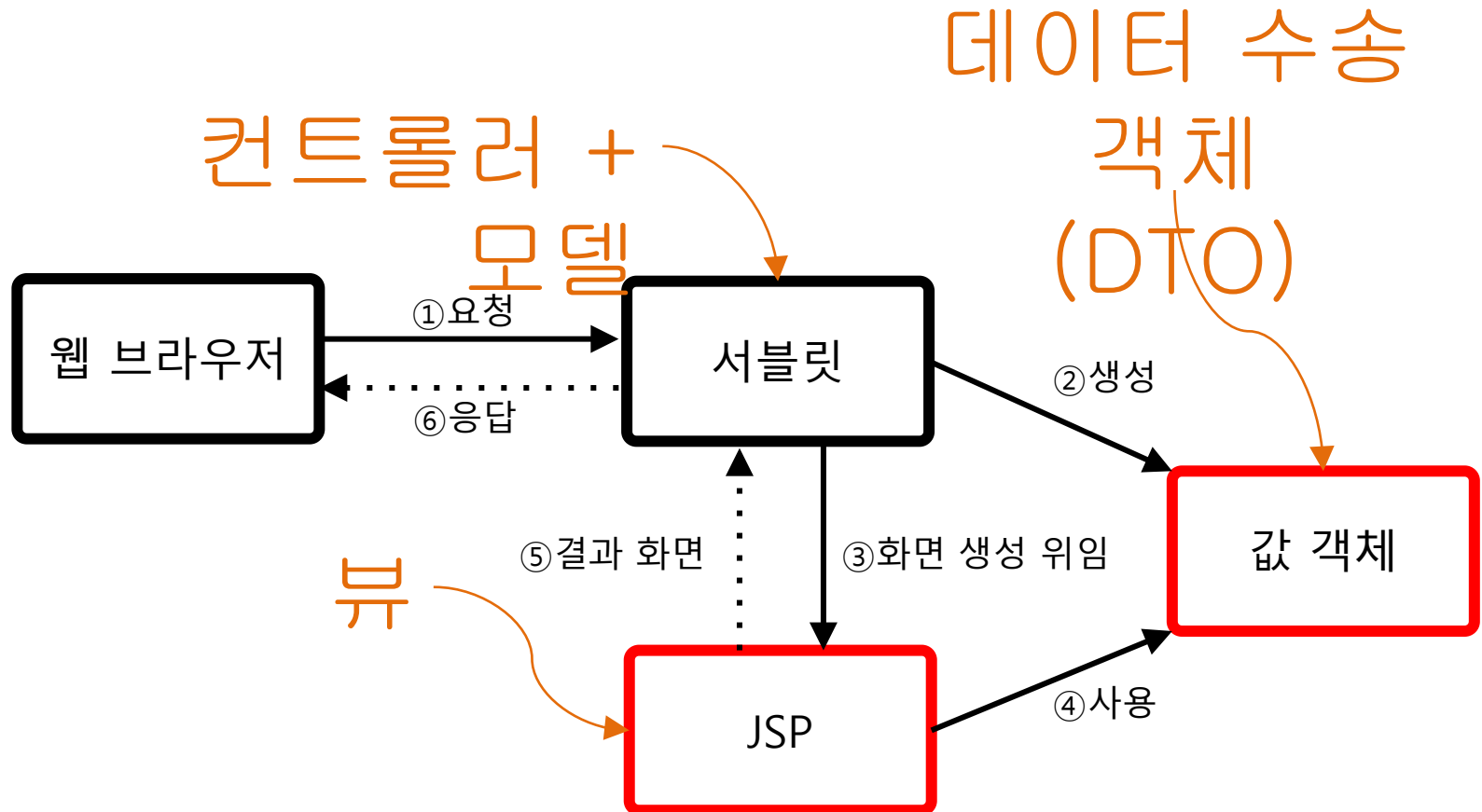
5.4 서블릿에서 뷰 분리하기



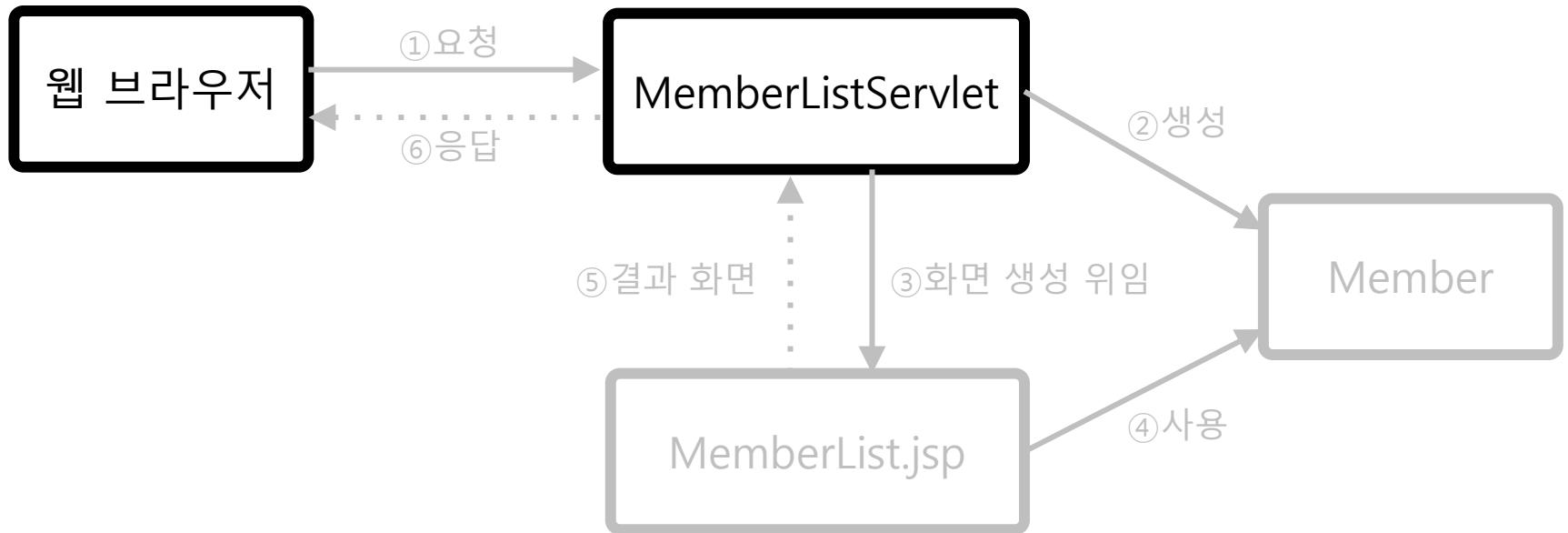
5.4 서블릿에서 뷰 분리하기



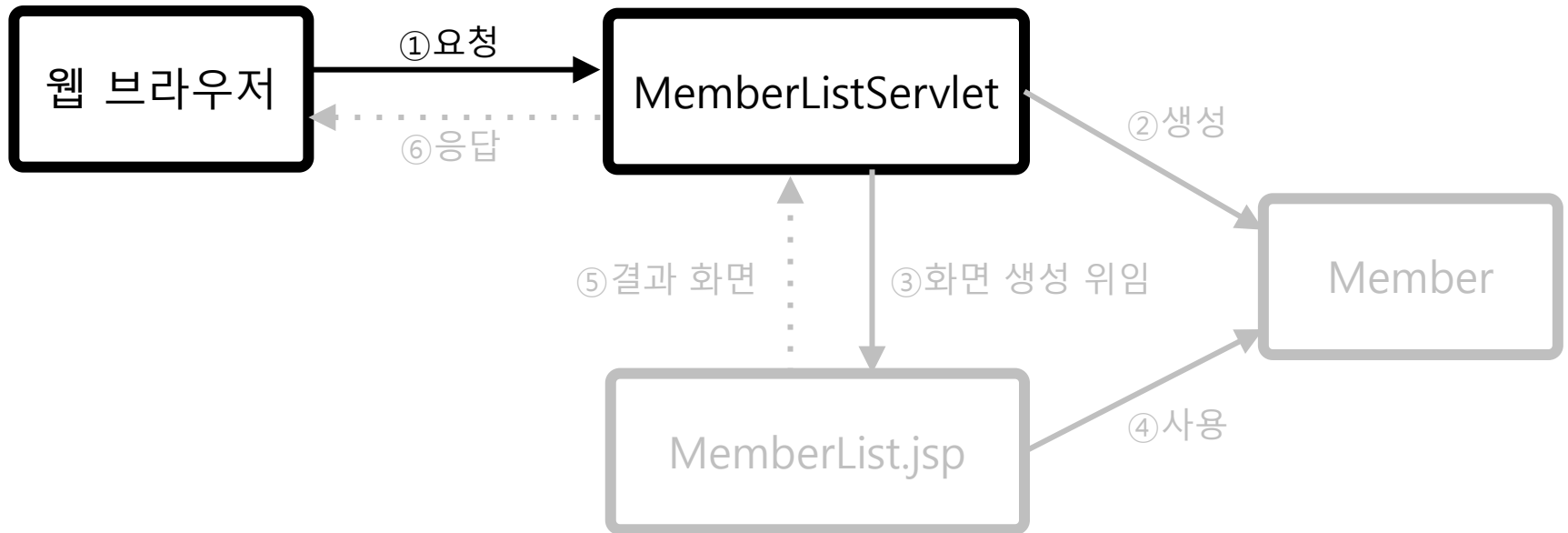
5.4 서블릿에서 뷰 분리하기



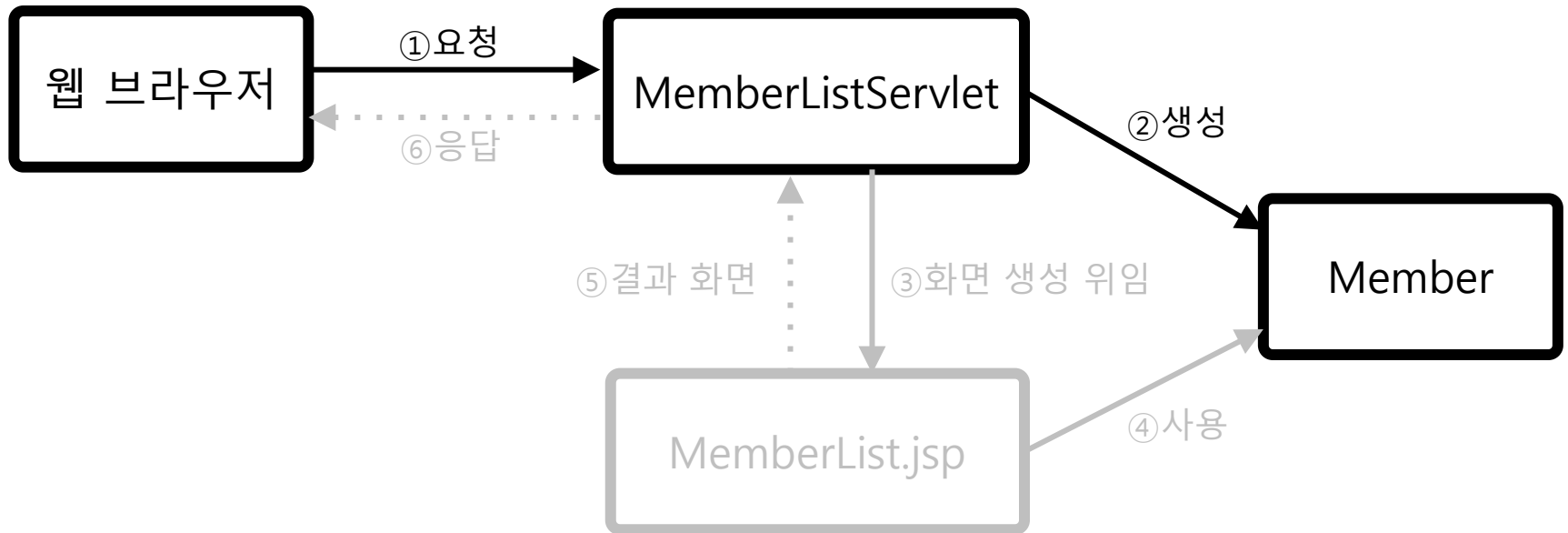
5.4 서블릿에서 뷰 분리하기



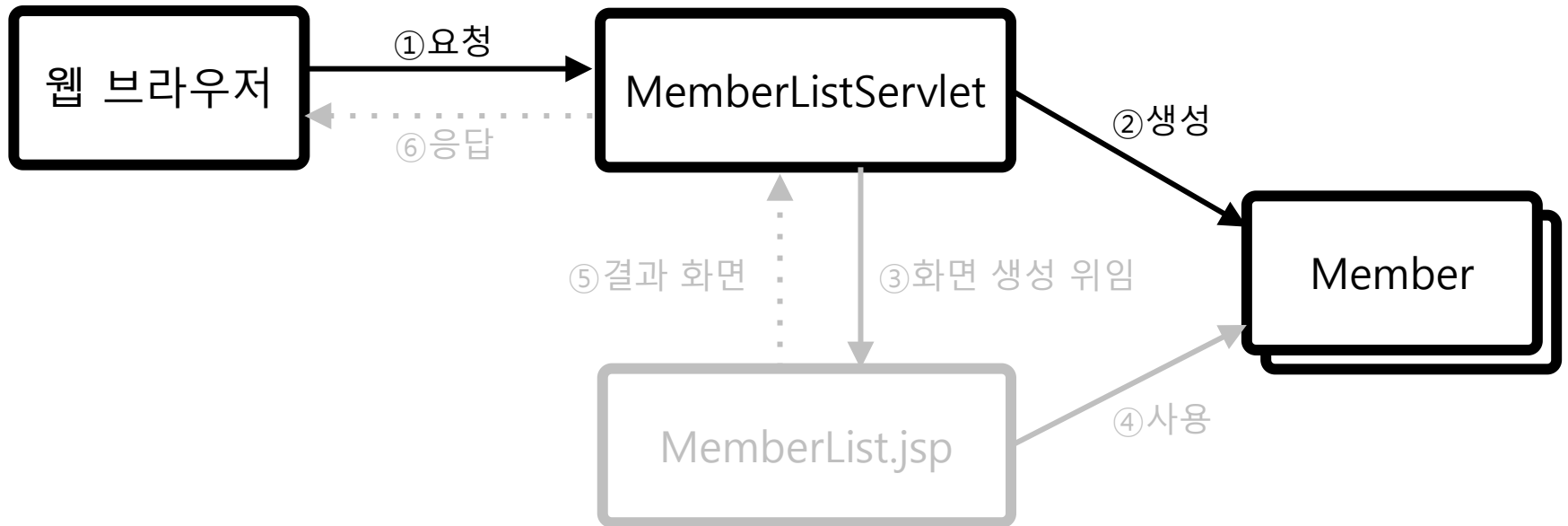
5.4 서블릿에서 뷰 분리하기



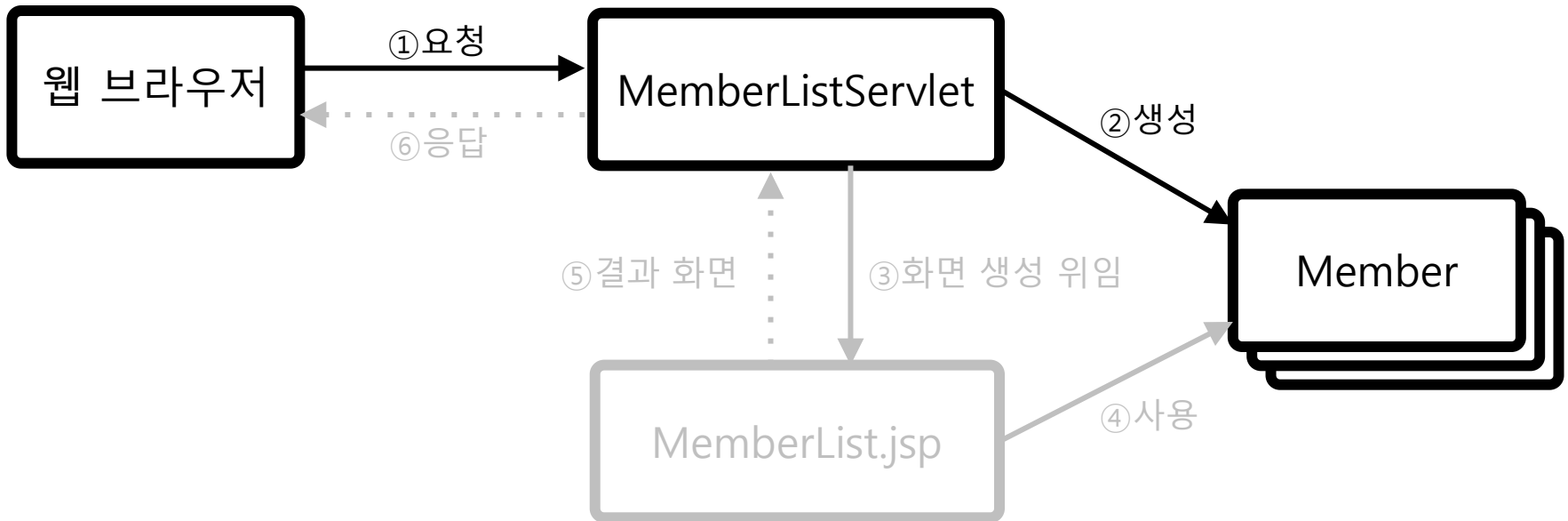
5.4 서블릿에서 뷰 분리하기



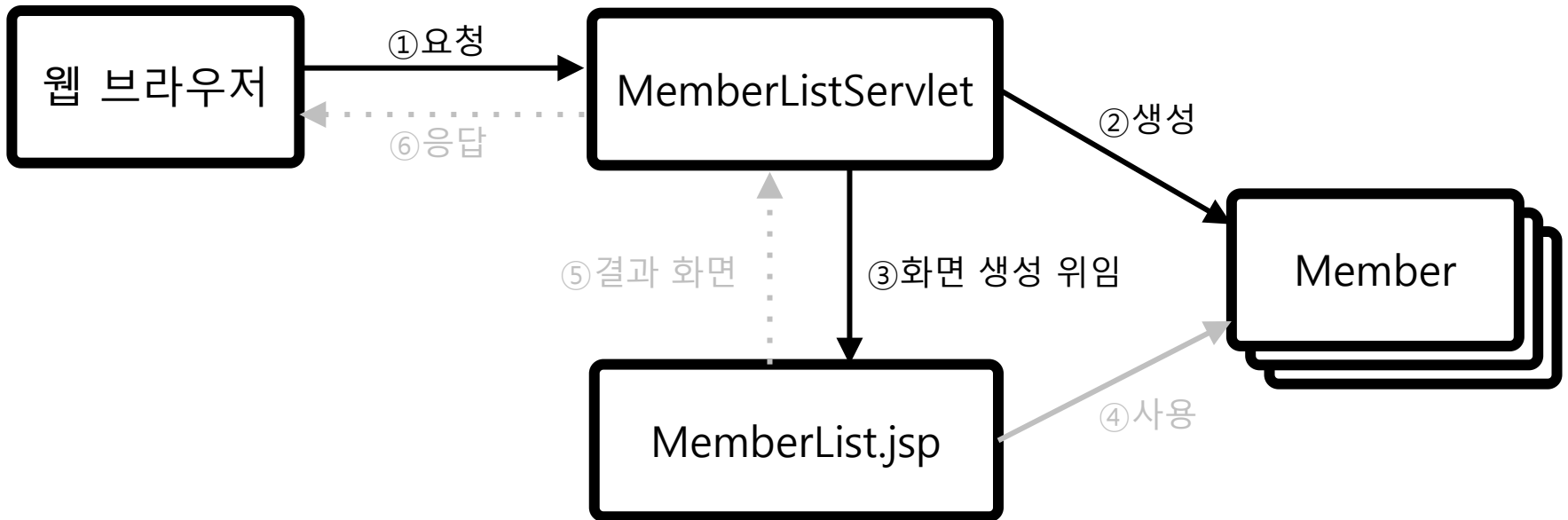
5.4 서블릿에서 뷰 분리하기



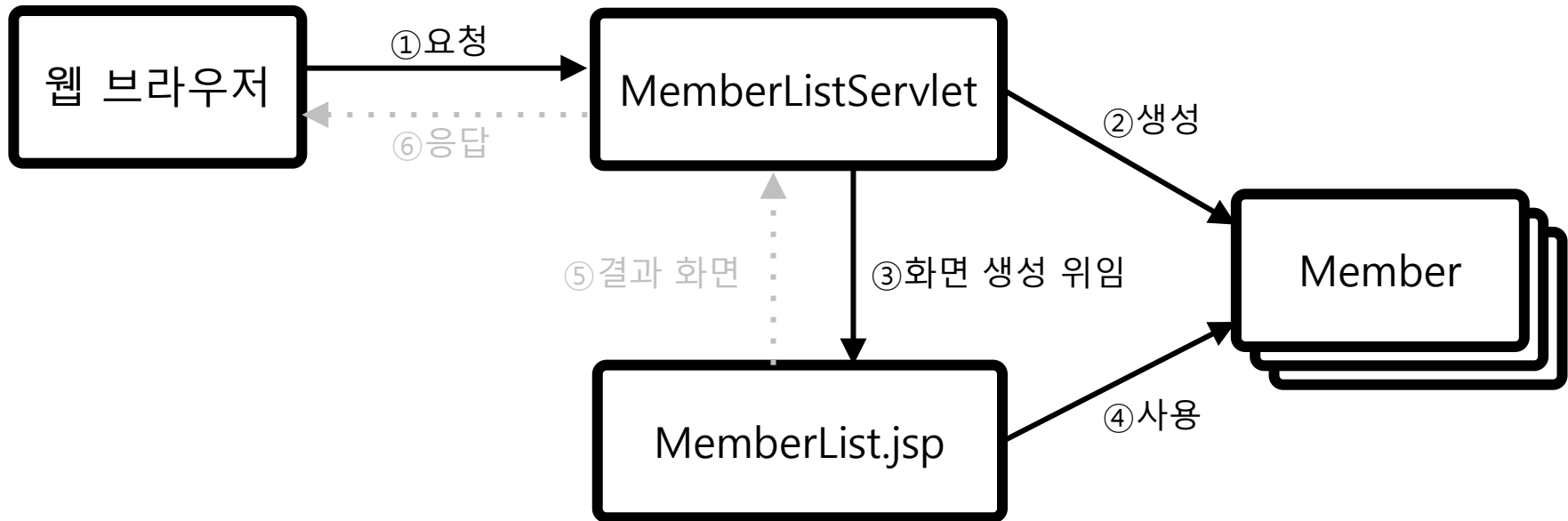
5.4 서블릿에서 뷰 분리하기



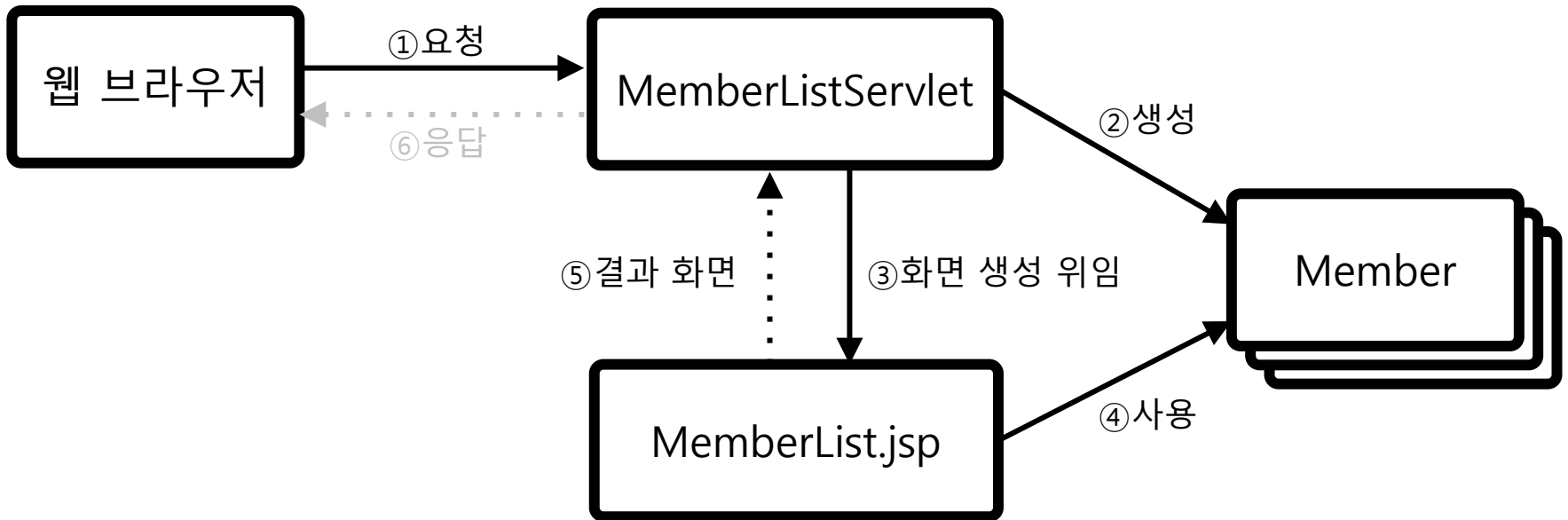
5.4 서블릿에서 뷰 분리하기



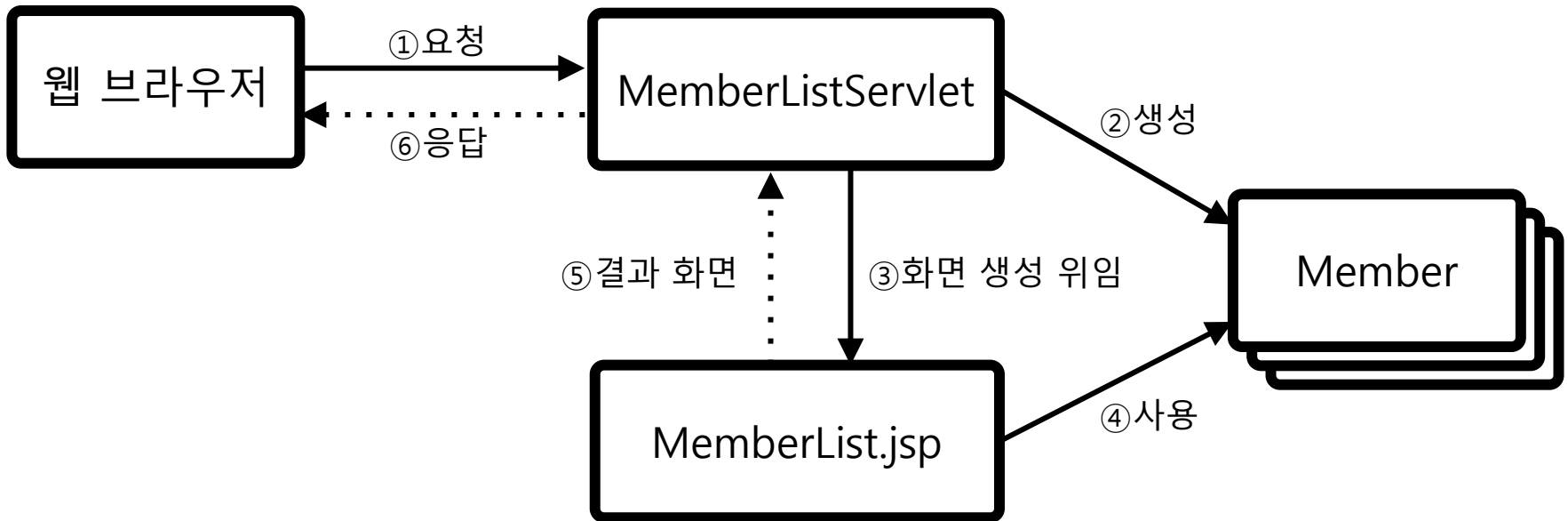
5.4 서블릿에서 뷰 분리하기



5.4 서블릿에서 뷰 분리하기

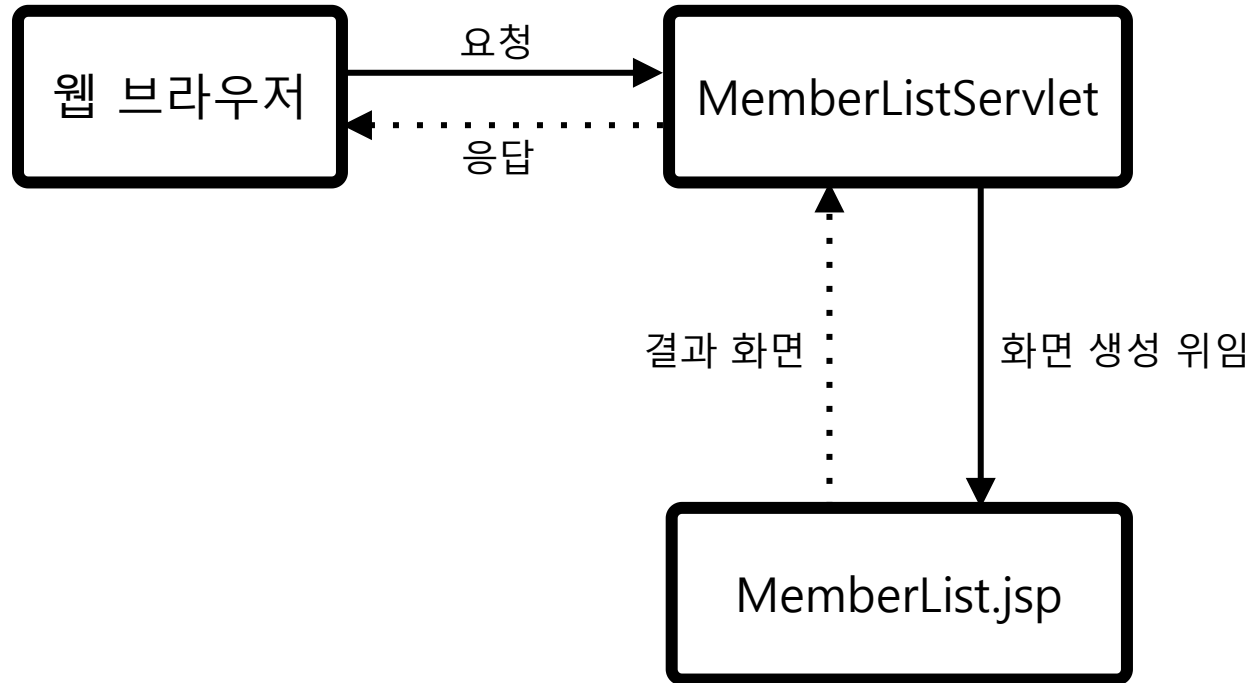


5.4 서블릿에서 뷰 분리하기

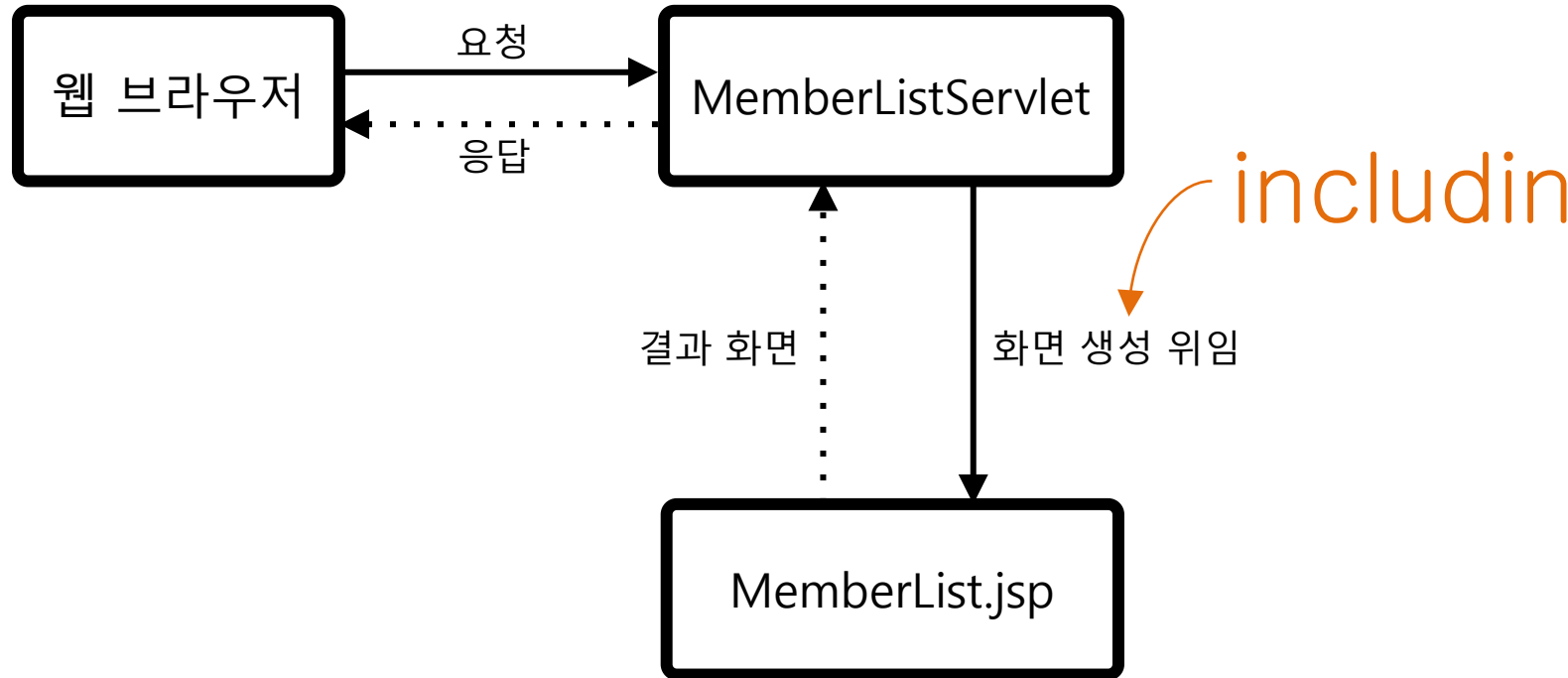


5.5 포워딩과 인클루딩

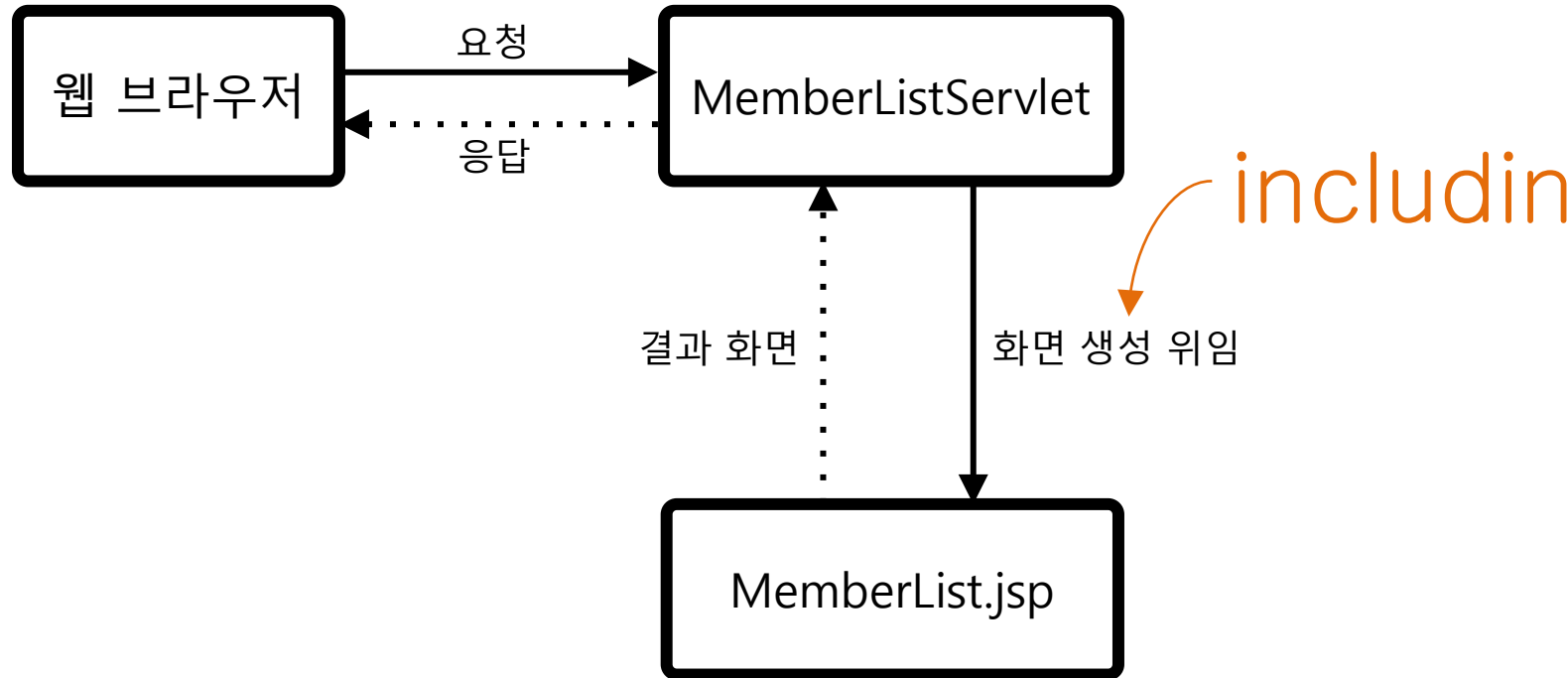
5.5 포워딩과 인클루딩



5.5 포워딩과 인클루딩



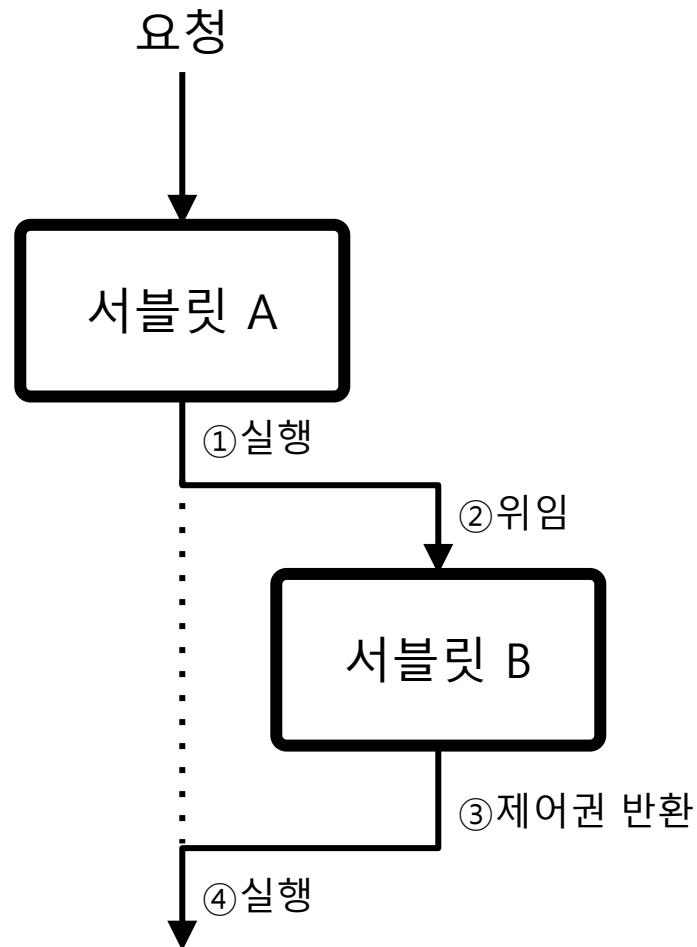
5.5 포워딩과 인클루딩



```
RequestDispatcher rd = request.getRequestDispatcher(  
    "/member/MemberList.jsp");  
rd.include(request, response);
```

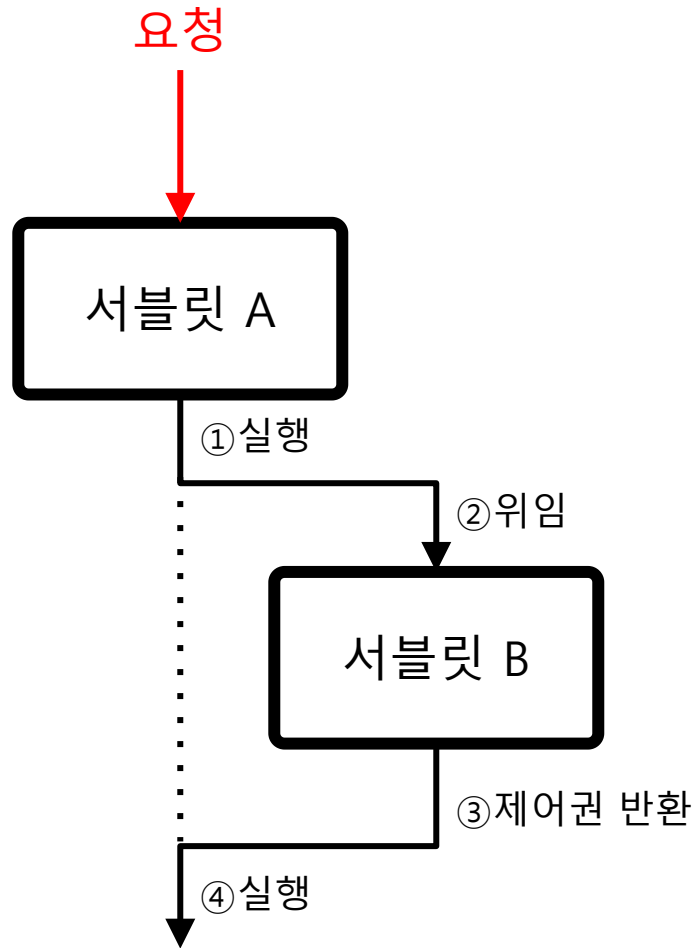
5.5 포워딩과 인클루딩

인클루딩



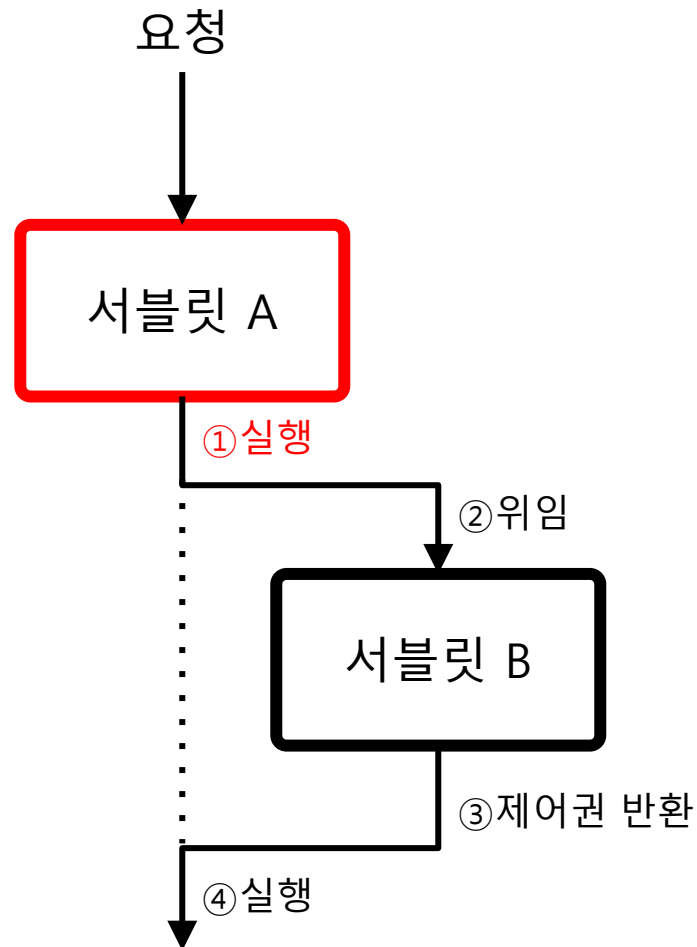
5.5 포워딩과 인클루딩

인클루딩



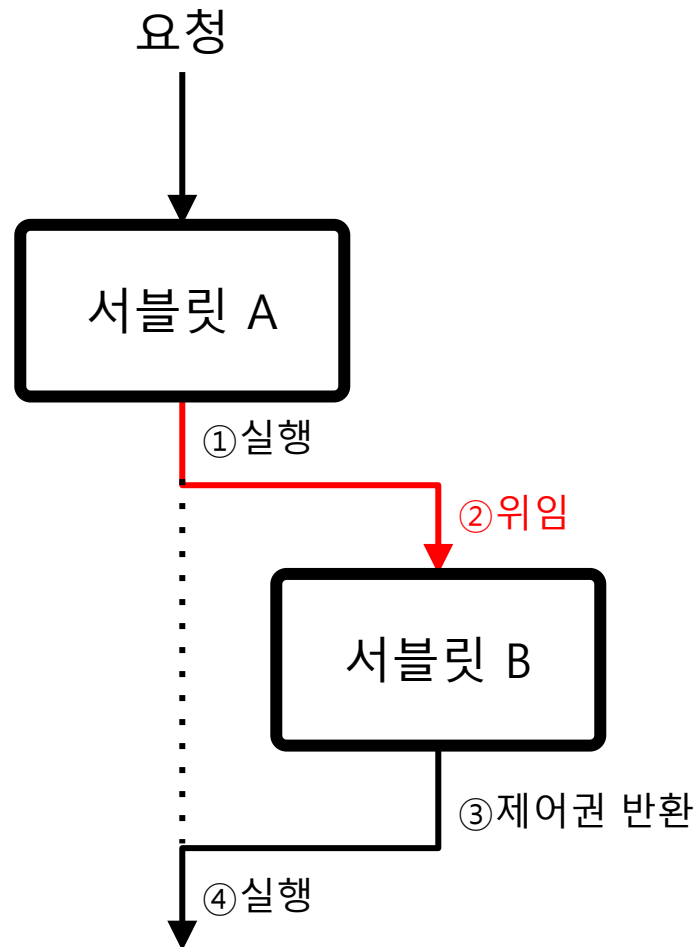
5.5 포워딩과 인클루딩

인클루딩



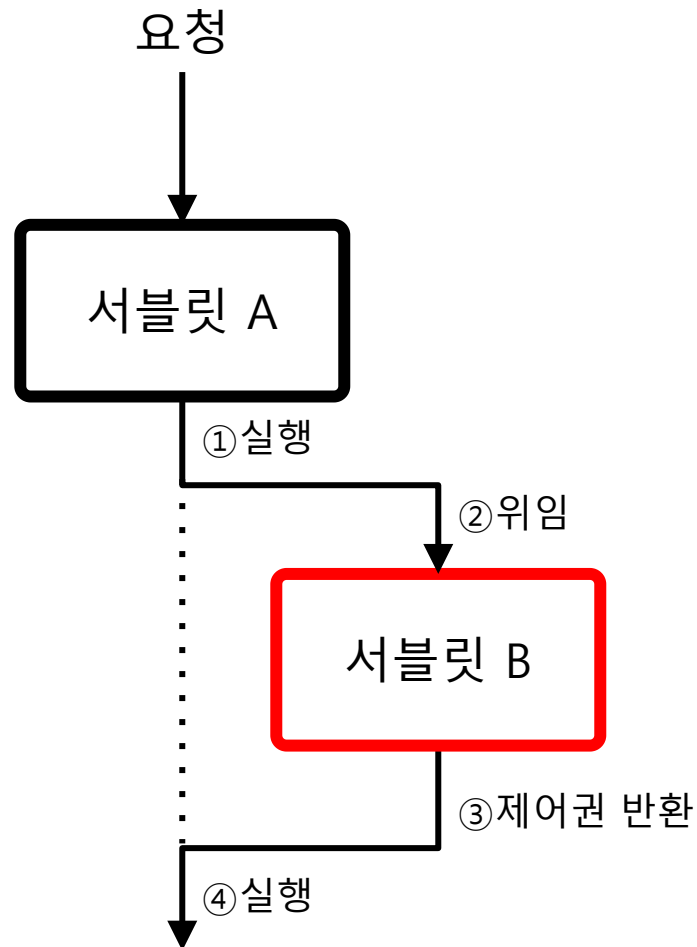
5.5 포워딩과 인클루딩

인클루딩



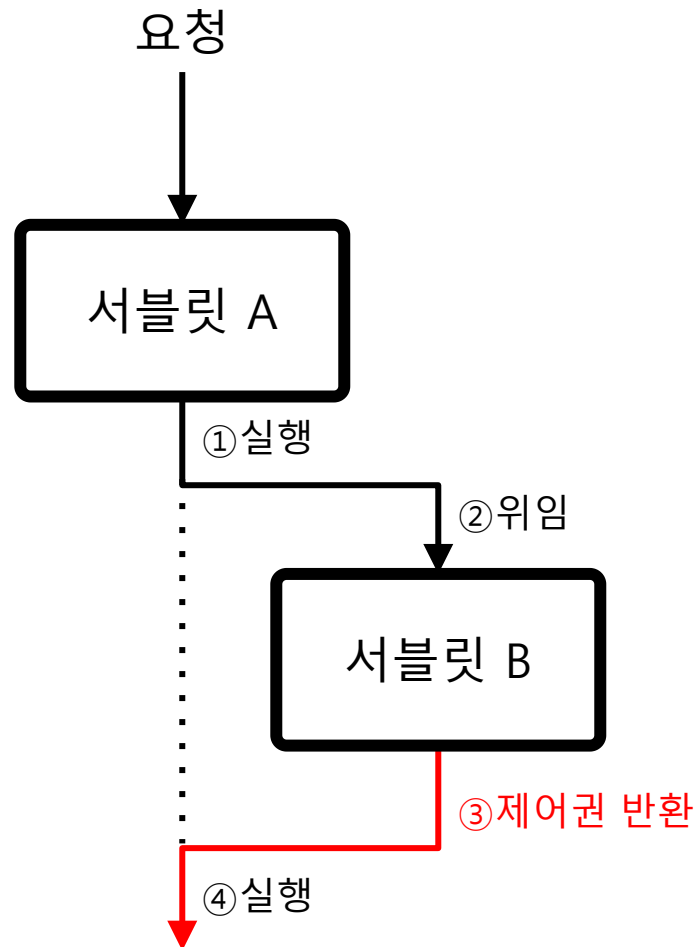
5.5 포워딩과 인클루딩

인클루딩



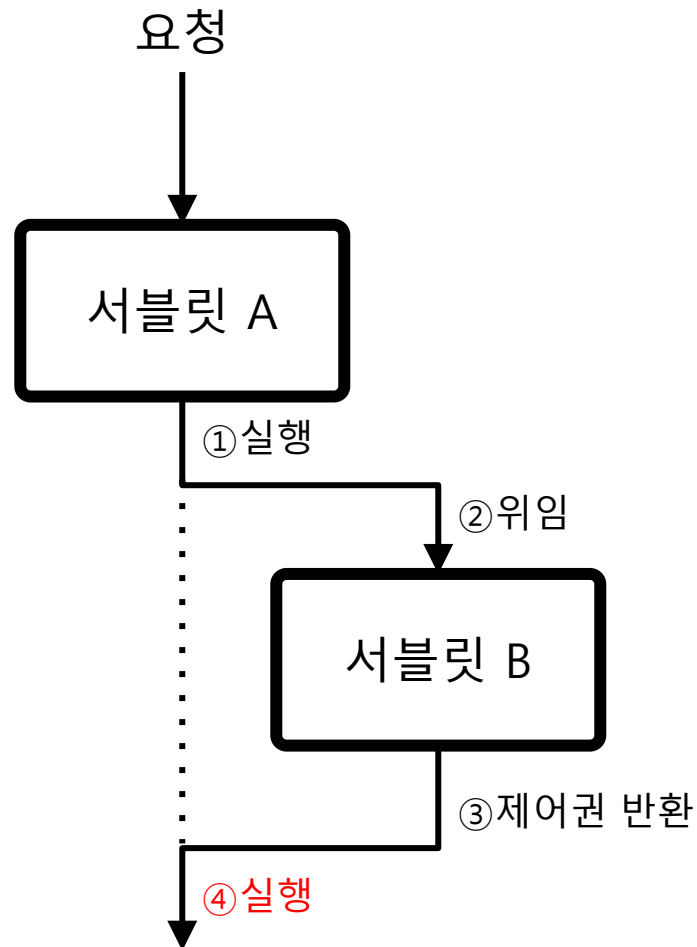
5.5 포워딩과 인클루딩

인클루딩



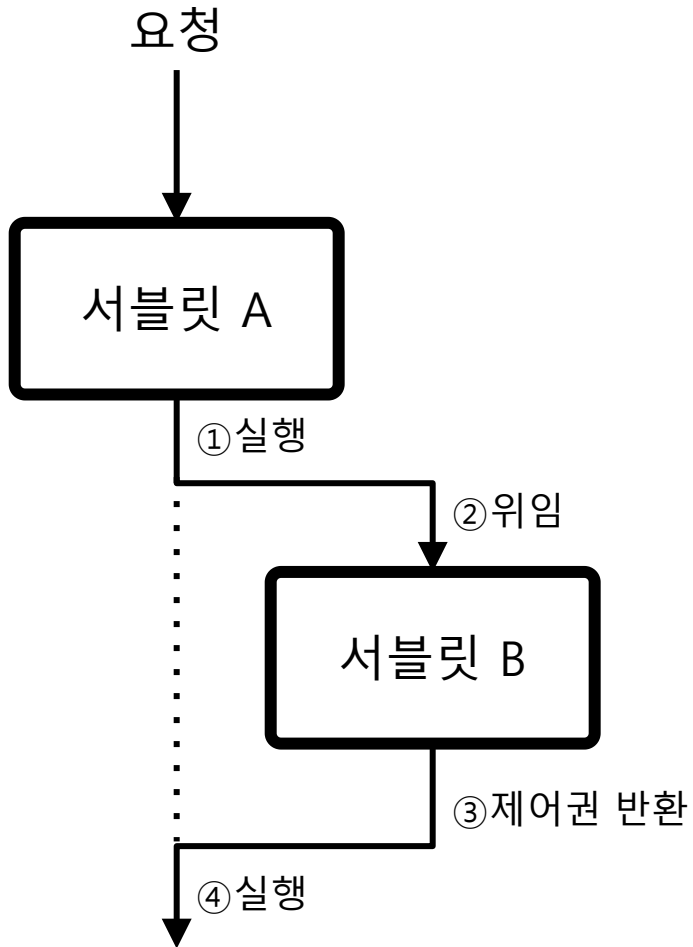
5.5 포워딩과 인클루딩

인클루딩

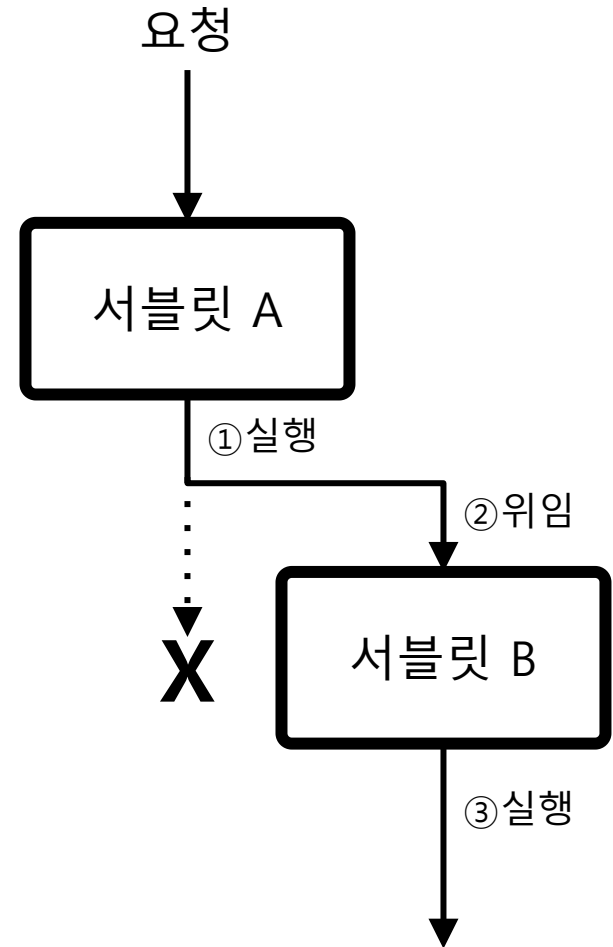


5.5 포워딩과 인클루딩

인클루딩

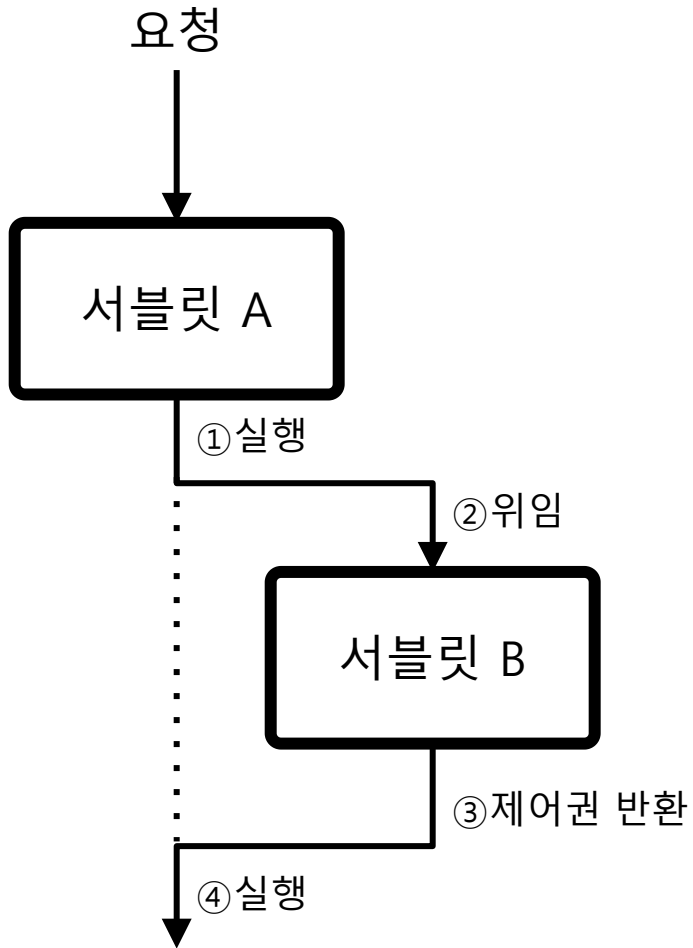


포워딩

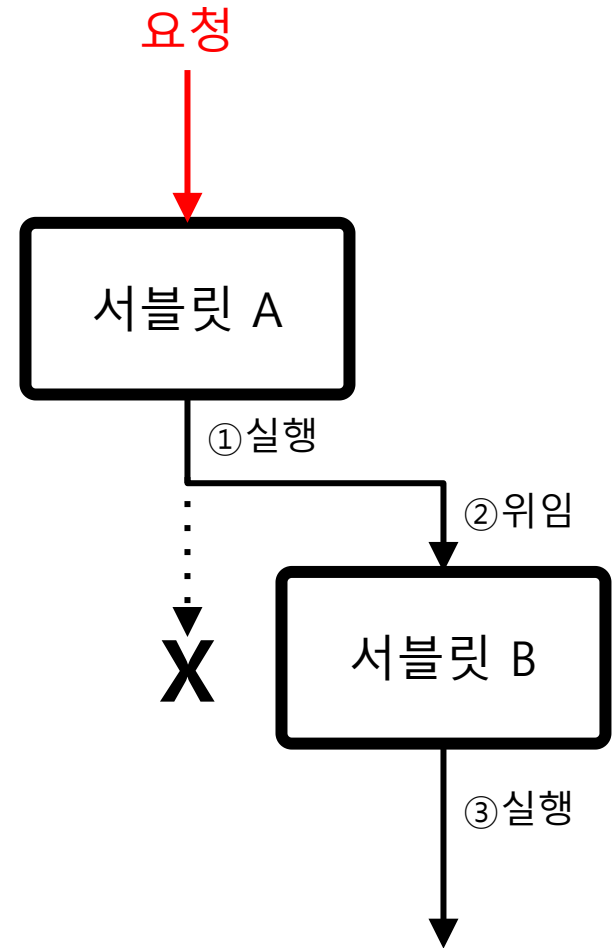


5.5 포워딩과 인클루딩

인클루딩

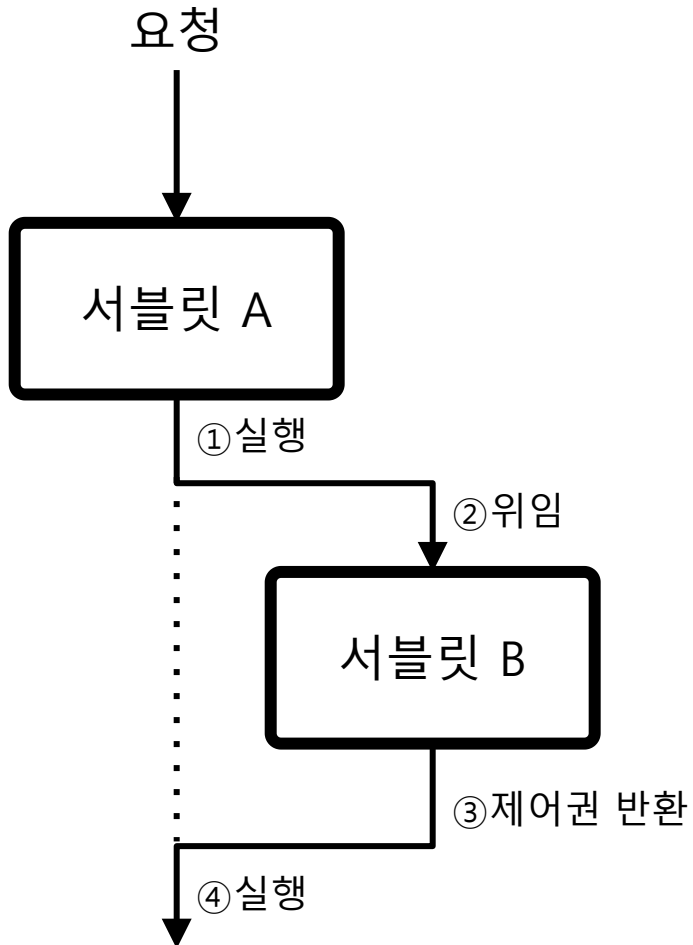


포워딩

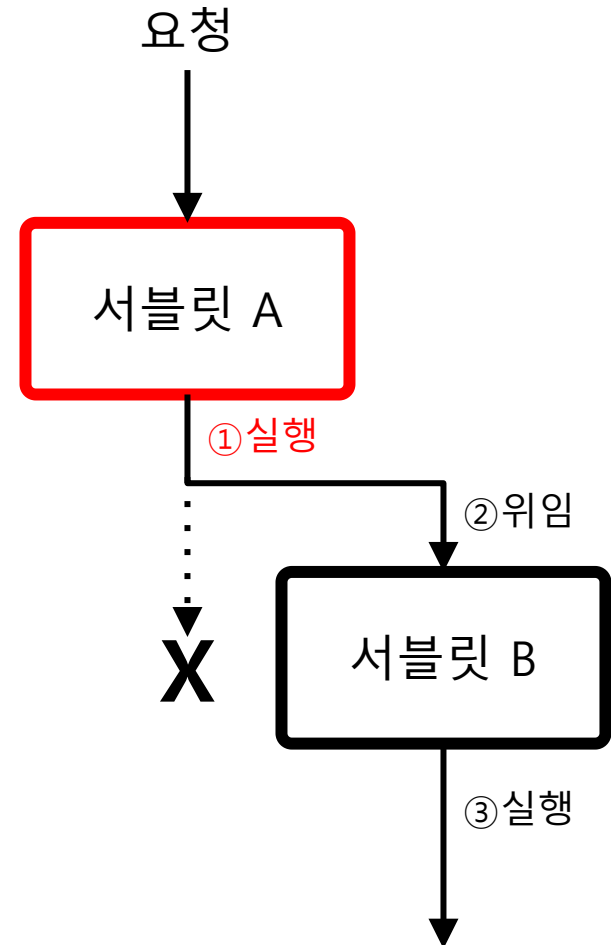


5.5 포워딩과 인클루딩

인클루딩

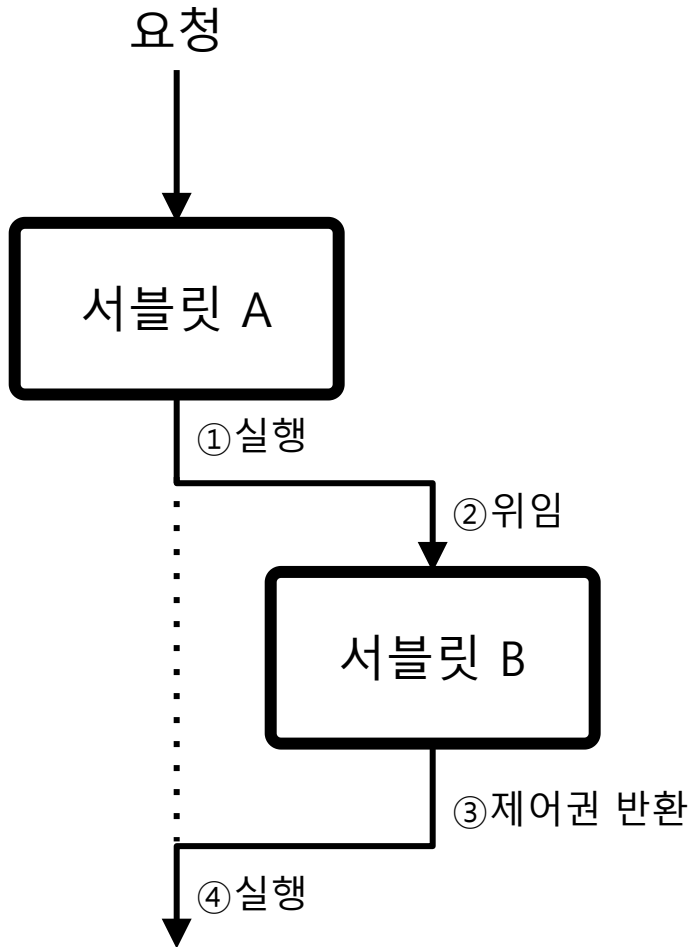


포워딩

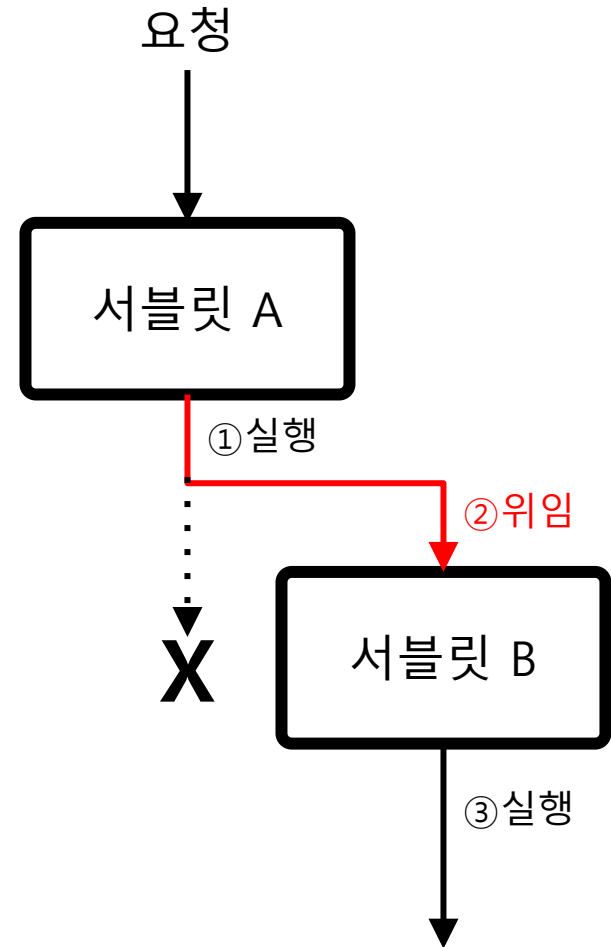


5.5 포워딩과 인클루딩

인클루딩

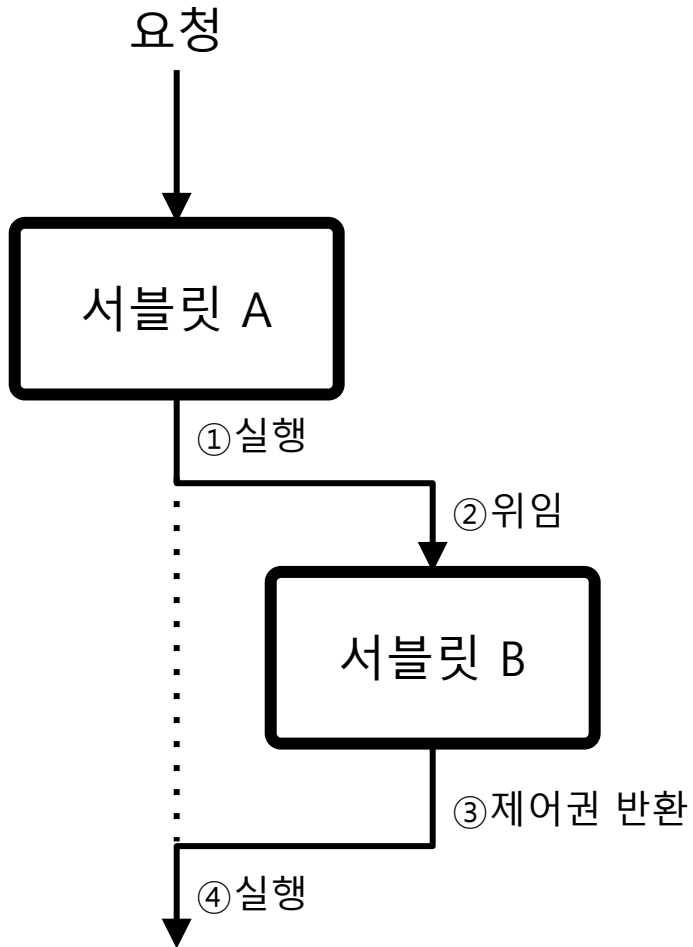


포워딩

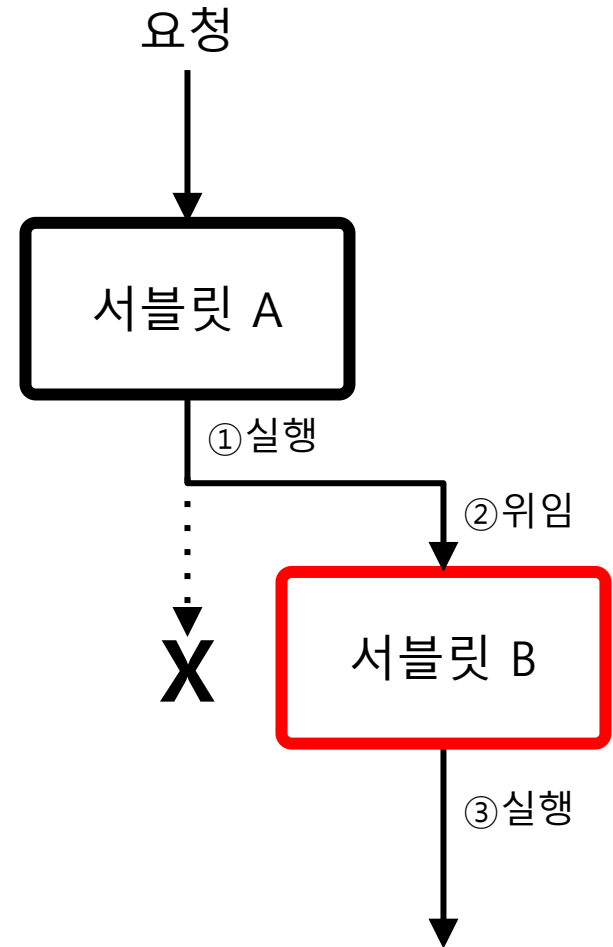


5.5 포워딩과 인클루딩

인클루딩

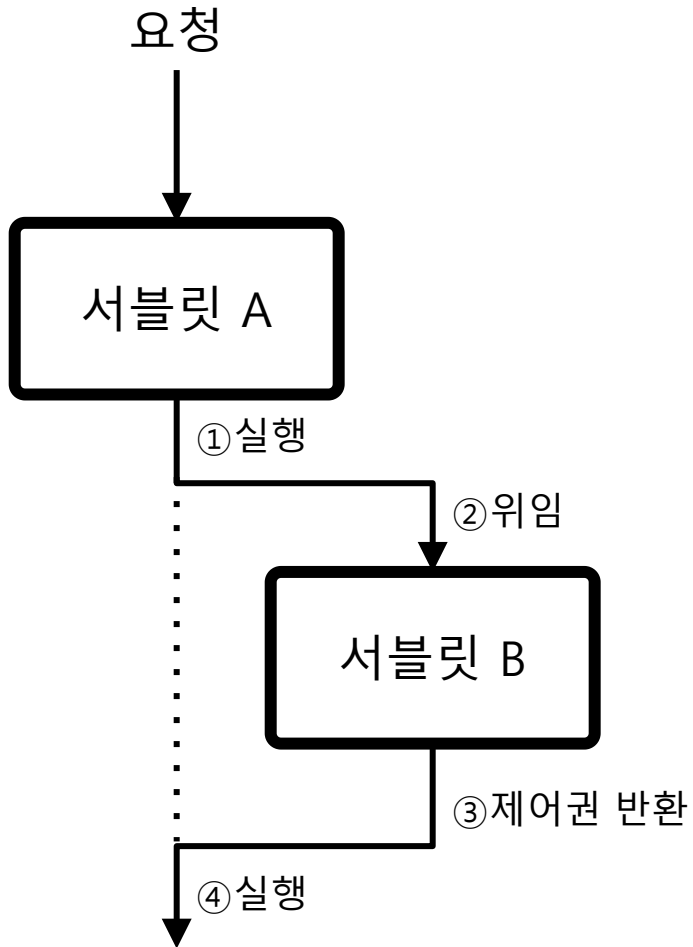


포워딩

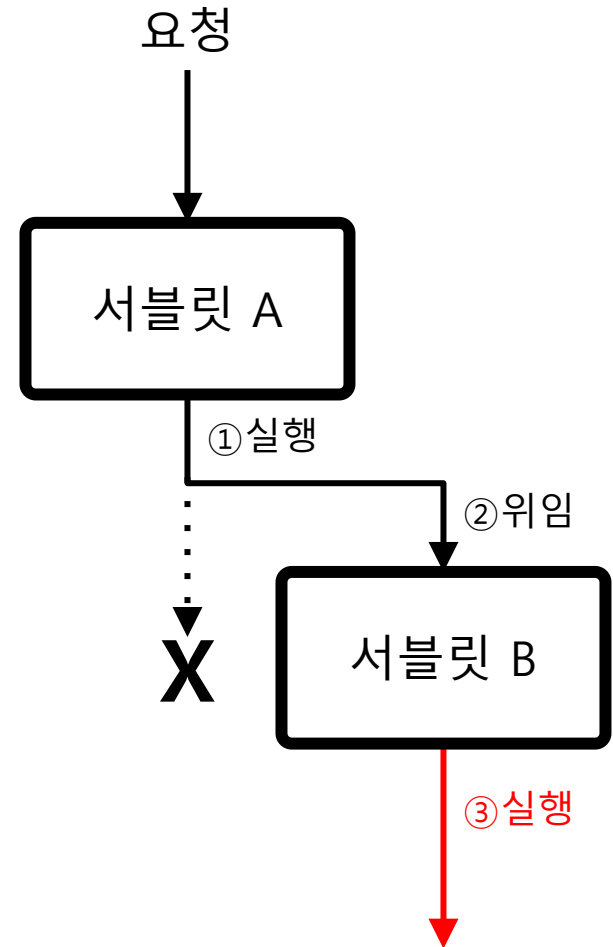


5.5 포워딩과 인클루딩

인클루딩

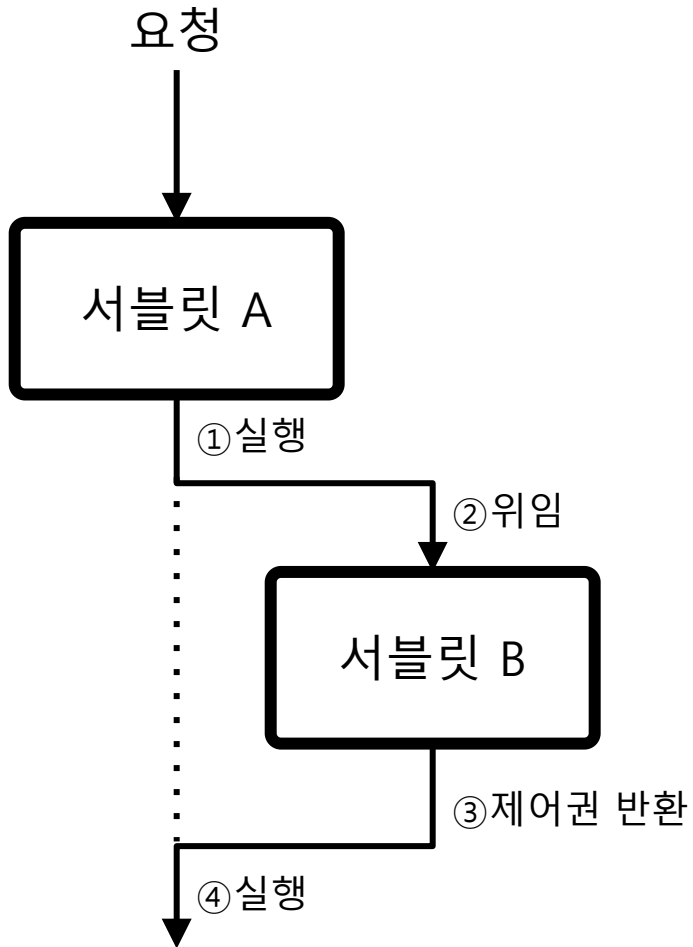


포워딩

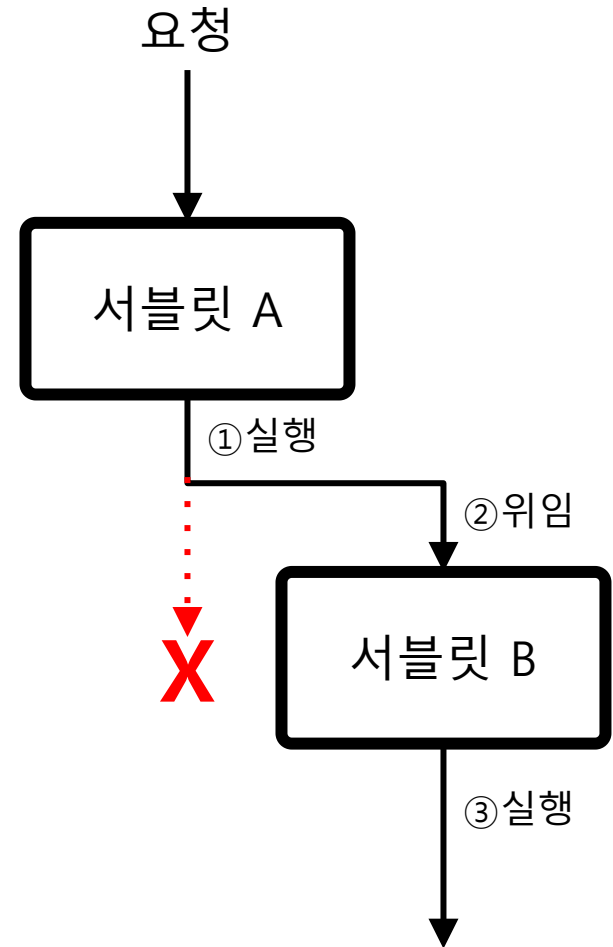


5.5 포워딩과 인클루딩

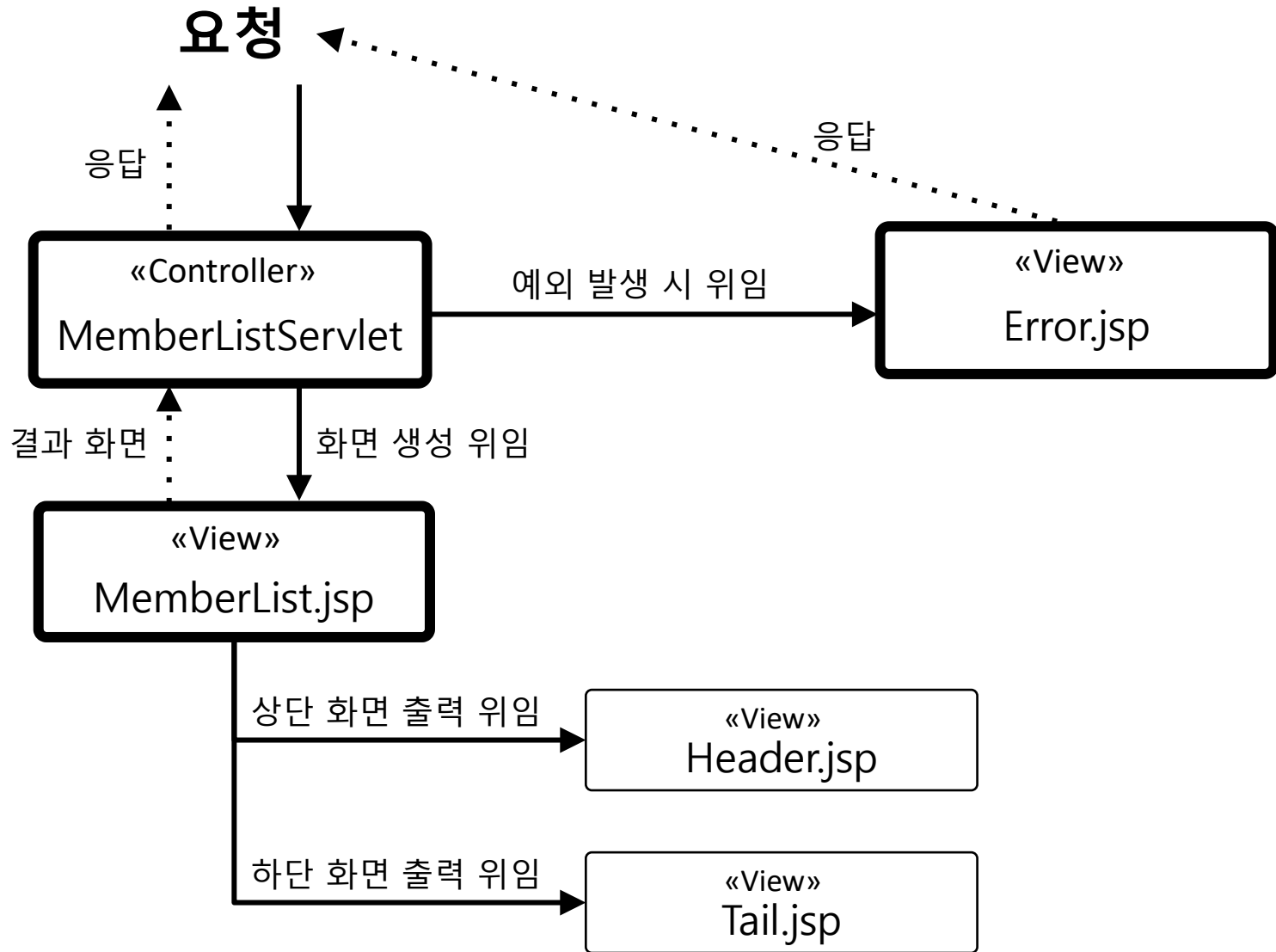
인클루딩



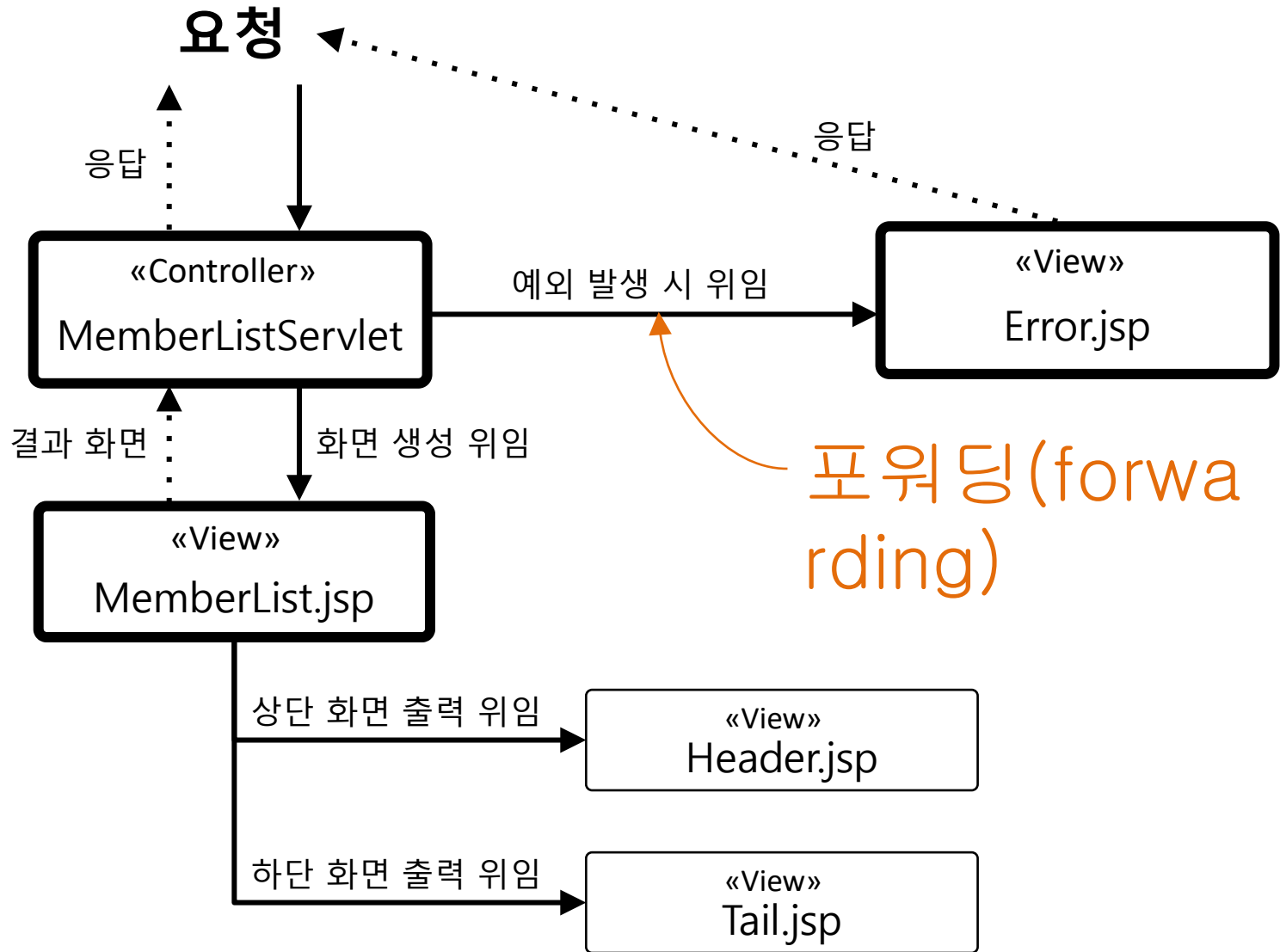
포워딩



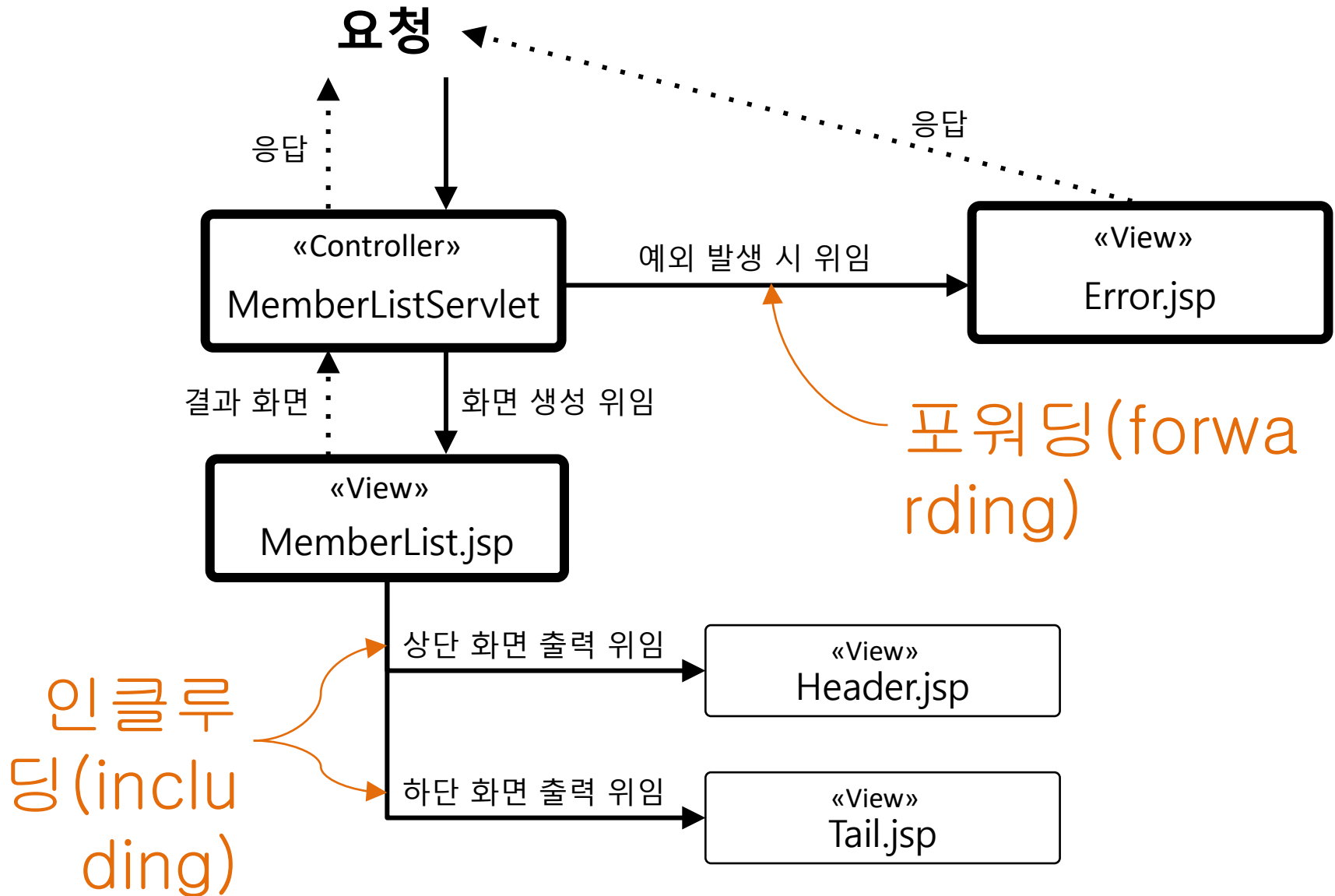
5.5 포워딩과 인클루딩



5.5 포워딩과 인클루딩



5.5 포워딩과 인클루딩



5.6 데이터 보관소

5.6 데이터 보관소

생성과 소멸 시점

5.6 데이터 보관소

생성과 소멸 시점

ServletContext

HttpSession

ServletRequest

JspContext

5.6 데이터 보관소

ServletContext

HttpSession

ServletRequest

JspContext



5.6 데이터 보관소

HttpSession

ServletRequest

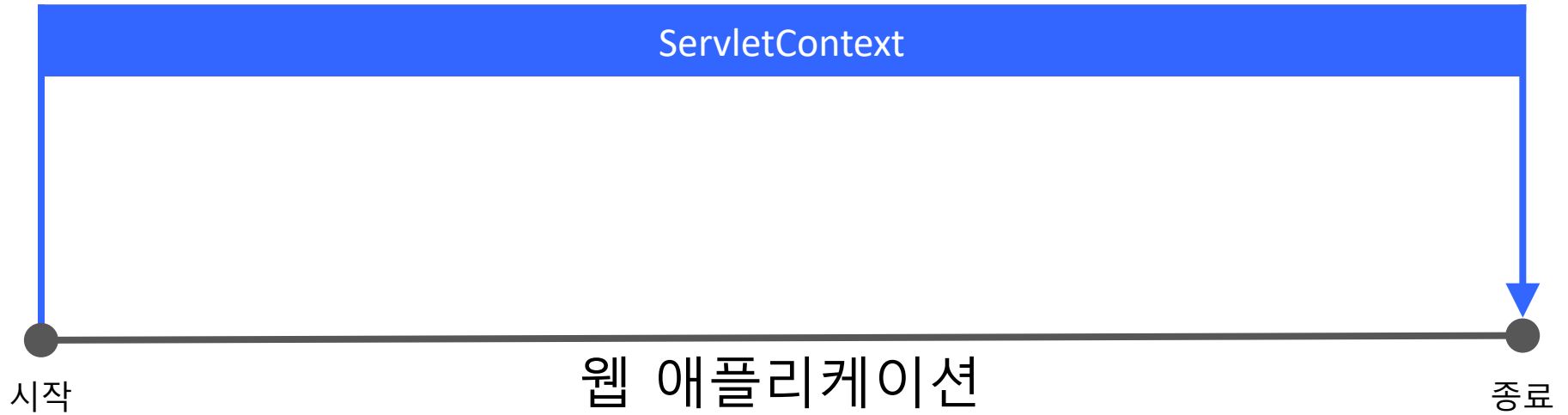
JspContext

ServletContext

시작

웹 애플리케이션

종료

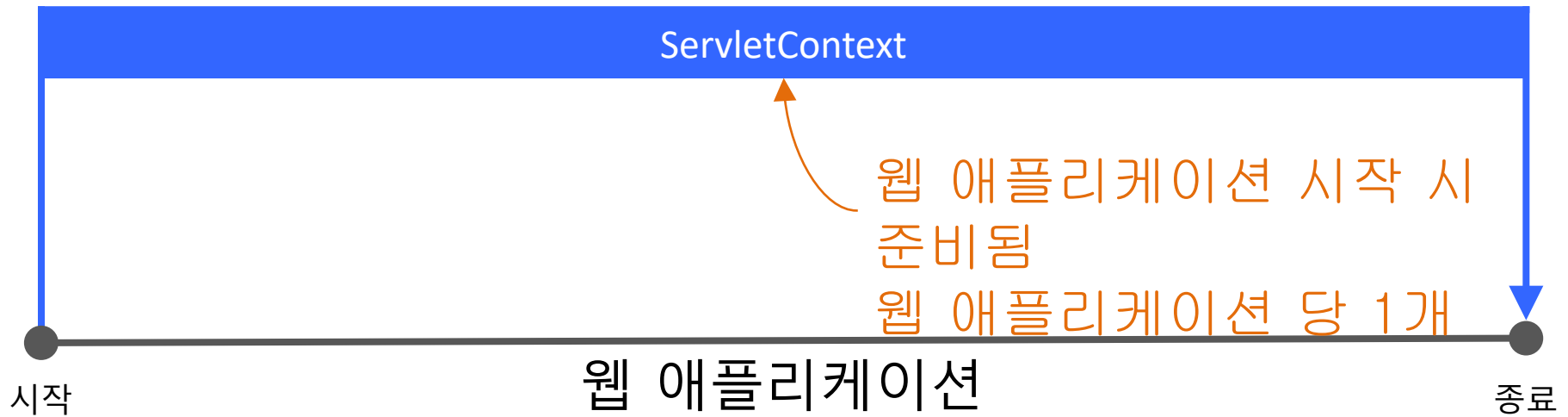


5.6 데이터 보관소

HttpSession

ServletRequest

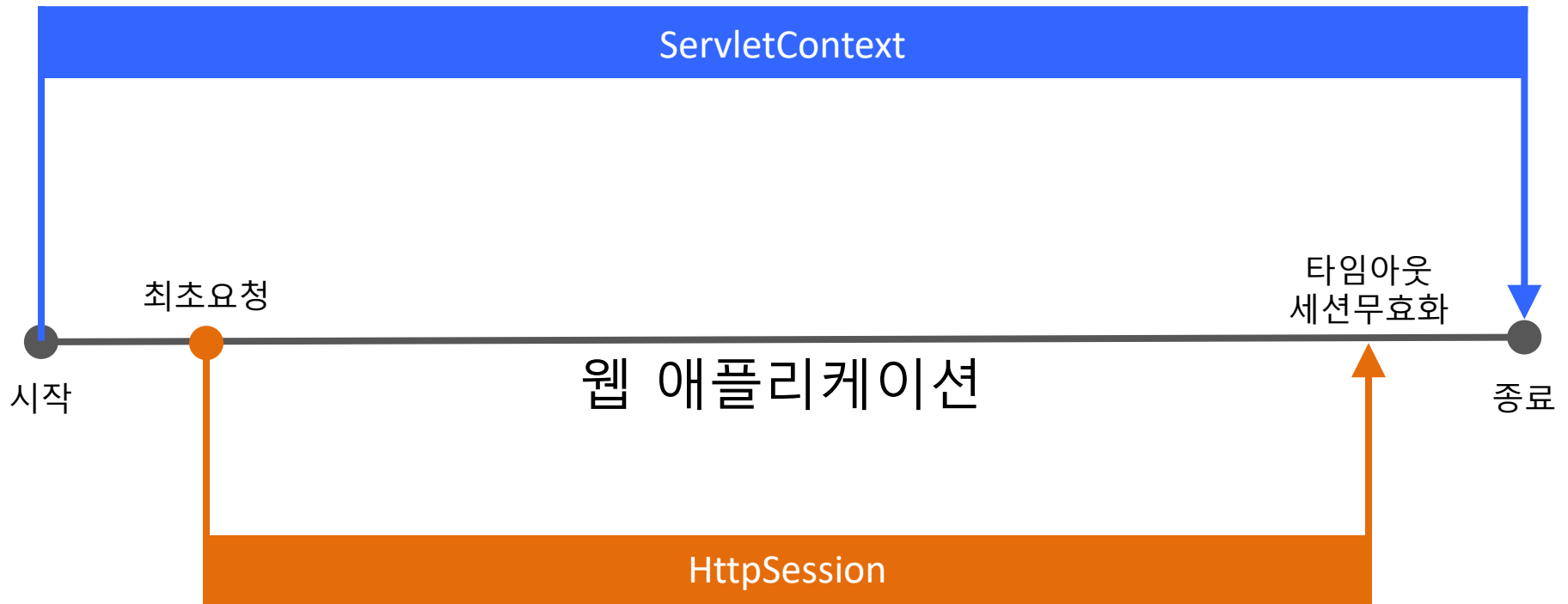
JspContext



5.6 데이터 보관소

ServletRequest

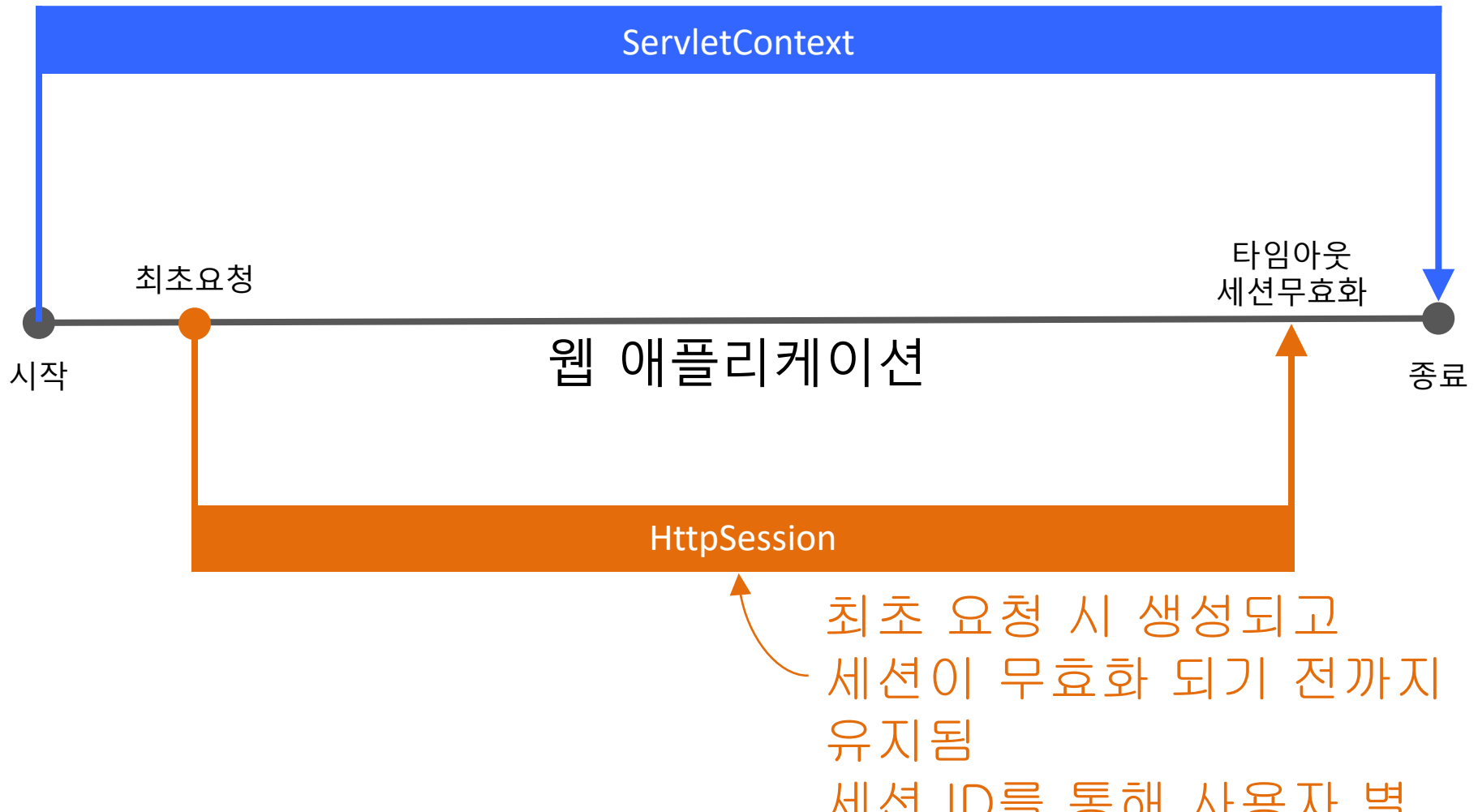
JspContext



5.6 데이터 보관소

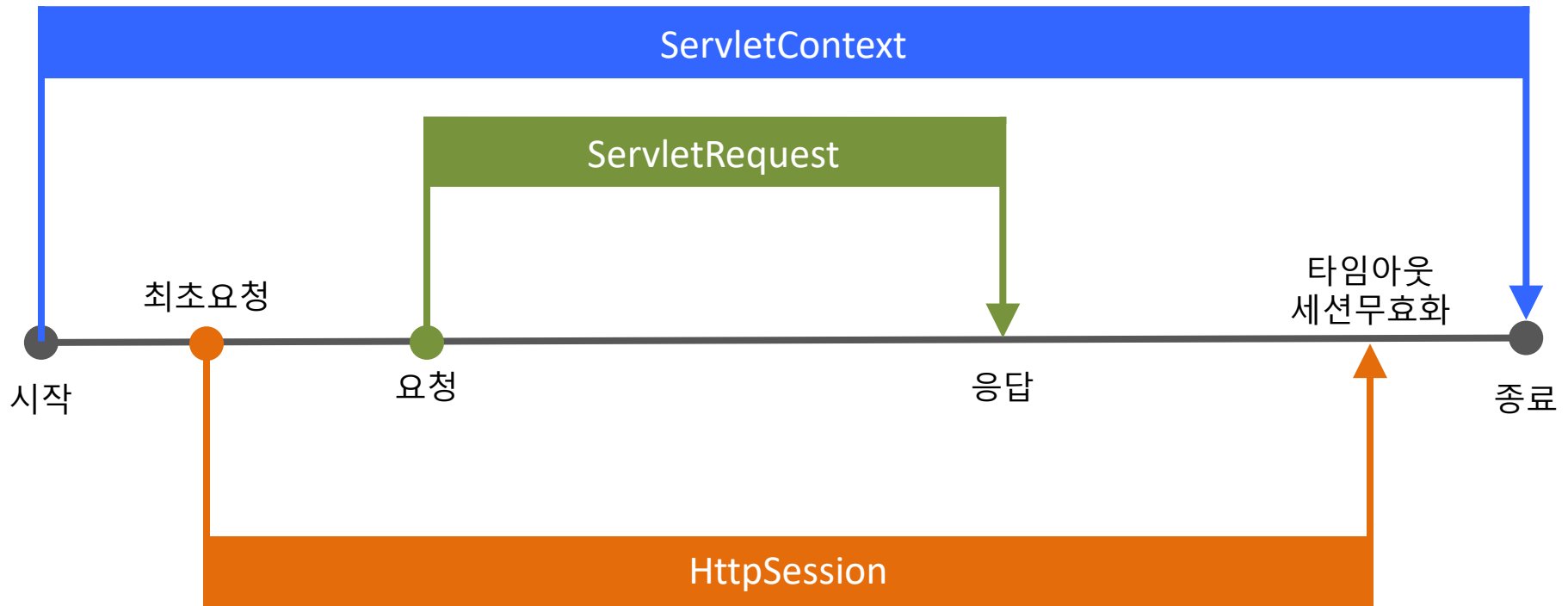
ServletRequest

JspContext



5.6 데이터 보관소

JspContext

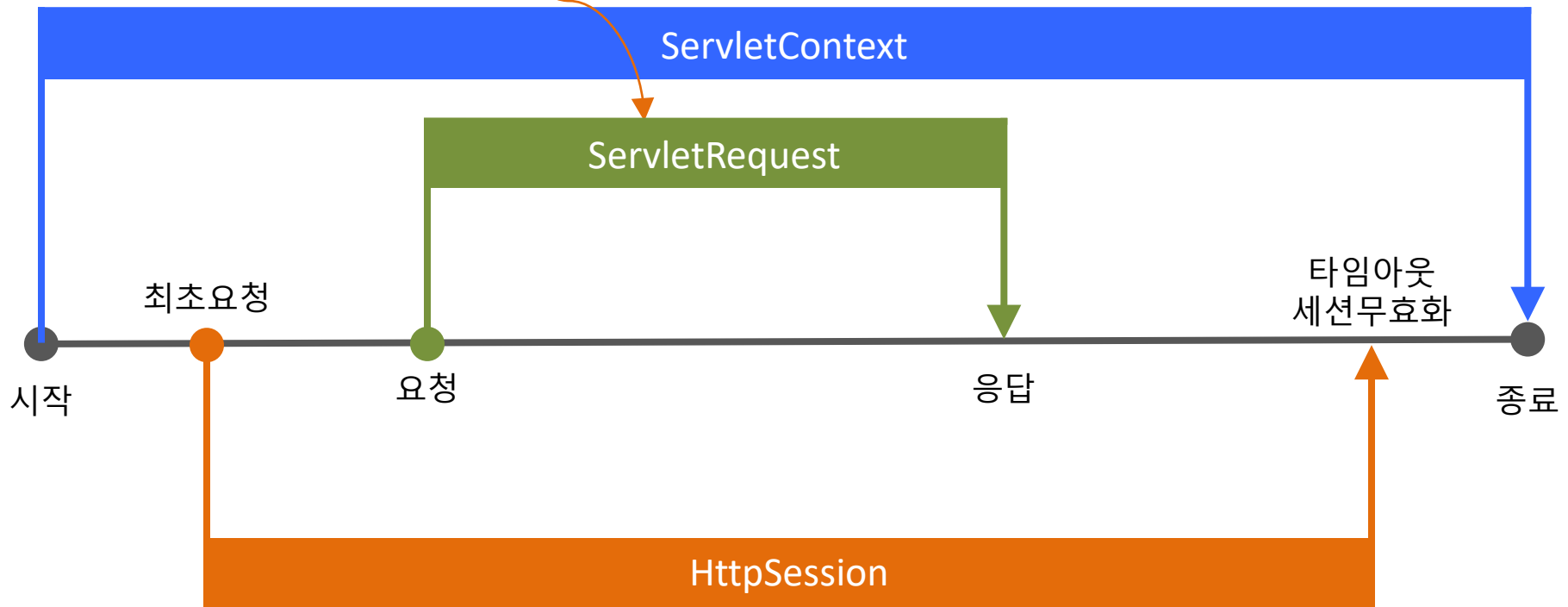


웹 애플리케이션

5.6 데이터 보관소

매 요청 때마다 생성되고
응답 전까지 유지됨

JspContext

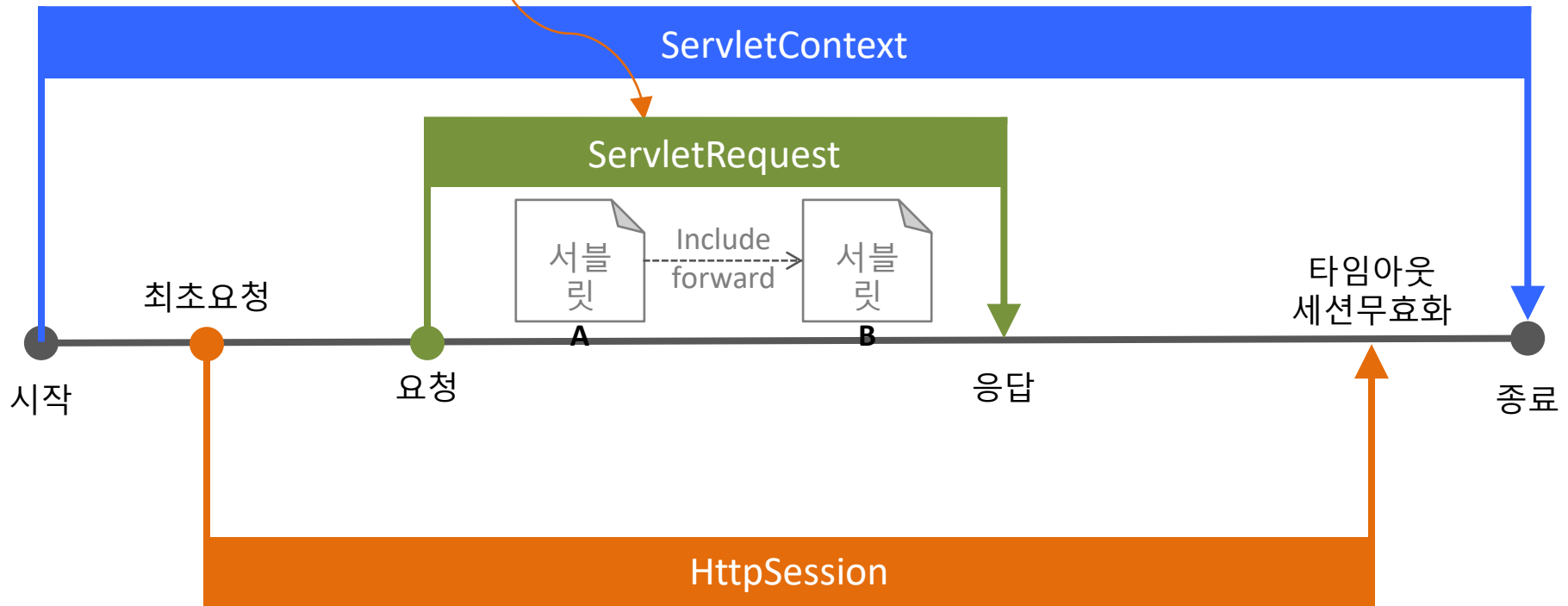


웹 애플리케이션

5.6 데이터 보관소

매 요청 때마다 생성되고
응답 전까지 유지됨
포워드, 인클루드 서블릿끼리 데이터
공유할 때 적합

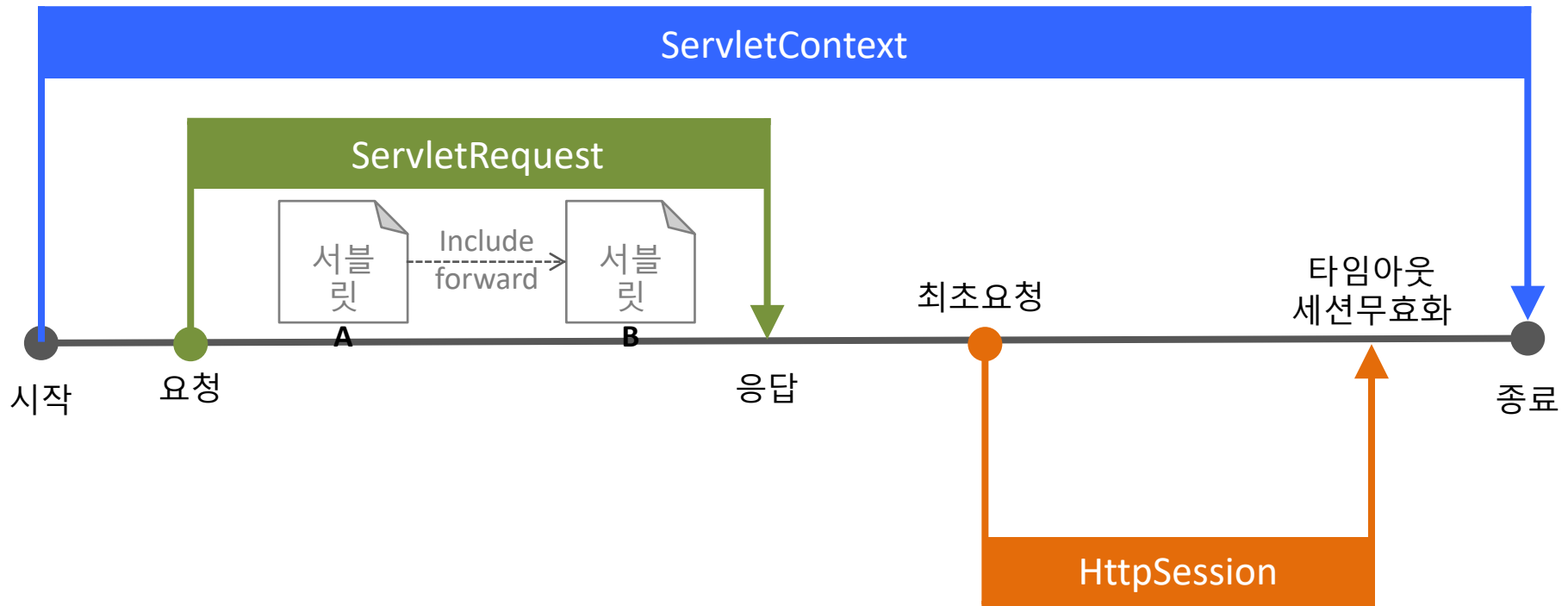
JspContext



웹 애플리케이션

5.6 데이터 보관소

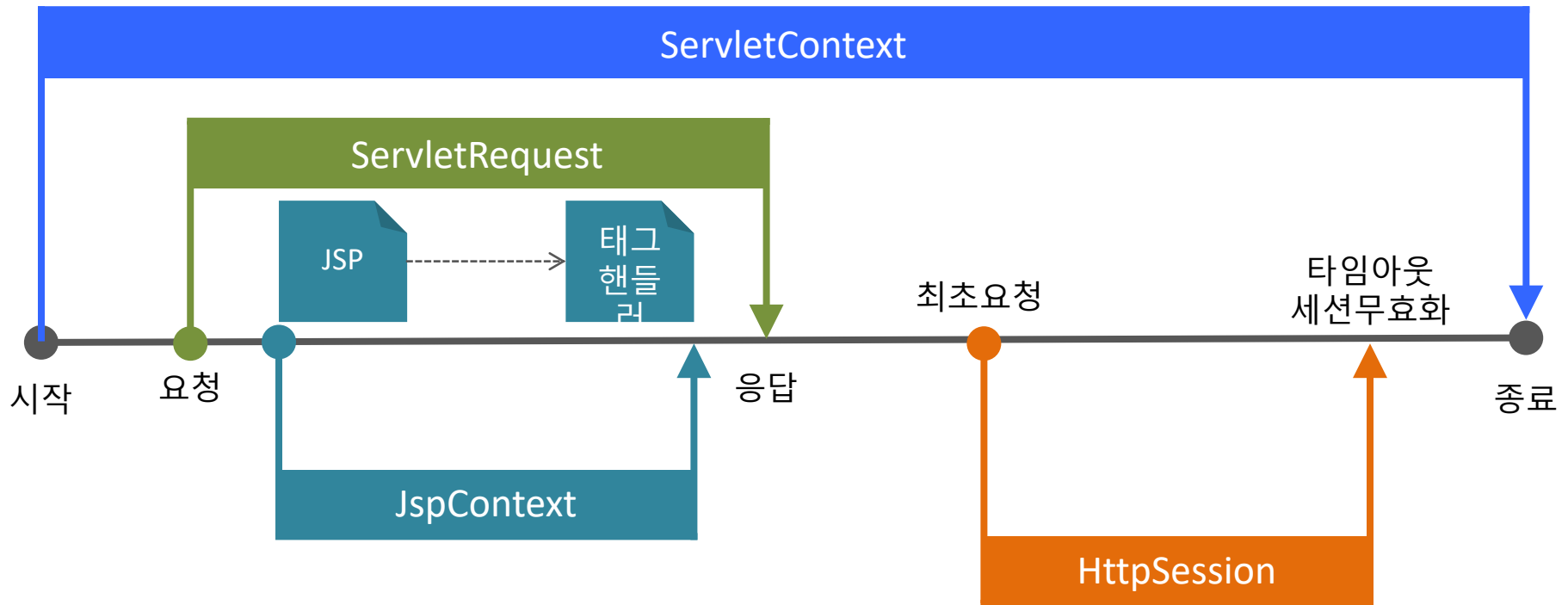
JspContext



웹 애플리케이션

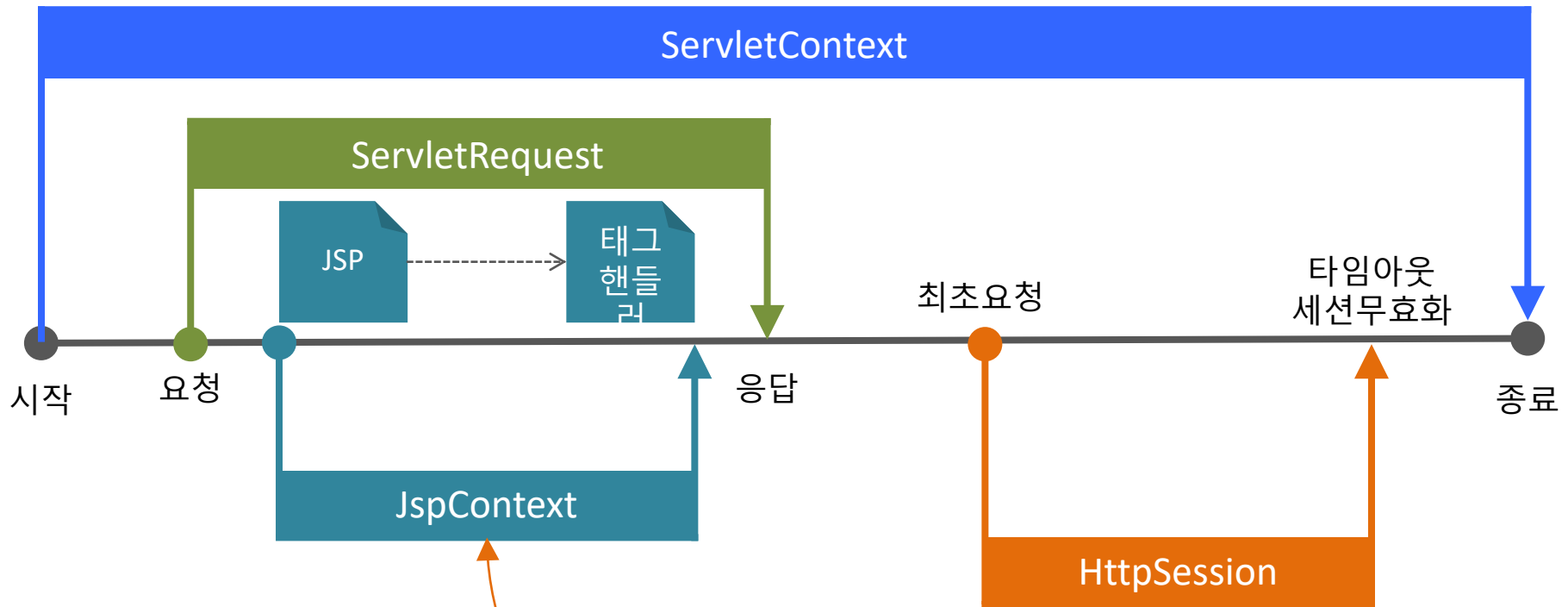
5.6 데이터 보관소

웹 애플리케이션



5.6 데이터 보관소

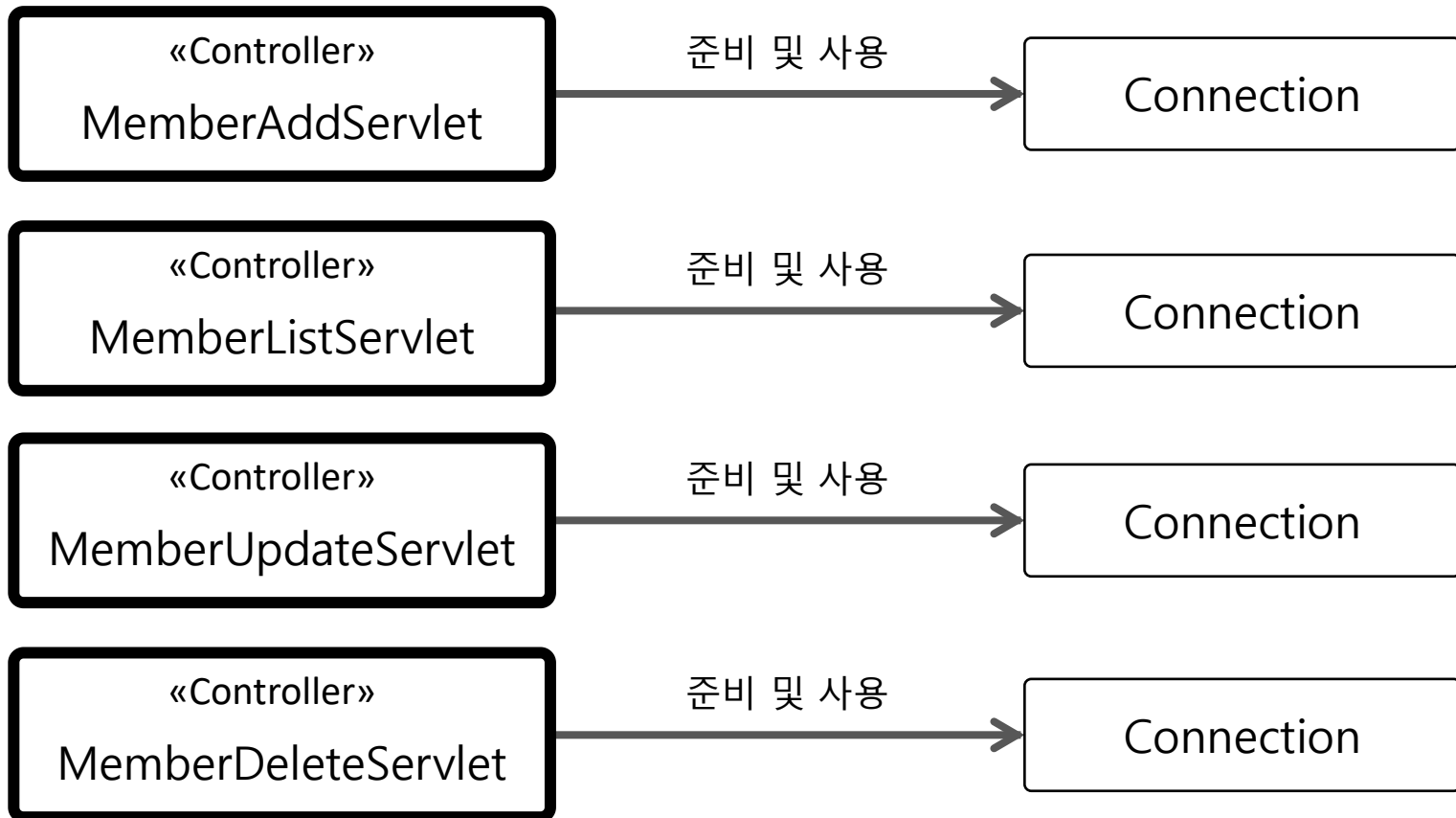
웹 애플리케이션



JSP 페이지를 실행하는 동안 존재함
태그 핸들러와 데이터를 공유할 때 사용

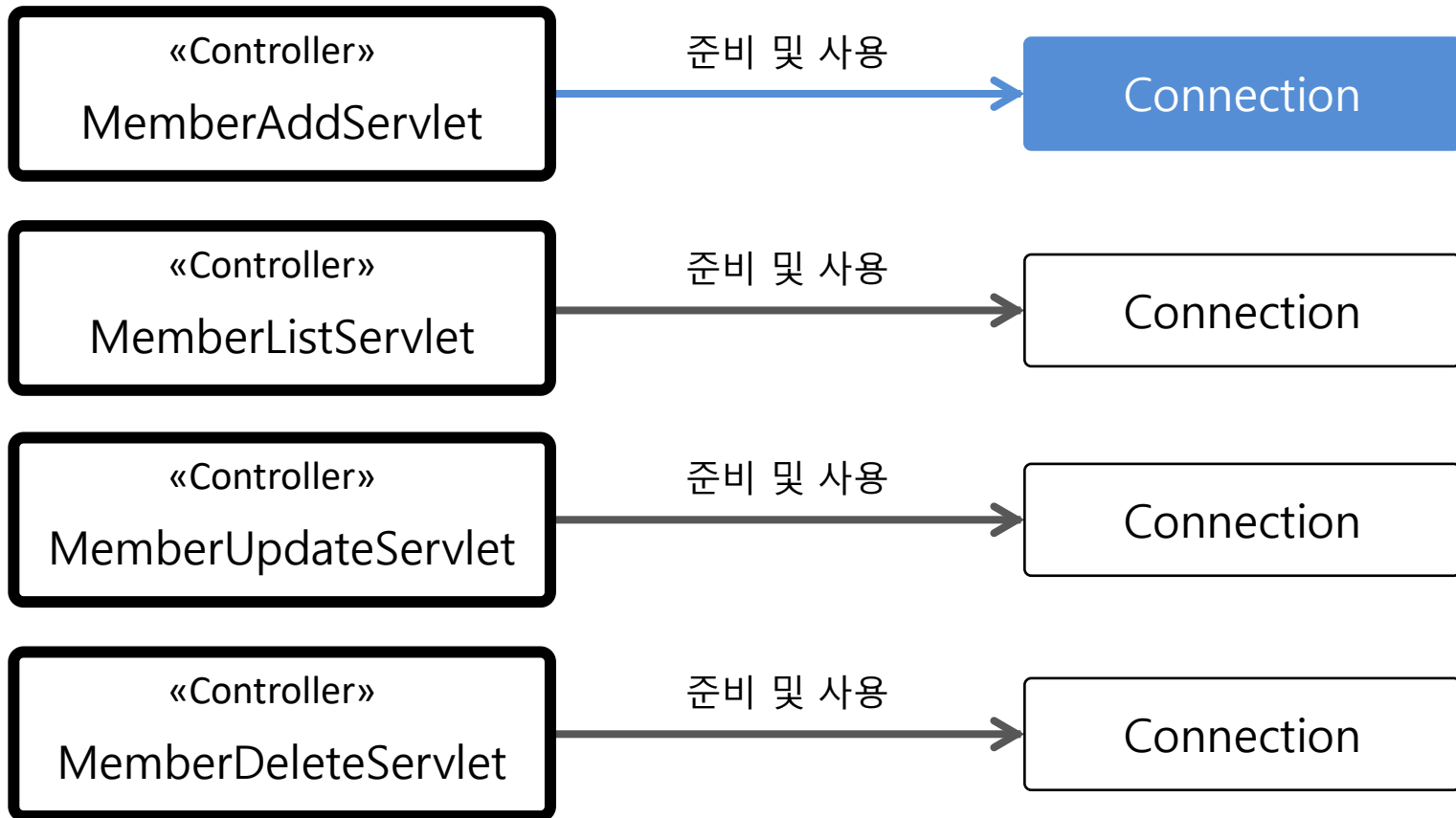
5.6 데이터 보관소

ServletContext 활용전



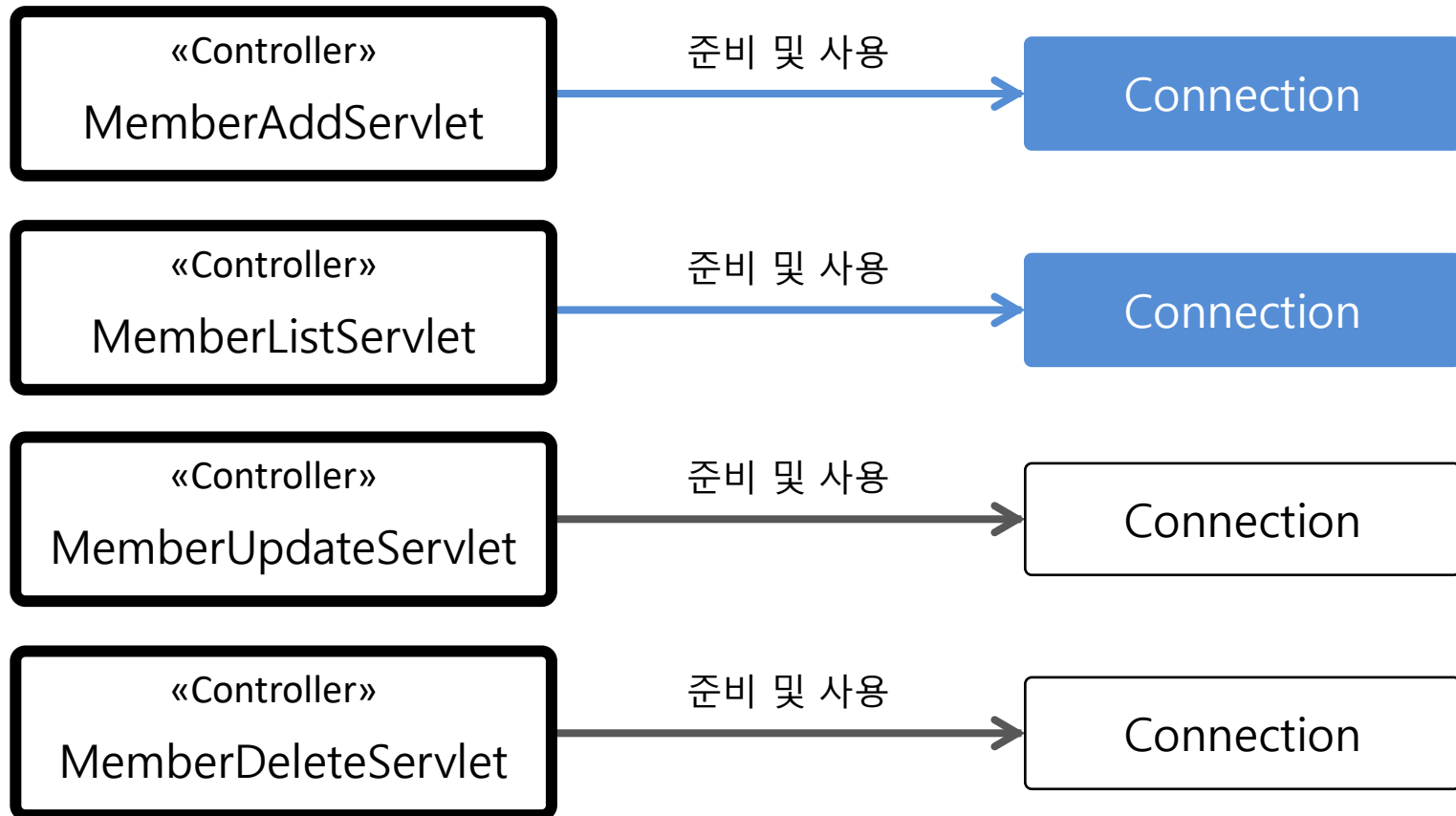
5.6 데이터 보관소

ServletContext 활용전



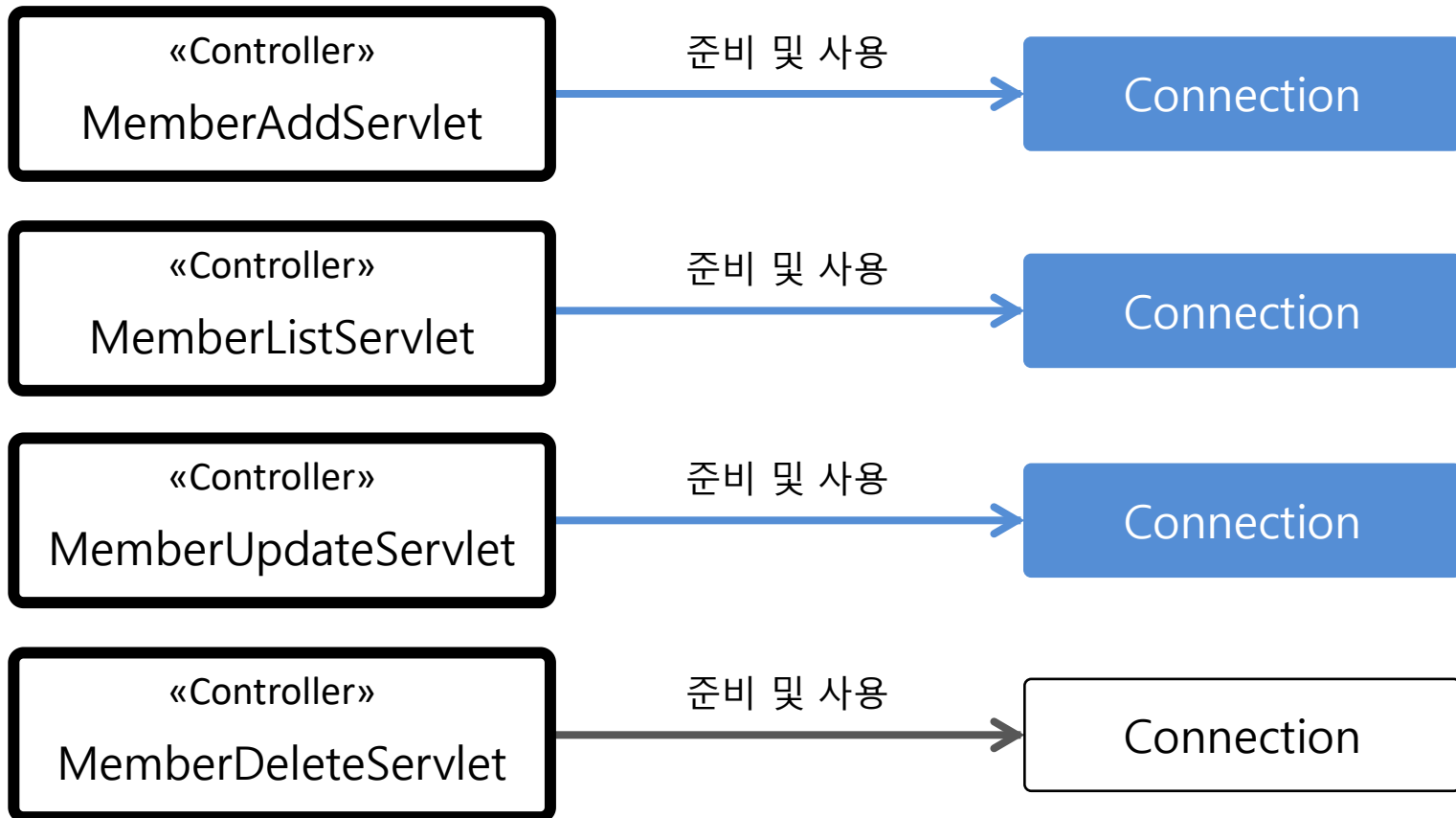
5.6 데이터 보관소

ServletContext 활용전



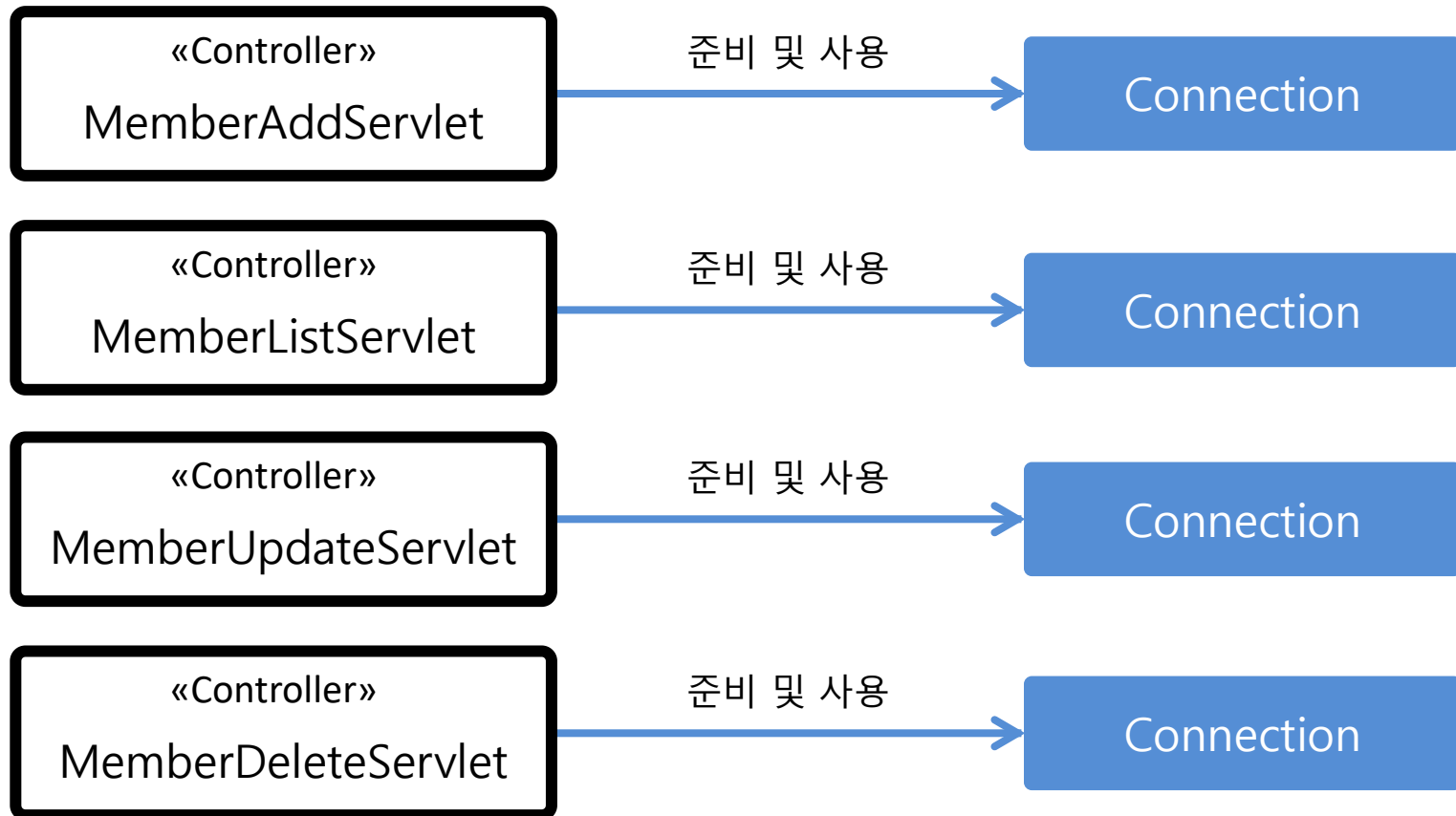
5.6 데이터 보관소

ServletContext 활용전



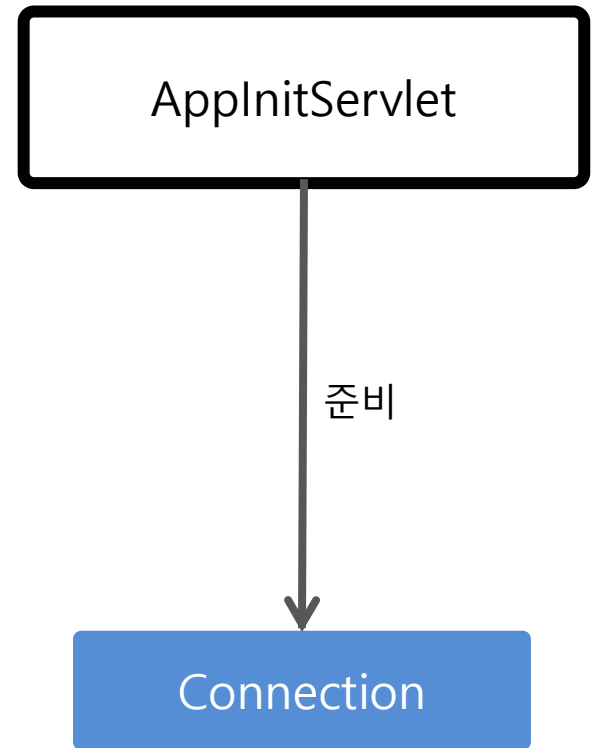
5.6 데이터 보관소

ServletContext 활용전

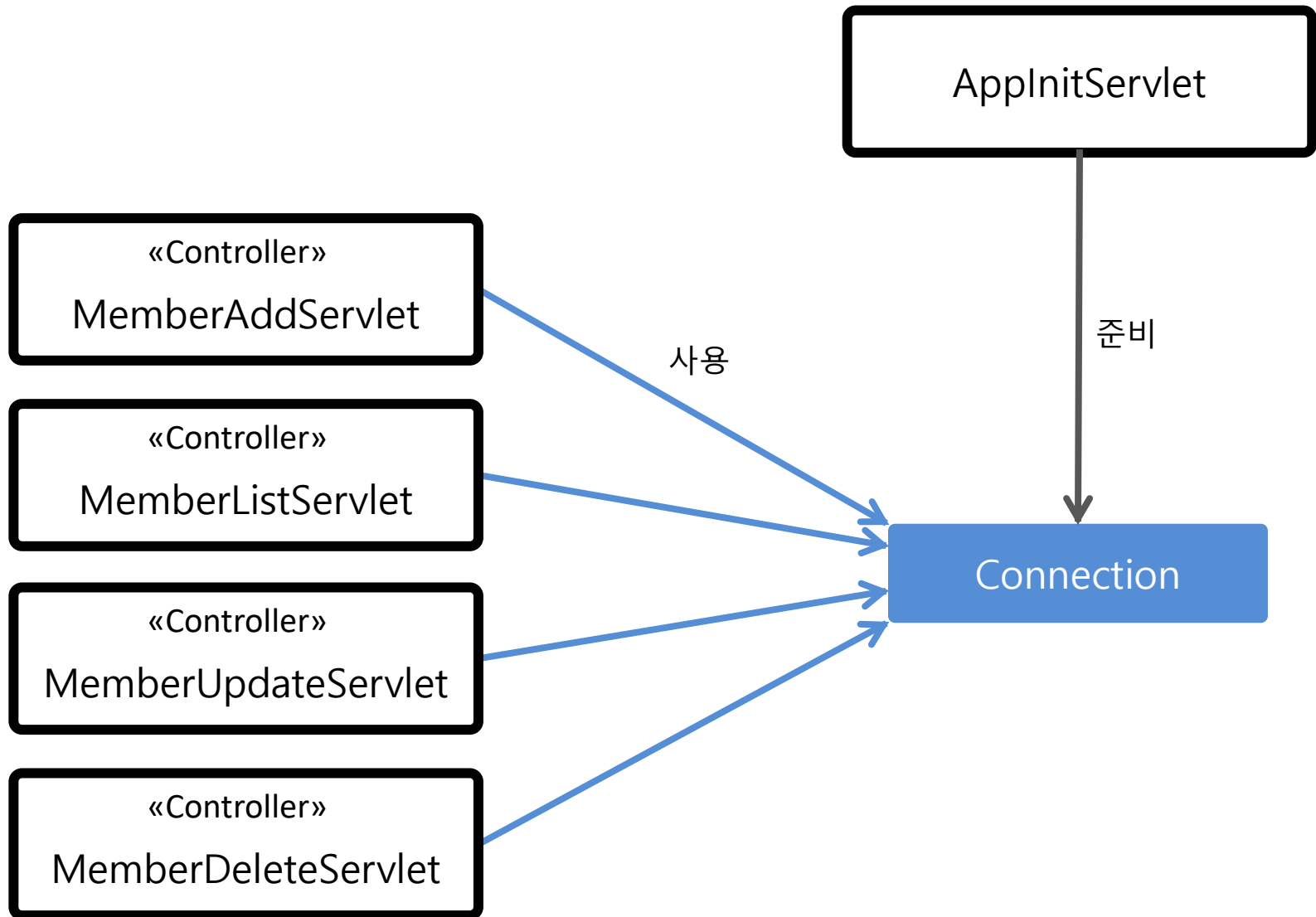


ServletContext **활용 후**

5.6 데이터 보관소

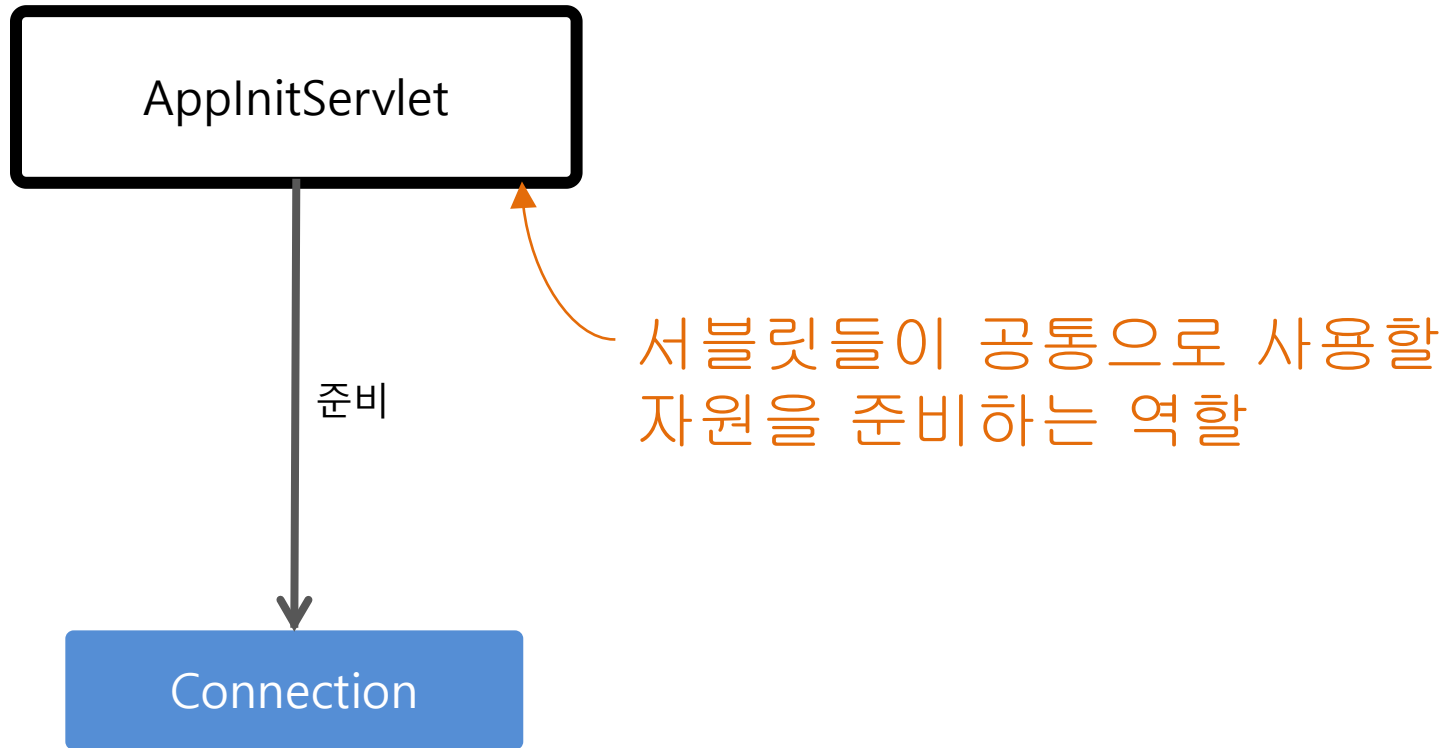


5.6 데이터 보관소

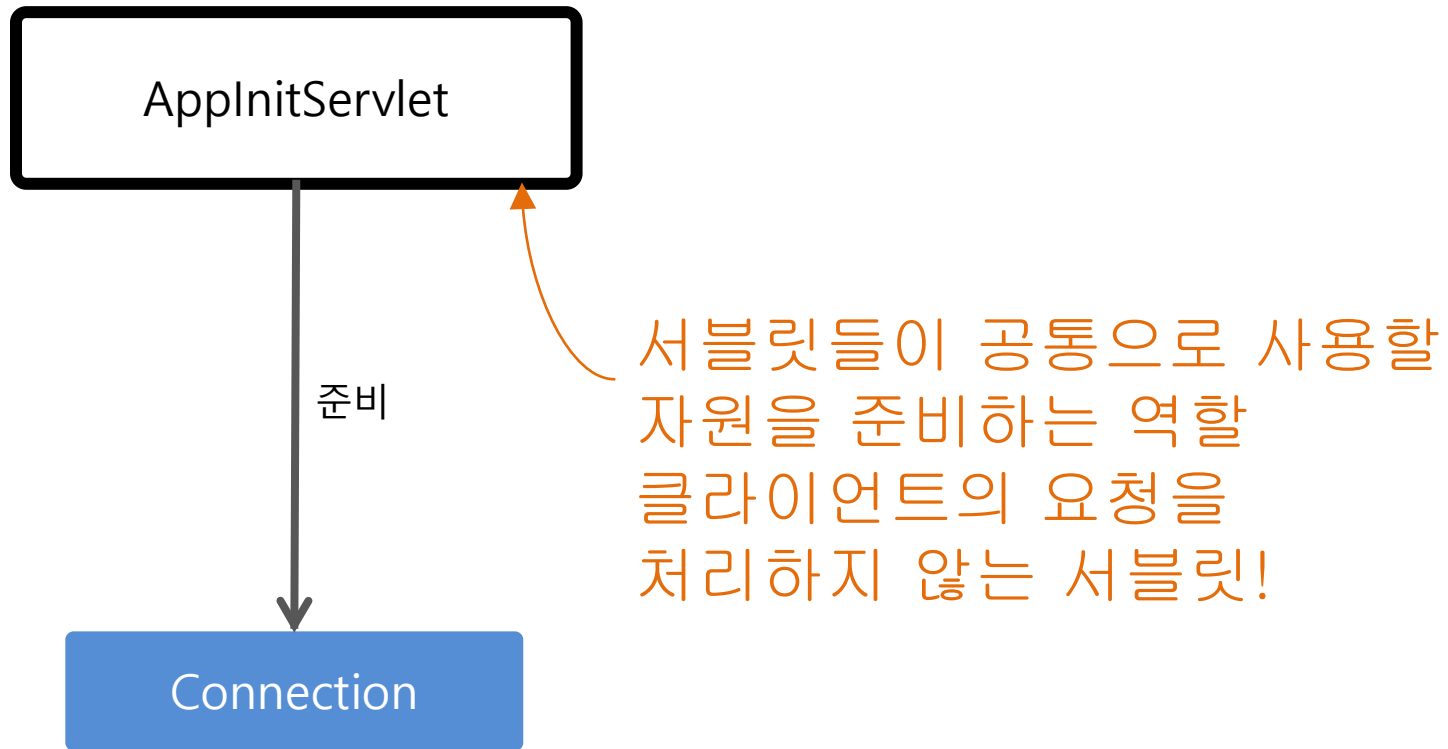


클라이언트 요청을 처리하지
않는 서블릿

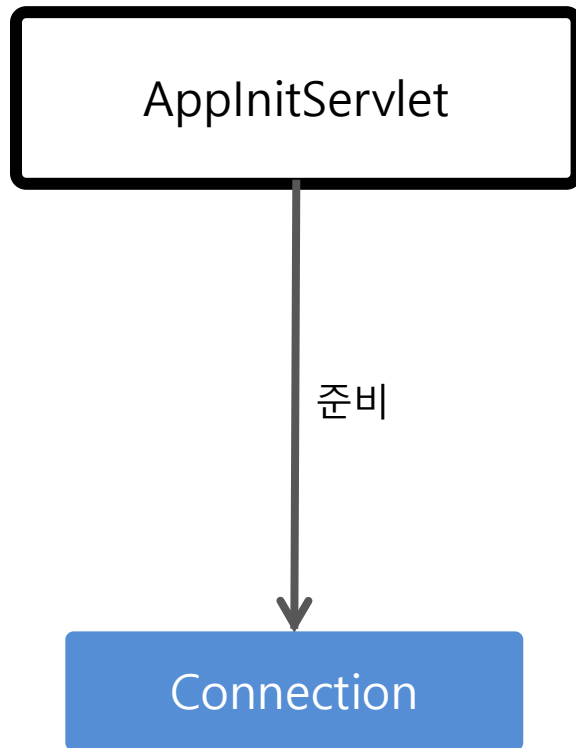
5.6 데이터 보관소



5.6 데이터 보관소



5.6 데이터 보관소



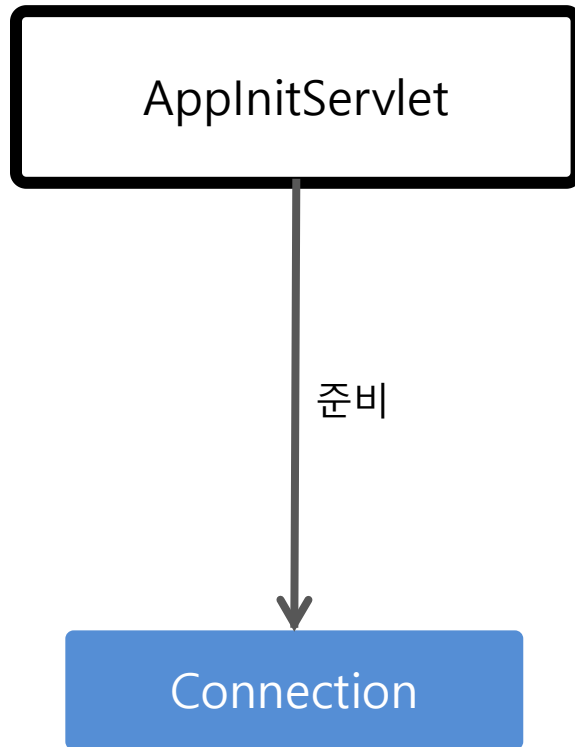
web.xml

```
<servlet>
  <servlet-name>ApplInitServlet</servlet-name>
  <servlet-class>
    spms.servlets.ApplInitServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  ...
</servlet-mapping>
```


5.6 데이터 보관소

web.xml



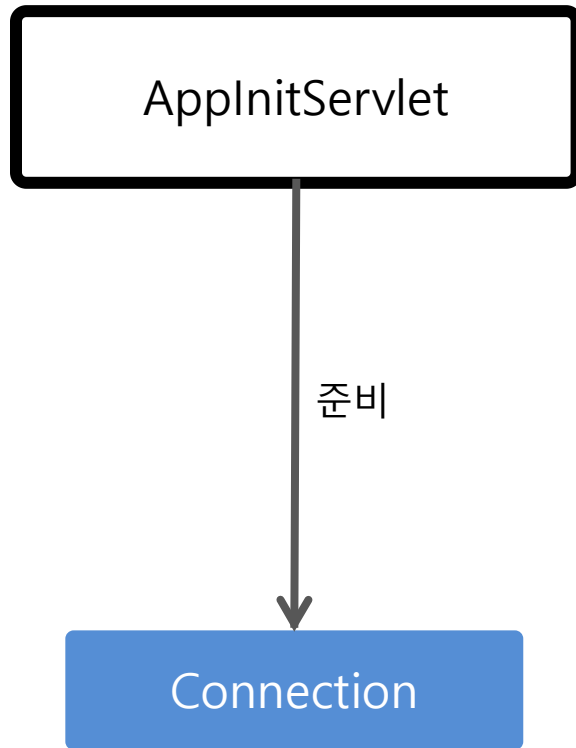
```
<servlet>
  <servlet-name>ApplInitServlet</servlet-name>
  <servlet-class>
    spms.servlets.ApplInitServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  ...
</servlet-mapping>
```

A large red 'X' is drawn over the `<servlet-mapping>` section of the XML code. An orange arrow points from the explanatory text below to the 'X'.

클라이언트의 요청을
처리하지 않기 때문에
URL을 매핑할 필요가 없음

5.6 데이터 보관소

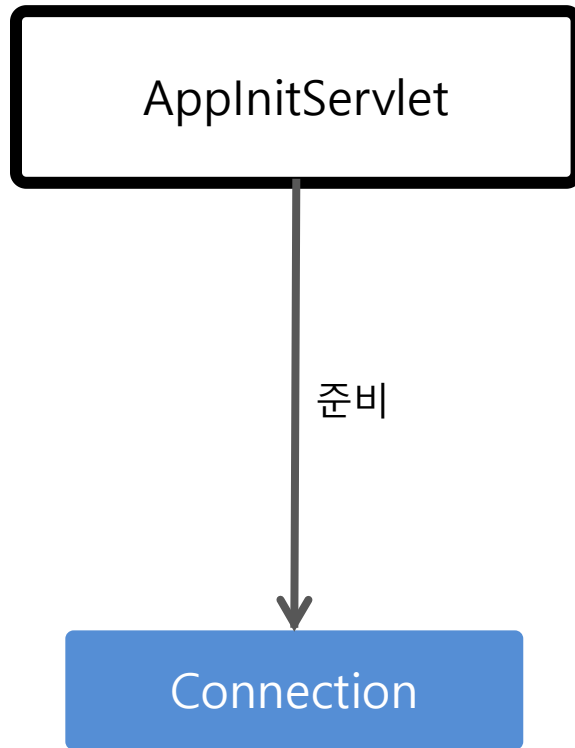


web.xml

```
<servlet>
  <servlet-name>ApplInitServlet</servlet-name>
  <servlet-class>
    spms.servlets.ApplInitServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

웹 애플리케이션 시작 시
서블릿 객체 생성 및 초기화

5.6 데이터 보관소

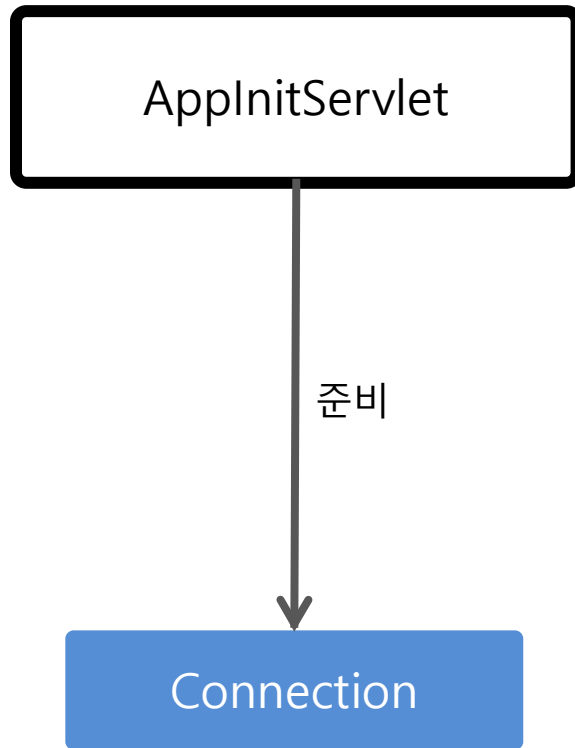


web.xml

```
<servlet>
  <servlet-name>ApplInitServlet</servlet-name>
  <servlet-class>
    spms.servlets.ApplInitServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

웹 애플리케이션 시작 시
서블릿 객체 생성 및 초기화
서블릿 객체 생성 → 생성자
호출 → `init()` 호출

5.6 데이터 보관소



web.xml

```
<servlet>
  <servlet-name>ApplInitServlet</servlet-name>
  <servlet-class>
    spms.servlets.ApplInitServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

서블릿 생성 및 초기화 순서
낮은 값부터 우선 생성됨

5.6 데이터 보관소

기존 서블릿 변경

5.6 데이터 보관소

«Controller»

MemberListServlet

```
public void doGet( ... ) ... {  
    Connection conn = null;  
    Statement stmt = null;  
    ResultSet rs = null;  
  
    try {  
        ServletContext sc = this.getServletContext();  
        Class.forName(sc.getInitParameter("driver"));  
        conn = DriverManager.getConnection(  
            sc.getInitParameter("url"),  
            sc.getInitParameter("username"),  
            sc.getInitParameter("password"));  
        stmt = conn.createStatement();  
        ...  
    }  
}
```

5.6 데이터 보관소

«Controller»

MemberListServlet

```
public void doGet( ... ) ... {  
    Connection conn = null;  
    Statement stmt = null;  
    ResultSet rs = null;  
  
    try {  
        ServletContext sc = this.getServletContext();  
        Class.forName(sc.getInitParameter("driver"));  
        conn = DriverManager.getConnection(  
            sc.getInitParameter("url"),  
            sc.getInitParameter("username"),  
            sc.getInitParameter("password"));  
        stmt = conn.createStatement();  
        ...  
    }  
}
```

5.6 데이터 보관소

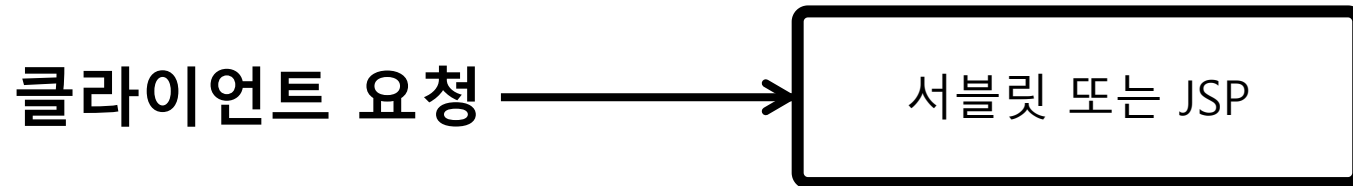
«Controller»

MemberListServlet

```
public void doGet( ... ) ... {  
    Connection conn = null;  
    Statement stmt = null;  
    ResultSet rs = null;  
  
    try {  
        ServletContext sc = this.getServletContext();  
        conn = (Connection) sc.getAttribute("conn");  
        stmt = conn.createStatement();  
        ...  
    }  
}
```

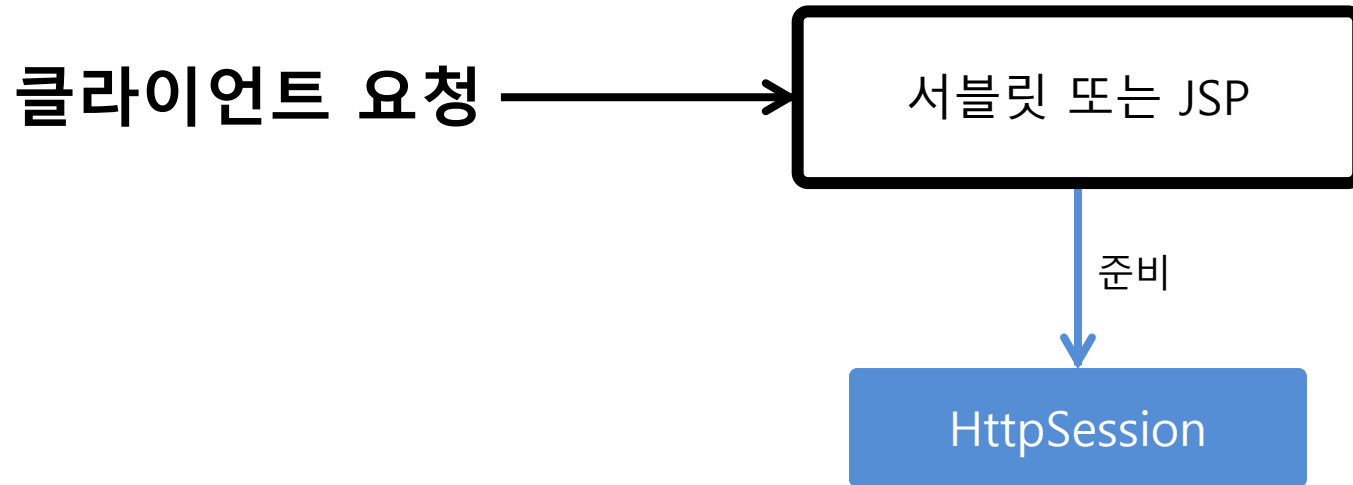

5.6 데이터 보관소

1) HttpSession 준비



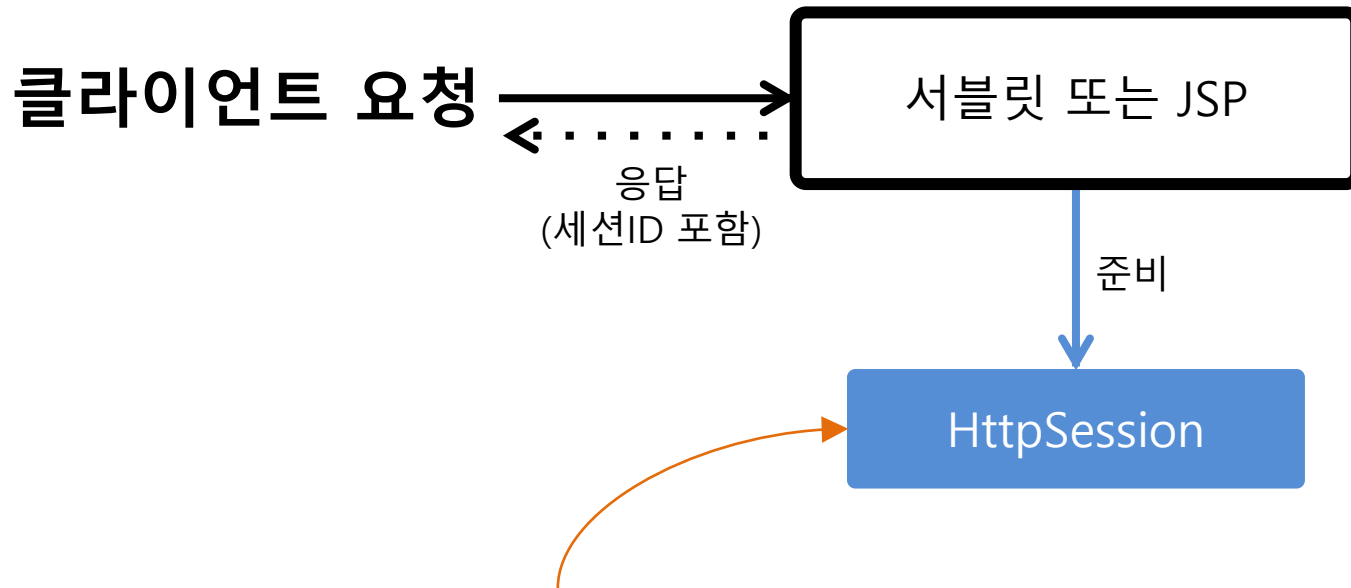
5.6 데이터 보관소

1) HttpSession 준비



5.6 데이터 보관소

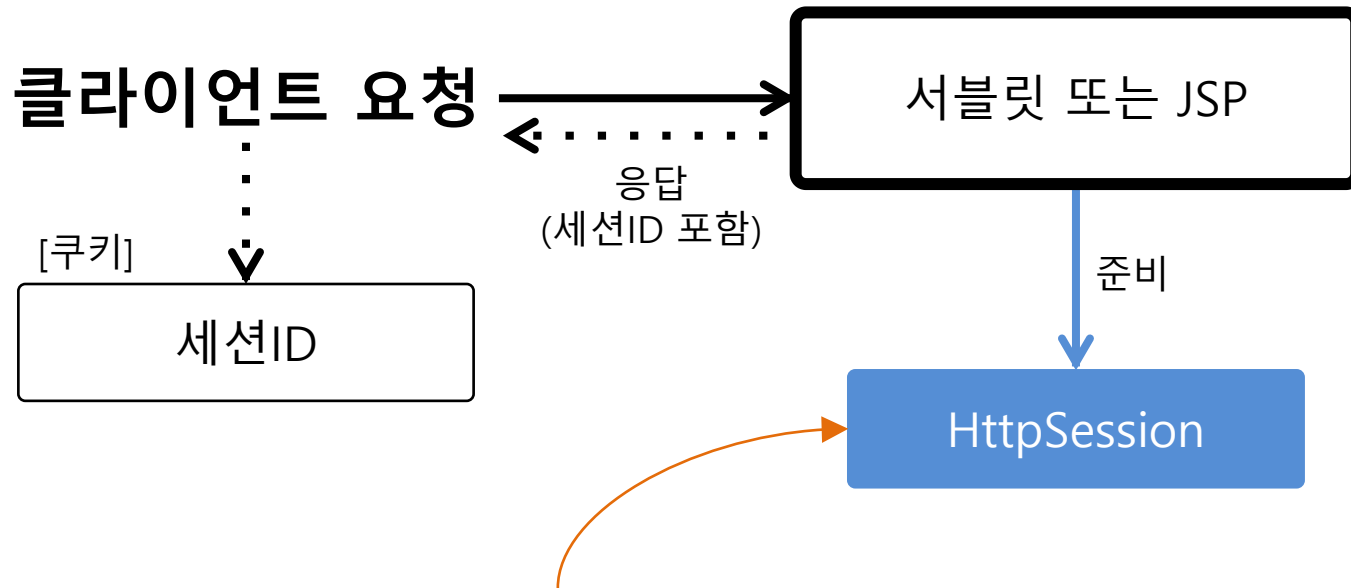
1) HttpSession 준비



i) 서블릿 컨테이너는 세션 식별자를 쿠키에 담아 클라이언트에 보냄

5.6 데이터 보관소

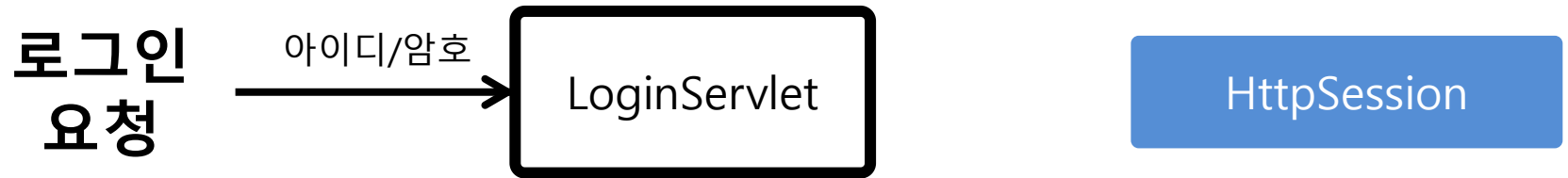
1) HttpSession 준비



- i) 서블릿 컨테이너는 세션 식별자를 쿠키에 담아 클라이언트에 보냄
- ii) 클라이언트는 세션 식별자를 메모리에 보관함
- iii) 이후 요청부터는 서버에 요청할 때마다 세션 식별자를 보냄

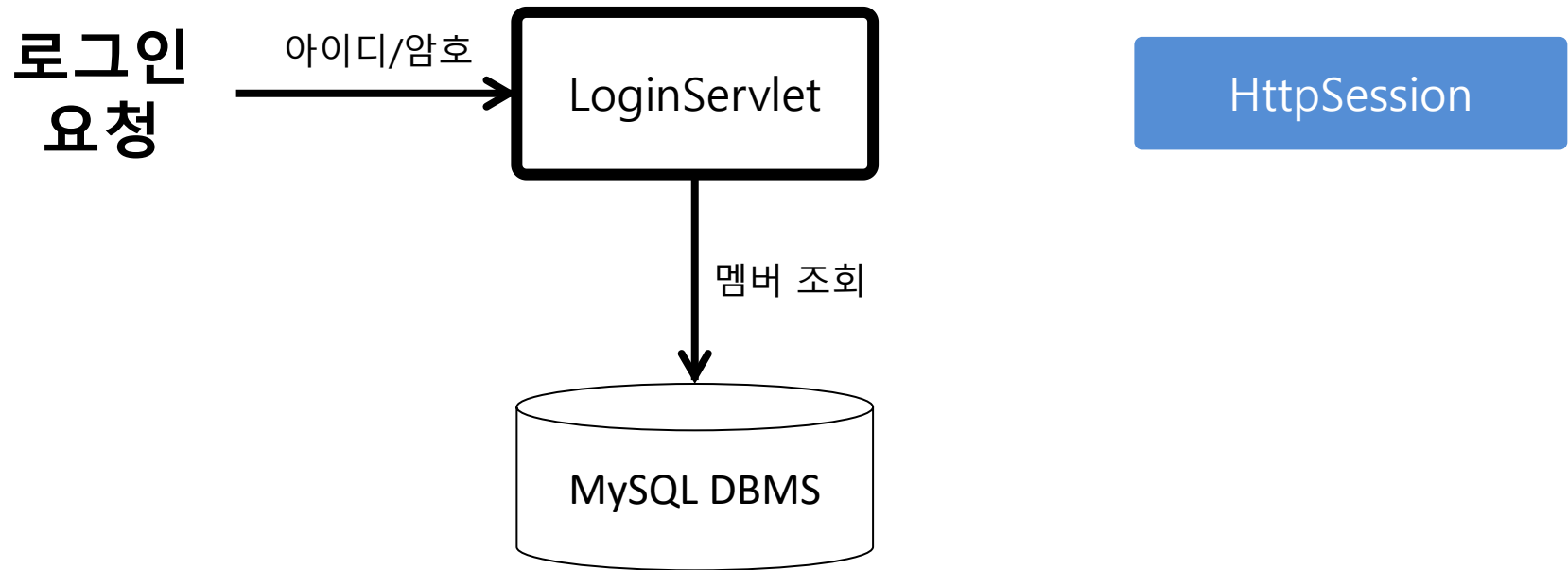
5.6 데이터 보관소

2) HttpSession에 데이터 보관



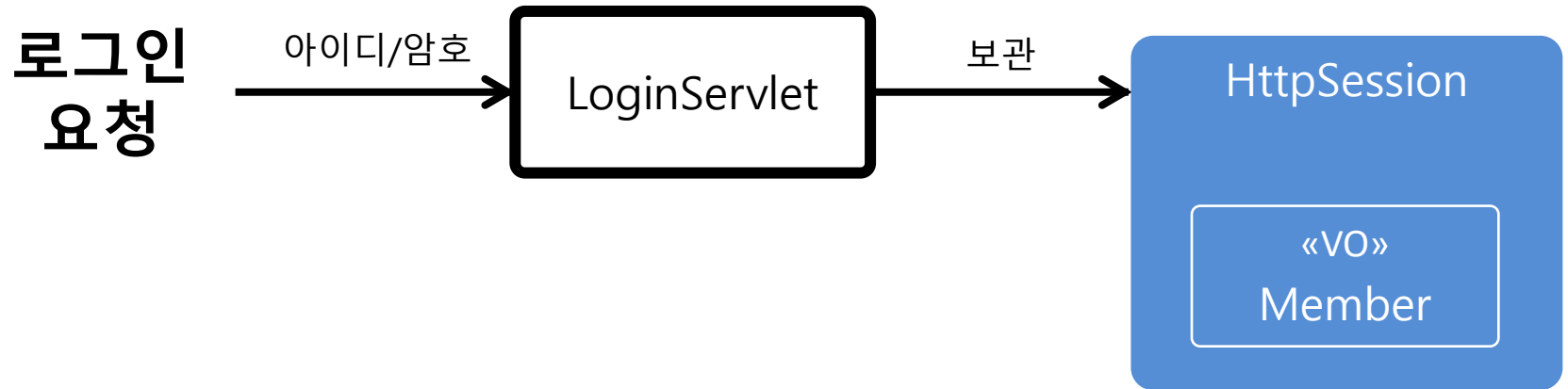
5.6 데이터 보관소

2) HttpSession에 데이터 보관



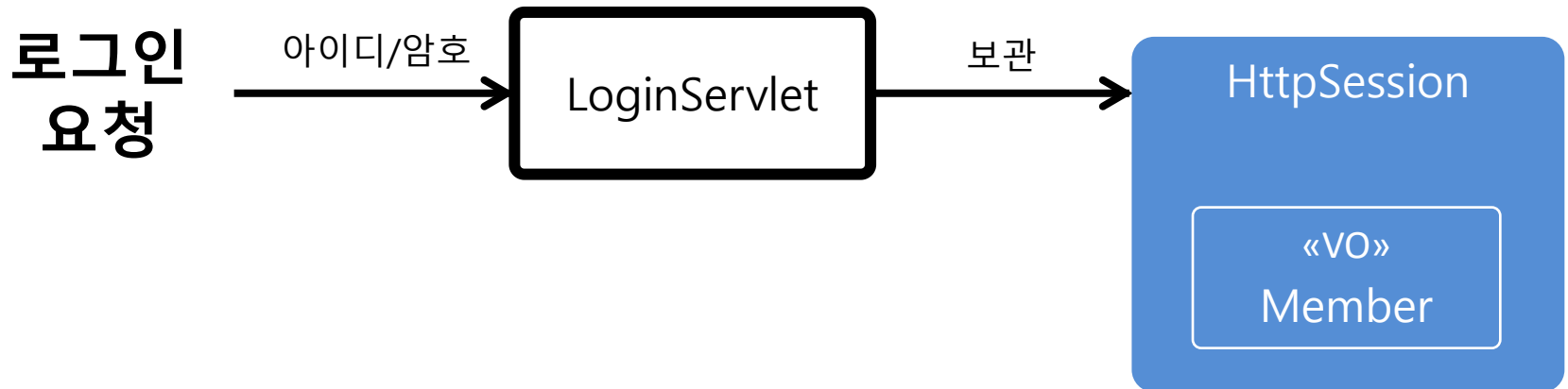
5.6 데이터 보관소

2) HttpSession에 데이터 보관



5.6 데이터 보관소

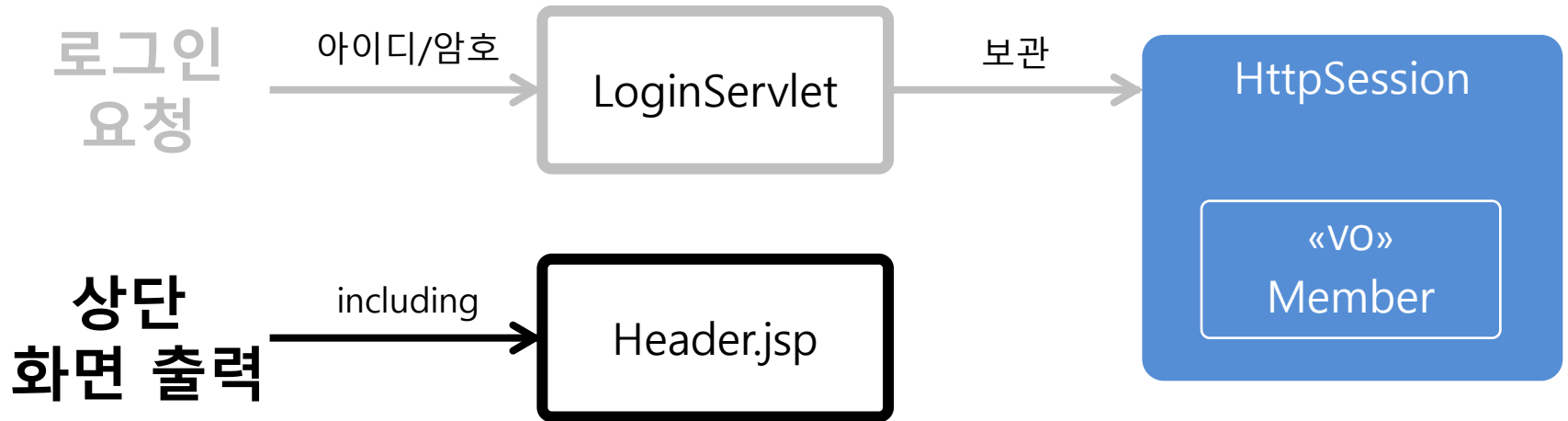
2) HttpSession에 데이터 보관



```
Member member = new Member()  
                .setEmail(rs.getString("EMAIL"))  
                .setName(rs.getString("MNAME"));  
HttpSession session = request.getSession();  
session.setAttribute("member", member);
```

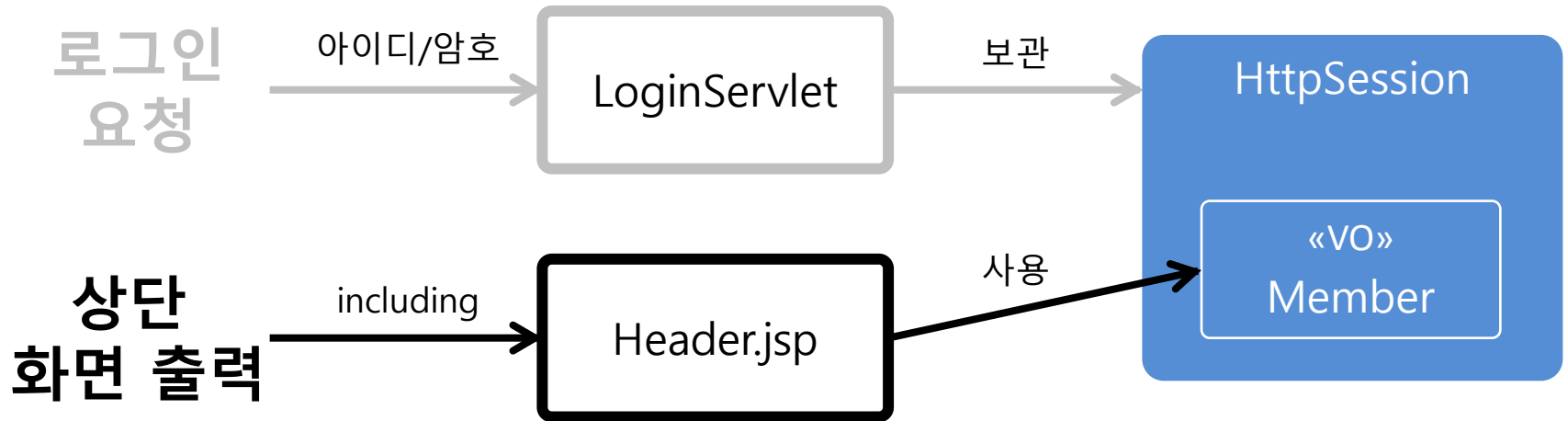

5.6 데이터 보관소

3) HttpSession에 보관된 데이터 사용



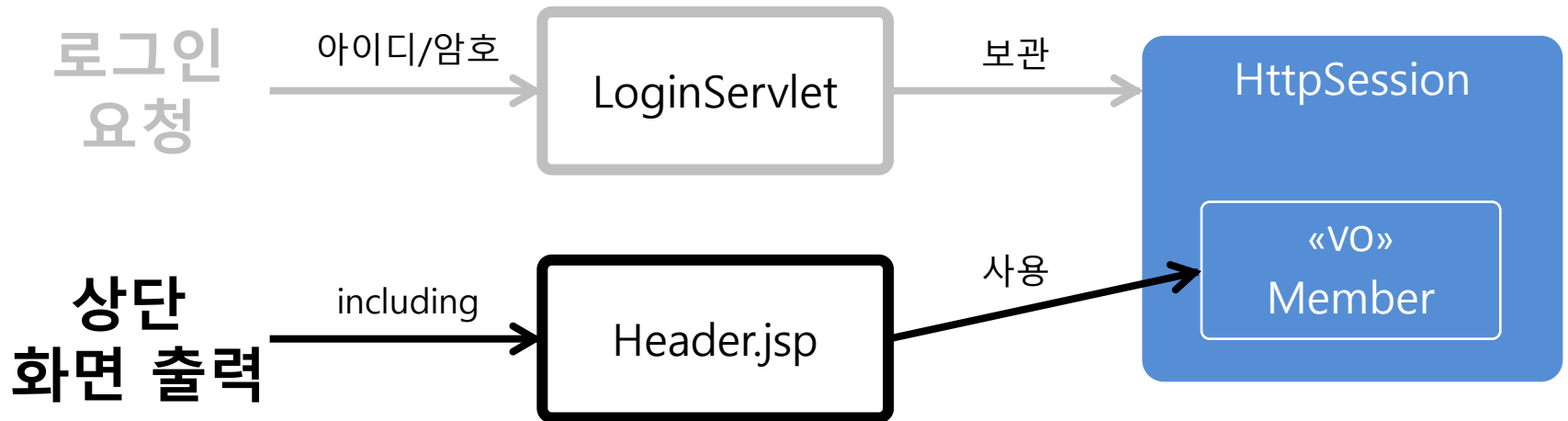
5.6 데이터 보관소

3) HttpSession에 보관된 데이터 사용



5.6 데이터 보관소

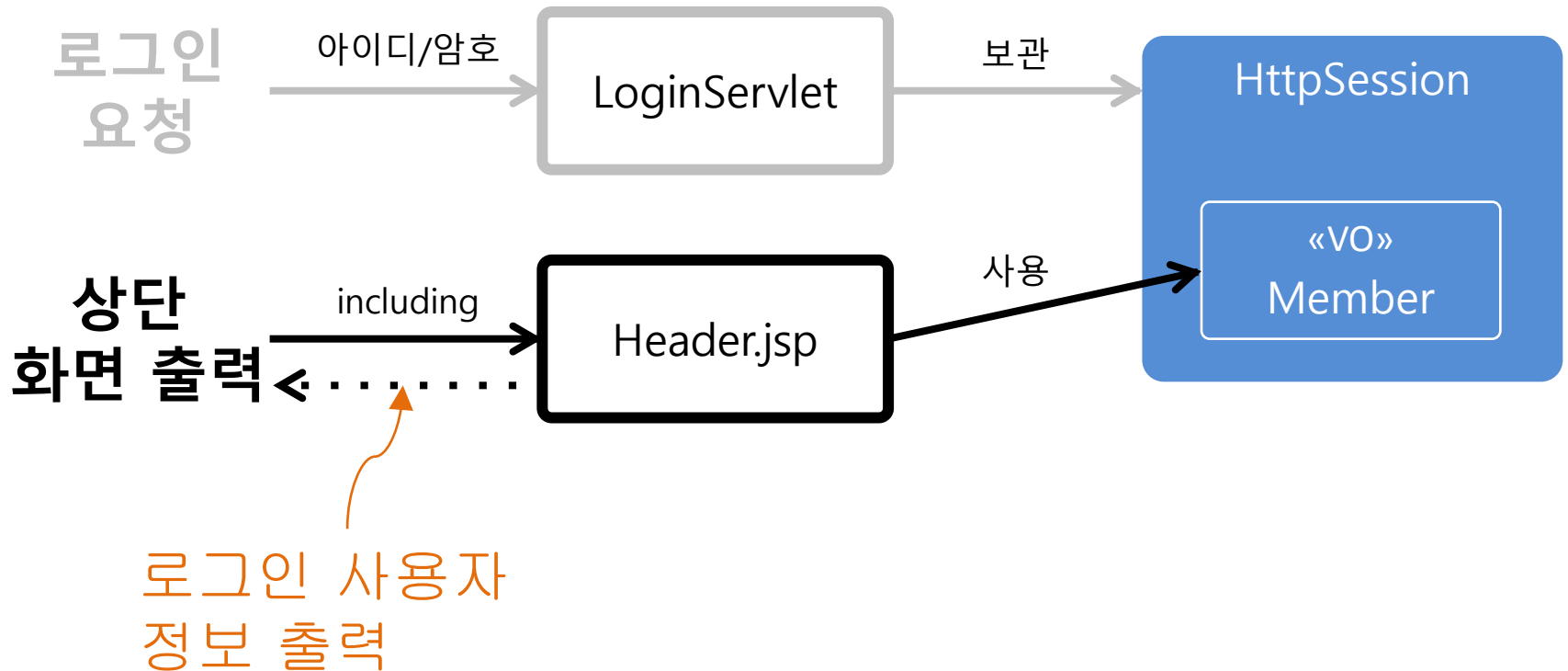
3) HttpSession에 보관된 데이터 사용



```
<%  
Member member = (Member)session.getAttribute("member");  
%>
```

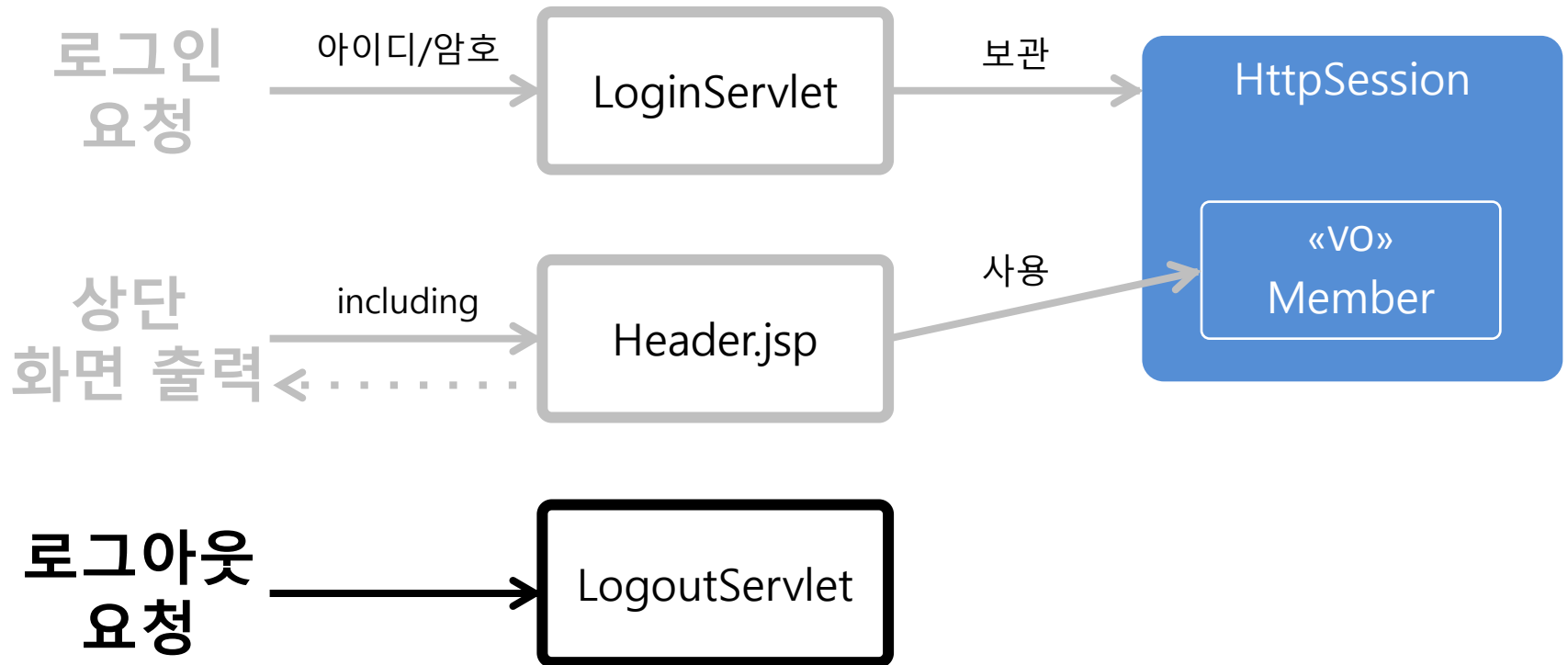
5.6 데이터 보관소

3) HttpSession에 보관된 데이터 사용



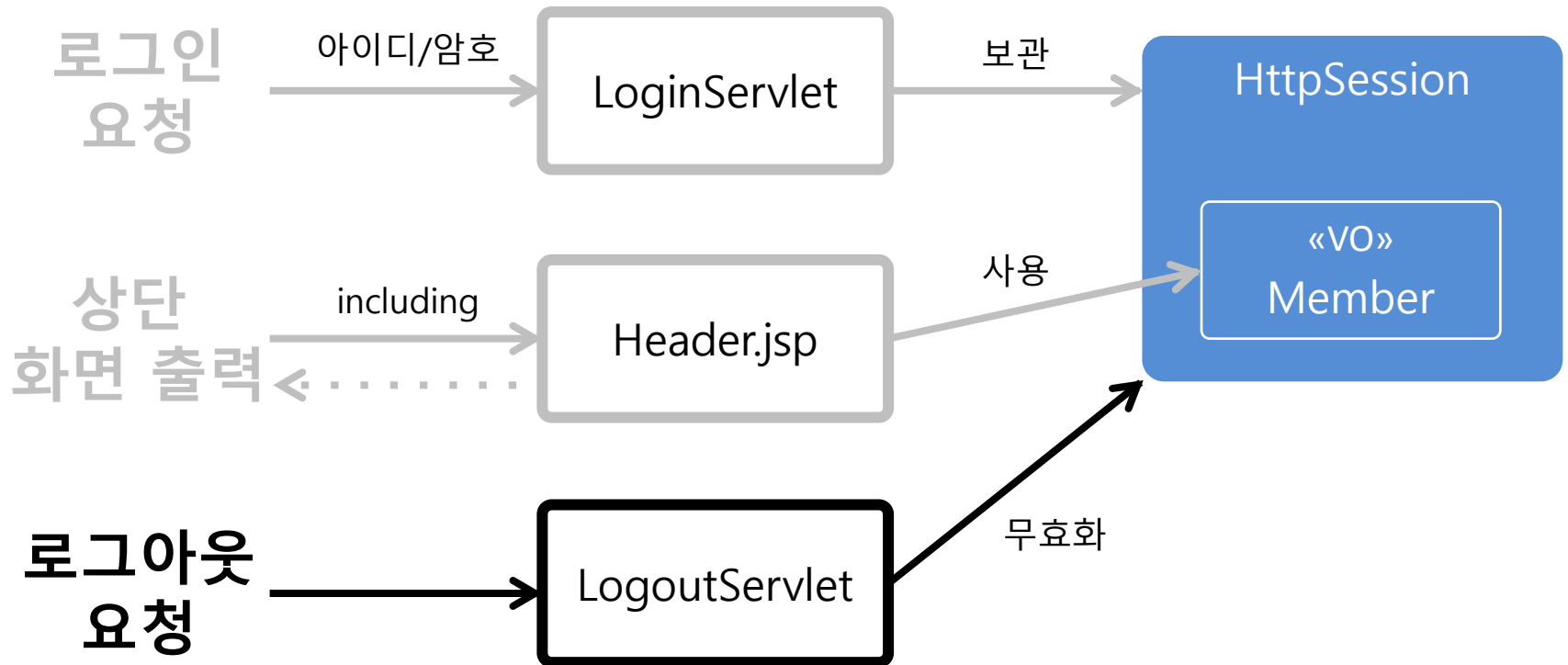
5.6 데이터 보관소

4) HttpSession 무효화



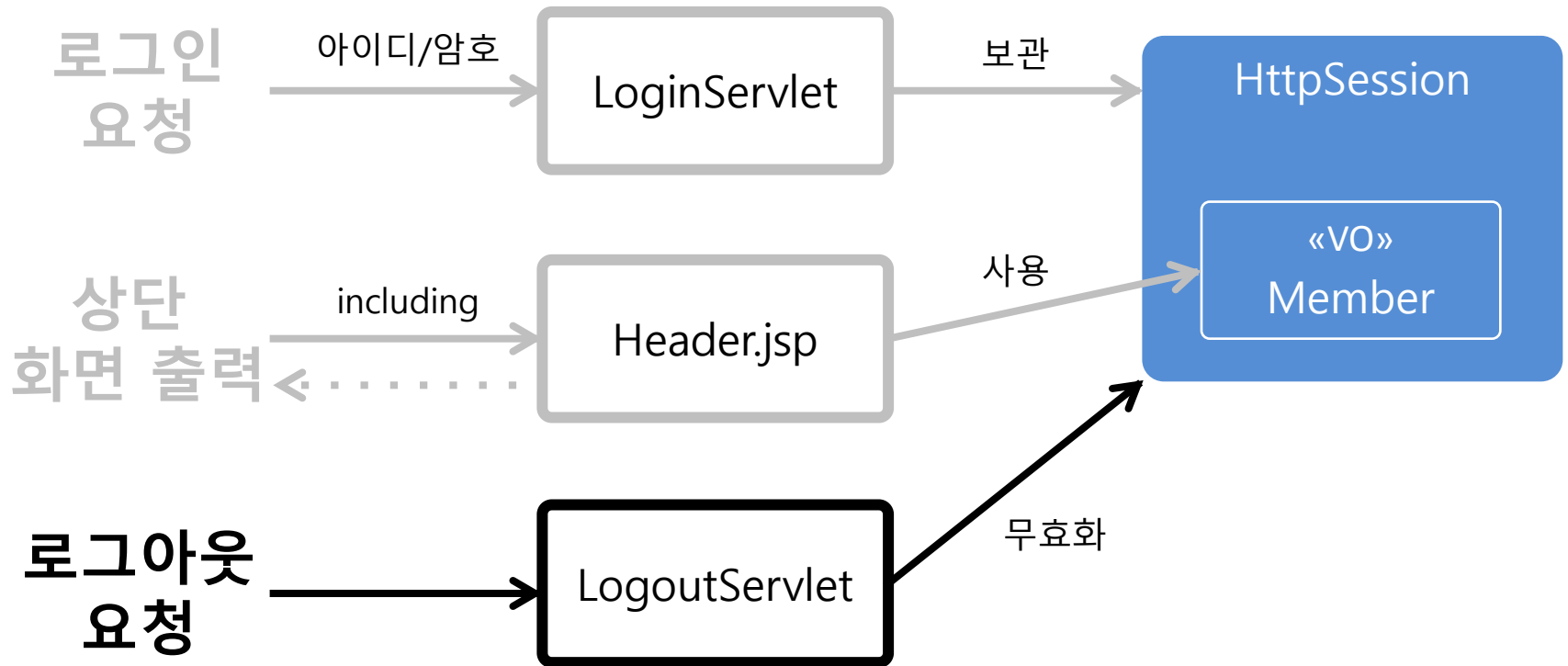
5.6 데이터 보관소

4) HttpSession 무효화



5.6 데이터 보관소

4) HttpSession 무효화



```
HttpSession session = request.getSession();  
session.invalidate();
```

5.6 데이터 보관소

4) HttpSession 무효화

