Intercepting HTTP Request with Filters
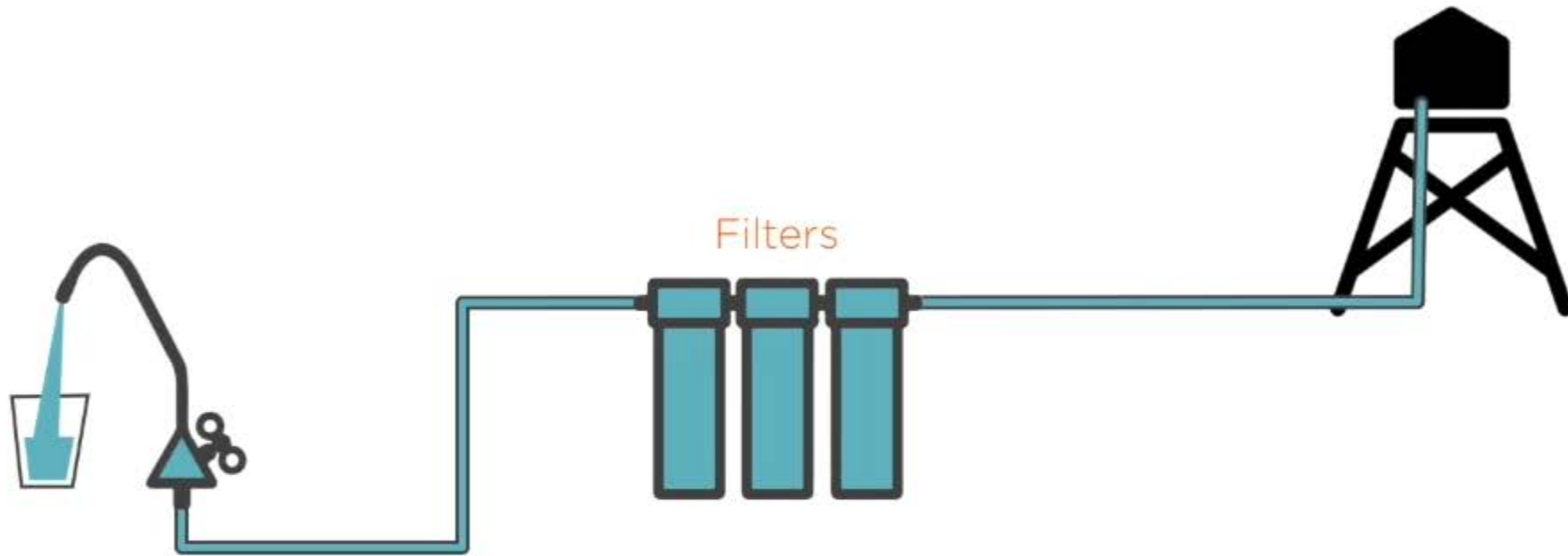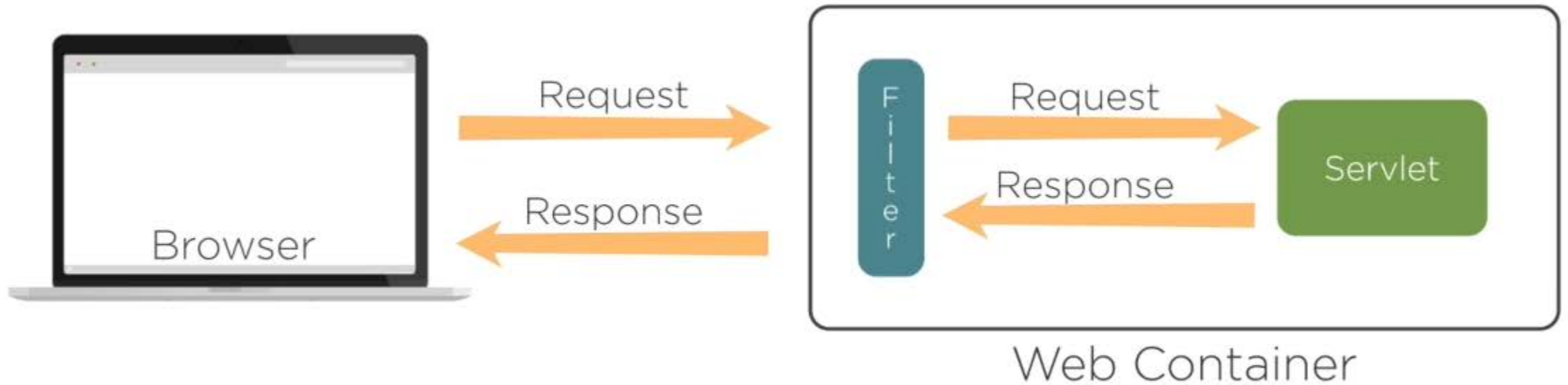
Sekhar Srinivasan
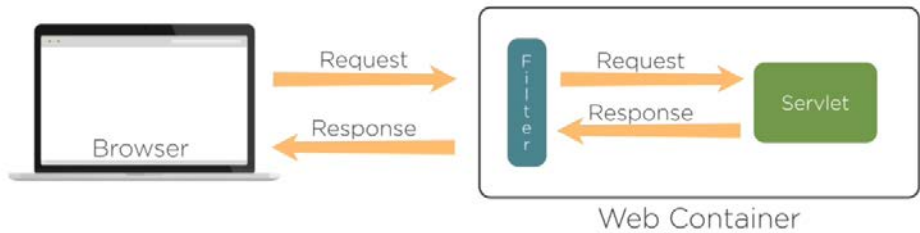
@sekharonline4u - www.sekhartheguru.net

What Are Filters?
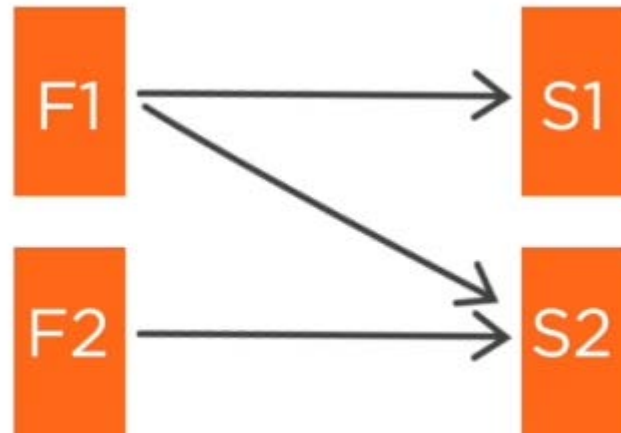

Filters

Servlets Filters

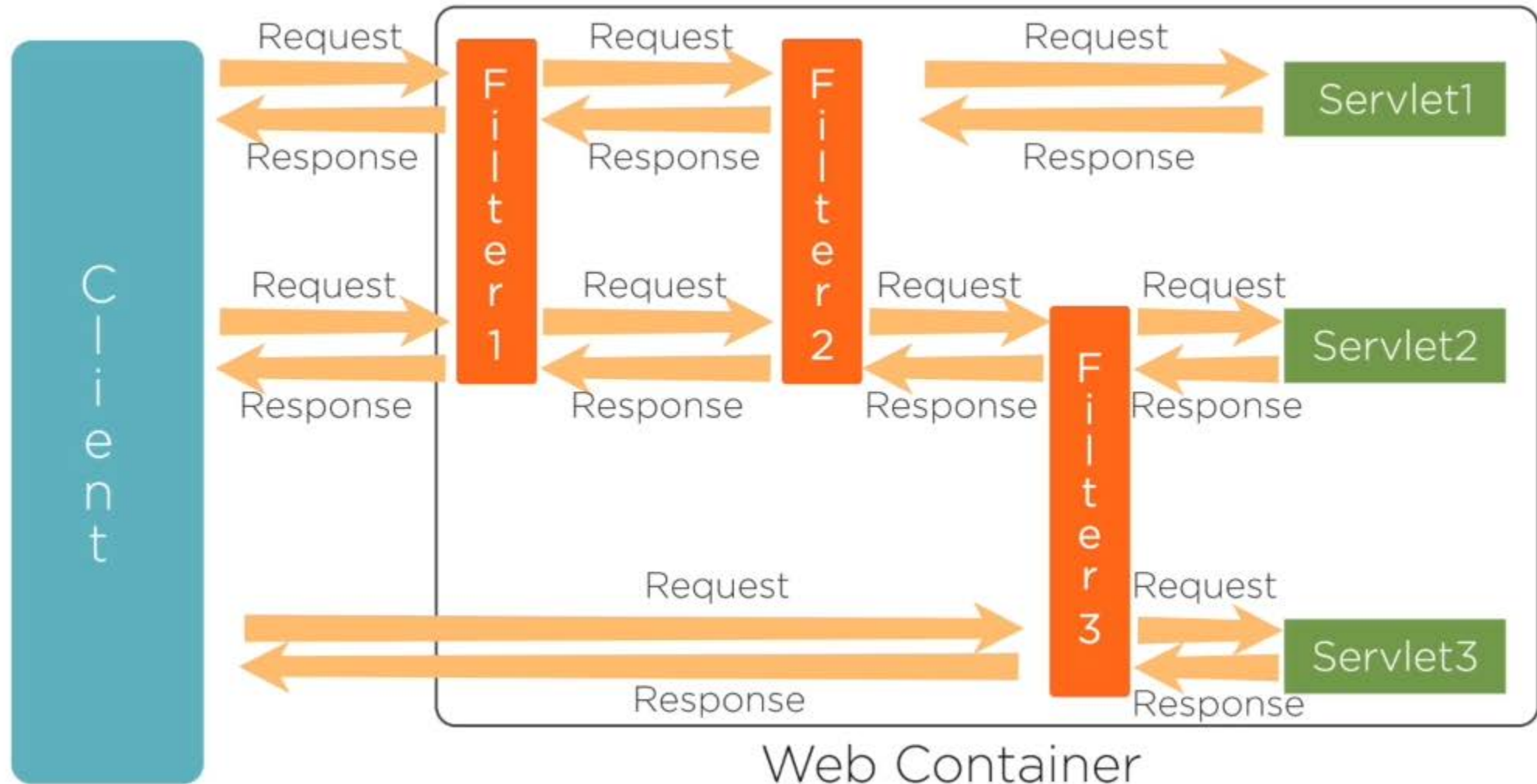Introduced from Servlets 2.3 Specification -Part of J2EE 2.3 Specification

Replaces Servlet Chaining or Chaining Servlets

What Are Servlet Filters?

Servlet filter is a small amount of code that executes before or after serving up a web resource

Servlet Filter

Why We Need a Servlet Filter?

Perform application task

Ability to reuse without rebuilding the application

Applied by configuration

Using the filter interface we can implement a task

They can be chained together

Initialization parameters

Have access to ServletRequest

Have access to ServletContext

Use of Servlet Filters

Record all the incoming requests

Log the IP addresses of the computers

Conversions

Data compressions

Encryption and Decryption

Input validations

Authentication and Authorization

Audit access to sensitive resources

Email system administrators on every application

To make the application perform better

Servlet Filter Life Cycle

```
public class ServletFilter implements javax.servlet Filter

{


}
```

```java
public class ServletFilter implements javax.servlet.Filter

{

    public void init(FilterConfig fConfig) throws ServletException

    {

    }

    public void doFilter(ServletRequest request, ServletResponse response,

        FilterChain chain) throws IOException, ServletException

    {

    }

    protect void destroy()

    {

    }

}
```

```java
public class ServletFilter implements javax.servlet.Filter

{

    public void init(FilterConfig fConfig) throws ServletException

    {

    }
    public void doFilter(ServletRequest request, ServletResponse response,
        FilterChain chain) throws IOException, ServletException
    {

    }
    protect void destroy()

    {

    }

}
```

```java
public class ServletFilter implements javax.servlet.Filter

{

    public void init(FilterConfig fConfig) throws ServletException

    {

    }

    public void doFilter(ServletRequest request, ServletResponse response,

        FilterChain chain) throws IOException, ServletException

    {

    }

    protect void destroy()

    {

    }

}
```

```java
public class ServletFilter implements javax.servlet.Filter

{

    public void init(FilterConfig fConfig) throws ServletException

    {

    }

    public void doFilter(ServletRequest request, ServletResponse response,

        FilterChain chain) throws IOException, ServletException

    {

    }

    protect void destroy()

    {

    }

}
```

```java
public class ServletFilter implements javax.servlet.Filter

{

    public void init(FilterConfig fConfig) throws ServletException

    {

    }

    public void doFilter(ServletRequest request, ServletResponse response,

        FilterChain chain) throws IOException, ServletException

    {

    }

    protect void destroy()

    {

    }

}
```

```java
public class ServletFilter implements javax.servlet.Filter


{



    public void init(FilterConfig fConfig) throws ServletException

    {

    }

    public void doFilter(ServletRequest request, ServletResponse response,

        FilterChain chain) throws IOException, ServletException
    {

    }

    protect void destroy()

    {

    }


}
```

```java
public class ServletFilter implements javax.servlet.Filter

{


    public void init(FilterConfig fConfig) throws ServletException

    {

    }

    public void doFilter(ServletRequest request, ServletResponse response,

        FilterChain chain) throws IOException, ServletException

    {

    }

    protect void destroy()

    {

    }

}
```

How a Filter Works

The web deployment descriptor(web.xml) declares that a filter should be invoked for a servlet

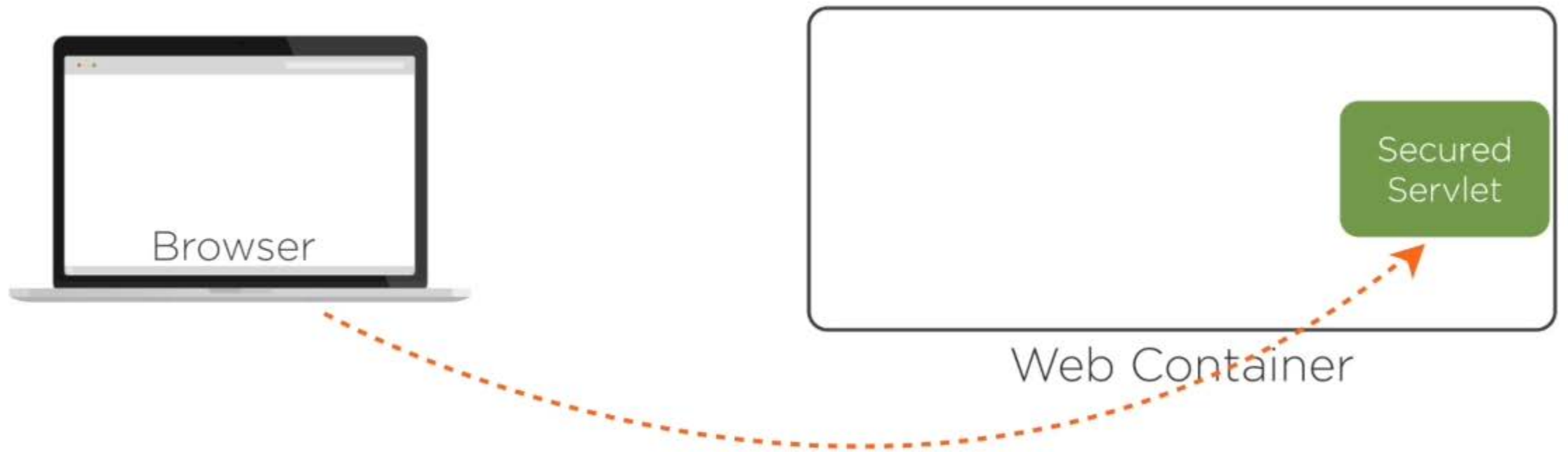Web.xml applies all the filters associated with servlet

For each filter the web container invokes
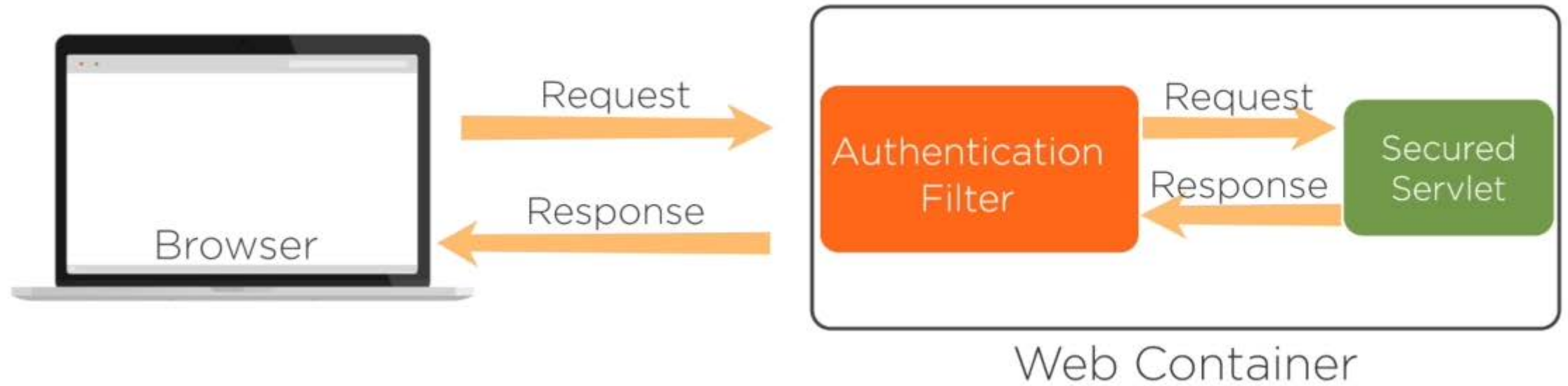
init(filterConfig)

doFilter(request,response,chain)

destroy()

The filter class should call chain.doFilter()

Demo: Servlets Filter Methods

Demo: Servlets Filter Methods

Filter Mapping in web.xml

```xml
<filter>

    <filter-name>FilterName</filter-name>

    <filter-class>FilterClassName</filter-class>

    <init-param>

    <param-name>test-param</param-name>

    <param-value>Initialization parameter value</param-value>
    </init-param>

</filter>

<filter-mapping>

    <filter-name>FilterName</filter-name>

    <url-pattern>/*</url-pattern>

</filter-mapping>
```

Summary

When to use Servlets Filter

How to create a Filter

Filter Mapping with Servlets