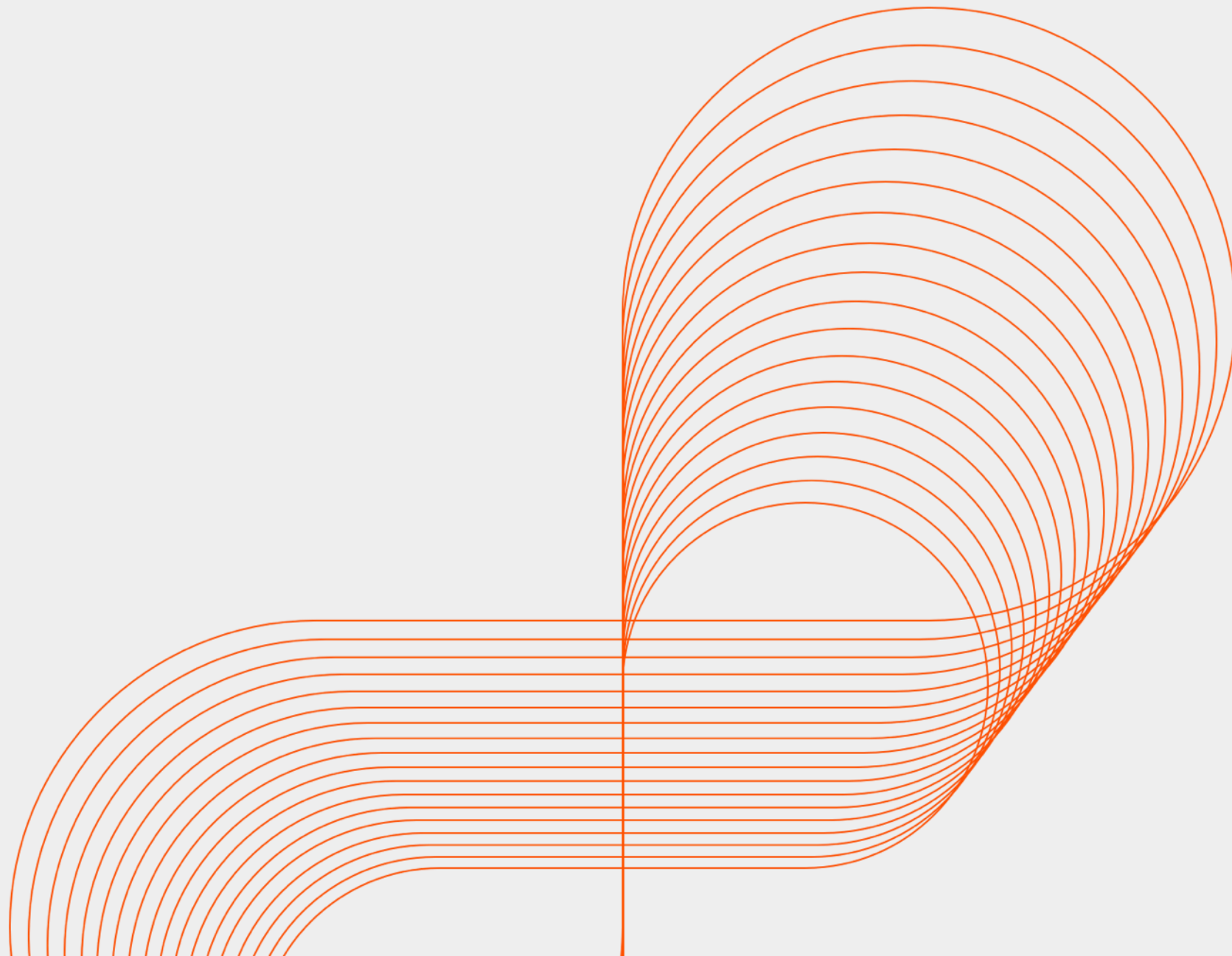




Persistent

Filters

Persistent University



Key learning points

1. What are filters

Filters

- Often an application requires that some code is executed whenever a request is sent to any resource within the application .
- For example if an application requires a user to log on then every resource in the application would need a security check to run before executing the resource.
- This is where Filters come into picture !
- Filters can intercept and modify the request coming into or the response going out of a Servlet or JSP page.

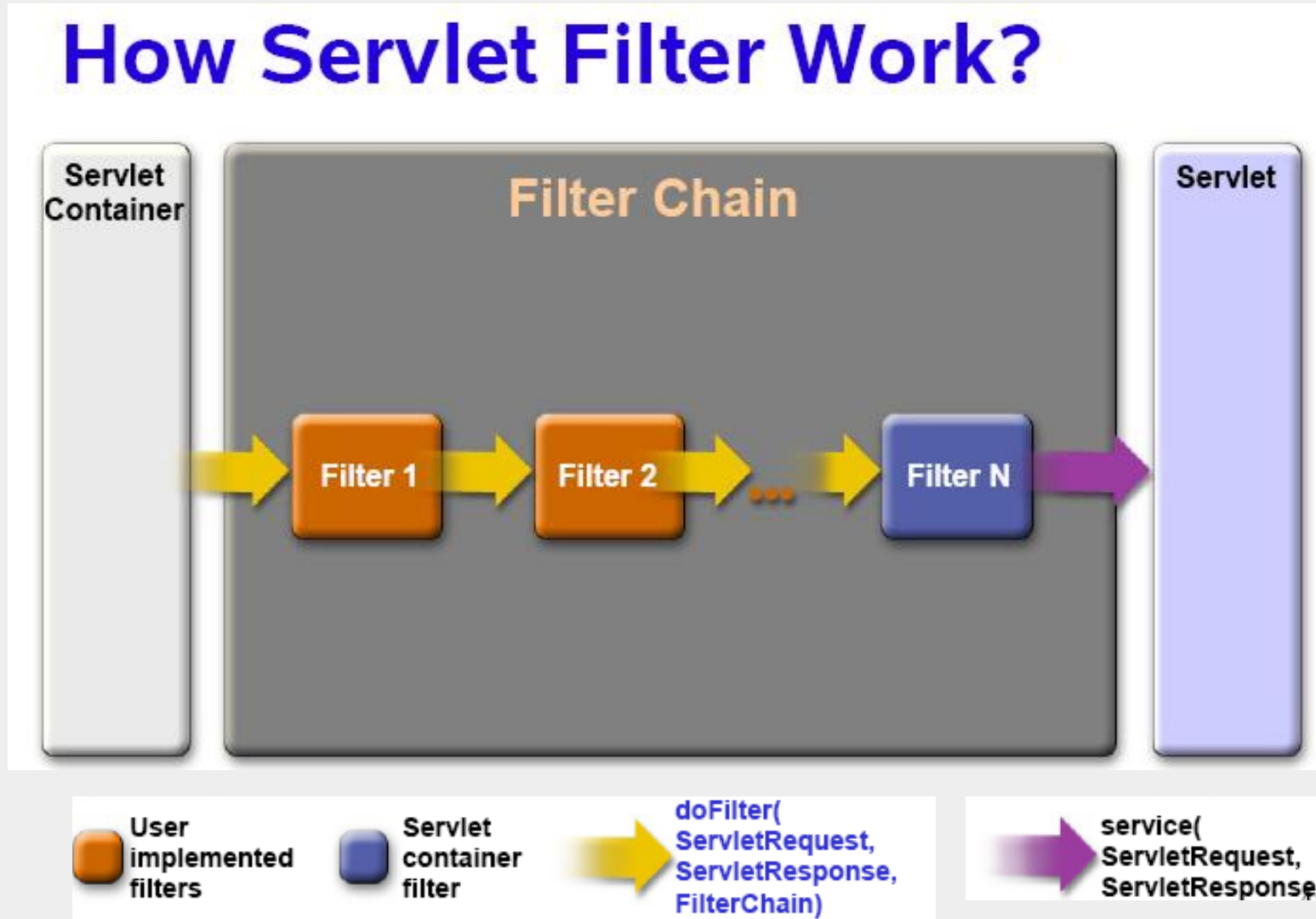
What are Java Servlet Filters?

- New component framework for intercepting and modifying requests and responses
 - Filters can be chained and plugged in to the system during deployment time
- Allows range of custom activities:
 - Marking access, blocking access
 - Caching, compression, logging
 - Authentication, access control, encryption
 - Content transformations

What Can a Filter Do?

- Examine the request headers
- Customize the request object if it wishes to modify request headers or data
- Customize the response object if it wishes to modify response headers or data
- Invoke the next entity in the filter chain
- Examine response headers after it has invoked the next filter in the chain

How servlet filter works?



How Filter Chain Works

- Multiple filters can be chained
 - order is dictated by the order of `<filter>` elements in the `web.xml` deployment descriptor
- The first filter of the filter chain is invoked by the container
 - via `doFilter(ServletRequest req, ServletResponse res, FilterChain chain)`
 - the filter then perform whatever filter logic and then call the next filter in the chain by calling `chain.doFilter(..)` method
- The last filter's call to `chain.doFilter()` ends up calling `service()` method of the Servlet

Steps for creating a Filter

- Create a class that implements the Filter interface. This interface comprises three methods: doFilter, init, and destroy.
- Put the filtering behavior in the doFilter method.
- Call the doFilter method of the FilterChain object.
- Register the filter with the appropriate servlets and JSP pages.

The Filter Interface

- All filters must implement `javax.servlet.Filter`.

public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)

- The `doFilter()` method contains the logic of the filter. The container will call this method each time an applicable request is being handled.

The init() method

```
public void init(FilterConfig filterConfig)
```

- The container calls the init() method on a filter instance once and only once during the lifetime of the filter.
- Gives the filter object a chance to initialize itself if required.

The destroy() method

Sample code ->

```
public void destroy(){  
    // do the clean up stuff  
}
```

The Container's rules for ordering filters:

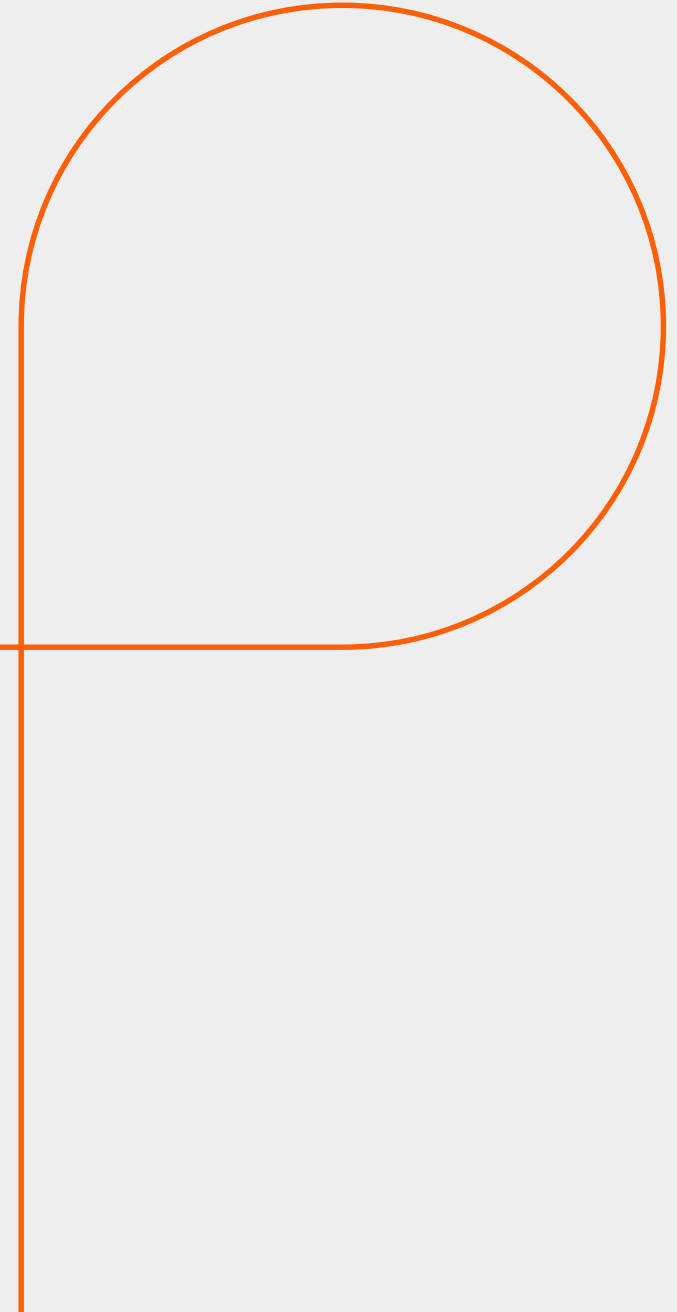
- All filters with matching URL patterns are located first. All filters that match will be placed in the chain in the order in which they are declared in the DD.
- Once all the matching filters are placed in the chain, the container does the same thing with filters that have a matching `<servlet-name>` in the DD.

Summary:

- With this we have come to an end of our session, where we discussed:
 - What are filters ?
 - What a filters can do?
 - How a filter chain works?
 - Steps for creating filters

Appendix

Thank You





Thank you!

Persistent University

