The slide features decorative curved lines in the corners. In the top-right corner, there is a thick, multi-layered arc in shades of orange and light blue. In the bottom-left corner, there is a similar thick, multi-layered arc in shades of light blue and orange. The main text is centered in the middle of the slide.

CS425: Software Engineering

Lesson 12

Security

Wholeness

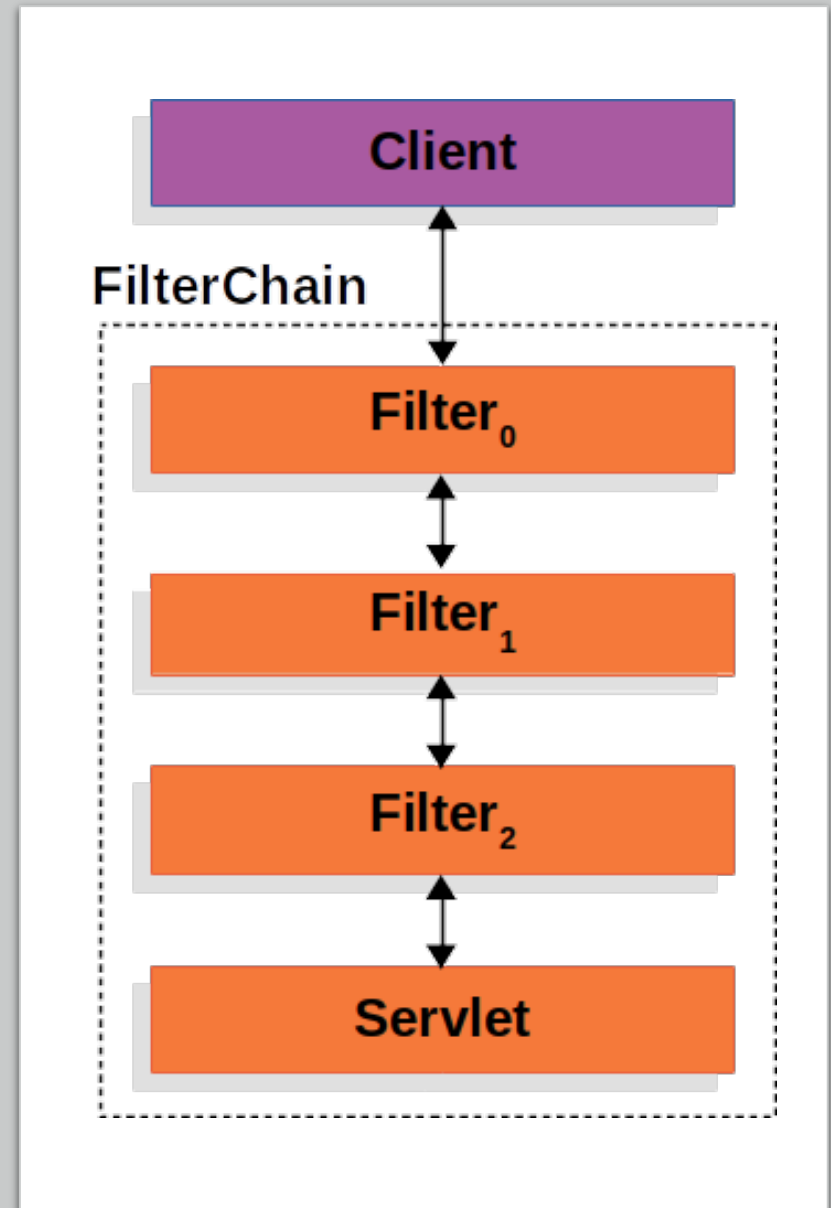
- Software Security describes methodologies, framework, processes and strategies for enhancing security and reducing vulnerabilities within software and its execution environment.
- Spring security offers a highly customizable authentication and access-control framework for Spring-based applications.
- Science of Consciousness: Action leads to Achievement. *Achievement leads to Fulfillment.*

Aspects of Security

- Authentication: process of verifying a user's identity. Ascertaining that someone is who they claim to be (e.g. userID and password)
- Authorization: mechanism for establishing access-level and granting appropriate rights and privileges.
- Cryptography: Techniques for securing communication through Data encryption

Key Concepts in Spring Security

- **SpringSecurityFilterChain:** For Servlet-driven Spring web applications, the Filter chain plays the central role for securing a given HTTP Request URI path.
- **Spring Boot Auto Configuration:** Simply adding the dependency named, `spring-boot-starter-security`, enables default config



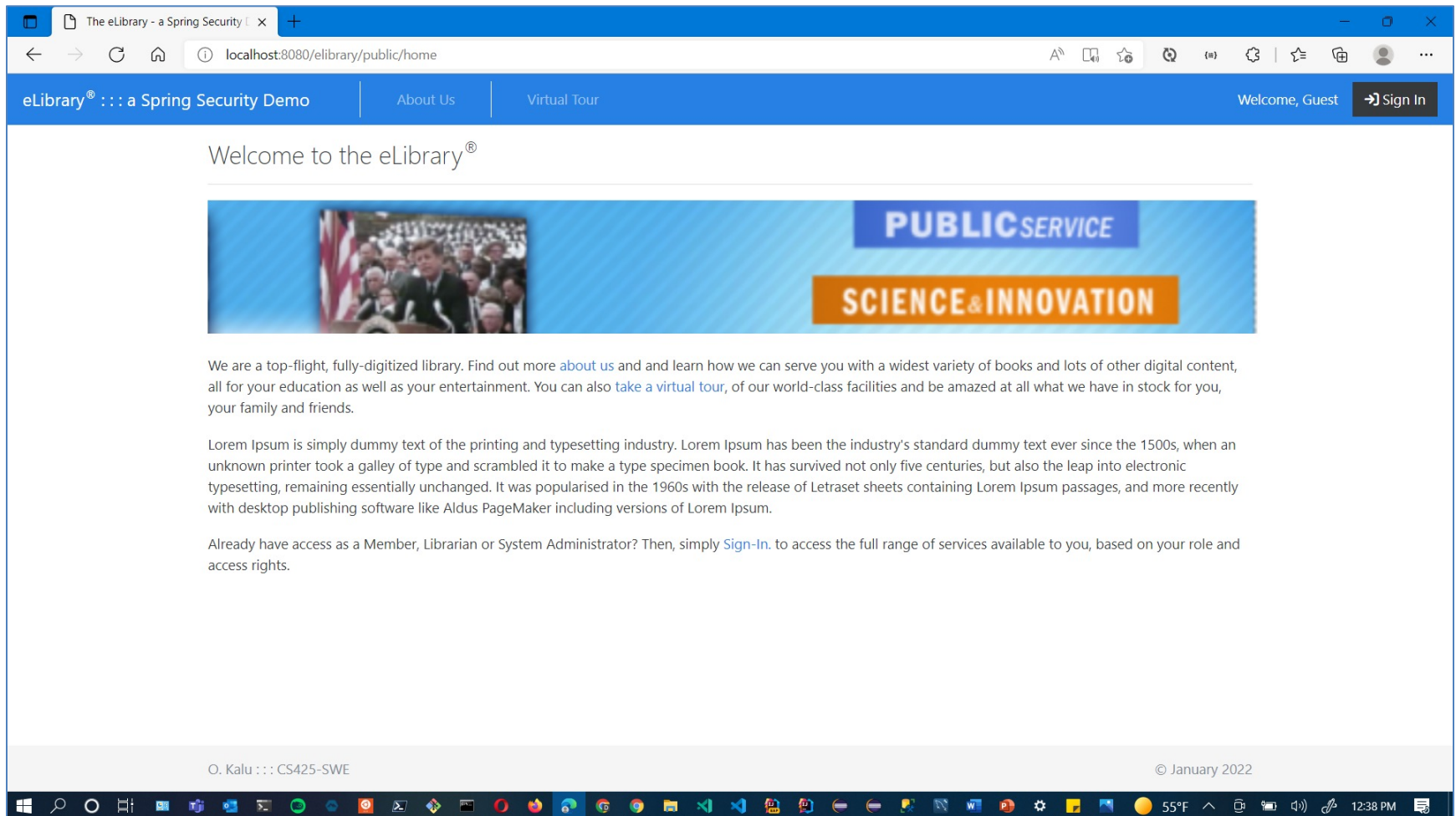
Types of Authentication Mechanisms supported by Spring Security

- HTTP Basic Authentication: enabled in HTTP server or Servlet container
- Form-based, custom Database Authentication: Where we use a Database to store/authenticate the user credentials.
- LDAP-driven Authentication. e.g. Microsoft Windows Active Directory service
- OAuth/OpenID Connect Authentication e.g. Login with Google or Microsoft or Github or Facebook account
- Token-based Authentication: Using JSON web token (JWT)

Spring Security API

- **AuthenticationManager**: is an interface with just one method named, `authenticate`, which processes an `Authentication` request.
- **AuthenticationProvider**: an interface that can be implemented by one or more classes that provide `Authentication` service using a given authentication mechanism e.g.
`DaoAuthenticationProvider`,
`LdapAuthenticationProvider`,
`JwtAuthenticationProvider` etc.

Implementing Spring Security for a webapp – Coding Demo




Implementing Spring Security for a webapp – Coding Demo

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/eLibrary/public/login'. The page has a blue header with the text 'eLibrary® :: a Spring Security Demo' and navigation links 'About Us' and 'Virtual Tour'. On the right of the header, it says 'Welcome, Guest' and has a 'Sign In' button. The main content area is titled 'Sign In' and contains two login options:

- Use your Google Account via OAuth**: This section includes the text 'Sign In with Google' and 'Simply click on the Google logo below and sign in using your Google Account.' Below this is a large blue button with the Google logo and the text 'Sign in with Google'.
- Use your eLibrary Account**: This section includes the text 'Sign In with your eLibrary Account' and 'Enter your eLibrary Username and Password below, to sign in.' It features two input fields: 'Username' and 'Password'. Below these fields are two buttons: 'Cancel' (orange) and 'Sign In' (green).

The bottom of the image shows a Windows taskbar with various application icons and a system tray displaying the date '© January 2022' and the time '12:40 PM'.

The slide features decorative curved lines in the corners. In the top right, there is a thick, multi-layered arc in shades of orange and grey. In the bottom left, there is a similar thick, multi-layered arc in shades of orange and grey. The main text is centered in a bold, dark blue font.

CS425: Software Engineering