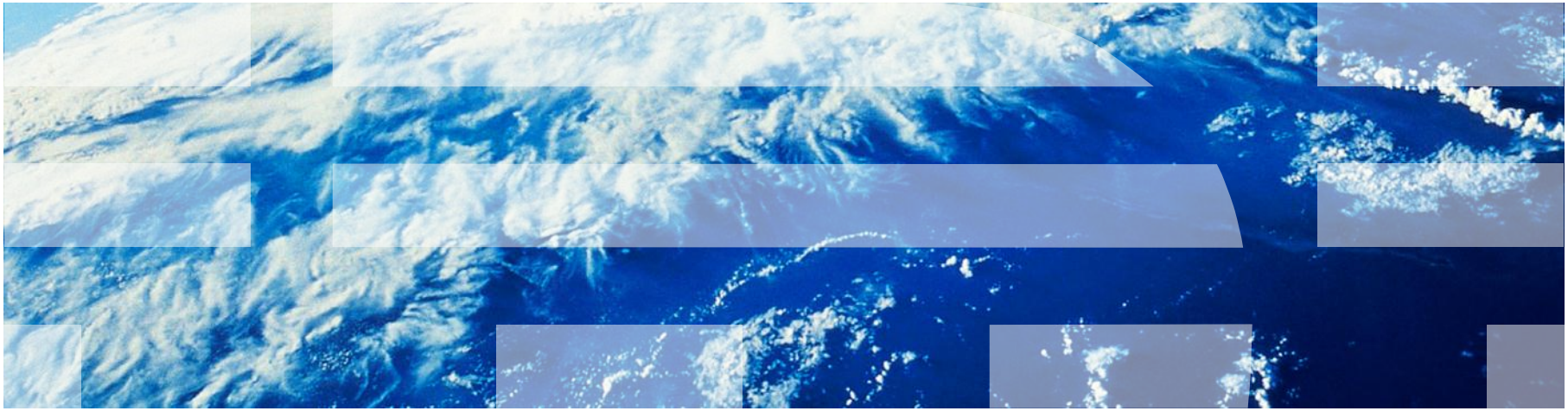
IBM Worklight Foundation V6.2.0 Getting Started

Authentication concepts



Trademarks

- IBM, the IBM logo, ibm.com, WebSphere, and Worklight are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
- Other company products or service names may be trademarks or service marks of others.
- This document may not be reproduced in whole or in part without the prior written permission of IBM.

About IBM®

- See http://www.ibm.com/ibm/us/en/

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

Authentication concepts and entities (1 of 10)

- IBM Worklight Foundation ® entities such as applications, adapter procedures, and static resources can be protected from unauthorized access.
- Entity protection rules are defined by a **security test** that contains one or more *authentication realms.*
- An **authentication realm** defines the process to be used to authenticate users.
- Each authentication realm consists of an **Authenticator** and a **Login Module**, which are server-side components.
- The same authentication realm can be used to protect several resources.
- Each authentication realm requires a **challenge handler** component on the client-side.
- Detailed definitions of all authentication components are given on the subsequent slides.

Authentication concepts and entities (2 of 10)
Authenticator

- An authenticator is a server-side entity that is responsible for collecting the credentials from the client application.
- An authenticator can collect any type of information that is accessible from an HTTP request object: cookies, headers, body, or any other properties.
- Worklight Server comes with a set of predefined authenticators, including:
  - A **form-based** authenticator that returns a challenge in the form of an HTML login form, making it useful for web environments and mobile applications.
  - An **adapter-based** authenticator that uses the Worklight adapter procedure to collect and validate the credentials from the client application.
  - A **header-based** authenticator that does not require interactive credentials collection, but checks the specific HTTP header instead.
- In addition to predefined authenticators, you can create your own **custom** authenticator by using the Java™ code.

Authentication concepts and entities (3 of 10)
Login modules

- A login module is a server-side entity that is responsible for verifying the user credentials and for creating a *user identity* object, which holds the user properties for the remainder of the session.
- The credentials validation can be done, for example, in one of the following ways:
  - By using a web service.
  - By looking up the user in a users table in a database.
  - By using the WebSphere® LTPA token.
- It is possible to add custom user properties according to the enterprise needs.
- A login module destroys the user identity object when the authenticated session terminates (logout or timeout).
- A login module can be configured to automatically record login attempts for audit purposes.
- In addition to predefined login modules, you can create your own custom login module by using the Java code.

Authentication concepts and entities (4 of 10)
Authentication realms

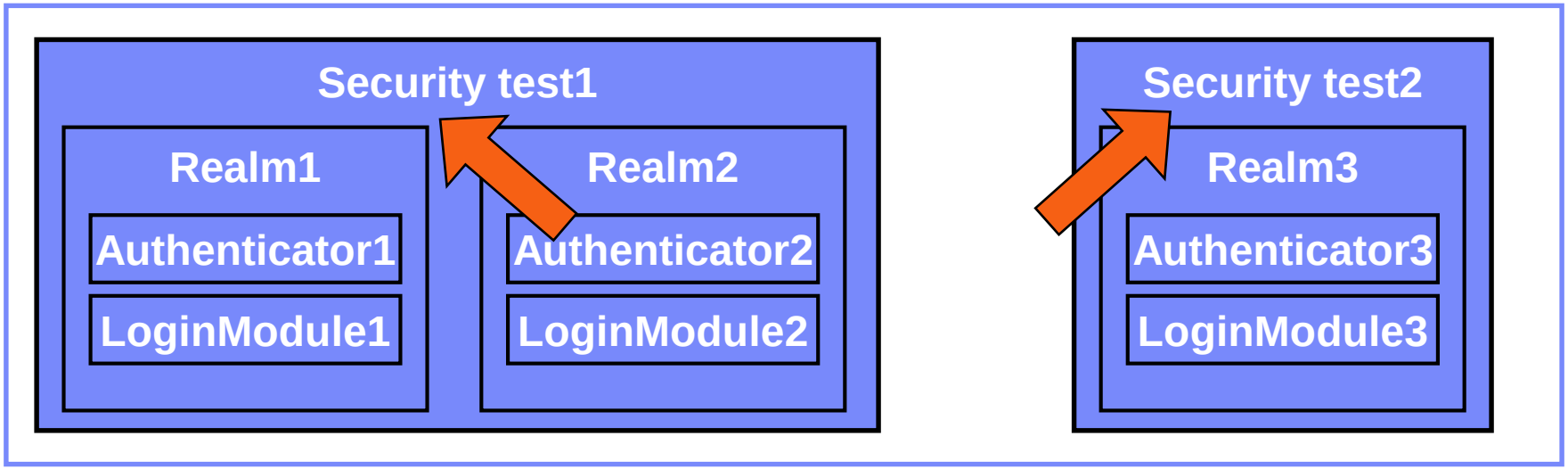
- An authentication realm is a combination of one authenticator and one login module.
- Each authentication realm defines its authentication flow:
  - What should happen after the authentication process is triggered?
  - What is the form of challenge that should be sent to the client application?
  - Which credentials should be collected?
  - How and when should credentials be collected?
  - How should credentials be sent to the server?
  - How should credentials be validated by the server?
  - What will be the result of the credentials validation?
  - What will be the properties of the user identity object?
- Worklight provides several predefined authentication realms for security features, such as a remote application disable, or an application authenticity.
- Each authentication realm that is defined in the server authentication configuration should have a corresponding challenge handler in the client application.

Authentication concepts and entities (5 of 10)
Security tests

- A security test is an ordered set of authentication realms that is used to protect a resource such as an adapter procedure, an application or a static URL.
- A security test defines the realms that the user must authenticate against to get access to the protected resource.
- A developer can define the order in which the authentication should be performed. For example: to request authentication in realm2 only after realm1 authentication succeeds.
- The IBM Worklight Foundation framework provides definitions of default security tests for mobile and web environments, and the ability to create custom security tests.
  - More in the following slides

Authentication concepts and entities (6 of 10)

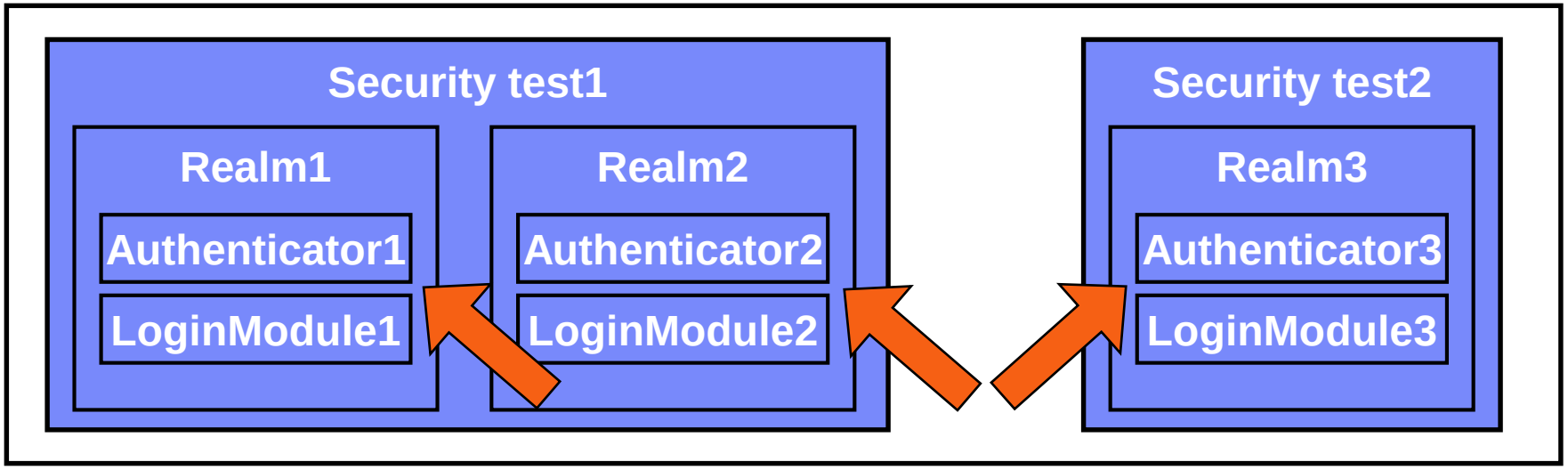
- Sample security configuration



- A resource - for example, an application or adapter procedure - can be protected by either of two security tests.
- Using Security test 1 means that the user must authenticate in both Realm1 and Realm2, each one with its own set of rules.
- Using Security test 2 means that the user must authenticate in Realm3 only.

Authentication concepts and entities (7 of 10)

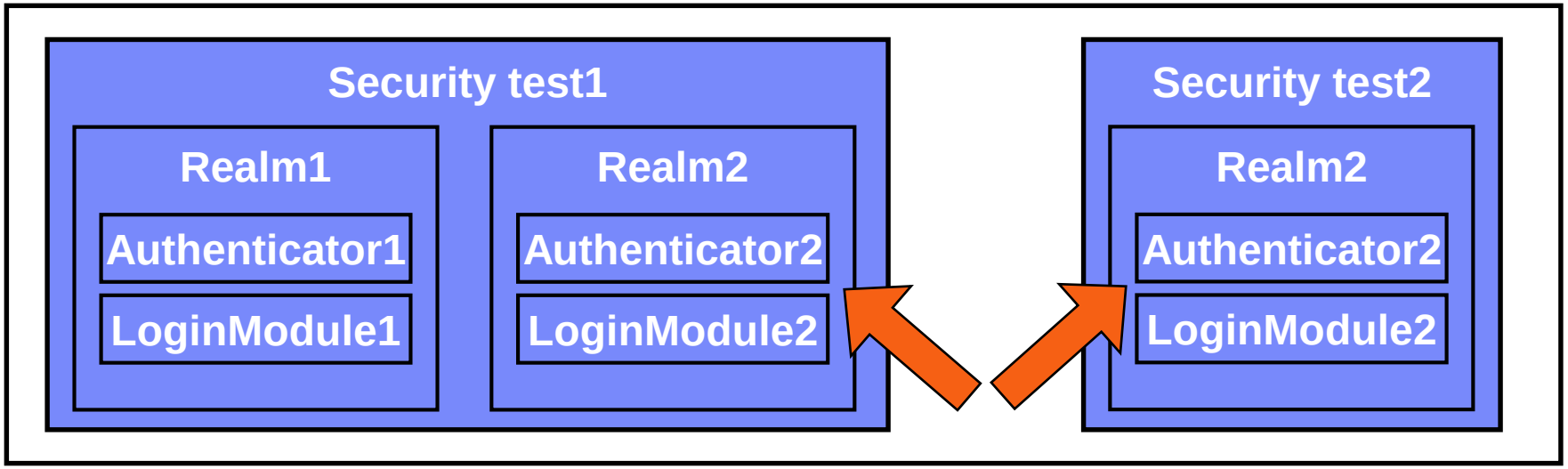
- Sample security configuration



- Each realm defines its own set of Authenticator and Login Module, meaning that each realm has its own rules for collecting credentials and validating them.

Authentication concepts and entities (8 of 10)

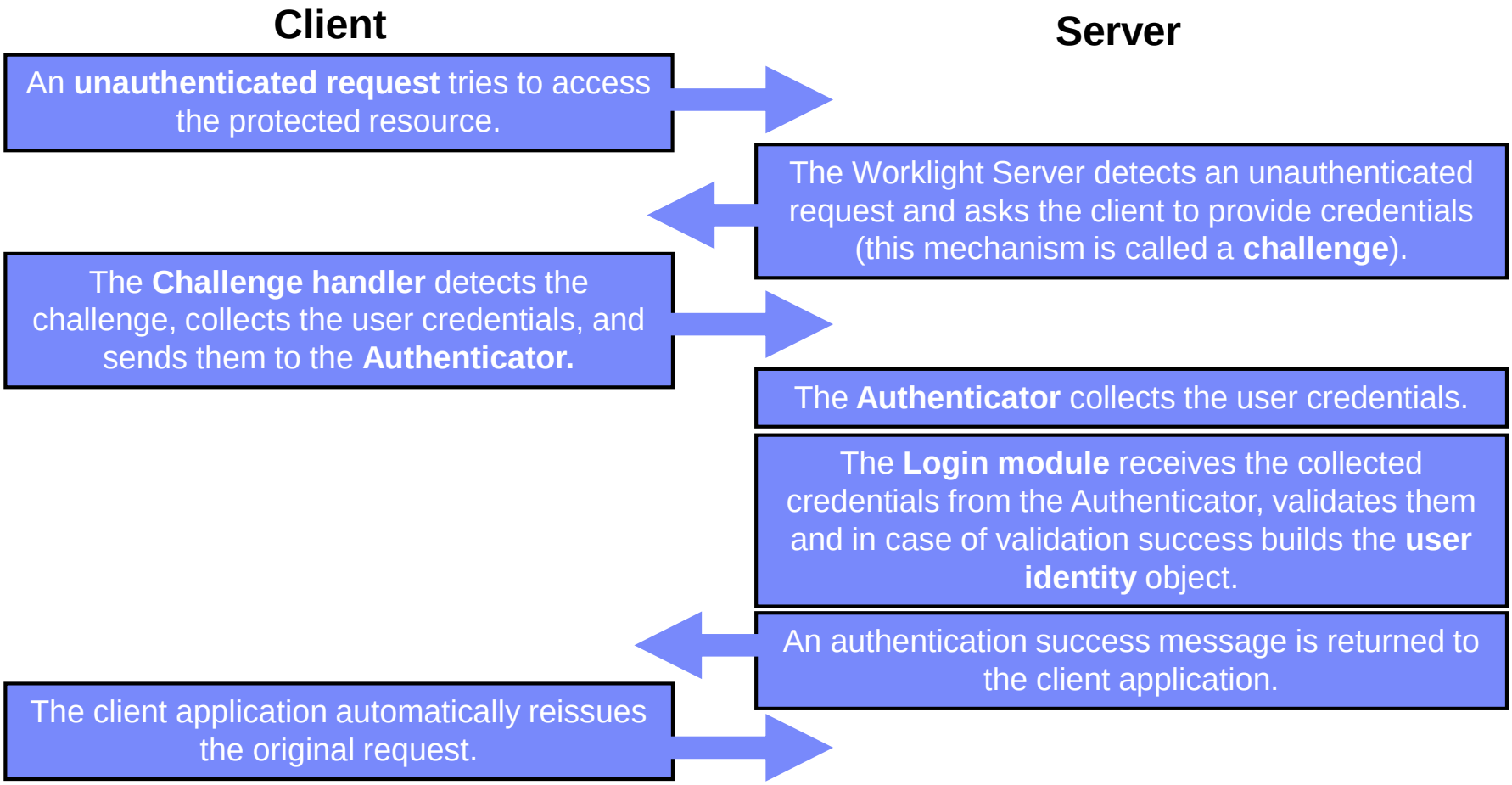
- Sample security configuration



- Realms, authenticators, and login modules can be reused.
- On an updated configuration above, Realm2 is reused.
- Protecting a resource with Security test1 means a must authenticate in both Realm1 and Realm2.
- Protecting a resource with Security test2 means a must authenticate in Realm2 only.

Authentication concepts and entities (9 of 10)

- When a request is made to the protected entity, IBM Worklight checks whether the session is already authenticated. If not, IBM Worklight automatically triggers a process to verify the user's identity.



Authentication concepts and entities (10 of 10)
Challenge handler

- A challenge handler is a client-side entity that controls the authentication process. It is used to detect the authentication challenges in the server responses and handle them.
- A separate challenge handler instance should be created for each realm that the application must authenticate in.
- A challenge handler can be used to detect and handle both the Worklight-related and the external authentication challenges, like the authentication proxies and the gateways.
- After a challenge handler detects an authentication challenge that is returned from the server, it is responsible for collecting the required credentials and for sending them back to the server.
- After the authentication flow completes, the challenge handler can send a notification back to the Worklight framework about the authentication success or failure.
- Though customizable, a challenge handler is created with a preset of methods that you can use to submit the credentials to the built-in user authentication types of the Worklight Server.

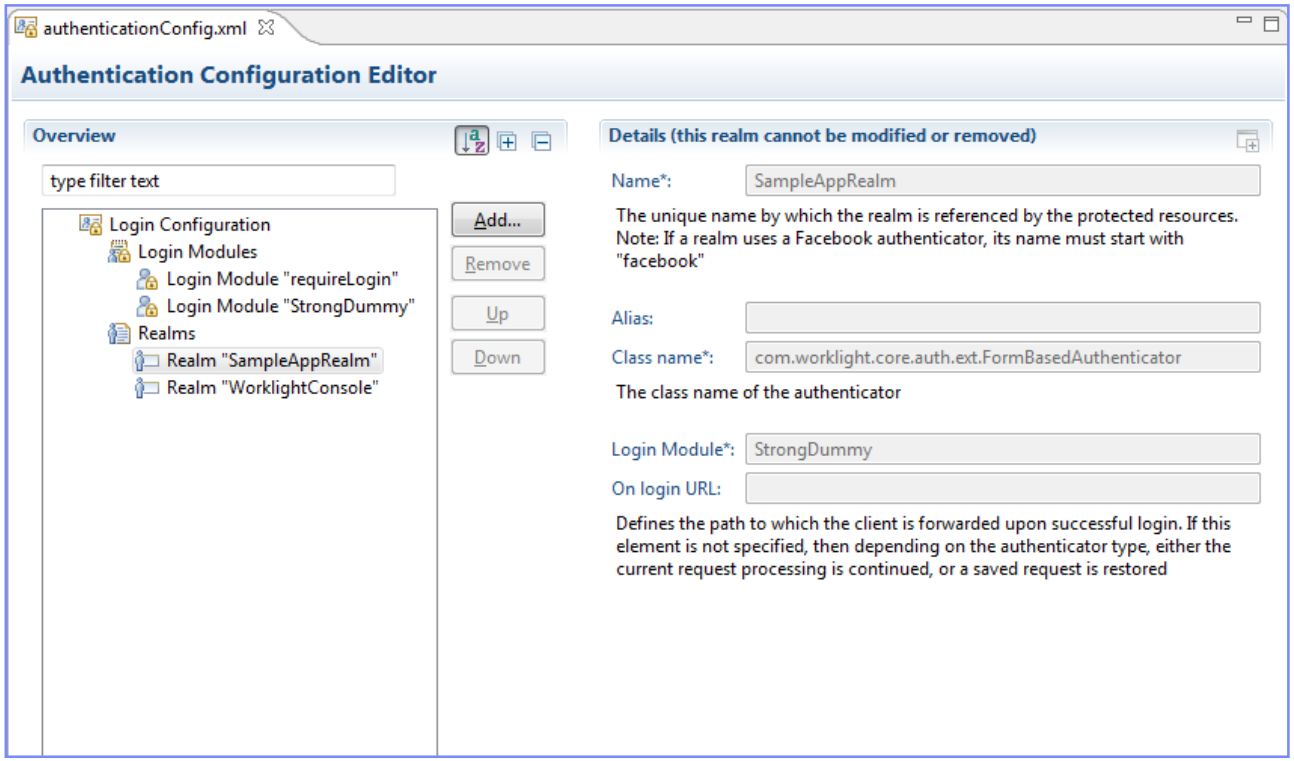
Create a challenge handler to define a customized authentication flow. In your challenge handler, do not add code that modifies the user interface when this modification is not related to the authentication flow.

Agenda

- Authentication concepts and entities
- Defining realms, authenticators, and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

Defining realms, authenticators, and login modules (1 of 3)

- Authentication settings are configured in the Worklight project in the **server\conf\authenticationConfig.xml** file.
- You can modify them by using the Authentication Configuration Editor.



Defining realms, authenticators, and login modules (2 of 3)

- Authentication settings are configured in the Worklight project in the **server\conf\authenticationConfig.xml** file.
- You can modify them by using the Authentication Configuration Editor.

```
<realms>
    <realm loginModule="StrongDummy" name="SampleAppRealm">
        <className>com.worklight.core.auth.ext.FormBasedAuthenticator</className>
    </realm>
    <realm loginModule="requireLogin" name="WorklightConsole">
        <className>com.worklight.core.auth.ext.FormBasedAuthenticator</className>
        <onLoginUrl>/console</onLoginUrl>
    </realm>
</realms>

<loginModules>
    <loginModule name="StrongDummy">
        <className>com.worklight.core.auth.ext.NonValid
    </loginModule>

    <loginModule name="requireLogin">
        <className>com.worklight.core.auth.ext.SingleId
    </loginModule>
</loginModules>
```

Each realm has a name, a **loginModule** specification, a **className** of an authenticator implementation and **optional** parameters.

Defining realms, authenticators, and login modules (3 of 3)

- Authentication settings are configured in the Worklight project in the **server\conf\authenticationConfig.xml** file.
- You can modify them by using the Authentication Configuration Editor.

```
<realms>
    <realm loginModule="StrongDummy" name="SampleAppRea
        <className>com.worklight.core.auth.ext.FormBase
    </realm>
    <realm loginModule="requireLogin" name="WorklightCo
        <className>com.worklight.core.auth.ext.FormBase
        <onLoginUrl>/console</onLoginUrl>
    </realm>
</realms>

<loginModules>
    <loginModule name="StrongDummy">
        <className>com.worklight.core.auth.ext.NonValidatingLoginModule</className>
    </loginModule>

    <loginModule name="requireLogin">
        <className>com.worklight.core.auth.ext.SingleIdentityLoginModule</className>
    </loginModule>
</loginModules>
```

Each **login module** has a name, a **className** of the implementation and **optional** parameters.

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

Defining security tests (1 of 5)

- With IBM Worklight Foundation, you can set up multiple realms for a security test.
- As a part of the security test setup, you must set which realms are considered a “**user realm**” and which are considered a “**device realm**”.
- Any identity from a realm that is defined as a **user realm** is used by IBM Worklight Foundation as a **user identity** for features that require one, such as push notification or application usage reports.
- An identity from a realm that is defined as a **device realm** is used by IBM Worklight Foundation as a **device identity** for features that require one, such as device provisioning, push notification, and SMS notification.

Defining security tests (2 of 5)

- After you set up your authentication realms, you must define the security tests to be used to protect your applications, adapter procedures, and static resources.
- Three types of security tests can be defined in the **authenticationConfig.xml** file:
  - **webSecurityTest** – a test that enables default web security-related realms.
  - **mobileSecurityTest** – a test that enables default mobile security-related realms.
  - **customSecurityTest** – a custom security test. Does not contain any default realm.

Defining security tests (3 of 5)
webSecurityTest

- Use the **webSecurityTest** to protect web applications.
- By default, the **webSecurityTest** includes a protection against XSRF attacks. For more information about this protection, see the IBM Worklight Foundation user documentation.
- Each **webSecurityTest** must contain one **<testUser>** element with a realm definition.
- This realm is considered a **user realm**.

```
<webSecurityTest name="SampleWebSecurityTest">
    <testUser realm="SampleRealm"/>
</webSecurityTest>
```

Defining security tests (4 of 5)
mobileSecurityTest

- Use the **mobileSecurityTest** to protect mobile applications.
- By default, the **mobileSecurityTest** includes:
  - A protection against XSRF attacks
  - An application authenticity test. For more information, see the user documentation.
  - An ability to disable mobile applications remotely from the Worklight Console.
- Each **mobileSecurityTest** must contain one **<testUser>** element with realm definition.
- This realm is considered a **user realm**.

```
<mobileSecurityTest name="SampleMobileSecurityTest">
    <testUser realm="SampleRealm"/>
</mobileSecurityTest>
```

Defining security tests (5 of 5)
customSecurityTest

- Use the **customSecurityTest** to dictate your own security preferences.
- Unlike the mobile and web security tests, the **customSecurityTest** does not include any predefined authentication realms. It includes only the tests that are defined by the developer.
- Any number of tests can be defined within the **customSecurityTest**.
- You can define which realm to be used as a user realm by adding the **isInternalUserId="true"** property.
- You can define the order of realms that the user must authenticate in.

```
<customSecurityTest name="SampleCustomSecurityTest">
    <test realm="SampleRealm1" step="1" />
    <test realm="SampleRealm2" step="2"/>
    <test realm="SampleRealm2" isInternalUserID="true" step="3"/>
</customSecurityTest>
```

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

Protecting applications

- Protecting an application means that authentication is required immediately when the application tries to connect to the Worklight Server.
- A separate **securityTest** can be defined for each application environment in the **application-descriptor.xml** file.

```
<common securityTest="SampleWebSecurityTest"/>

<android version="1.0" securityTest="SampleMobileSecurityTest">
    <worklightSettings include="true"/>
    <pushSender key="a" senderId="b"/>
            <security>
        <encryptWebResources enabled="true"/>
        <testWebResourcesChecksum enabled="true"/>
    </security>
</android>
```

- If no **securityTest** is defined for a specific environment, only a minimal set of default platform tests are carried out.

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

Protecting adapters

- Protecting an adapter procedure means that authentication is required when this adapter procedure is called by a client application.
- A separate **securityTest** can be defined for each adapter procedure in the adapter XML file.

```
<wl:adapter xmlns:wl="http://www.worklight.com/integration" xmlns:http="http://w

    <displayName>DummyAdapter</displayName>
    <description>DummyAdapter</description>
    <connectivity>
        <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
            <protocol>http</protocol>
            <domain>rss.cnn.com</domain>
            <port>80</port>
        </connectionPolicy>
        <loadConstraints maxConcurrentConnectionsPerNode="2"/>
    </connectivity>

    <procedure name="getSecretData" securityTest="DummyAdapter-securityTest"/>

</wl:adapter>
```

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

Protecting static resources

- A static resource is a URL that is loaded from the Worklight Server:
  - For example: the Worklight Console or Mobile Web application.
- Protecting a static resource means that the Worklight server requires authentication at any attempt to browse to the specified URL.
- The static resources and their protection can be defined in the **authenticationConfig.xml** file.

```
<staticResources>
    <resource id="worklightConsole" securityTest="WorklightConsoleSecurityTest">
        <urlPatterns>/console*</urlPatterns>
    </resource>
</staticResources>
```

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Check yourself questions

What is next

- In the following modules, you will implement several authentication types:
  - Form-based authentication
  - Adapter-based authentication
  - Custom Java authenticator and a login module
  - LDAP login module
  - LPTA token in WebSphere Application Server
- For more information about authentication, see the user documentation.

Agenda

- Authentication concepts and entities
- Defining realms, authenticators and login modules
- Defining security tests
- Protecting applications
- Protecting adapters
- Protecting static resources
- What is next
- Test your knowledge

Quiz (1 of 3)

Test your knowledge of the material in this module.
Answers on slide 3 of the quiz.

- What is the difference between an Authenticator and a Login module?
  - An Authenticator is a server-side entity that is used to collect and validate credentials. A Login module is a server-side entity that is used to create a userIdentity.
  - An Authenticator is a server-side entity that is used to collect credentials and create a user identity. A Login module is a server-side entity that is used to validate credentials.
  - An Authenticator is a server-side entity that is used to collect credentials. A Login module is a server-side entity that is used to validate credentials and create a user identity.
  - An Authenticator is a client side entity that performs basic credentials validation. A Login module is a server-side entity that performs deep credentials validation.
- A developer created two adapter procedures. Each procedure is protected by its own security test with different realms. What would be the consequence of this approach?
  - When a user authenticates in one realm, that user will be automatically authenticated in a second one.
  - A user will not be able to use these procedures together in the same application.
  - A user will have to log in to each realm separately.
  - A user will have to log out from one realm before that user can use a procedure that is protected by another realm.

Quiz (2 of 3)

Test your knowledge of the material in this module.
Answers on slide 3 of the quiz.

- What is the dependency between realm, authenticator, and login module in the authenticationConfig.xml file?
  - Each authenticator element must specify its className, realm, and loginModule.
  - Each realm element must specify a className of its authenticator and a loginModule name.
  - Each loginModule element must specify a className of its realm and an authenticator name.
  - Each authenticator element must specify its realm and its loginModule.

Quiz (3 of 3)

- The difference between an Authenticator and a Login module is the following one:
  - An Authenticator is a server-side entity that is used to collect and validate credentials. A Login module is a server-side entity that is used to create a userIdentity.
  - An Authenticator is a server-side entity that is used to collect credentials and create a user identity. A Login module is a server-side entity that is used to validate credentials.
  - An Authenticator is a server-side entity that is used to collect credentials. A Login module is a server-side entity that is used to validate credentials and create a user identity.
  - An Authenticator is a client side entity that performs basic credentials validation. A Login module is a server-side entity that performs deep credentials validation.
- A developer created two adapter procedures. Each procedure is protected by its own security test with different realms. What would be the consequence of this approach?
  - When a user authenticates in one realm, that user will be automatically authenticated in a second one.
  - A user will not be able to use these procedures together in the same application.
  - A user will have to log in to each realm separately.
  - A user will have to log out from one realm before that user can use a procedure that is protected by another realm.
- What is the dependency between realm, authenticator and login module in the authenticationConfig.xml file?
  - Each authenticator element must specify its className, realm, and loginModule.
  - Each realm element must specify a className of its authenticator and a loginModule name.
  - Each loginModule element must specify a className of its realm and an authenticator name.
  - Each authenticator element must specify its realm and its loginModule.

Notices

- Permission for the use of these publications is granted subject to these terms and conditions.
- This information was developed for products and services offered in the U.S.A.
- IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
- IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
  - IBM Director of Licensing
    IBM Corporation
    North Castle Drive
    Armonk, NY 10504-1785
    U.S.A.
- For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
  - Intellectual Property Licensing
    Legal and Intellectual Property Law
    IBM Japan Ltd.
    1623-14, Shimotsuruma, Yamato-shi
    Kanagawa 242-8502 Japan
- **The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.
- This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
- IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
- Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
  - IBM Corporation
    Dept F6, Bldg 1
    294 Route 100
    Somers NY 10589-3216
    USA
- Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.
- The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.
- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

- This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
- Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:
  - © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Privacy Policy Considerations

- IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.
- Depending upon the configurations deployed, this Software Offering may use session cookies that collect session information (generated by the application server). These cookies contain no personally identifiable information and are required for session management. Additionally, persistent cookies may be randomly generated to recognize and manage anonymous users. These cookies also contain no personally identifiable information and are required.
- If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent. For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the sections entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Support and comments

- For the entire IBM Worklight documentation set, training material and online forums where you can post questions, see the IBM website at:
  - http://www.ibm.com/mobile-docs
- **Support**
  - Software Subscription and Support (also referred to as Software Maintenance) is included with licenses purchased through Passport Advantage and Passport Advantage Express. For additional information about the International Passport Advantage Agreement and the IBM International Passport Advantage Express Agreement, visit the Passport Advantage website at:
    - http://www.ibm.com/software/passportadvantage
  - If you have a Software Subscription and Support in effect, IBM provides you assistance for your routine, short duration installation and usage (how-to) questions, and code-related questions. For additional details, consult your IBM Software Support Handbook at:
    - http://www.ibm.com/support/handbook
- **Comments**
  - We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.
  - For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.
  - When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state.
  - Thank you for your support.
  - Submit your comments in the IBM Worklight Developer Edition support community at:
    - https://www.ibm.com/developerworks/mobile/worklight/connect.html
  - If you would like a response from IBM, please provide the following information:
    - Name
    - Address
    - Company or Organization
    - Phone No.
    - Email address

Thank You