

# PingFederate SDK Development Training

Ping Identity Education



The Identity Security Company

## Agenda

- Review of PingFederate Web SSO and Administrator UI
- Introducing the PingFederate SDK
- Getting Started with the SDK
- Anatomy of an SDK Component
- Developing IdP Adapters
- Miscellaneous Topics (logging, debugging, 3rd party libraries)
- Developing SP Adapters
- Using Existing integration Kits (into pluggable frameworks)
- Developing Adapter Selectors
- Developing Data Source Drivers
- Developing Password Credential Validators



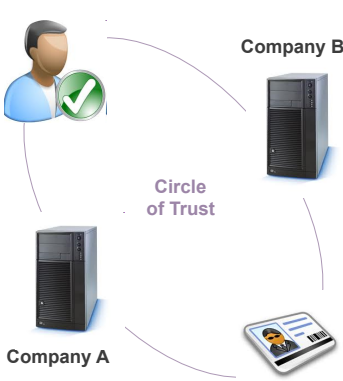
PingFederate SDK Development

## REVIEW OF PINGFEDERATE WEB SSO AND ADMINISTRATOR UI



**The Identity Security Company**

### So what is Federation?




- Users authenticate on the Company A domain
- Both companies agree upon user identities and attributes via a standard security protocol.
- User's identity is verified on the Company B domain and they are granted appropriate access
- The sites are commonly referred to as the Identity Provider and the Service Provider

***“Federation is a standards-based method for sharing and managing identity data and establishing single sign-on across security domains and organizations.”***

4

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## PingFederate Services

# PingFederate®

Administration Console      Management Services

Integration Kits      **Cloud Single Sign-On & Federated Identity**


SaaS Connectors      **Secure Mobile Access**

Cloud Identity Connectors      **Automated Cloud User Provisioning**

Token Translators      **API Security**

CloudDesktop      **Runtime Services**

Logging & Monitoring

5      Copyright ©2013 Ping Identity Corporation. All rights reserved.      

## Administrator's Console

# PingFederate®

Help | About | Logout (Administrator)

**IdP Configuration**      **Server Configuration**      **SP Configuration**

**APPLICATION INTEGRATION SETTINGS**      **SYSTEM SETTINGS**      **SECURITY**      **APPLICATION INTEGRATION SETTINGS**

- Adapters
- Adapter Selection
- Default URL
- Application Endpoints

**FEDERATION SETTINGS**      **ADMINISTRATIVE FUNCTIONS**      **AUTHENTICATION**      **FEDERATION SETTINGS**


- Protocol Endpoints
- Metadata Export
- XML File Signatures
- Configuration Archive
- Account Management
- License Management
- Virtual Host Names
- Application Authentication
- Password Credential Validators
- Active Directory Domains/Kerberos
- Realms
- Protocol Endpoints

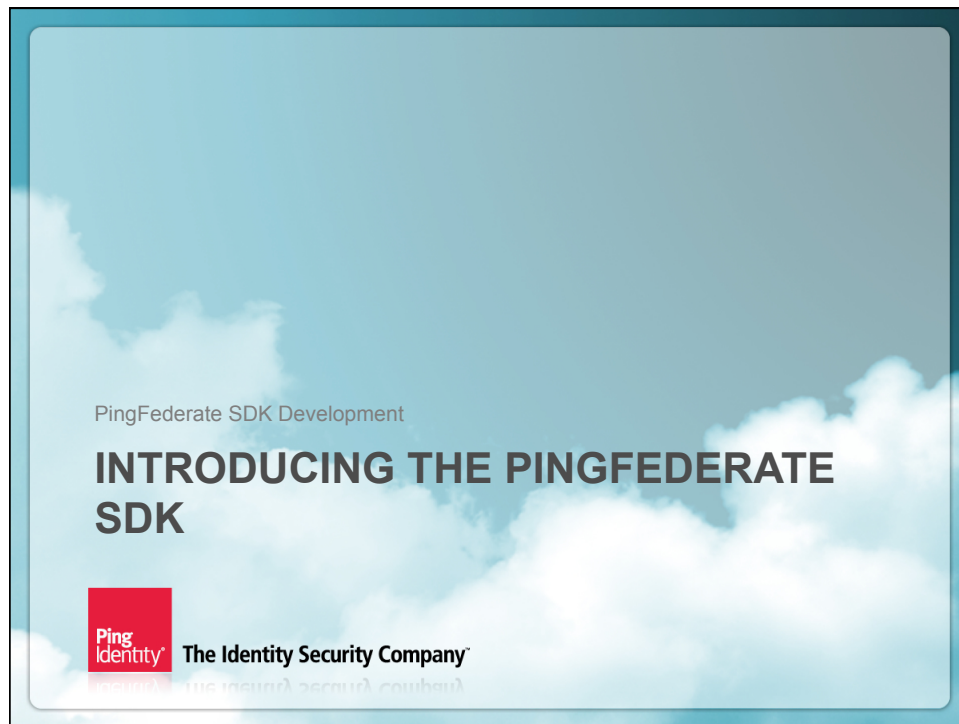
**SP CONNECTIONS (0)**      **IDP CONNECTIONS (0)**

- Manage All SP    Create New
- Manage All IdP    Create New

**SP AFFILIATIONS (0)**

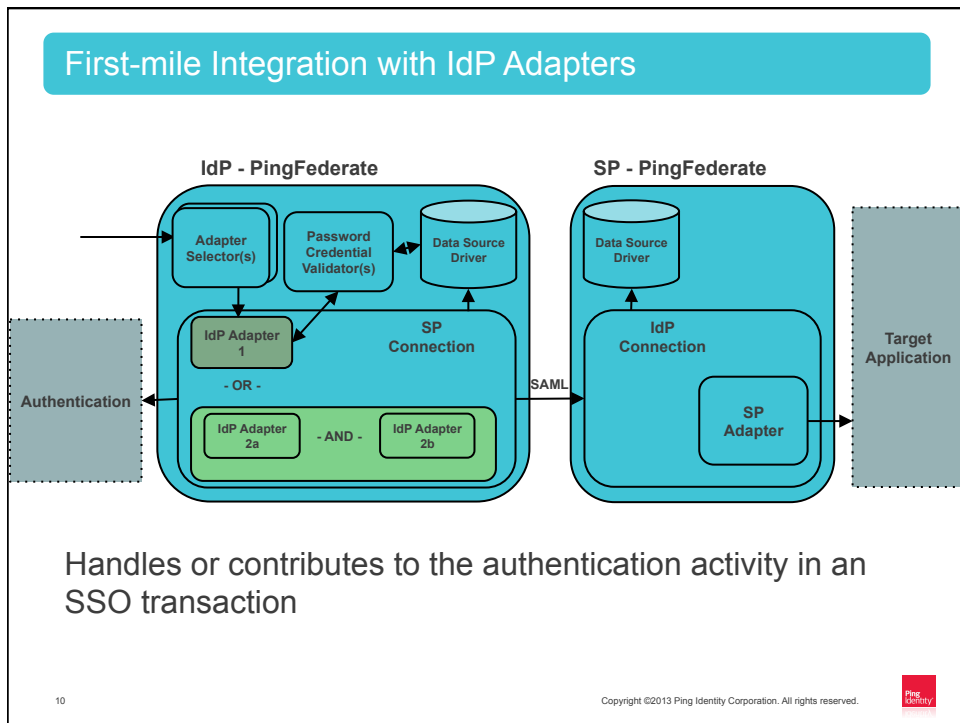
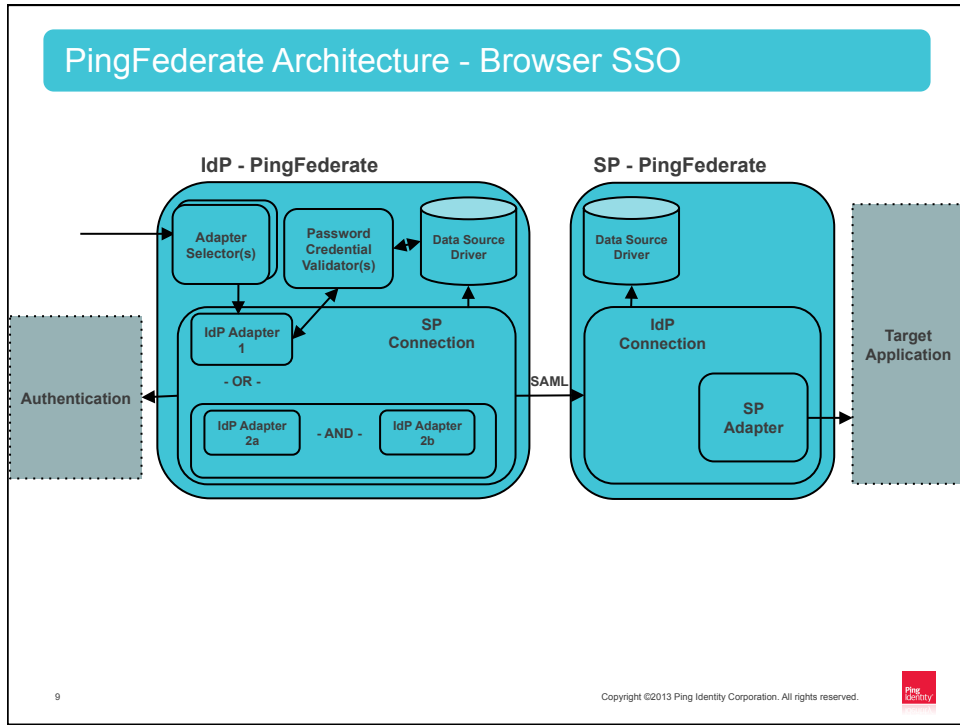
- Manage All Affiliations    Create New

6      Copyright ©2013 Ping Identity Corporation. All rights reserved.      

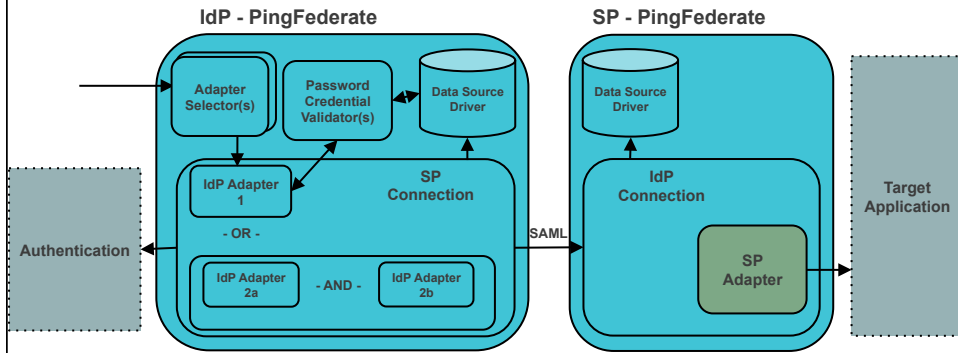


### Introducing the PingFederate SDK

- Set of Java packages and interfaces that enable developers to build custom components to meet unique requirements
  - First-mile integration - IdP Adapters
  - Last-mile integration - SP Adapters
  - Attribute lookup
  - Adapter selection
  - Credential validation
  - Custom token processors/generators
- Custom components are first-class citizens in PingFederate runtime and administration
  - Configuration UI plugs in to the administration console screens
  - Custom runtime behavior plugs in to end-user SSO transaction processing



## Last-mile Integration - SP Adapters



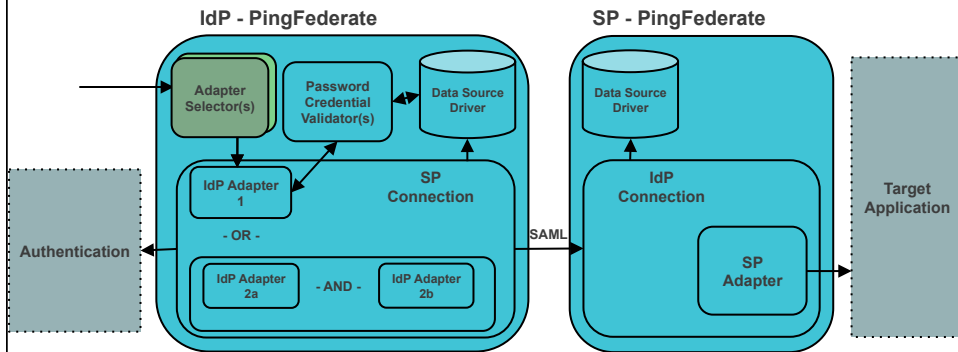
Passes information from an SSO transaction to a target application to establish a security context for the user

11

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Adapter Selectors

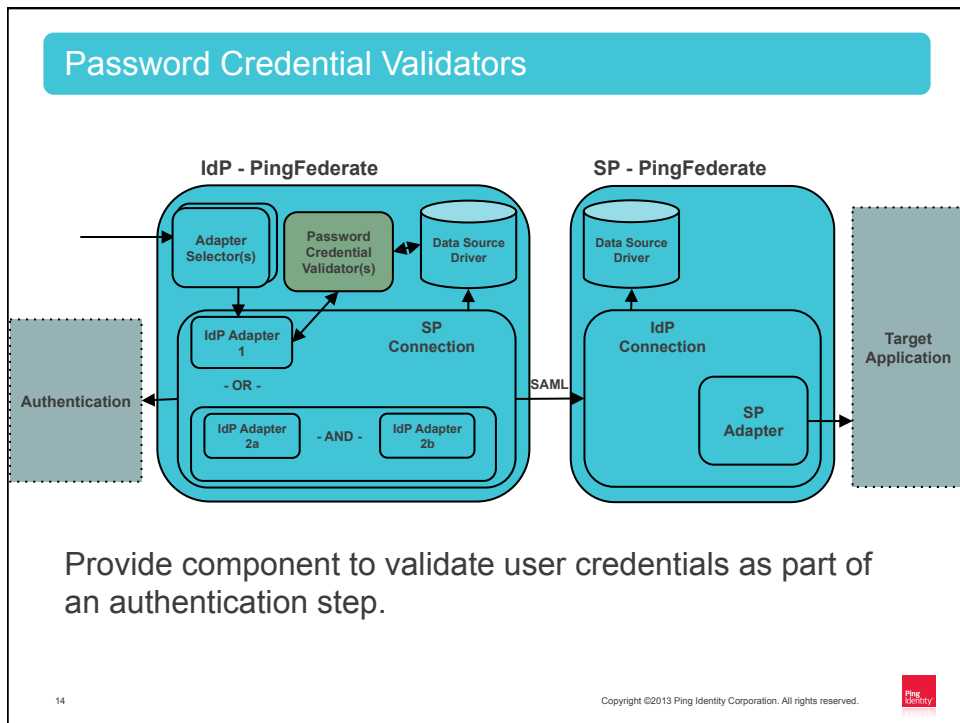
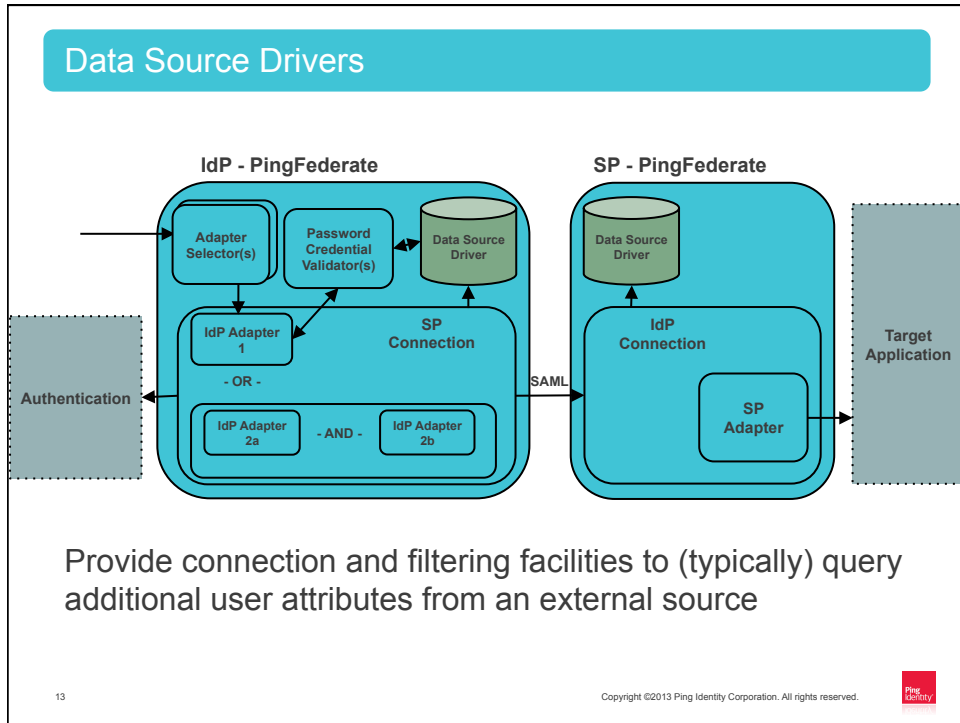


Provide the capability to choose among multiple configured IdP Adapters based on the current user context

12

Copyright ©2013 Ping Identity Corporation. All rights reserved.





## Documentation

- Javadocs
  - <PingFederate Install>/pingfederate/sdk/doc/index.html
- Sample Implementations
  - <PingFederate Install>/pingfederate/sdk/plugin-src
- Developer's Guide
  - <http://documentation.pingidentity.com/display/PF/SDK+Developer%27s+Guide>

15

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

## GETTING STARTED WITH THE SDK



The Identity Security Company



## Setting up a Development Environment

- Prerequisites
  - PingFederate
  - JDK 1.7+
  - Ant
- SDK Folder Structure <PingFederate Install>/pingfederate/sdk
  - /plugin-src
    - /your-project
      - /java (your source code)
      - /lib (3rd party JARs)
  - /doc
  - /lib
  - build.local.properties (Ant overrides for build/deploy)
  - build.properties
  - build.xml (Ant script)

17

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Build and Deploy a Component (Ant)

- Edit and save build.local.properties
  - target-plugin-name=idp-adapter-example
- Run the Ant script from a cmd prompt
  - ant deploy-plugin (jar-plugin, clean-plugin)
  - Check for BUILD SUCCESSFUL message
  - (Optional) Find the JAR in the PingFederate deploy folder
    - <PingFederate Install>/pingfederate/server/default/deploy
- Force PingFederate to re-scan the deploy folder
  - Restart PingFederate

18

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Build and Deploy a Component (Alternate)

- Build the JAR file
- Copy the JAR file to the PingFederate deploy folder
  - <PingFederate Install>/pingfederate/server/default/deploy
- Force PingFederate to re-scan the deploy folder
  - Restart PingFederate

19

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

## ANATOMY OF AN SDK COMPONENT



The Identity Security Company

## Anatomy of an SDK Component

- Descriptor Method
  - Invoked when PingFederate loads the component
  - Component Name
  - Configuration UI Screen Design
  - Additional Component-specific Properties
- Configuration Retrieval Method
  - Invoked on-demand when PingFederate determines the component needs configuration
  - Typically, values retrieved in this method are stored as private instance variables
- Runtime Processing Methods
  - Invoked per SSO transaction
  - Persisting state across runtime calls must be handled with `SessionStateSupport` or a persistent store

21

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Descriptor Method

- The component description is designed by returning implementations of the following (or their sub-classes):
  - `com.pingidentity.sdk.PluginDescriptor`
  - `com.pingidentity.sdk.GuiConfigDescriptor`

22

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Descriptor Method: com.pingidentity.sdk.PluginDescriptor

- PluginDescriptor takes 3 parameters at a minimum
  - String name
  - ConfigurablePlugin plugin (i.e. "this")
  - GuiConfigDescriptor guiConfigDescriptor
- Component-specific sub-classes may take additional parameters
  - Adapter Contract
  - Adapter Selector Result Values
  - Data Source Filter Values
- Example for Adapter Selector

```
AdapterSelectorDescriptor adapterSelectorDescriptor = new AdapterSelectorDescriptor(COMPONENT_NAME, this, guiConfigDescriptor, results);
adapterSelectorDescriptor.setSupportsExtendedResults(false);
return adapterSelectorDescriptor;
```

23

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Descriptor Method: com.pingidentity.sdk.GuiConfigDescriptor

- GuiConfigDescriptor is the "UI Designer" class
  - UI Controls
    - org.sourceid.saml20.adapter.gui
  - UI Validation
    - org.sourceid.saml20.adapter.gui.validation
    - org.sourceid.saml20.adapter.gui.validation.impl
  - UI Description/Instructions
- Example for Adapter Selector

```
GuiConfigDescriptor guiConfigDescriptor = new GuiConfigDescriptor();
guiConfigDescriptor.setDescription(COMPONENT_DESC);

// GUI for HTTP Header Name
TextFieldDescriptor headerNameDescriptor = new TextFieldDescriptor(FIELD_NAME_HEADER_NAME, FIELD_DESC_HEADER_NAME);
headerNameDescriptor.addValidator(new RequiredFieldValidator());
guiConfigDescriptor.addField(headerNameDescriptor);
```

## Descriptor Method: Example

```

public PluginDescriptor getPluginDescriptor()
{
    GuiConfigDescriptor guiConfigDescriptor = new GuiConfigDescriptor();
    guiConfigDescriptor.setDescription(COMPONENT_DESC);

    // GUI for HTTP Header Name
    TextFieldDescriptor headerNameDescriptor = new TextFieldDescriptor(FIELD_NAME_HEADER_NAME, FIELD_DESC_HEADER_NAME);
    headerNameDescriptor.addValidator(new RequiredFieldValidator());
    guiConfigDescriptor.addField(headerNameDescriptor);

    TextFieldDescriptor headerValueDescriptor = new TextFieldDescriptor(FIELD_NAME_HEADER_VALUE, FIELD_DESC_HEADER_VALUE);
    headerValueDescriptor.addValidator(new RequiredFieldValidator());
    guiConfigDescriptor.addField(headerValueDescriptor);

    // Add our pre-coded results
    Set<String> results = new HashSet<String>();
    results.add(RESULT_YES);
    results.add(RESULT_NO);

    AdapterSelectorDescriptor adapterSelectorDescriptor = new AdapterSelectorDescriptor(COMPONENT_NAME, this, guiConfigDescriptor, results);
    adapterSelectorDescriptor.setSupportsExtendedResults(false);
    return adapterSelectorDescriptor;
}

```

25

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Descriptor Method: org.sourceid.sam20.adapter.gui

### UI Controls

Standard UI	Upload-Download	PingFederate Objects	Table
CheckBoxFieldDescriptor RadioGroupFieldDescriptor SelectFieldDescriptor TextAreaFieldDescriptor TextFieldDescriptor	ActionDescriptor UploadFileFieldDescriptor	ClientCertKeypairFieldDescriptor DsigKeypairFieldDescriptor EncryptionCertificateFieldDescriptor TrustedCAFieldDescriptor  CustomSourceFieldDescriptor JdbcDatastoreFieldDescriptor LdapDatastoreFieldDescriptor  PasswordCredentialValidatorFieldDescriptor	TableDescriptor

26

Copyright ©2013 Ping Identity Corporation. All rights reserved.



Descriptor Method:  
`org.sourceid.saml20.adapter.cui.validation.(impl)`

## UI Validation

Simple	Complex
DoubleValidator EmailValidator FloatValidator HttpURLValidator IntegerValidator LongValidator RegExValidator RequiredFieldValidator URLValidator	ConfigurationValidator EnhancedRowValidator

27

Copyright ©2013 Ping Identity Corporation. All rights reserved.

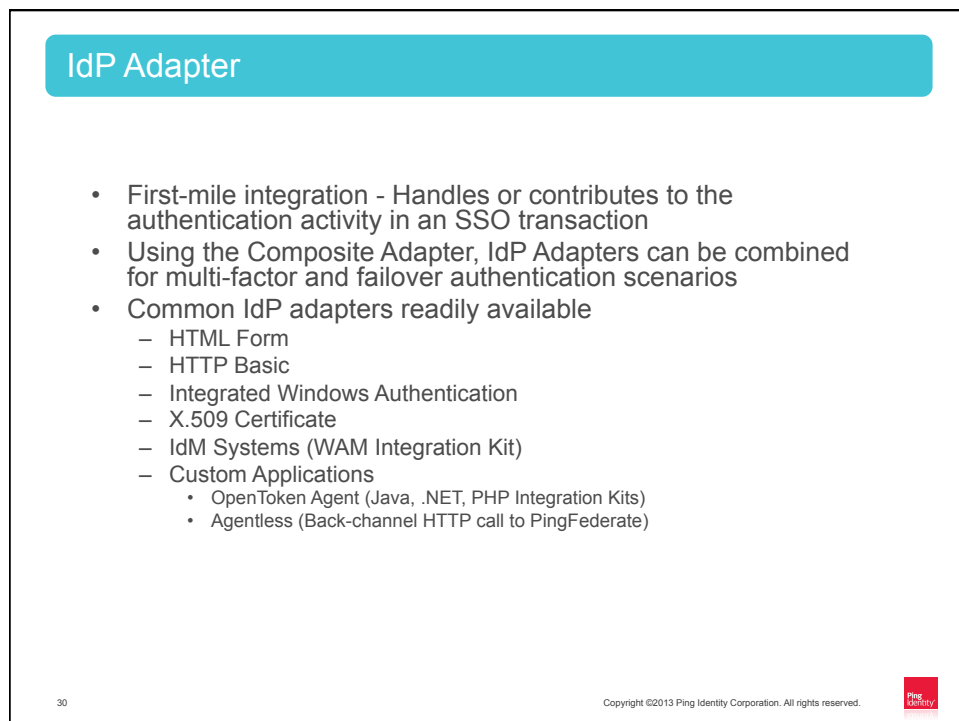
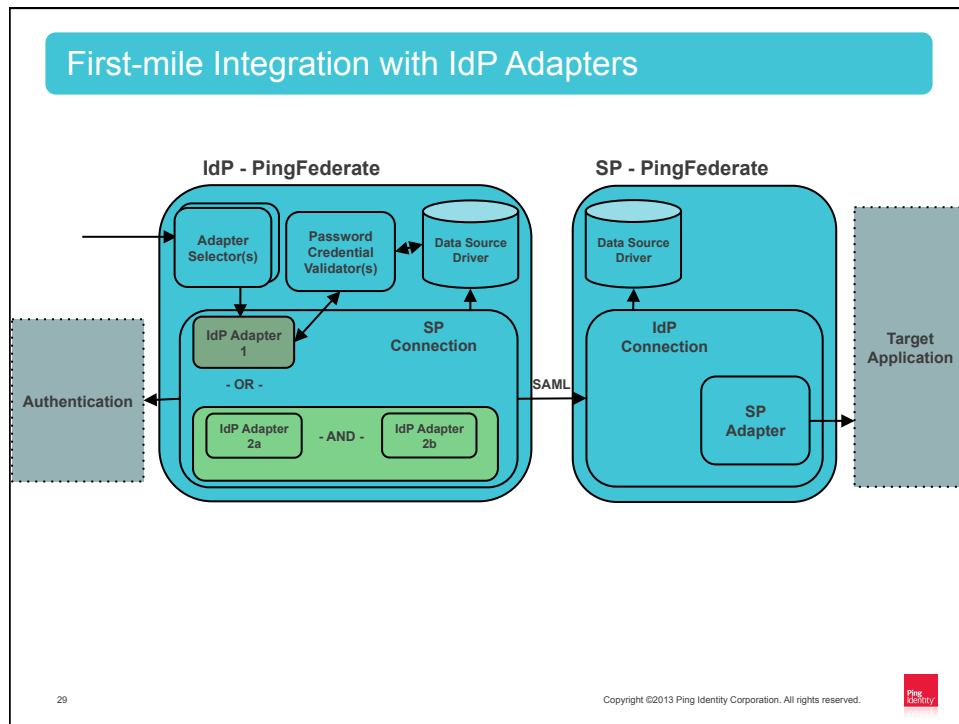


PingFederate SDK Development

## DEVELOPING IDP ADAPTER



The Identity Security Company



## Building a Custom IdP Adapter

- Interface
  - implements IdpAuthenticationAdapterV2
- Descriptor
  - IdpAuthnAdapterDescriptor getAdapterDescriptor()
- Configuration Retrieval
  - void configure(Configuration configuration)
- Session Lookup
  - AuthnAdapterResponse lookupAuthN(HttpServletRequest req, HttpServletResponse resp, Map<String, Object> inParameters)
- Session Logout
  - boolean logoutAuthN(Map authnIdentifiers, HttpServletRequest req, HttpServletResponse resp, String resumePath)
- Session Lookup (**Deprecated** - implement and return null)
  - Map lookupAuthN(HttpServletRequest req, HttpServletResponse resp, String partnerSpEntityId, AuthnPolicy authnPolicy, String resumePath)

31

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

## CODE REVIEW & LAB IDP ADAPTER



The Identity Security Company™

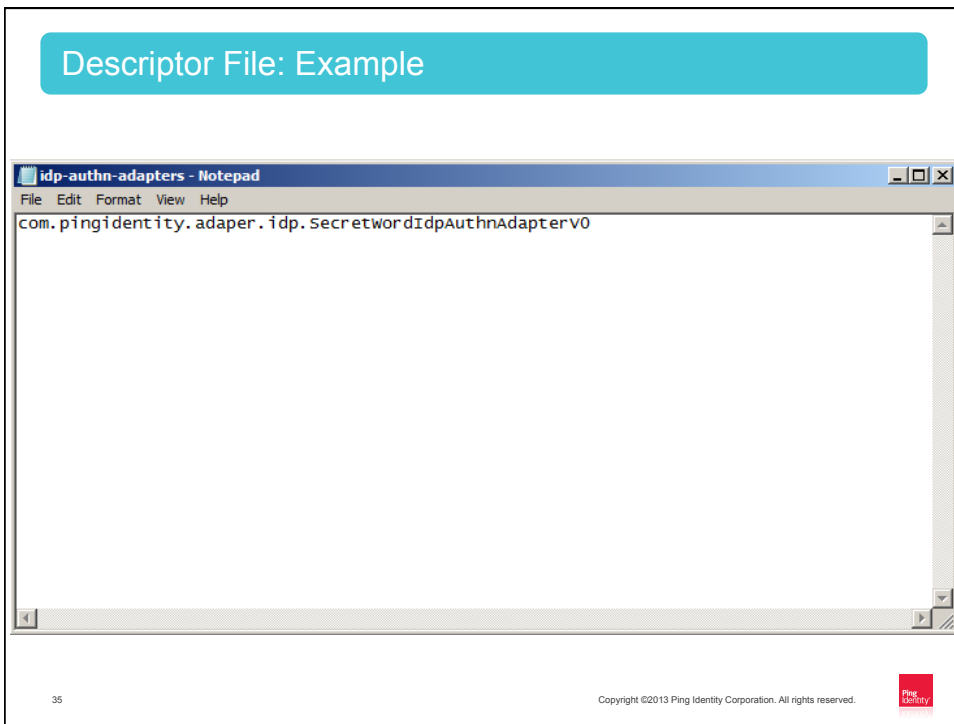




### Descriptor File

- Required file that tells PingFederate what SDK components are contained in the JAR.
    - Using the Ant script creates and packages this file for you automatically.
  - ProjectStructure <PingFederate Install>/pingfederate/sdk
    - /plugin-src
      - /your-project
        - /java (your source code)
        - /lib (3rd party JARs)
        - /resources
          - » /PF-INF
            - **DescriptorFile**
- File must be named according to the component type and the contents must be the fully-qualified class name of the component
  - idp-authn-adapters
  - sp-authn-adapters
  - adapter-selectors
  - custom-drivers

## Descriptor File: Example



## Logging

- Administrative logging controls
  - Runtime logging is configured in <PingFederate Install>/pingfederate/server/default/conf/log4j.xml
- Logging from a custom component
  - import org.apache.commons.logging.Log;
  - import org.apache.commons.logging.LogFactory;
  - private Log log = LogFactory.getLog(this.getClass());
  - log.debug("message to log");
- Log output is written by default to server.log
  - <PingFederate Install>/pingfederate/log/server.log

## log4j Basics

- Loggers – create log messages
- Log Levels – control level of logging detail
  - TRACE, DEBUG, INFO, WARN, ERROR, FATAL
- Appenders – where log messages go
- Layouts – control format of messages

37

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Remote Debugging with Eclipse

- Enable debugging on the JVM
  - Modify the JVM command line options by uncommenting the following line in <PingFederate>/pingfederate/bin/run.sh
    - `JAVA_OPTS="-Xdebug -Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n $JAVA_OPTS"`
  - Restart PingFederate
- Attach the Eclipse debugger
  - Manage Debug Configurations and create a new Remote Java Application configuration
  - Connect to localhost:8787

38

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Third-Party Libraries

- SDK Folder Structure <PingFederate Install>/pingfederate/sdk
  - /plugin-src
    - /your-project
      - /java (your source code)
      - /lib (3rd party JARs)
  - /doc
  - /lib
  - build.local.properties (Ant overrides for build/deploy)
  - build.properties
  - build.xml (Ant script)

39

Copyright ©2013 Ping Identity Corporation. All rights reserved.

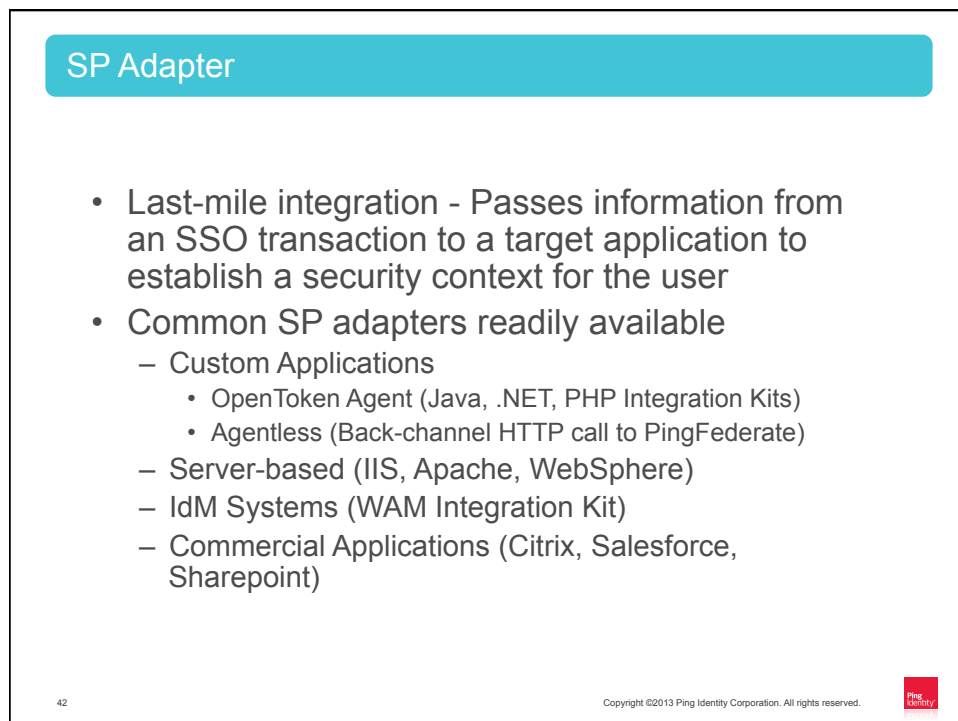
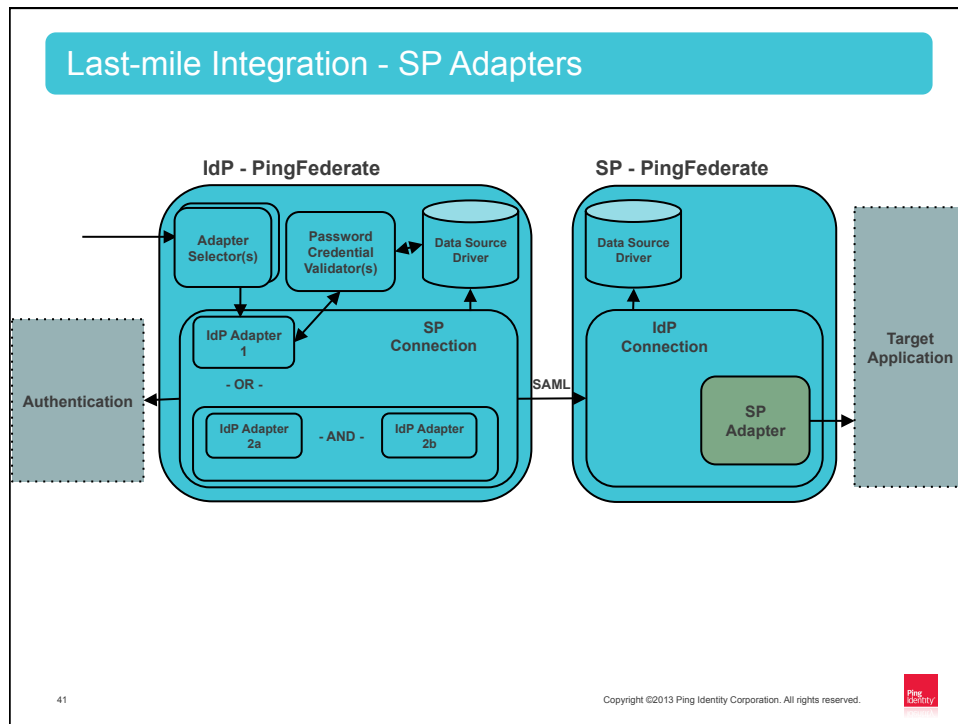


PingFederate SDK Development

## DEVELOPING SP ADAPTER



The Identity Security Company



## Building a Custom SP Adapter

- Interface
  - implements SpAuthenticationAdapter
- Descriptor
  - AuthnAdapterDescriptor getAdapterDescriptor()
- Configuration Retrieval
  - void configure(Configuration configuration)
- Session Creation
  - Serializable createAuthN(SsoContext ssoContext, HttpServletRequest req, HttpServletResponse resp, String resumePath)
- Session Logout
  - boolean logoutAuthN(Serializable authnBean, HttpServletRequest req, HttpServletResponse resp, String resumePath)
- Account Linking
  - String lookupLocalUserId(HttpServletRequest req, HttpServletResponse resp, String partnerIdpEntityId, String resumePath)

43

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

## CODE REVIEW & LAB SP ADAPTER



The Identity Security Company

PingFederate SDK Development

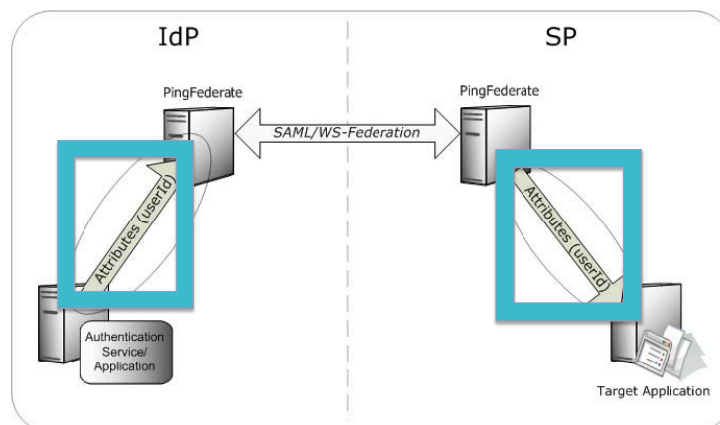
## USING EXISTING INTEGRATION KITS (INTO PLUGGABLE FRAMEWORKS)



The Identity Security Company

### Components of Integration

- Adapters plug into the PingFederate server
- Agents interface with local systems and servers



## Leveraging The “Developer” Integration Kits

- Language Integration (Java, .Net, PHP)
  - OpenToken Adapter
- Agentless Integration
  - ReferenceID Adapter
- Supports IdP and SP Integration

47

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## OpenToken – Java, .Net, or PHP

48

Copyright ©2013 Ping Identity Corporation. All rights reserved.





## OpenToken Security

- The OpenToken contains the attributes as they get passed between the application and PingFederate (on the IdP) or between PingFederate and the application (on the SP)
- Secure Way to Exchange Identity Information
  - Advanced Encryption Standard (AES)
    - AES-128/CBC
    - AES-256/CBC
    - 3DES-168/CBC
- Name/Value Pairs – Supports multi-value attributes
- Multiple Transport Options

49

Copyright ©2013 Ping Identity Corporation. All rights reserved.



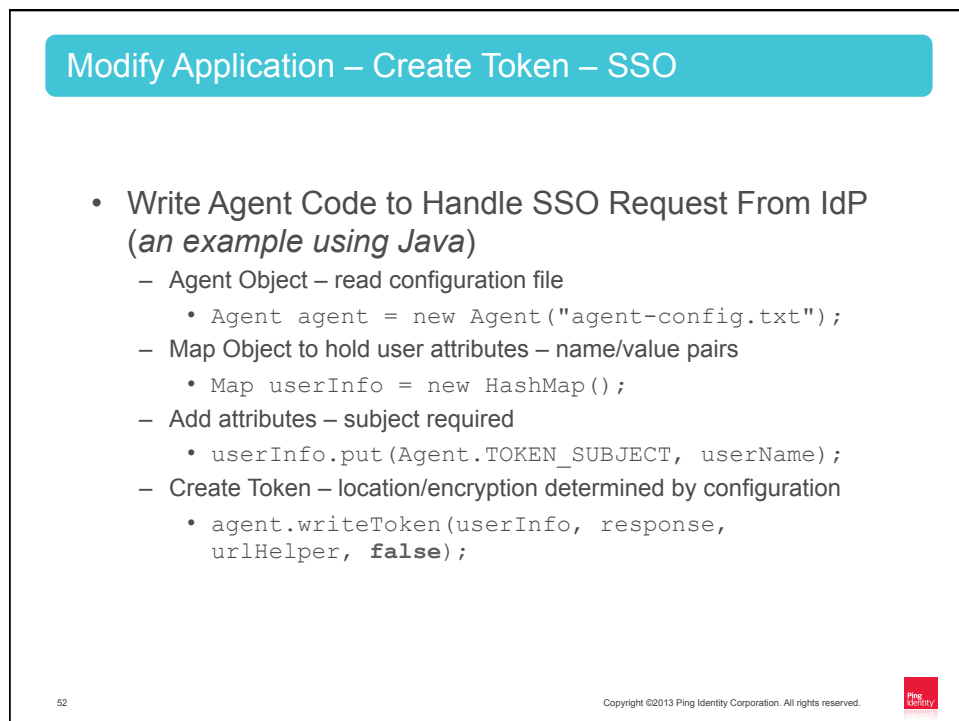
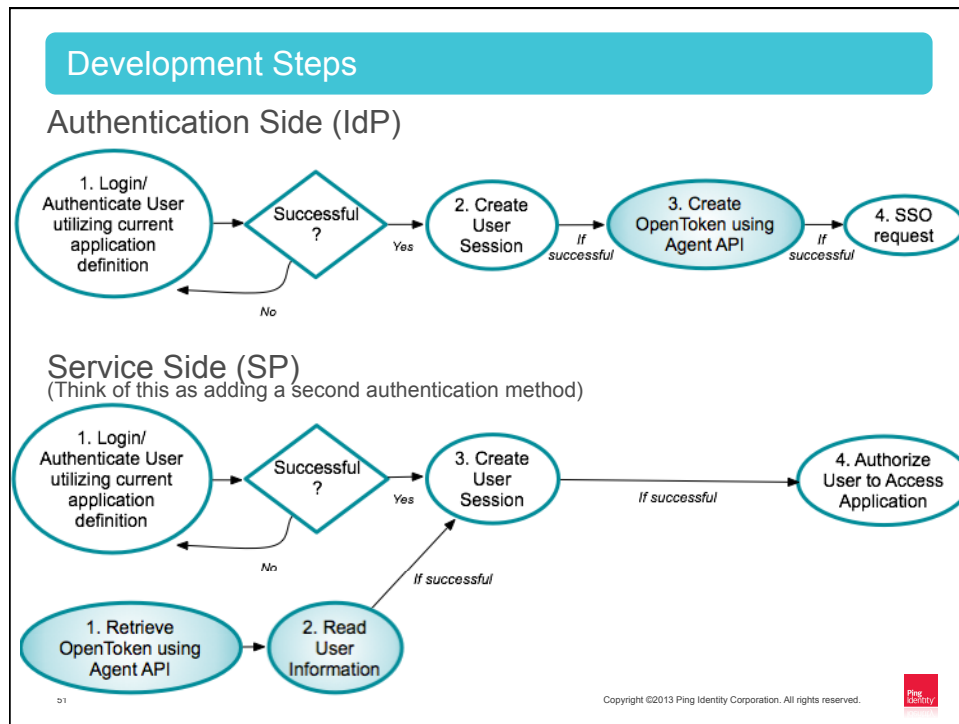
## Transport Options for OpenToken

- Cookie
  - Must be in same domain
  - 4K size limit
  - Web servers tend not to log this info
  - Might not work on mobile or non-browser clients
- Form POST
  - Unlimited data size (browser-dependent)
  - Needs JavaScript enabled (for auto-submit)
- Query Parameter
  - Not recommended for production
  - OpenToken is placed on Browser URL!!
  - Works in everyenvironment

50

Copyright ©2013 Ping Identity Corporation. All rights reserved.





## Modify Application – Read Token – SSO

- Write Agent Code to Receive SSO Request from SP  
(*an example using C#*)
  - Agent Object – read configuration file
    - `Agent agent = new Agent(Server.MapPath("agent-config.txt"));`
  - Read Token – get user attributes
    - `iDictionary userInfo = agent.ReadToken(Request);`
  - Process user attributes – name/value pair
    - `String username = (String)userInfo[Agent.TOKEN_SUBJECT];`
  - Provide services

53

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Things to watch for

- If edit or open the agent-config.txt, make sure you save the file with the proper line ending format. (*Notepad not a good editor to use!*)
- Using a different Java SDK other than Sun's is unsupported and may give you issues
- Remember, Java is Case Sensitive (*best to always assume case sensitivity*)
- Depending on encryption requirements, you may need to install Java Cryptographic Extension
- If using cookies, watch for the domains...
- Remember, the name/value pairs in OpenToken are strings
- Do not let agent-config.txt be downloadable from your application

54

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## ReferenceID – Any Languages that supports HTTP request/response

55

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Development Steps

- Authentication Side
  - Authenticate User, utilizing current application definition
  - Create User Session
  - Create Attribute object then send through backchannel HTTP Request
  - Redirect User passing Response REF as Query Parameter
- Service Side (think of it as adding a second authentication method)
  - Retrieve REF Query Parameter from request
  - Retrieve Attribute Object using backchannel HTTP request with REF Value
  - Read User Information from returned attribute object
  - Create User Session
  - Authorize User to access application

56

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## IdP Side of Things

- ReferenceID Adapter Dropoff Endpoint
  - `http[s]://<pf-host>:<pf-port>/ext/ref/dropoff`
- Authenticate using 1 of 3 ways
  - HTTP Basic
  - Parameters
    - ping.username and ping.password
  - Mutual TLS/SSL
- Reference Attribute Transport Mode
  - **Form Post**
  - Query Parameter
- Send Attributes
  - **JSON**
  - Query Parameters
- Receive Attributes (SLO)
  - **JSON**
  - Properties

57

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## SP Side of Things

- ReferenceID Adapter Dropoff Endpoint
  - `http[s]://<pf-host>:<pf-port>/ext/ref/pickup`
- Authenticate using 1 of 3 ways
  - HTTP Basic
  - Parameters
    - ping.username and ping.password
  - Mutual TLS/SSL
- Reference Attribute Transport Mode
  - **Form Post**
  - Query Parameter
- Send Attributes (Account Linking)
  - **JSON**
  - Query Parameters
- Receive Attributes
  - **JSON**
  - Properties

58

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## A Good Development Approach

- Start Page – normal entry page for your application
  - May login user to application
- Auth Page – add this as a new page, PingFederate comes here
  - If user already authenticated then create attribute object then make call
  - If user not authenticated then go to authentication page, if successful create create attribute object then make call
- SLO Separate Page – add this new page, PingFederate comes here
  - If you need to implement SLO for your application.

59

Copyright ©2013 Ping Identity Corporation. All rights reserved.

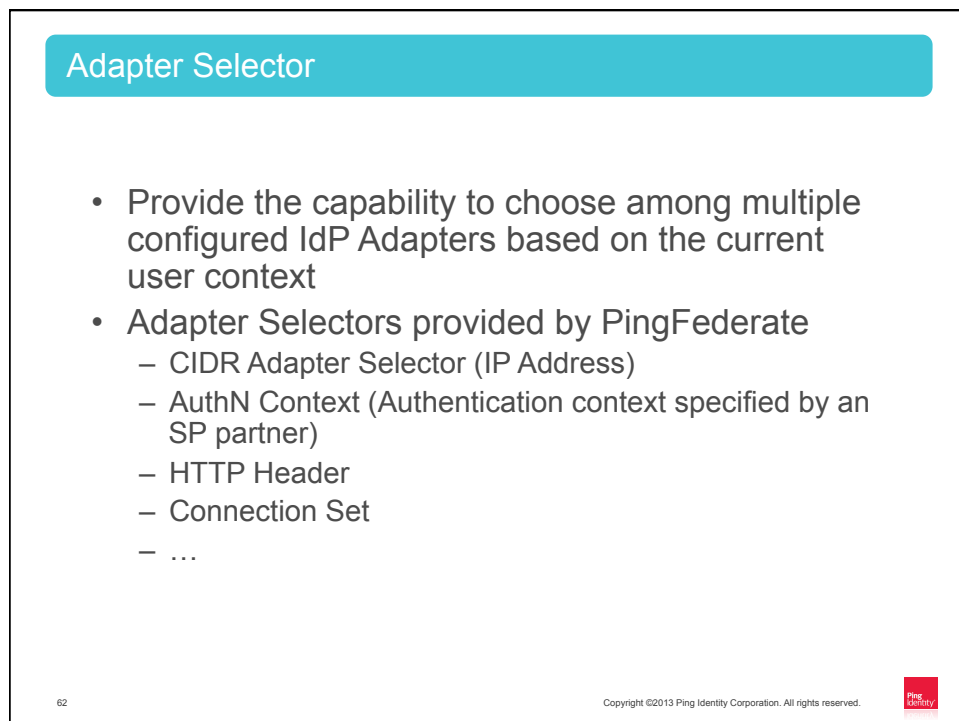
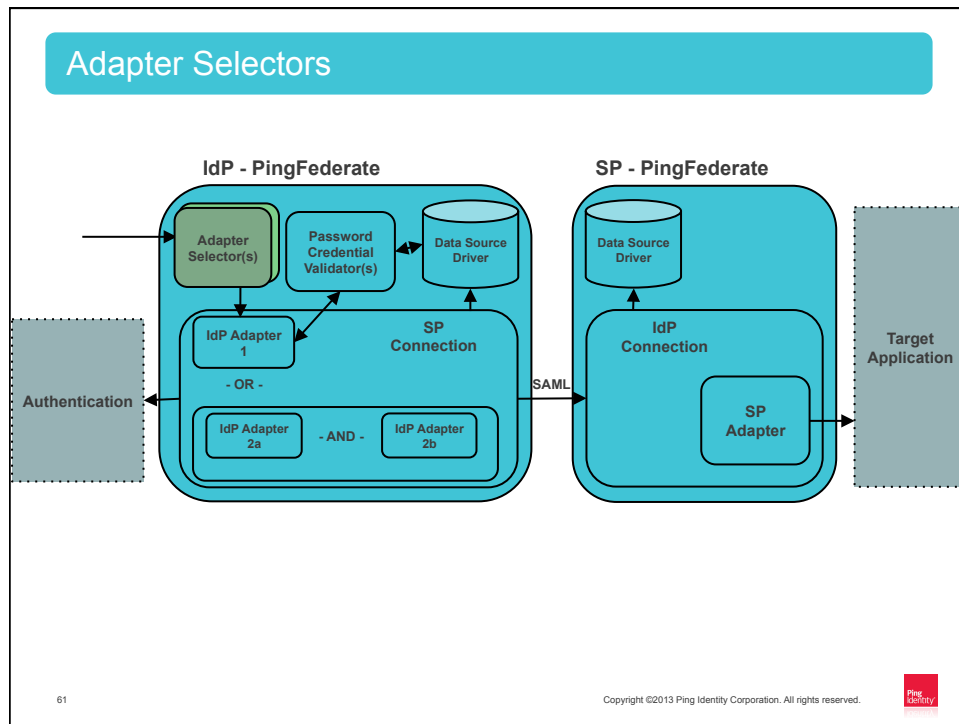


PingFederate SDK Development

## DEVELOPING ADAPTER SELECTOR



The Identity Security Company



## Building a Custom Adapter Selector

- Interface
  - implements AdapterSelector
- Descriptor
  - PluginDescriptor getPluginDescriptor()
- Configuration Retrieval
  - void configure(Configuration configuration)
- Adapter Selection
  - AdapterSelectorContext selectContext(HttpServletRequest req, HttpServletResponse resp, Map<String, String> mappedAdapterIdsNames, Map<String, Object> extraParameters, String resumePath)
- Post-Processing Callback
  - void callback(HttpServletRequest req, HttpServletResponse resp, Map authnIdentifiers, String adapterInstanceId, AdapterSelectorContext adapterSelectorContext)

63

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

## CODE REVIEW & LAB ADAPTER SELECTOR



The Identity Security Company™

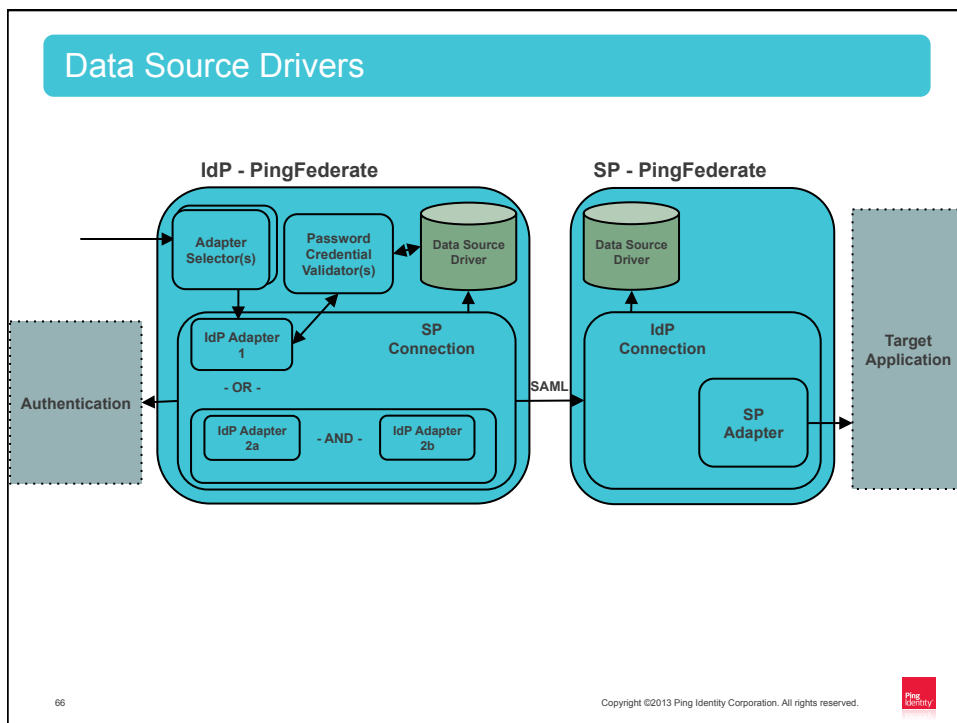


PingFederate SDK Development

## DEVELOPING DATA SOURCE DRIVER



The Identity Security Company



## Data Source Driver

- Provide connection and filtering facilities to (typically) query additional user attributes from an external source
- Data Source Drivers provided by PingFederate
  - LDAP
  - JDBC

67

Copyright ©2013 Ping Identity Corporation. All rights reserved.



## Building a Custom Data Source Driver

- Interface
  - implements CustomDataSourceDriver
- Descriptor
  - SourceDescriptor getSourceDescriptor()
  - CustomDataSourceDriverDescriptor(ConfigurableDriver adapter, String type, AdapterConfigurationGuiDescriptor adapterConfigurationGuiDesc, FilterFieldsGuiDescriptor fieldsDescriptor)
- Configuration Retrieval
  - void configure(Configuration configuration)
- Connection Testing
  - boolean testConnection()
- Available Fields Retrieval
  - List<String> getAvailableFields()
- Query Handling
  - Map<String, Object> retrieveValues(Collection<String> attributeNamesToFill, SimpleFieldList filterConfiguration)

68

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

# CODE REVIEW & LAB CUSTOM DATA SOURCE DRIVER

 **The Identity Security Company**

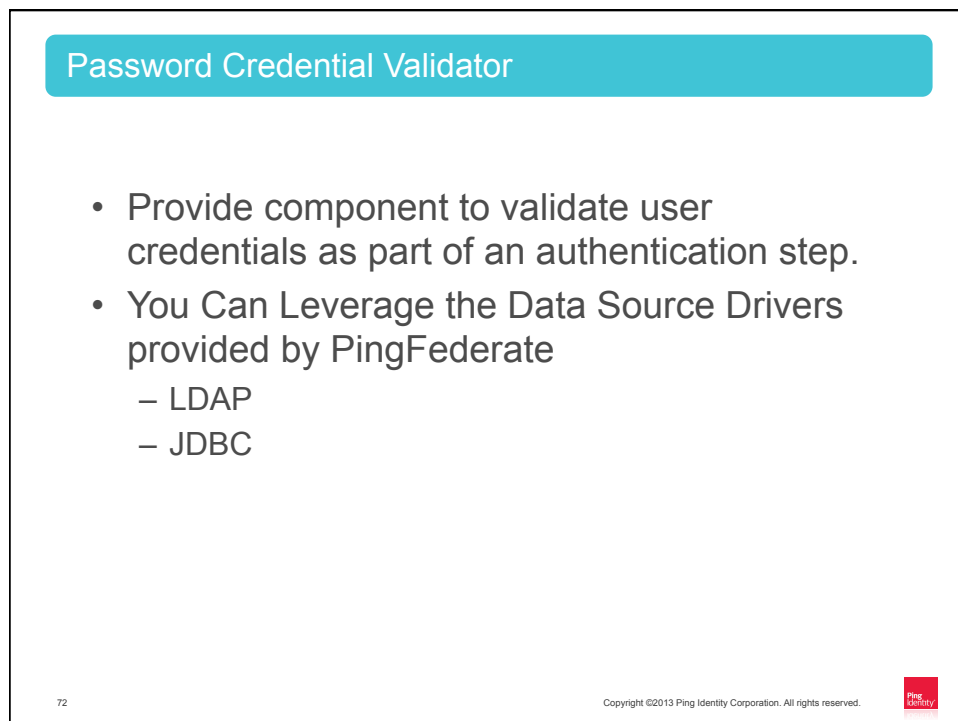
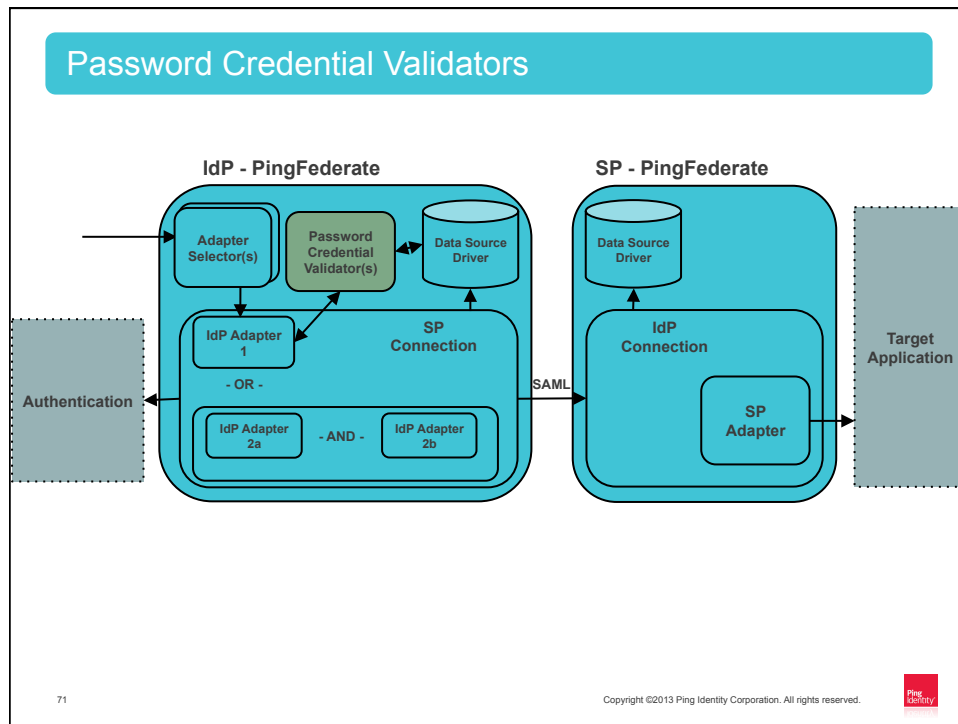
Κατασκευή της ιστοσελίδας ασφαλιών ασφαλιών

PingFederate SDK Development

# DEVELOPING PASSWORD CREDENTIAL VALIDATOR

 **The Identity Security Company**

Κατασκευή της ιστοσελίδας ασφαλιών ασφαλιών



## Building a Password Credential Validator

- **Interface**
  - implements PasswordCredentialValidator
- **Descriptor**
  - PluginDescriptor getPluginDescriptor ()
  - pluginDescriptor.setAttributeContractSet(Collections.singleton(USERNAME))
  - pluginDescriptor.setSupportsExtendedContract(false)
- **Configuration Retrieval**
  - void configure(Configuration configuration)
- **Process Password**
  - AttributeMap processPasswordCredential(String username, String password) throws PasswordValidationException

73

Copyright ©2013 Ping Identity Corporation. All rights reserved.



PingFederate SDK Development

## CODE REVIEW & LAB CUSTOM PASSWORD CREDENTIAL VALIDATOR



The Identity Security Company