Introduction to OpenEdge REST

Session 426 – OE REST, Part 1 of 2

Dustin Grau – dgrau@progress.com

Principal Solutions Consultant

PROGRESS BravePoint

PUG CHALLENGE EXCHANGE AMERICAS

Introductions

PUG CHALLENGE EXCHANGE AMERICAS

“The times they are a changin“

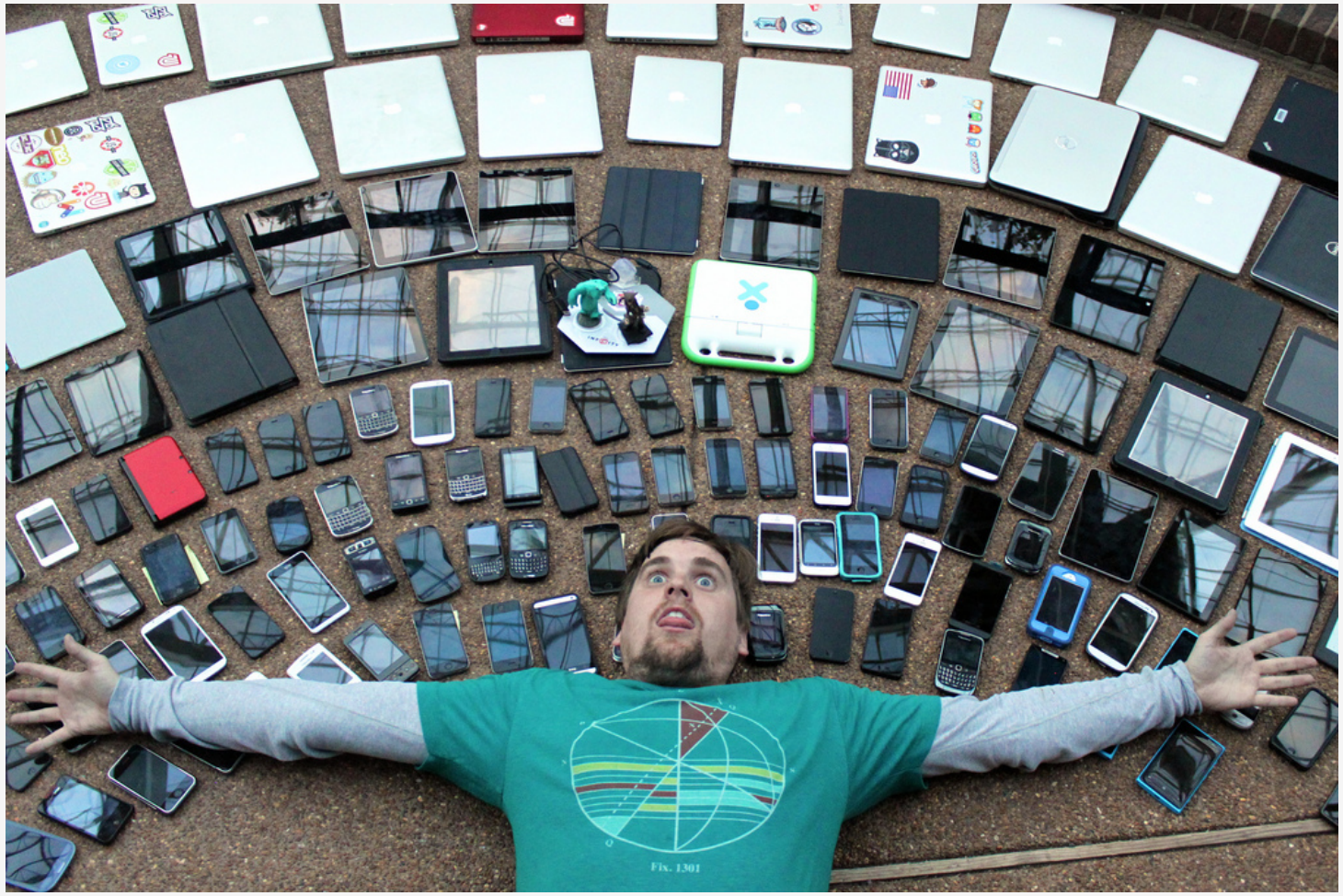
“The times they are a changin“

REST is ReST: Representational State Transfer

- Resource-based methodology that uses verbs to interact with nouns
  - GET (read) http://localhost:8080/app/customer
  - POST (create), PUT (update), Delete (remove)
- Content may be part of the URI or the request body
  - Depends on the HTTP verb used
  - http://localhost:8080/app/customer?CustNum=1
  - More on this in Part 2
- Many URI's may refer to the same resource, for different purposes
  - GET http://localhost:8080/app/invoice/customer
  - GET http://localhost:8080/app/order/customer

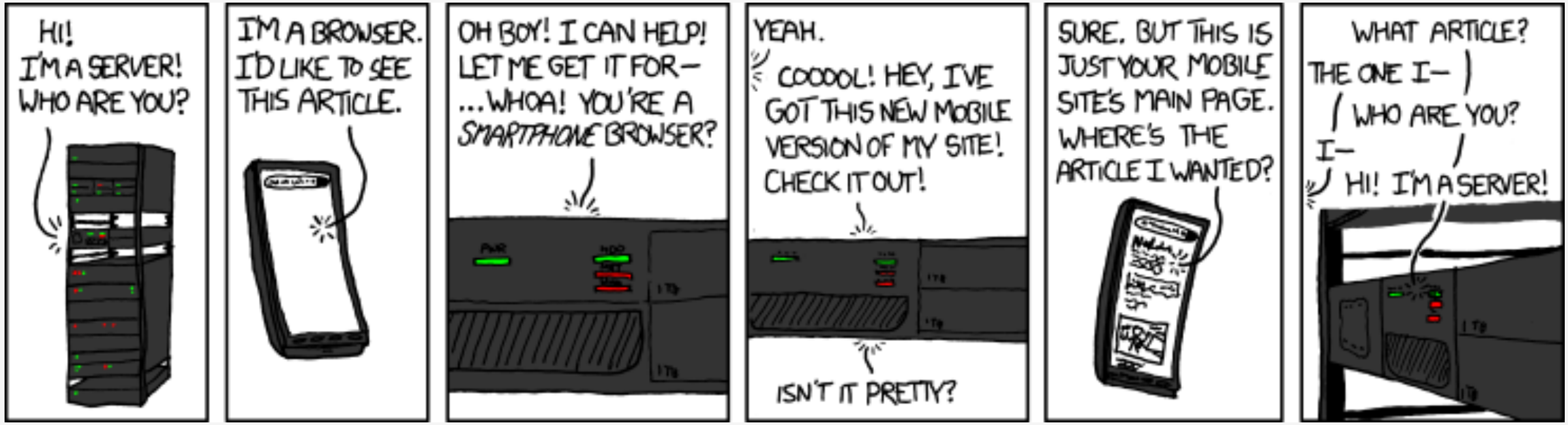
REST Doesn't Care

- The server should not care how the data is ultimately presented to the user



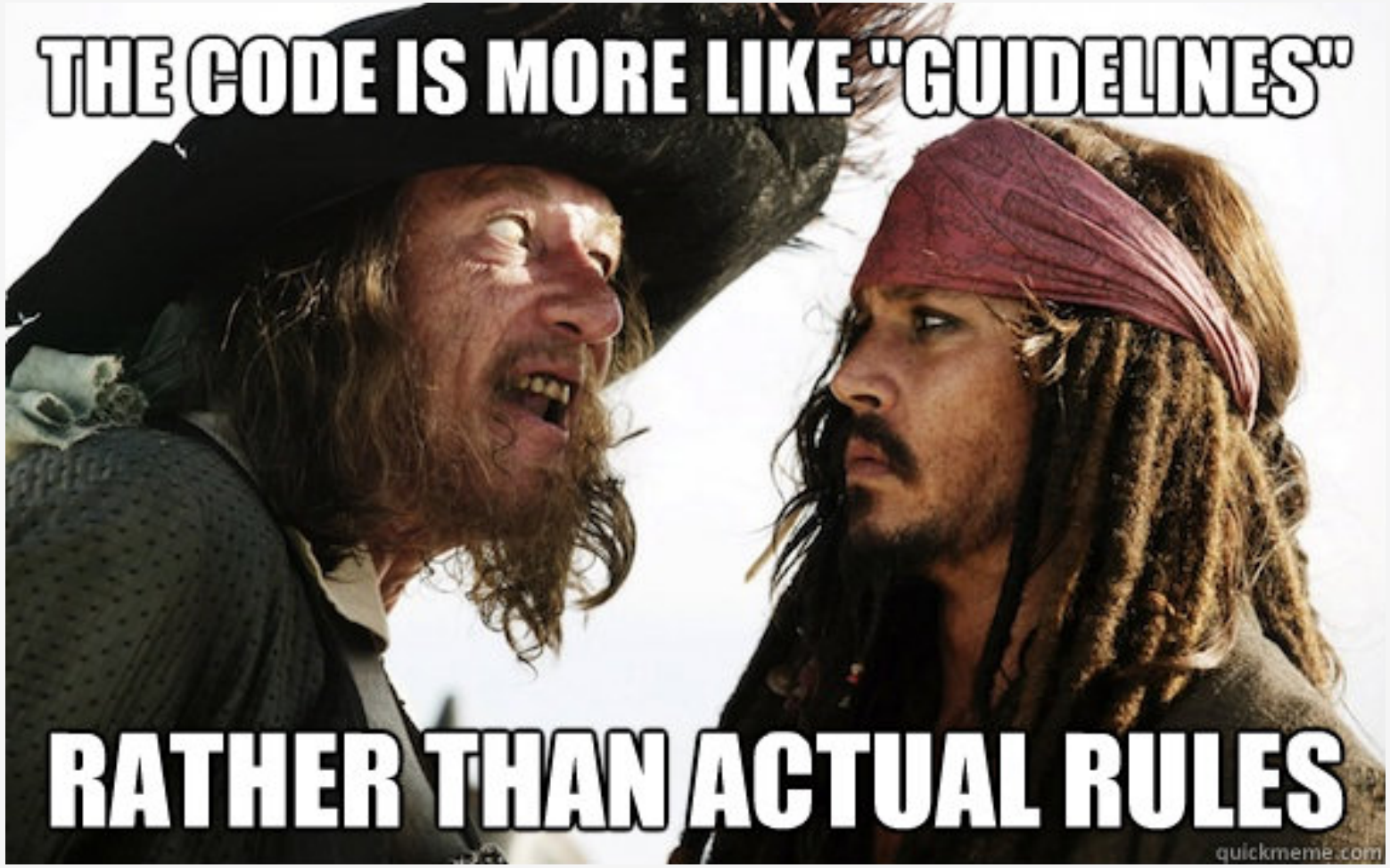
Persistence is not RESTful

- Each request should have just enough information to complete a request



REST Code of Conduct

- Data is requested and delivered in a uniform manner (eg. JSON), but open to interpretation



OpenEdge 11.5

PUG CHALLENGE EXCHANGE AMERICAS

OpenEdge REST Adapter

- Introduced several versions ago (11.2 w/ OE Mobile)
  - Provides performance and scalability
  - Means of direct data access via the web
- Utilizes Apache Tomcat as HTTP front-end
  - Security via Spring framework in Tomcat
  - Alternative to WSA or WebSpeed
- OE 11.5 adds Pacific AppServer (PAS)
  - Retains the “Classic AppServer”
  - We will focus on the “Classic” aspect
    - Roy Ellis has a full presentation on PASOE

Progress Developer Studio

- PDSOE comes with “Tomcat in the Box”
  - Not meant for production use!
  - Has limited configuration changes (ie. None)
- REST Service vs. Mobile Service project types
  - Manual mapping vs. annotation-driven mapping
  - Design-time catalog file (mobile service)
  - More on this in Part 2
- Generation of service definition (PAAR file)
  - More on this in Part 2
- Support for PASOE
  - Similar to WebSpeed (Messenger + Broker)
  - AppServer = blocking, WebSpeed = streaming

Configuration

PUG CHALLENGE EXCHANGE AMERICAS

AppServer Configurations

- Examples in terms of Classic AppServer
- Remember that Tomcat is involved
  - URI's reflect the webapp in use
  - http://<server>:<port>/<webapp>/rest/<service>/<resource>[/<sub-resource>]
- Configure the AppServer
  - State-free operating mode (remember session != state)
  - Tomcat will handle our session management
  - Configure server/port in runtime.properties
- Multiple AppServers may be used
  - Primary application
  - Security (e.g. realm auth)

Sample runtime.properties

```
<?xml version="1.0" encoding="UTF-8"?>
<AppServer:AppServerEndpointBean …>
  <AppServer:userName xs:nil="true"/>
  <AppServer:password xs:nil="true"/>
  <AppServer:extraInfo xs:nil="true"/>
  <AppServer:sessionMode>1</AppServer:sessionMode>
  <bpm:ApplicationRuntimeProperties>
    <bpm:appServiceProtocol>AppserverDC</bpm:appServiceProtocol>
    <bpm:appServiceHost>localhost</bpm:appServiceHost>
    <bpm:appServicePort>3066</bpm:appServicePort>
    <bpm:appServiceName>yourbroker</bpm:appServiceName>
    …
  </bpm:ApplicationRuntimeProperties>
</AppServer:AppServerEndpointBean>
```

Tomcat Configurations

- Use PDSOE's Tomcat for development
  - Production requires Tomcat be installed
  - HTTPS is crucial for security (credentials)
- Set your security model
  - WEB-INF/web.xml
  - contextConfigLocation in context-param block
- Apply security to URI's via security model
  - WEB-INF/appSecurity-*.xml
  - End-points are controlled via intercept-url rules
- Test via http://<server>:<port>/<webapp>/rest (WADL)
- Deploy/Undeploy vs. Republish (Windows has gotchas)

Spring Framework

- Identity management
- AuthN (who) vs. AuthZ (what)
  - Think: passport vs. keys
- Basic vs. Form authentication models
  - Basic requires a special header w/ token on each request
  - Form provides true logoff enforcement (avoids replay attack)
- Anonymous access – first default, simplest
- Tomcat Users – adding auth complexity
- OE Realm – true SSO potential
- Client-Principal Object (CP Token)
  - Created automatically by Tomcat
  - Even anonymous users get a token!

Sample web.xml

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    <!-- USER EDIT: Select which application security model to employ
      /WEB-INF/appSecurity-anonymous.xml
      /WEB-INF/appSecurity-basic-local.xml
      …
    -->
    /WEB-INF/appSecurity-form-oerealm.xml
  </param-value>
</context-param>
```

Sample appSecurity-form-oerealm.xml

```
<intercept-url pattern="/rest/si/catalog/**"
               access="hasAnyRole('ROLE_ANONYMOUS', 'ROLE_EndUser')"/>

<intercept-url pattern="/rest/si/customer/*”
               access=”hasRole(‘ROLE_CustAdmin’)"/>

<intercept-url pattern="/static/js/*" method="GET"
               access="permitAll()"/>

<anonymous enabled="true" />
```

OERealm Security

- Still relies on Spring security framework (an industry standard)
  - OE Realm is an information conduit, not the actual authenticator
- Uses a pre-defined interface to access an ABL class (IHybridRealm)
  - Performs lookup of user by some UserID (numeric)
  - Confirms account is NOT locked, NOT expired, IS enabled
  - Compares password via your hash process
- Spring manages a Tomcat session (+CP token)
  - CP token provides identification for authorization, access to URI's
- You should secure the access between Tomcat and authenticating AppServer
  - Use a private, pre-generated client-principal object
  - Mike Jacobs covers this in his session on OE Realm Security

Sample appSecurity-form-oerealm.xml

```
<b:bean id="OERealmUserDetails"
        class="com.progress.rest.security.OERealmUserDetailsImpl" >
        <b:property name="realmURL" value="AppServerDC://localhost:3066/oerealm" />
        <b:property name="realmClass" value="Path.To.Security.RealmClass" />
        <b:property name="grantedAuthorities" value="ROLE_EndUser" />
        <b:property name="rolePrefix" value="ROLE_" />
        <b:property name="roleAttrName" value="ATTR_ROLES" />
        <b:property name="enabledAttrName" value="ATTR_ENABLED" />
        <b:property name="lockedAttrName" value="ATTR_LOCKED" />
        <b:property name="expiredAttrName" value="ATTR_EXPIRED" />
        <b:property name="realmPwdAlg" value="0" />
       <b:property name="realmTokenFile" value="AuthRealm.cp" />
        <b:property name="certLocation" value="" />
  </b:bean>
```

The IHybridRealm Interface

```
method public character GetAttribute ( input piUserID as integer, input pcAttrName as character ).
method public character extent GetAttributeNames ( input piUserID as integer ).
method public character extent GetUsernames ( ).
method public character extent GetUsernamesByQuery ( input pcQueryString as character ).
method public character extent GetUsernamesByQuery ( input pcAttrName  as character, input pcAttrValue as character ).
method public logical RemoveAttribute ( input piUserID as integer, input pcAttrName as character ).
method public logical SetAttribute ( input piUserID as integer, input pcAttrName as character, input pcAttrValue as character ).
method public logical ValidatePassword ( input piUserID as integer, input pcPassword as character ).
method public logical ValidatePassword ( input piUserID as integer, input pcDigest as character,
                                         input pcNonce as character, input pcTimestamp as character ).
method public integer ValidateUser ( input pcUsername as character ).
```

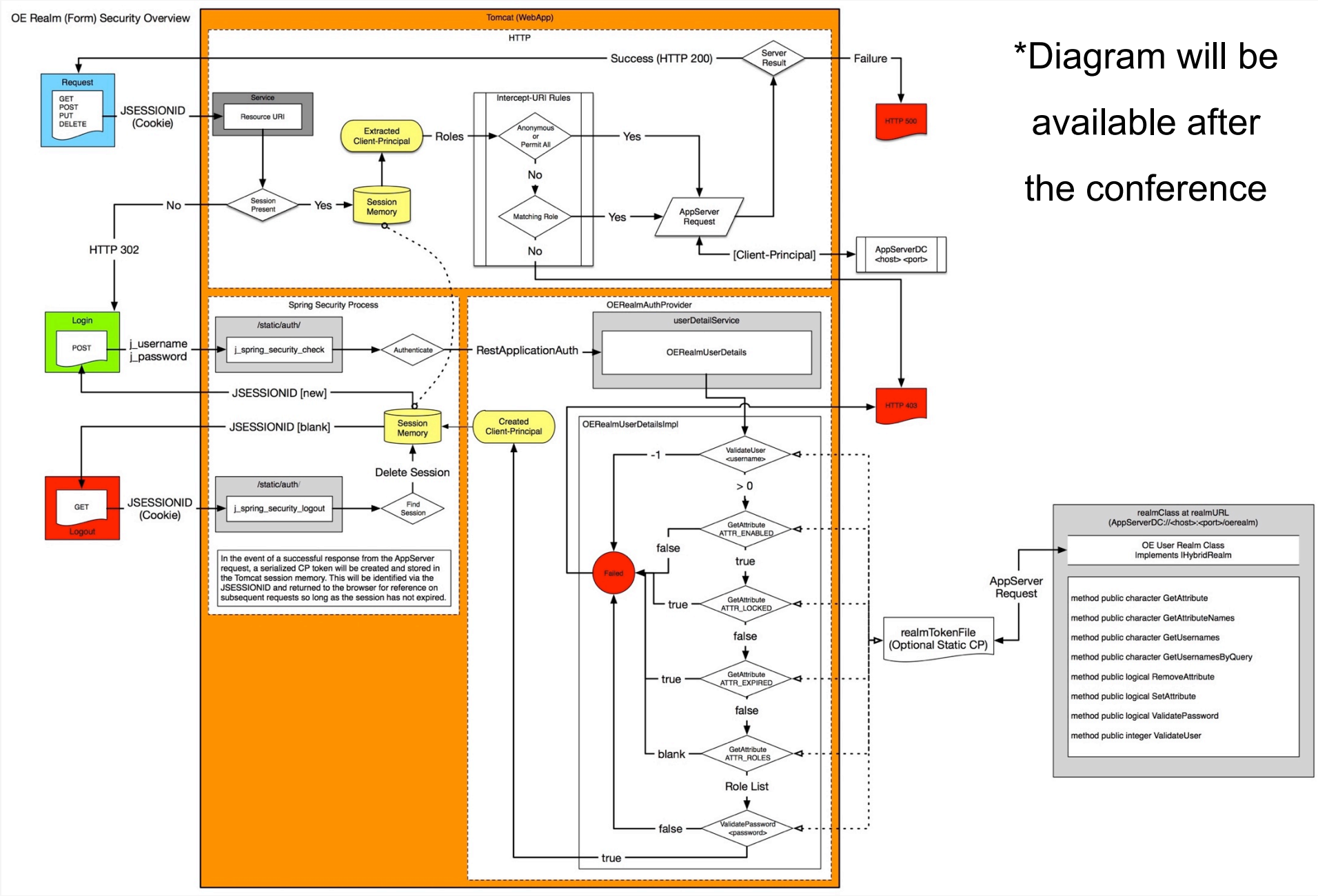


*Diagram will be available after the conference

Management

PUG CHALLENGE EXCHANGE AMERICAS

Deploying to Non-Development Servers

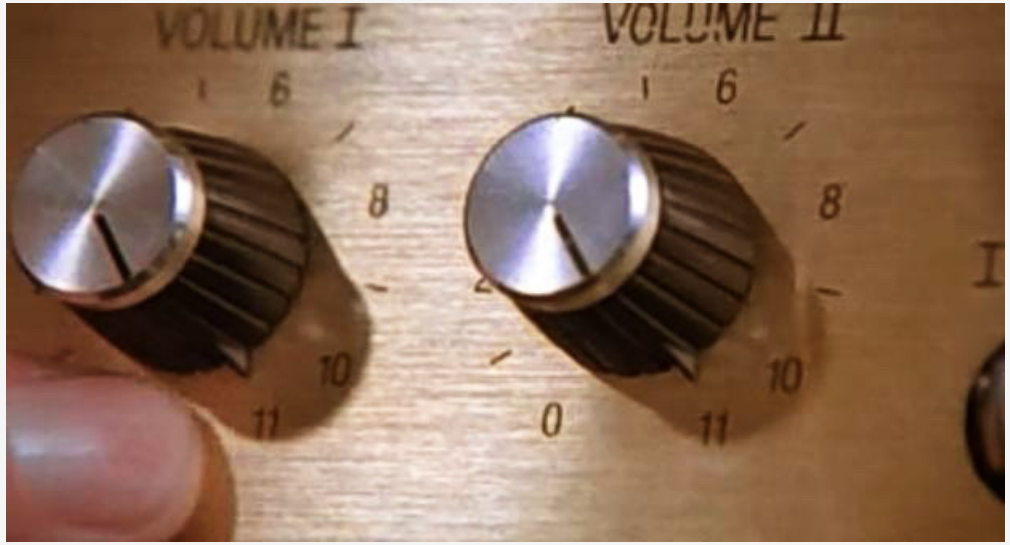
- Install minimum versions Java 1.7 and Tomcat 7
  - Else errors will be thrown about mismatched libraries
  - Java libraries are copied to any WAR files created
- When bundling a WAR file, deploy as WebApp
  - Right-click on a defined service in PDSOE project
  - Select “Export Services Incrementally”
  - Use Tomcat management (http://localhost:8080)
- Configure any “Classic AppServer” instances normally
  - OpenEdge Management Console (http://localhost:9090)
  - Directly via ubroker.properties files in $DLC/properties

Accessing a REST Service

- JavaScript libraries (e.g. jQuery)
  - $.ajax(…)
- Postman or RESTclient
  - Browser plugins for Chrome, Firefox
- Just use your browser!
  - Ok, this is mainly for GET's
- If it can speak HTTP…

When Things Go Sideways

- Where is my log file?!
  - /WEB-INF/adapters/logs/<service>.log
  - PDSOE Tomcat: <DLC>/servers/tomcat/webapps/<webapp>
  - Standalone Tomcat: <CATALINA_HOME>/webapps/<webapp>
- When in doubt, use TRACE/DEBUG modes
  - Found in WEB-INF/classes/log4j.properties



Demonstration

Quick setup of a new REST project

Thank You!

- “REST Support for B2B Access to Your OpenEdge AppServer”
  - Kumar Navneet & David Cleary, Progress Exchange 2014
- “210: OE Realm and Your Application’s Authentication Process”
  - Kumar Navneet & Mike Jacobs, PUG Challenge Americas 2015
- “402: OpenEdge REST for Any Application”
  - Matt Baker, PUG Challenge Americas 2015
- Part 2 of this presentation covers actual development!

PUG
CHALLENGE
EXCHANGE
AMERICAS