

Webservices
Soap – Rest

Problématiques d'Intégration ...

Laurent KIEFFER

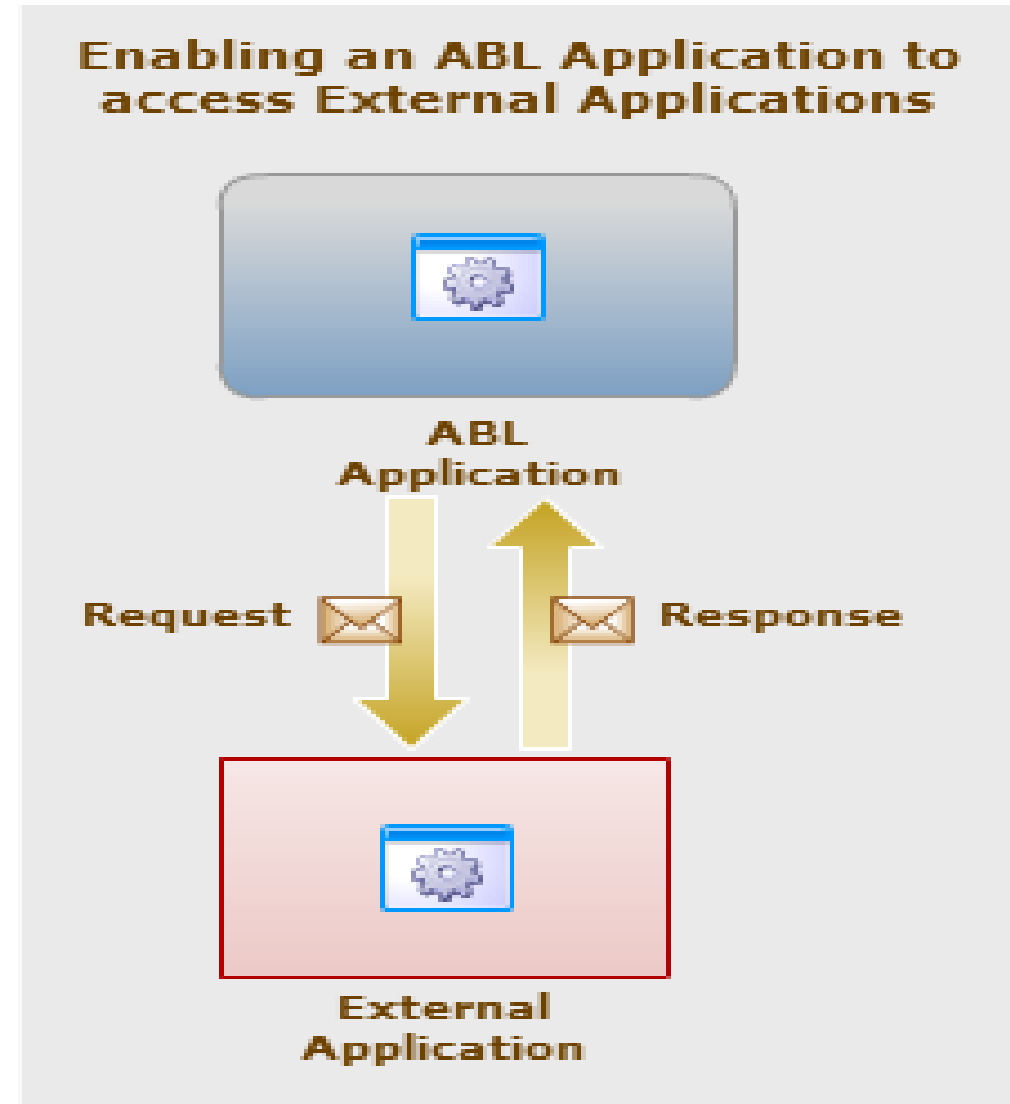
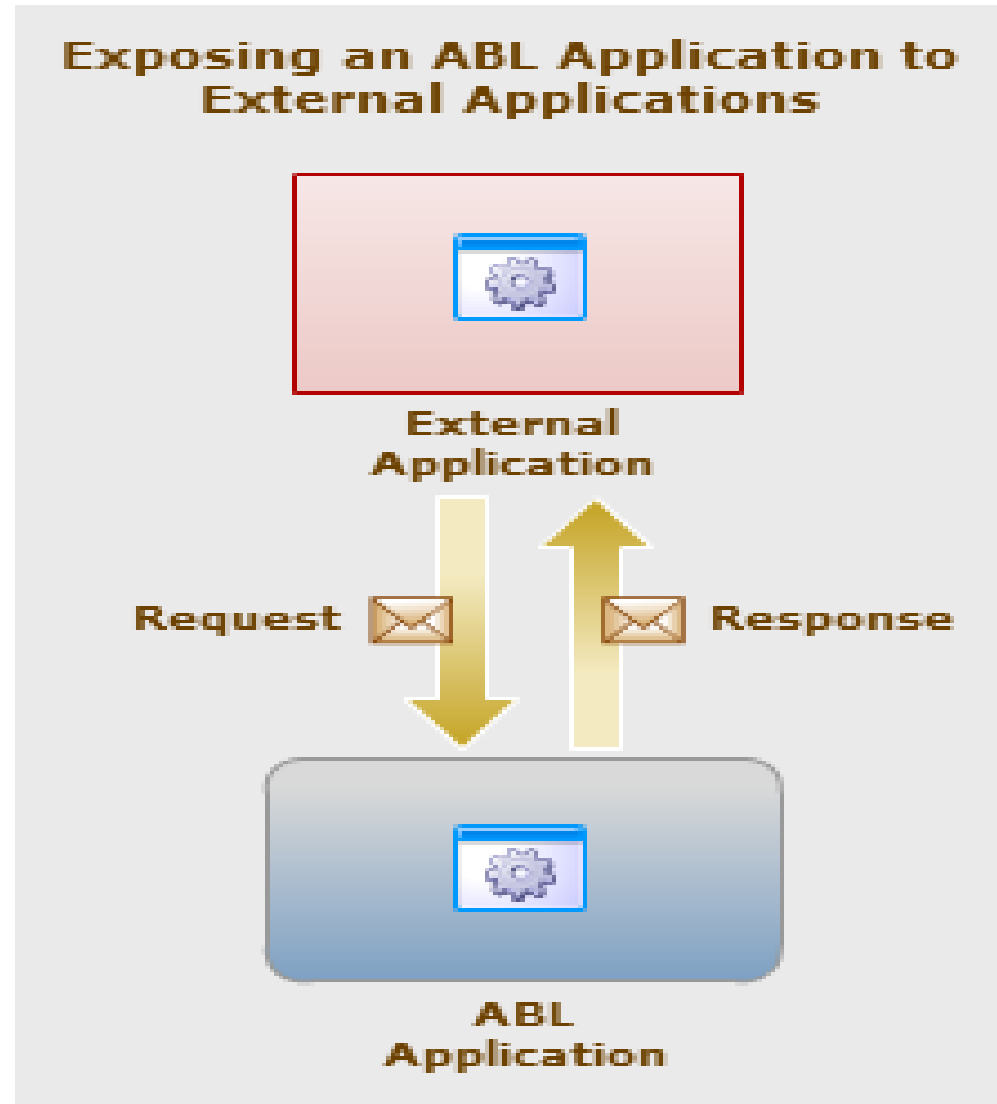
16 Décembre 2014



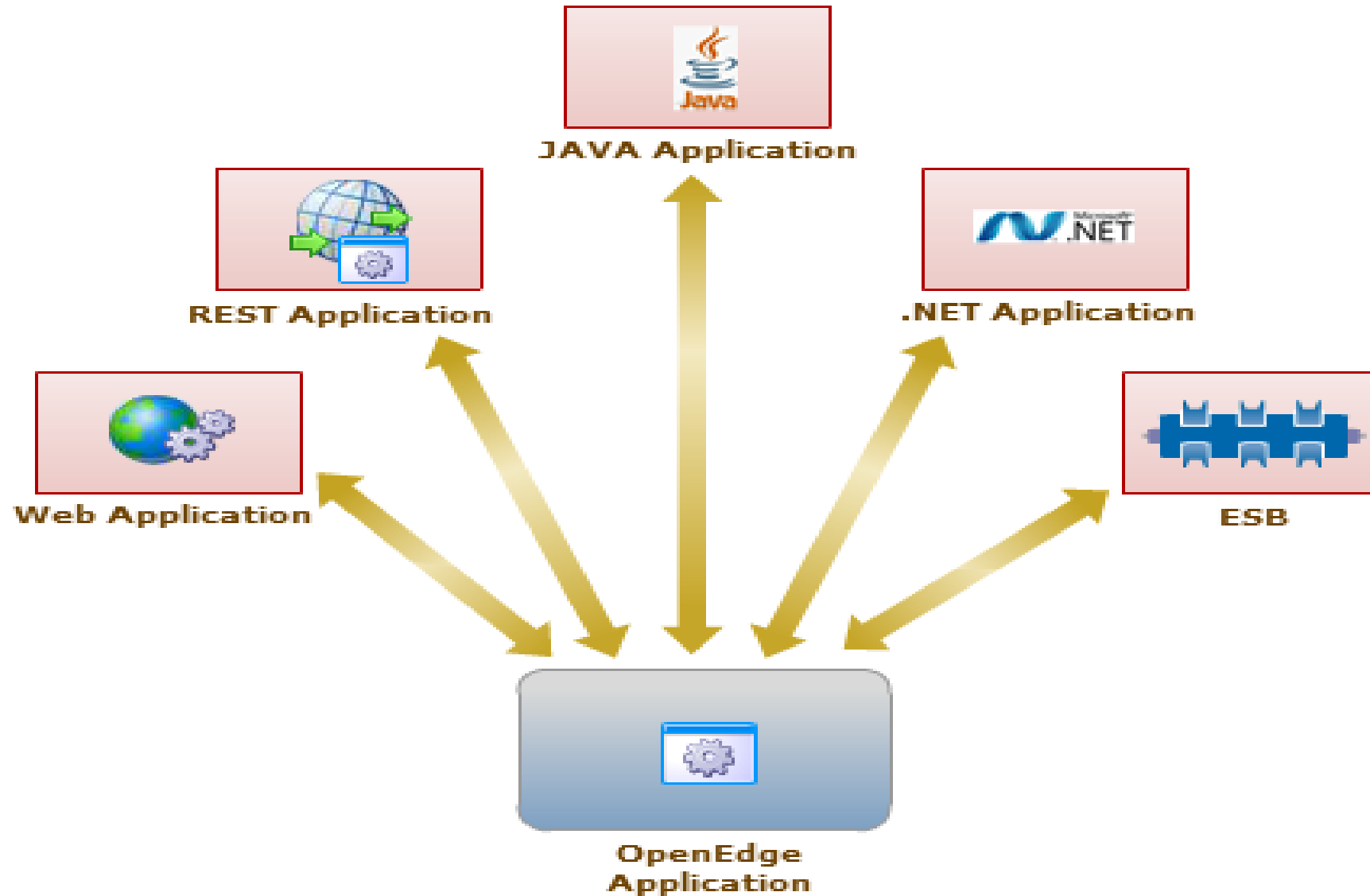
Introduction

- Exposer une application ABL à d'autres applications
- Permettre à une application ABL d'accéder à d'autres applications

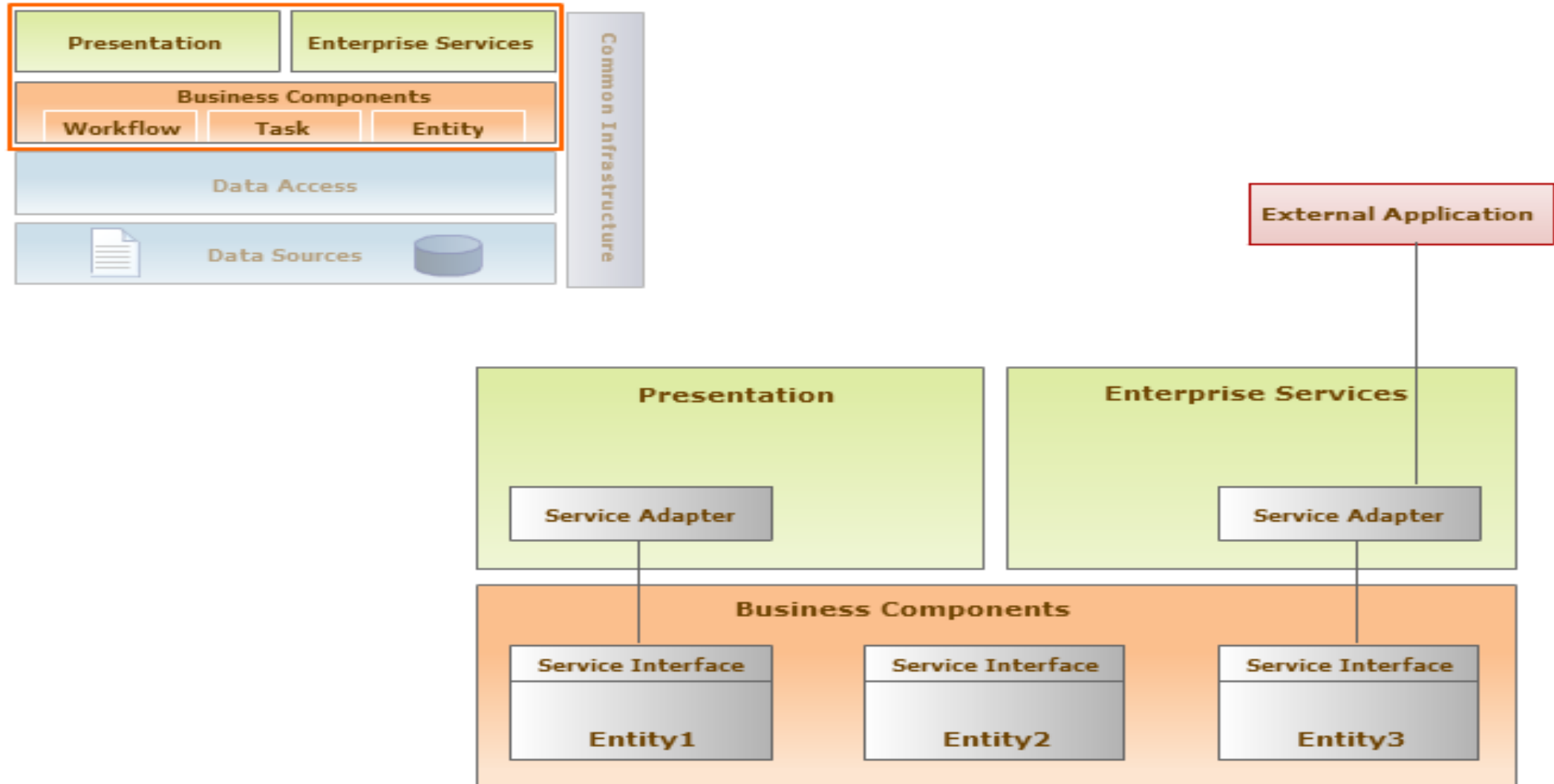
Problématique d'intégration?



Exposer une application ABL

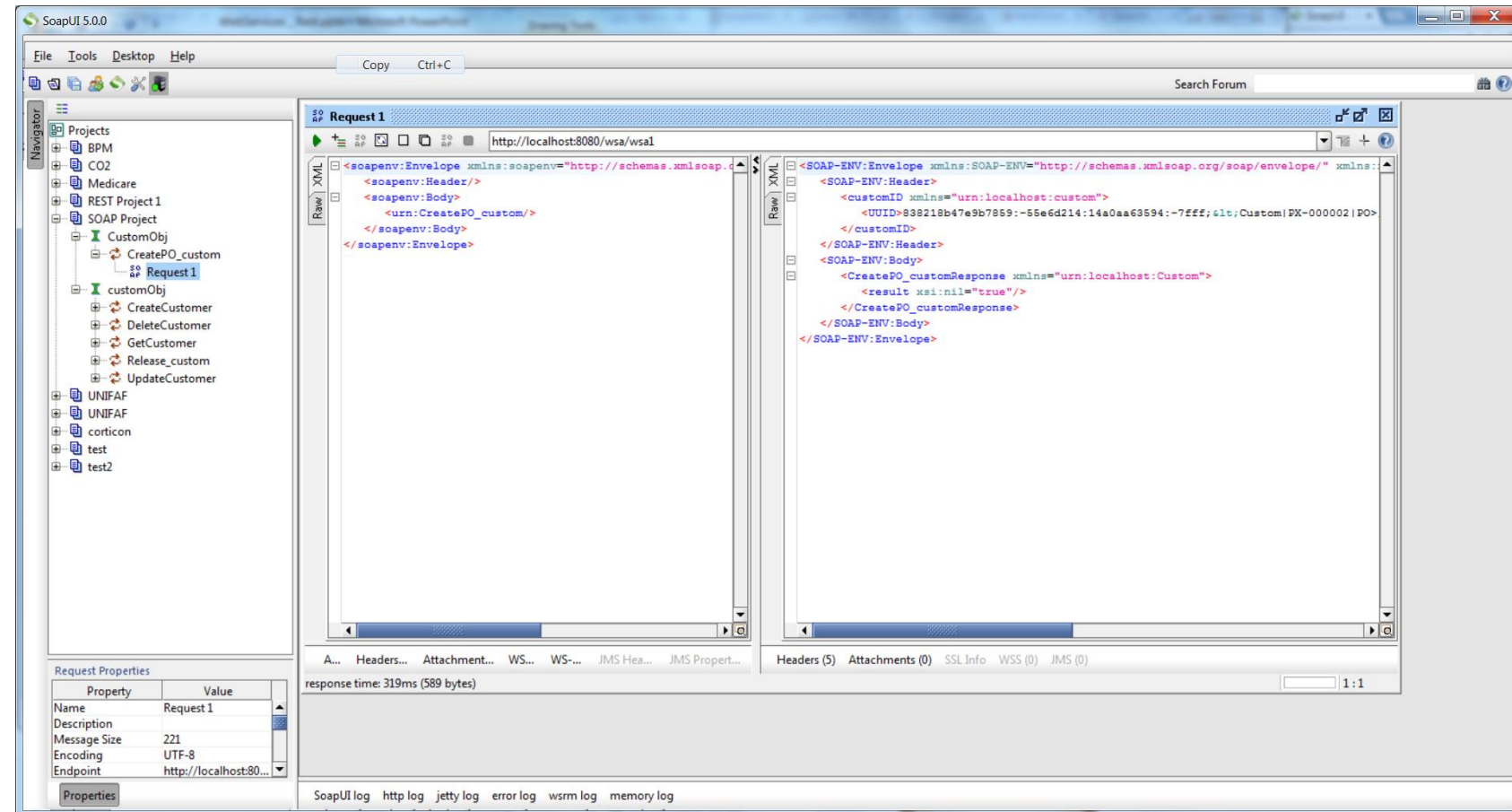


Comment une application basée sur OERA fournit des Services



Outil de test utilisé en exemple : SOAP UI

- Permet de tester
 - Les Webservices Soap
 - Les Webservices Rest

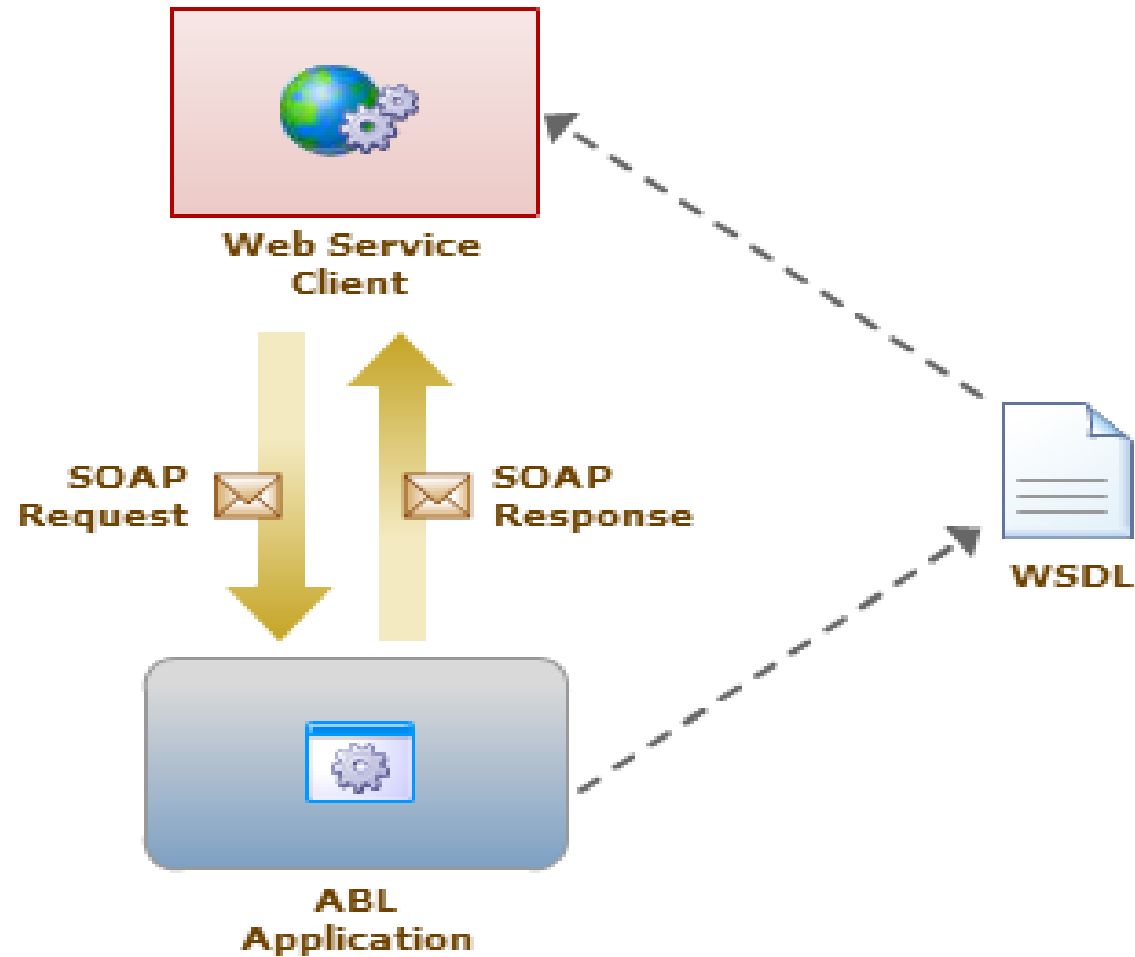


Procédure utilisée pour les exemples

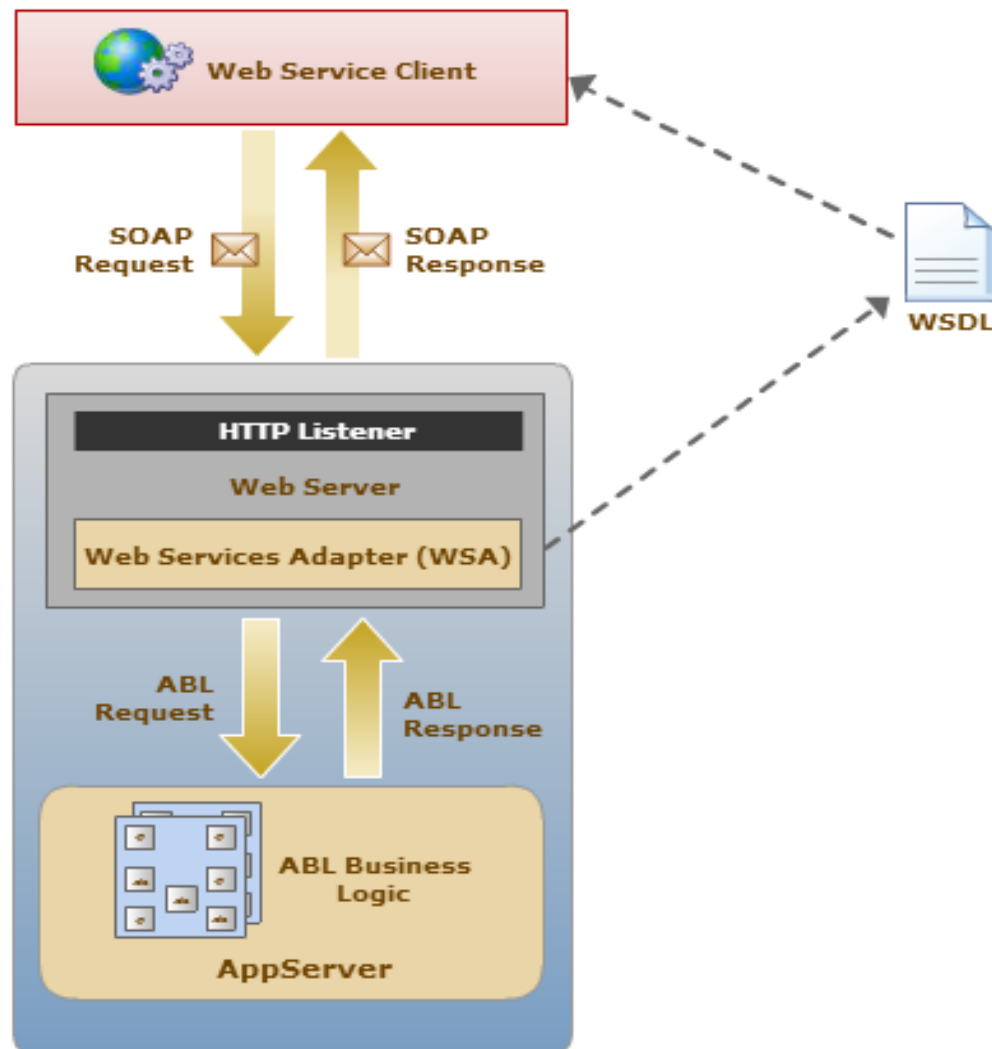
- Custom.p qui comporte une procédure interne : GetCustomer

```
@openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="false").  
PROCEDURE GetCustomer:  
    DEFINE INPUT PARAMETER custId AS INTEGER NO-UNDO.  
    DEFINE OUTPUT PARAMETER TABLE FOR ttCustomer.  
    FIND FIRST Customer WHERE Customer.CustNum = custId NO-ERROR.  
    IF AVAILABLE Customer THEN  
        DO:  
            CREATE ttCustomer.  
            BUFFER-COPY Customer TO ttCustomer.  
        END.  
  
    RELEASE Customer.  
  
END PROCEDURE.
```

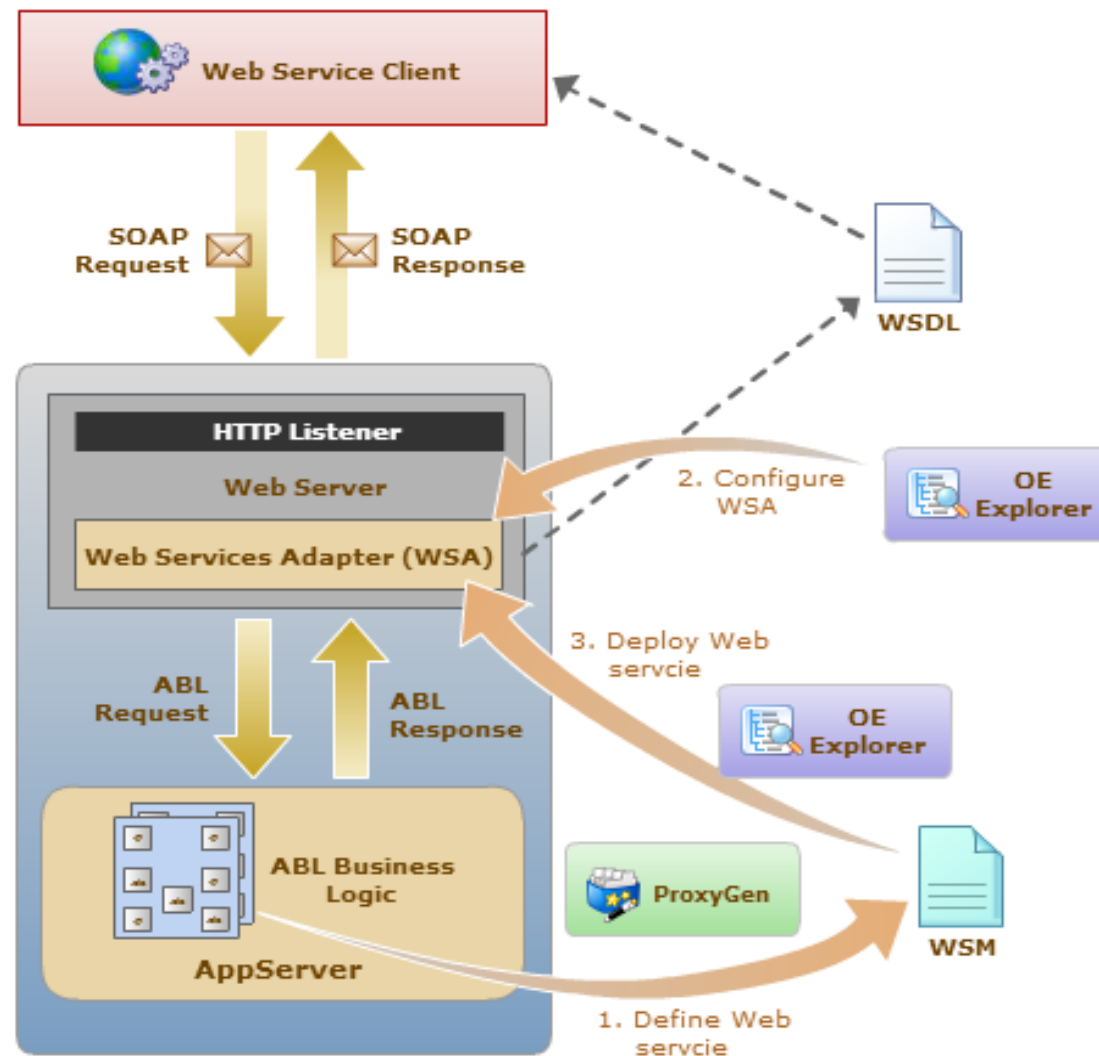
Exposer une application ABL sous forme de Webservices SOAP



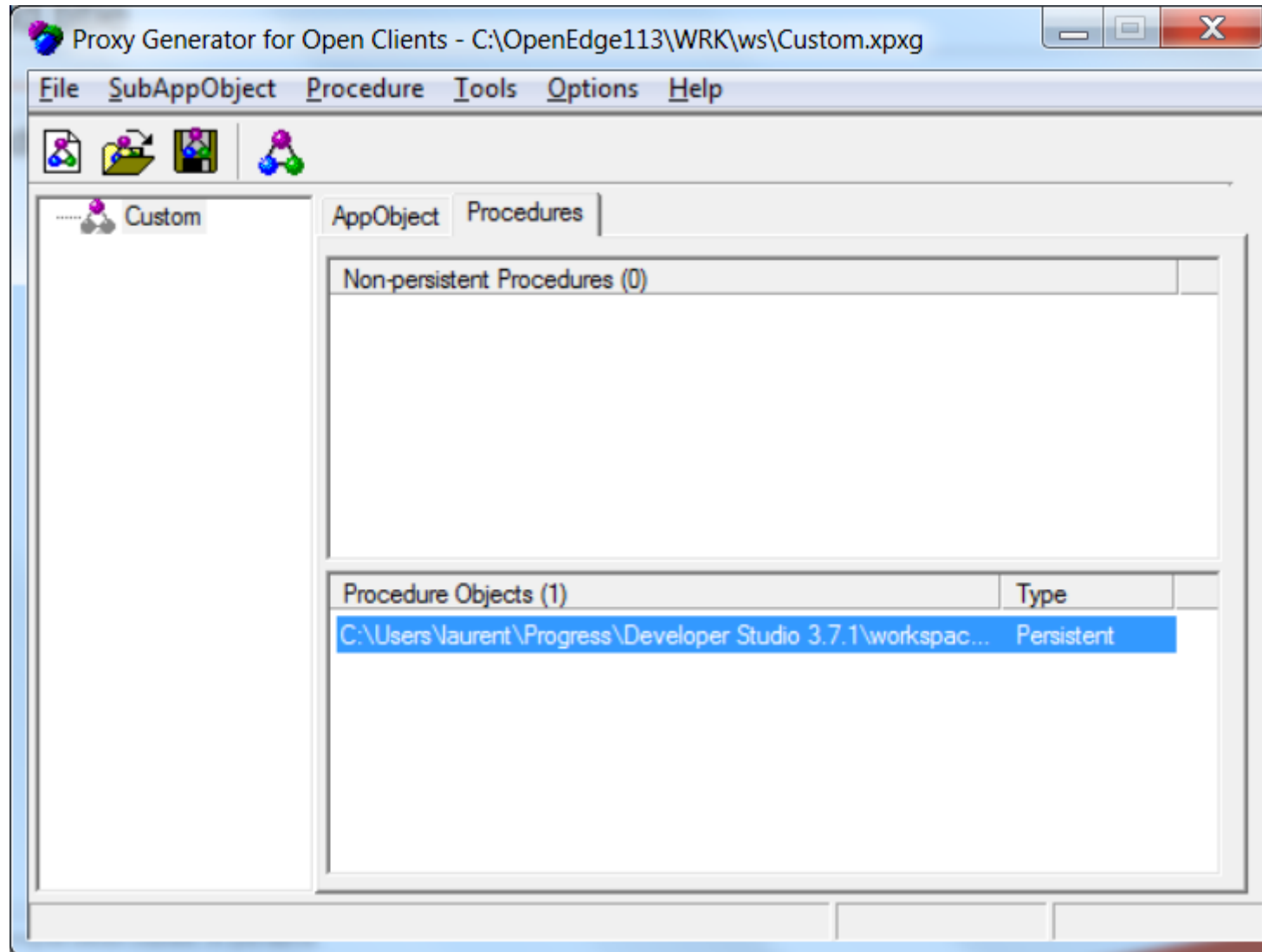
Architecture d'une application ABL exposée sous forme de Web service



Comment exposer une application ABL sous forme de Webservice SOAP



Proxy Generator



Generation WSM

Generate Proxies for Custom

Client Type: **Web Service**

Client Details

Namespace: **um:localhost**

URL for WSA: **http://localhost:8080/wsa/wsa1**

SOAP Action:

- ☒ Blank
- ☐ User-defined

☐ Append Object Name

WSDL Style

- ☒ Doc/Literal
- ☐ RPC/Literal
- ☐ RPC/Encoded

☒ Generate Test WSDL

Suffix

PortType/Binding: **Obj** ☒ Use Default

Web Service: **Service** ☒ Use Default

AppService Name: **restbroker1** ☐ Use Default

Output Directory: **c:\openedge113\wrk\ws** ...

Author: Version:

☐ Return ABL RETURN-VALUE on Connect

Session Model

- ☒ Free
- ☐ Managed

☒ Verbose Logging

☒ Leave Proxy Files

OK **Cancel** **Help**

OE Explorer – OE Management

Resources > OpenEdge > Web Services Adapter > wsa1

Container: nbparlaurent1
Web Services Adapter: wsa1

Collections:

WebServices Adapter Status

Container: nbparlaurent1
Host: NBPARLAURENT1
Adapter: ACTIVE

Command and Control

Control Enable or disable adapter	Monitoring Plans Adapter conditions on which to alert
Configuration Advisor Suggests rule thresholds based on data in the FathomTrendDatabase	Log File Monitor Log file errors on which to alert
Log File Viewer Examine the adapter log file	Configuration Property file configuration associated with this adapter
Login Set WSA Web Server username/password	Log Off Log off of the WSA Web Server
List List the deployed web services	Deploy Deploy a new web service
Import Import a web service	Defaults Set web service defaults

Deployed Web Services

Custom ▼ Select

Status Enablement Deployed web service status and enablement	Update Update a deployed web service
Application Properties Deployed web service application properties	Export Export a deployed web service
Statistics Deployed web service run-time statistics	Undeploy Undeploy a web service

Operational Views

Status Web Service Adapter status information	Statistics Web Services Adapter run-time statistic information
--	---

WebService Client : test

The screenshot displays the SoapUI 5.0.0 application window. The interface includes a menu bar (File, Tools, Desktop, Help), a toolbar, and a search bar. On the left, a 'Navigator' pane shows a project tree with folders like 'BPM', 'CO2', 'Medicare', 'REST Project 1', and 'SOAP Project'. Under 'SOAP Project', there is a 'CustomObj' folder containing 'CreatePO_custom' and 'customObj'. 'customObj' has several methods: 'CreateCustomer', 'DeleteCustomer', 'GetCustomer', 'Request 1', 'Release_custom', and 'UpdateCustomer'. Below the navigator is the 'Request Properties' table.

Property	Value
Name	Request 1
Description	
Message Size	221
Encoding	UTF-8
Endpoint	http://localhost:80...

The main workspace is divided into two panels. The left panel shows the raw XML of the request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:CreatePO_custom/>
  </soapenv:Body>
</soapenv:Envelope>
```

The right panel shows the raw XML of the response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<TV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:localhost:Custom">
  <host:custom">
    -55e6d214:14a0aa63594:-7fff;&lt;t;Custom|PX-000004|PO>;+REwLU0vC6WeWhgfU/IXaA==</UUID>
  </host:custom>
</TV>
```

At the bottom, a status bar shows 'response time: 3524ms (589 bytes)' and a log area with buttons for 'SoapUI log', 'http log', 'jetty log', 'error log', 'wsrm log', and 'memory log'.

Appel méthode GetCustomer

The screenshot displays the SoapUI 5.0.0 interface. On the left, the 'Navigator' pane shows a project structure with a 'SOAP Project' containing a 'CustomObj' and a 'Request 1' for the 'GetCustomer' method. The 'Request Properties' pane at the bottom left shows details for 'Request 1', including its name, description, message size (472), encoding (UTF-8), and endpoint (http://localhost:8080/...). The main workspace is split into two panels. The left panel shows the raw XML of the SOAP request, which includes a header with a custom ID and a body with a 'GetCustomer' call. The right panel shows the raw XML of the SOAP response, which includes a 'GetCustomerResponse' with a 'result' set to 'true' and a 'ttCustomer' object containing various customer details like name, address, phone, and credit limit. The status bar at the bottom indicates a response time of 11ms and 958 bytes.

Request 1

URL: http://localhost:8080/wsa/wsa1

Request XML

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <urn:customID>
      <urn:UUID>838218b47e9b7859:-55e6d214:14a0aa63594:-7fff;&lt;Custor
    </urn:customID>
  </soapenv:Header>
  <soapenv:Body>
    <urn:GetCustomer>
      <urn:custId>10</urn:custId>
    </urn:GetCustomer>
  </soapenv:Body>
</soapenv:Envelope>
```

Response XML

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:="urn:localhost:custom">
  <SOAP-ENV:Body>
    <GetCustomerResponse xmlns="urn:localhost:custom">
      <result xsi:nil="true"/>
      <ttCustomer>
        <ttCustomerRow>
          <CustNum>10</CustNum>
          <Name>Just Joggers Limited</Name>
          <Address>Fairwind Trading Est</Address>
          <Address2>Shoe Lane</Address2>
          <City>Ramsbottom</City>
          <State>Sussex</State>
          <Country>United Kingdom</Country>
          <Phone>070 682 2887</Phone>
          <Contact>George Lacey</Contact>
          <SalesRep>SLS</SalesRep>
          <Comments/>
          <CreditLimit>22000</CreditLimit>
          <Balance>1222.11</Balance>
          <Terms>Net30</Terms>
          <Discount>20</Discount>
          <PostalCode>BL0 9ND</PostalCode>
          <Fax/>
          <EmailAddress/>
        </ttCustomerRow>
      </ttCustomer>
    </GetCustomerResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	472
Encoding	UTF-8
Endpoint	http://localhost:8080/...

Auth Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0)
response time: 11ms (958 bytes)

SoapUI log http log jetty log error log wsrm log memory log

Headers (5) Attachments (0) SSL Info WSS (0) JMS (0)

1:1

Demo



Créer des APIs RESTful




Le bon design d'une API RESTful est essentiel

- Choisir entre un service REST “monolithique” ou des services REST multiples
 - Diviser les API [URL] en application web hébergeant les services REST associés
 - (exemple: services d'administration versus services de données)
 - Chaque URL d'un service REST définit la hiérarchie des ressources associées
 -



RESTful Web Application Design

- Chaque application web(OpenEdge REST) a
 - Un nom d'application
 - Un ou plusieurs services REST
 - Une configuration de sécurité(authentification utilisateur et droit d'accès)
 - Une connexion à un AppServer de type State-free [service]
- Exemple (pour une application OpenEdge REST):

http://**host:port/webAppName**/rest/

Deployment site defined part OE
defined part

RESTful Service et URL

■ Qu'est ce qu'un Service REST

- A un nom de service qui apparait dans l'URL
- Un nom de service contient une ou plusieurs ressources
- Chaque ressources a une URL unique pour le service
- Chaque URL de ressource peut avoir
 - Des paramètres d'entrée optionnels et/ou des options



Vous êtes ici...

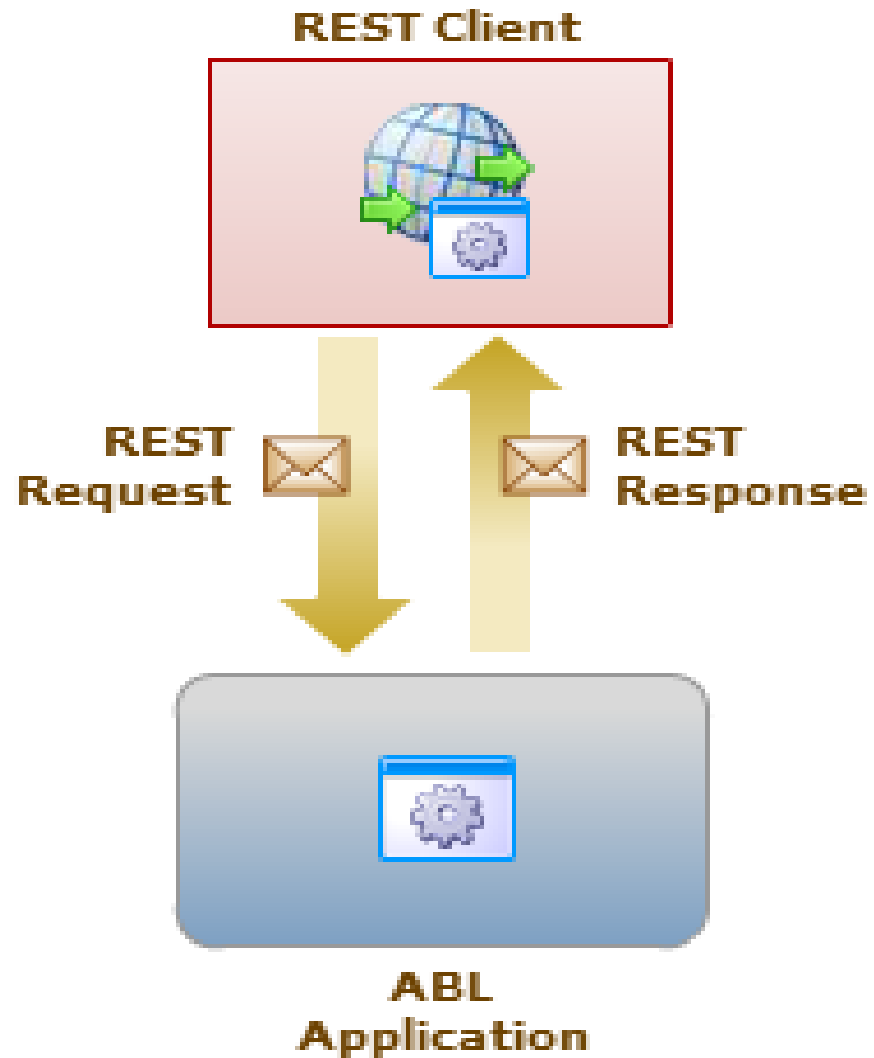
■ Exemple

http://host:port/webAppName/rest/<service-name>/<resource-path>

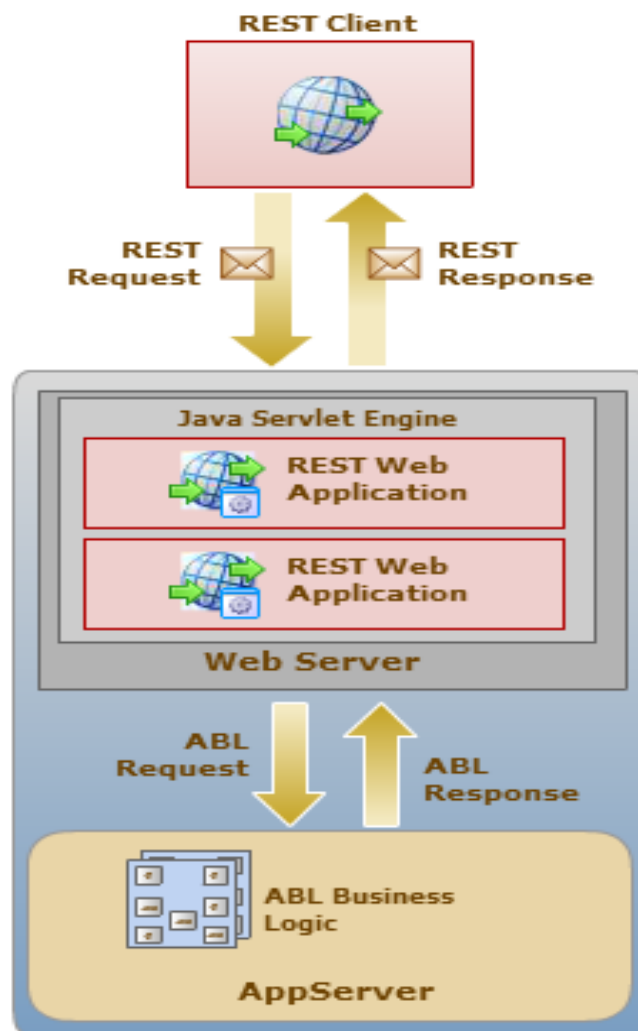


Developer designed part

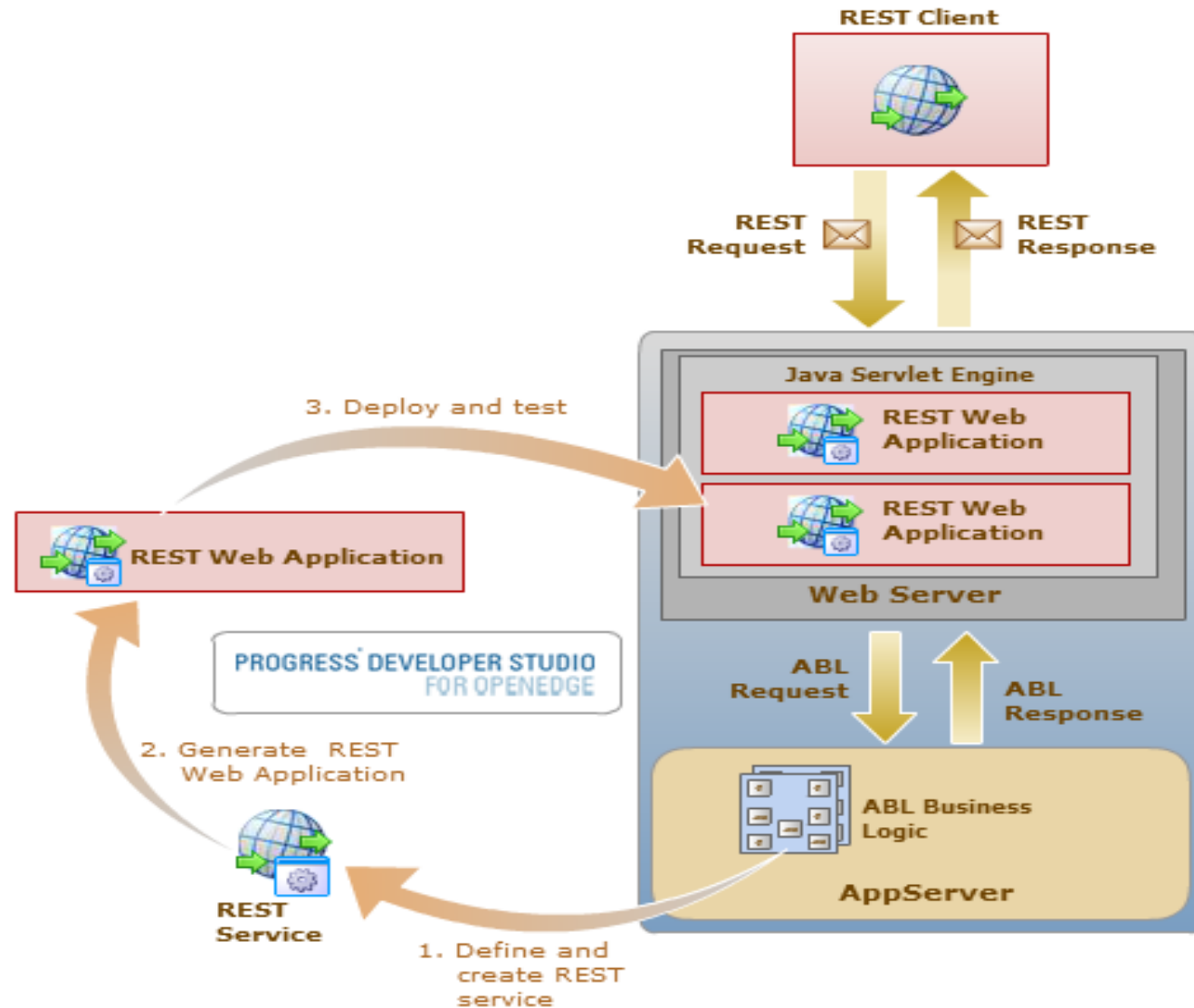
Exposer une application ABL sous forme de Webservice REST



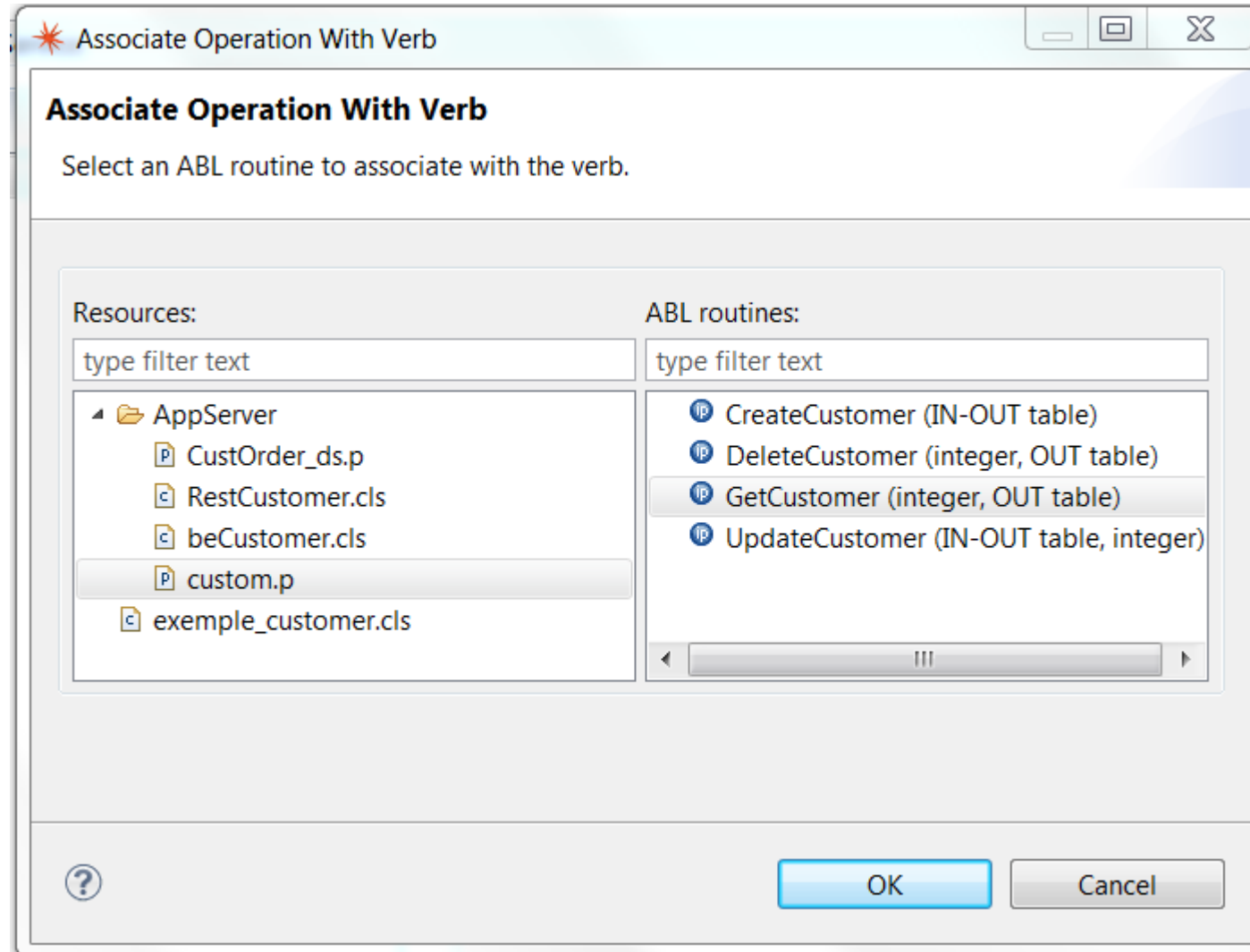
Architecture d'une application ABL exposée sous forme de Webservice REST



Comment exposer une application ABL sous forme de Webservice REST



Utilisation du code ABL pour le Webservice Rest



Mapping Paramètres en Input

REST Resource URI Editor

Service relative URI: Edit...

Resources + -

Verb Association

Verb='GET'	custom..GetCustomer	... ✖
Verb='PUT'		... ✖
Verb='POST'		... ✖
Verb='DELE...		... ✖

Mapping Definitions

Input

Output

Request

- URL Parameters
 - Complete URL
 - Query String Parameters
 - custId
- HTTP Message
 - Method
 - Headers
 - [Drop a parameter here...]
 - Cookies
- Server Contexts
 - Servlet Request
 - Servlet Response
 - Servlet Context
 - Servlet Config

Parameters

Interface Parameters

`{java:Integer}` ✖

custId

Mapping paramètre en Output

REST Resource URI Editor

Service relative URI: Edit...

Resources + ×

/customer

Verb Association

Verb='GET'	custom..GetCustomer	... ×
Verb='PUT'		... ×
Verb='POST'		... ×
Verb='DELE...		... ×

Mapping Definitions

Input

Parameters

Interface Parameters

- ttCustomer {java:Object}

Output

Response

HTTP Message

- Response Code
- Headers
- [Drop a parameter here...]
- Cookies
- Body
- [Drop a parameter here...]

Exemple :

<http://localhost:8980/ExchangeService/rest/ExchangeService/customer?custId=1>

The screenshot displays a REST client interface with the following details:

- Request 1** tab is active.
- Method:** GET
- Endpoint:** http://localhost:8980
- Resource:** /ExchangeService/rest/ExchangeService/customer
- Parameters:** ?custId=1

The **Raw** tab is selected, showing the request details:

Name	Value	Style	Level
custId	1	QUERY	RESOURCE

Below the table, there are fields for **Required:** (checkbox), **Type:** (dropdown), and **Options:** (list with Add, Edit, Remove buttons).

The **Raw** tab also shows the XML response:

```
<?xml version='1.0' encoding='utf-8'>
<Response xmlns="http://localhost/ExchangeService/rest/ExchangeService/customer">
  <ttCustomer>
    <e>
      <Address>276 North Drive</Address>
      <Address2/>
      <Balance>65000.0</Balance>
      <City>Burlington</City>
      <Comments>This customer is on credit hold. Lift</Comments>
      <Contact>Gloria Shepley</Contact>
      <Country>USA</Country>
      <CreditLimit>66700.0</CreditLimit>
      <CustNum>1</CustNum>
      <Discount>35</Discount>
      <EmailAddress/>
      <Fax/>
      <Name>Lift Tours</Name>
      <Phone>(617) 450-0086</Phone>
      <PostalCode>01730</PostalCode>
      <SalesRep>HXM</SalesRep>
      <State>MA</State>
      <Terms>Net30</Terms>
    </e>
  </ttCustomer>
</Response>
```

The bottom status bar shows: Auth Headers (0) Attachments (0) Representations (0) JMS Headers JMS Property (0) Headers (6) Attachments (0) SSL Info Representations (1) Schema (conflicts) JMS (0)

Demo



Autre exemple

<http://localhost:8980/CustomersServices/rest/CustomersServices/beCustomer>

The screenshot displays a REST client interface with the following components:

- Request 1** tab at the top.
- Method:** GET
- Endpoint:** http://localhost:8980
- Resource:** /CustomersServices/rest/CustomersServices/beCustomer
- Parameters:** (empty)
- Raw** view selected on the left sidebar.
- Response** tab on the right sidebar showing XML data.

The XML response is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns="http://localhost/CustomersServices/rest/CustomersServices/beCustomer">
  <dsCustomer>
    <eCustomer>
      <e>
        <Address>276 North Drive</Address>
        <Balance>65000.0</Balance>
        <City>Burlington</City>
        <Comments>This customer is on credit hold. Lift</Comments>
        <Contact>Gloria Shepley</Contact>
        <Country>USA</Country>
        <CreditLimit>66700.0</CreditLimit>
        <CustNum>1</CustNum>
        <Discount>35</Discount>
        <EmailAddress/>
        <Fax/>
        <Name>Lift Tours</Name>
        <Phone>(617) 450-0086</Phone>
        <PostalCode>01730</PostalCode>
        <SalesRep>HXM</SalesRep>
        <State>MA</State>
        <Terms>Net30</Terms>
      </e>
    </eCustomer>
    <e>
      <Address>Rattipolku 3</Address>
      <Balance>28000.0</Balance>
      <City>Oslo</City>
      <Comments>Ship all products 2nd Day Air. lift</Comments>
      <Contact>Urpo Leppakoski</Contact>
      <Country>Finland</Country>
      <CreditLimit>27600.0</CreditLimit>
      <CustNum>2</CustNum>
      <Discount>35</Discount>
      <EmailAddress/>
      <Fax/>
    </e>
  </dsCustomer>
</Response>
```

Avec : ?filter=name begins 'l'

The screenshot shows a REST client interface with the following details:

- Request 1** tab is active.
- Method:** GET
- Endpoint:** http://localhost:8980
- Resource:** /CustomerServices/rest/CustomerServices/beCustomer
- Parameters:** ?filter=name begins 'l'
- Raw** tab is selected, showing the XML response.

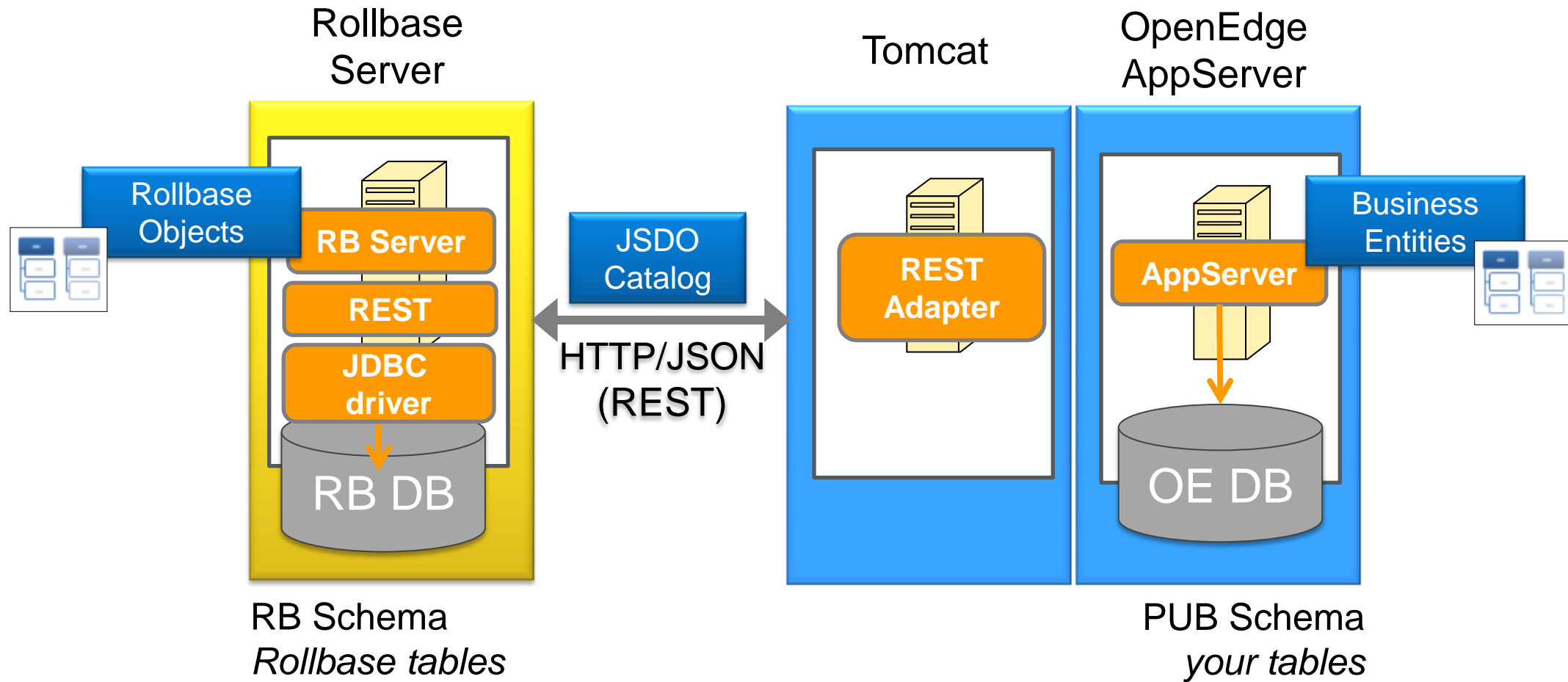
Name	Value	Style	Level
filter	name begins 'l'	QUERY	RESOURCE

```
<Response xmlns="http://localhost/CustomerServices/rest/CustomerServices/beCustomer">
  <dsCustomer>
    <eCustomer>
      <e>
        <Address>276 North Drive</Address>
        <Balance>65000.0</Balance>
        <City>Burlington</City>
        <Comments>This customer is on credit hold. Lift</Comments>
        <Contact>Gloria Shepley</Contact>
        <Country>USA</Country>
        <CreditLimit>66700.0</CreditLimit>
        <CustNum>1</CustNum>
        <Discount>35</Discount>
        <EmailAddress/>
        <Fax/>
        <Name>Lift Tours</Name>
        <Phone>(617) 450-0086</Phone>
        <PostalCode>01730</PostalCode>
        <SalesRep>HXM</SalesRep>
        <State>MA</State>
        <Terms>Net30</Terms>
      </e>
    </e>
    <e>
      <Address>rue de la gare</Address>
      <Balance>624.34</Balance>
      <City>Lyon</City>
      <Comments>Ship all products overnight.</Comments>
      <Contact>Mr Roulemousse</Contact>
      <Country>France</Country>
      <CreditLimit>13000.0</CreditLimit>
      <CustNum>25</CustNum>
      <Discount>30</Discount>
      <EmailAddress/>
      <Fax/>
    </e>
  </dsCustomer>
</Response>
```

Demo



Exemple Rollbase : OpenEdge Service Objects



Ecran Rollbase avec OpenEdge Service

New Application

Install Applications

RECENT ITEMS

La voile en folie (eCustomer)

Credit by Customer (Chart)

eCustomer (Object)

Customer (Chart)

Customer per Country (Chart)

LK_Appserver (Application)

Locker Room (eCustomer)

La Boule Lyonnaise (eCustomer)

Statistic (Object)

Report (Object)

Lift Tours (eCustomer)

CRM (Application)

Recycle Bin 0

eCustomers

Statistics

New

Design This Page

All eCustomers

New eCustomer

Customer by Salesrep

Filter

Select

More actions...

eCustomers 1-20 of 1,117

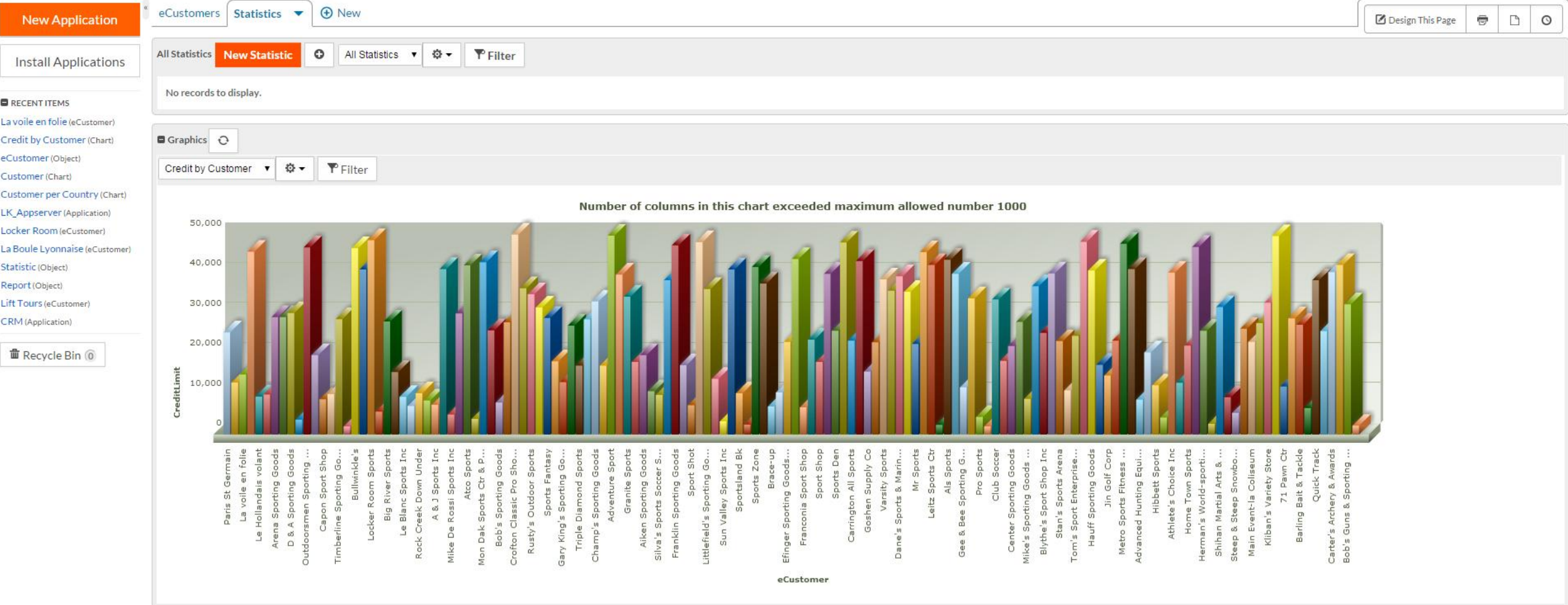
Page 1 of 56

Action	Name	SalesRep	Address	Balance	City	Comments	Contact	Country	CreditLimit	CustNum	Discount	EmailAddress
<input type="checkbox"/> Edit Del	Lift Tours	HXM	276 North Drive	903.64	Burlington	This customer is on credit hold.	Gloria Shepley	USA	66,700.00	1	35	
<input type="checkbox"/> Edit Del	Urpon Frisbee	DKP	Rattipolku 3	437.63	Oslo	Ship all products 2nd Day Air.	Urpo Leppakoski	Finland	27,600.00	2	35	
<input type="checkbox"/> Edit Del	Hoops	HXM	Suite 415	1,199.95	Atlanta	This customer is now OFF credit hold.	Michael Traitser	USA	75,000.00	3	10	
<input type="checkbox"/> Edit Del	Go Fishing Ltd	SLS	Unit 2	14,235.14	Harrow		Alan Frogbrook	United Kingdom	15,000.00	4	10	
<input type="checkbox"/> Edit Del	Match Point Tennis	JAL	66 Homer Pl	0.00	Boston		Robert Dorr	USA	11,000.00	5	50	
<input type="checkbox"/> Edit Del	Fanatical Athletes	SLS	20 Bicep Bridge Rd	1,202.66	Montgomery	Ship C.O.D. at all times.	Andrew Strong	United Kingdom	38,900.00	6	0	
<input type="checkbox"/> Edit Del	Aerobics valine Ky	DKP	Peltolantie 3	1,112.44	Tikkurila		Elli Ilmanen	Finland	13,500.00	7	15	
<input type="checkbox"/> Edit Del	Game Set Match	RDR	Box 60	8,254.00	Deatsville	This customer is back on Credit Hold.	Tore Break	USA	15,000.00	8	50	
<input type="checkbox"/> Edit Del	Pihtiputaan Pyora	DKP	Putikontie 2	1,242.14	Pihtipudas	Call for shipping instructions.	Markku Ventolinen	Finland	29,900.00	9	10	
<input type="checkbox"/> Edit Del	Just Joggers Limited	SLS	Fairwind Trading Est	1,222.11	Ramsbottom		George Lacey	United Kingdom	22,000.00	10	20	
<input type="checkbox"/> Edit Del	Keilailu ja Biljardi	DKP	Vattuniemenkuja 8 A	1,186.80	Helsinki		Riitta Hammermaki	Finland	10,900.00	11	25	
<input type="checkbox"/> Edit Del	Surf Lautaveikkoset	DKP	Venemestarinkatu 35	0.00	Salo		Veikko Lehtisoini	Finland	6,500.00	12	35	
<input type="checkbox"/> Edit Del	Biljardi ja tennis	DKP	Urheilutie 1	1,227.58	Mantsala	Do not ship through US Mail.	Tuomo A.t. Mattinen	Finland	18,200.00	13	50	
<input checked="" type="checkbox"/> Edit Del	Paris St Germain	DOS	113, avenue du Stade	1,215.25	Paris		Philippe Dasgil	France	25,500.00	14	20	
<input type="checkbox"/> Edit Del	Hoopla Basketball	HXM	87 Calumnet St	2,619.00	Egg Harbor	This customer is OFF credit hold!	Robert Johns	USA	8,500.00	15	45	
<input type="checkbox"/> Edit Del	Thundering Surf Inc.	JAL	354 Market St.	1,237.73	Coffee City		Fran Horan	USA	16,300.00	16	40	
<input type="checkbox"/> Edit Del	High Tide Sailing	SLS	178 Schooner Hill	2,648.02	Immingham	Speak with Captain Bob before shipping any products.	Angela Tackle	United Kingdom	10,500.00	17	25	
<input type="checkbox"/> Edit Del	Antin Metsastysase	DKP	Vanhainkodinkuja 1	1,188.39	Sysma		Antti Oyrinmetsa	Finland	21,300.00	18	35	
<input type="checkbox"/> Edit Del	Buffalo Shuffleboard	HXM	155 Carolina Ave	1,228.02	Buffalo	Ship to Green Bay warehouse.	George Buitkus	USA	35,800.00	19	20	
<input type="checkbox"/> Edit Del	Espoon Pallokeskus	DKP	Sinikalliontie 18	0.00	Espoo		Pekka Maukajarvi	Finland	12,500.00	20	25	

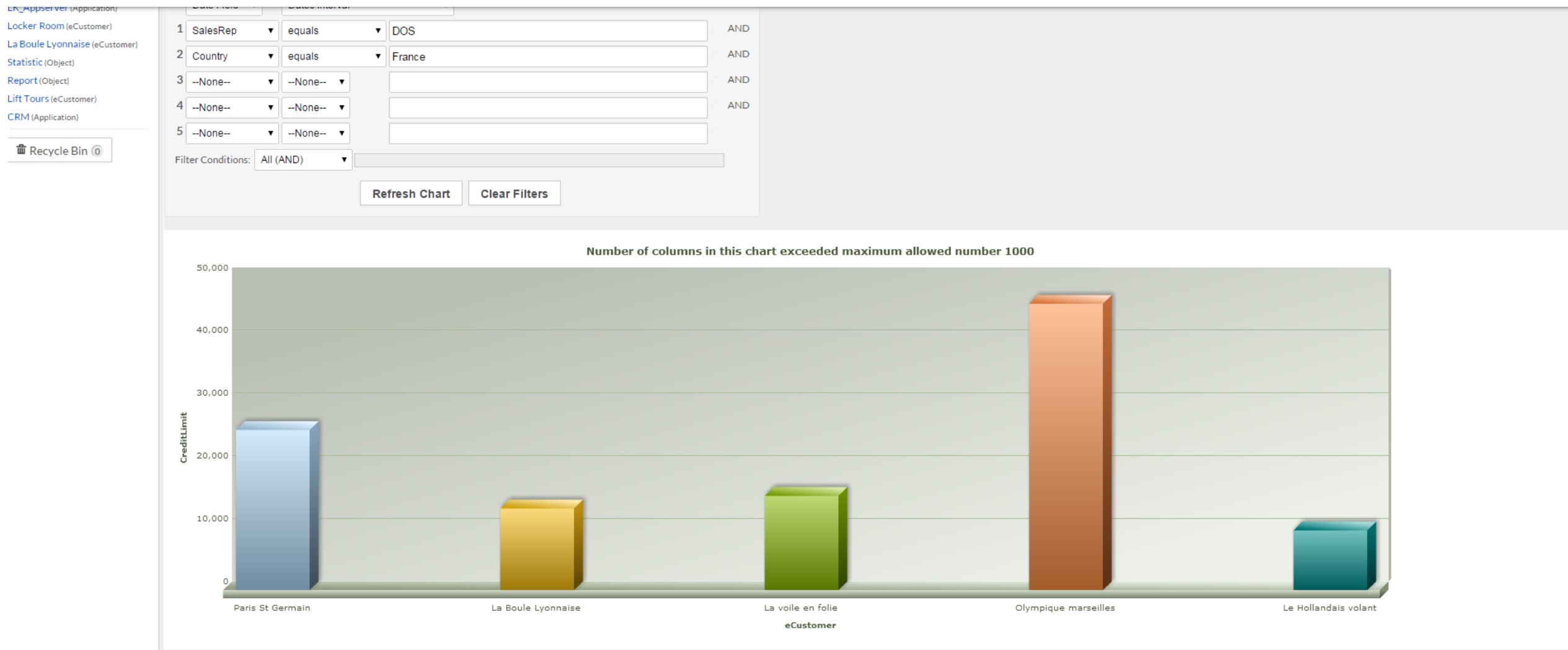
eCustomers 1-20 of 1,117

Page 1 of 56

Graphique Rollbase sur les données OpenEdge



Filtre sur les données OpenEdge via l'OpenEdge Service



Mettre en Production



Déployer votre service REST

1	Exporter service REST Service en .war (New Deployment) / .zip (Incremental Deployment).
2	Utiliser un Tomcat configuré en production. Deployer .war en utilisant un nom différent
3	Editer default logging, AppServer connection properties, security configuration
4	Prévoir d'activer AppServer SSO & CORS (pour les clients javascripts)
5	Enable/Disable Services REST

Remote Management Tools

- Nécessite REST Management Agent (oerm.war) installé sur Tomcat
- Aide au Déploiement & Administration des Applications REST à distance
- Peut être utilisé en production



Graphical Management Center

```
proenv>restman -help
Command Tool Usage for AppServer REST Adapter
=====
-help or -h                Display command line help.
-name or -i (Name) (Required) Name of AppServer REST Adapter
-user or -u (UserName)      User name
-host or -r                 Host name where AdminServer is running
-port                       Port number of running AdminServer
-webserverauth              User name for Web Server Authentication
-war                        location of WAR file
-appname                    friendly name of a Web Service
-prop                       name of property to be set
-value                      new value of property
-query or -q                query named WSA or Web Service
-deploy                     deploy a Web Service
-undeploy                   undeploy a named Web Service
-list                       lists all deployed Web Services
-getdefaults                displays the default properties
-setdefaults                sets a default properties
-resetdefaults              resets the default properties
-enable                     enables a Web Service
-disable                    disables a Web Service
-getstats                   displays statistics
-resetstats                 resets statistics
-getprops                   displays the run-time properties
-setprops                   sets a run-time properties
-resetprops                 reset the run-time properties
-unpublish                  unpublish a named REST Service
-republish                  republish change-set to a deployed app
```

Command Line Utility (restman)

Quand les Choses vont de travers



Guide Debug

■ Tools

- Un bon outil de “proxy debug”
- Activer HTTP message tracking dans les applications web

■ Logs

- Web server logs
- Web application logs
- Web application logging configuration

■ Flow

- Ping de l'application Web
- Obtenir les descriptions des API RESTful
- Ping de l'AppServer
- Accéder aux API RESTful de l'Appserver

Debugging Services REST

- On peut identifier les informations nécessaires pour invoquer un service REST OpenEdge en envoyant une requête GET d'un client REST avec le format URI suivant:

```
http[s]://<host_name>:<port>/<rest_application_name>/rest
```

```
{
  "AppServerStatus": {
    "PingStatus": "[true|False]",
    "ABLReturnVal": "[return string from ABL code]"
    "Error": {
      _errorMsg: "[Error Message (if any) while running the
                  ABL code]"
      _errorNum: "[Error Number associated with the error
                  message]"
    }
  },
  "RESTServices": [
    { [REST Service1 JSON] },
    { [REST Service2 JSON] },
    ...
    { [REST ServiceN JSON] }
  ]
}
```

Opération fréquemment utilisées

```
{
  "AppServerStatus":{
    "PingStatus":"true",
    "ABLReturnVal":"A developer can customize this"
  },
  "RESTServices":[
    {
      "http://localhost:8980/ping/rest/pingService":{...},
      "http://localhost:8980/ping/rest/BenefitService":{
        "Benefits":{
          "Benefits..DeleteBenefits":{...},
          "Benefits..UpdateBenefits":{
            "Method":"PUT",
            "Resource-Path":"/Benefits",
            "Http-Request-Type":"application/json",
            "Http-Response-Type":"application/json",
            "ABL-Resource-Name":"Benefits.cls",
            "ABL-Resource-path":"WRKDIR\\",
            "Procedure-Type":"Internal Procedure",
            "Input":[
              {
                "HTTP-Source":"http.body",
                "ABL-Target":"dsBenefits",
                "ABL-Type":"DATASET"
              }
            ],
            "Output":[
              {
                "HTTP-Target":"http.body",
                "ABL-Source":"dsBenefits",
                "ABL-Type":"DATASET"
              }
            ]
          },
          "Benefits..ReadBenefits":{...},
          "Benefits..CreateBenefits":{...}
        }
      }
    }
  ]
}
```

1

Identifying the AppServer and ABL code execution status

2

Constructing the URL to invoke an internal procedure

3

Setting media types for your REST request

4

Sending a value to the ABL input parameter

5

Receiving a value from the ABL output parameter

Pacific AppServer for OpenEdge



Pacific AppServer for OpenEdge

- Basé sur Tomcat 7.0.55
- Configuré en production par défaut
- Adapteurs intégrés
 - APSV (AIA Remplacement)
 - REST
 - SOAP
- Supporte les fichiers .paar et .wsm



PROGRESS