



OpenEdge and Mobile Applications

Gus Björklund. Wizard. Progress.

PUG-Norway, Oslo, Norge, tirsdag 05.mars 2013

FinPUG, S/S Borea, Turku, Finland, 7-8 .3.2013

Reminder: Turn your cell phones ON!


```
var customers = new JSPO (customer);
```

on CompleteFn -

on Successfn

on Σ for F_A

on Successful Cust;

customers.hill();

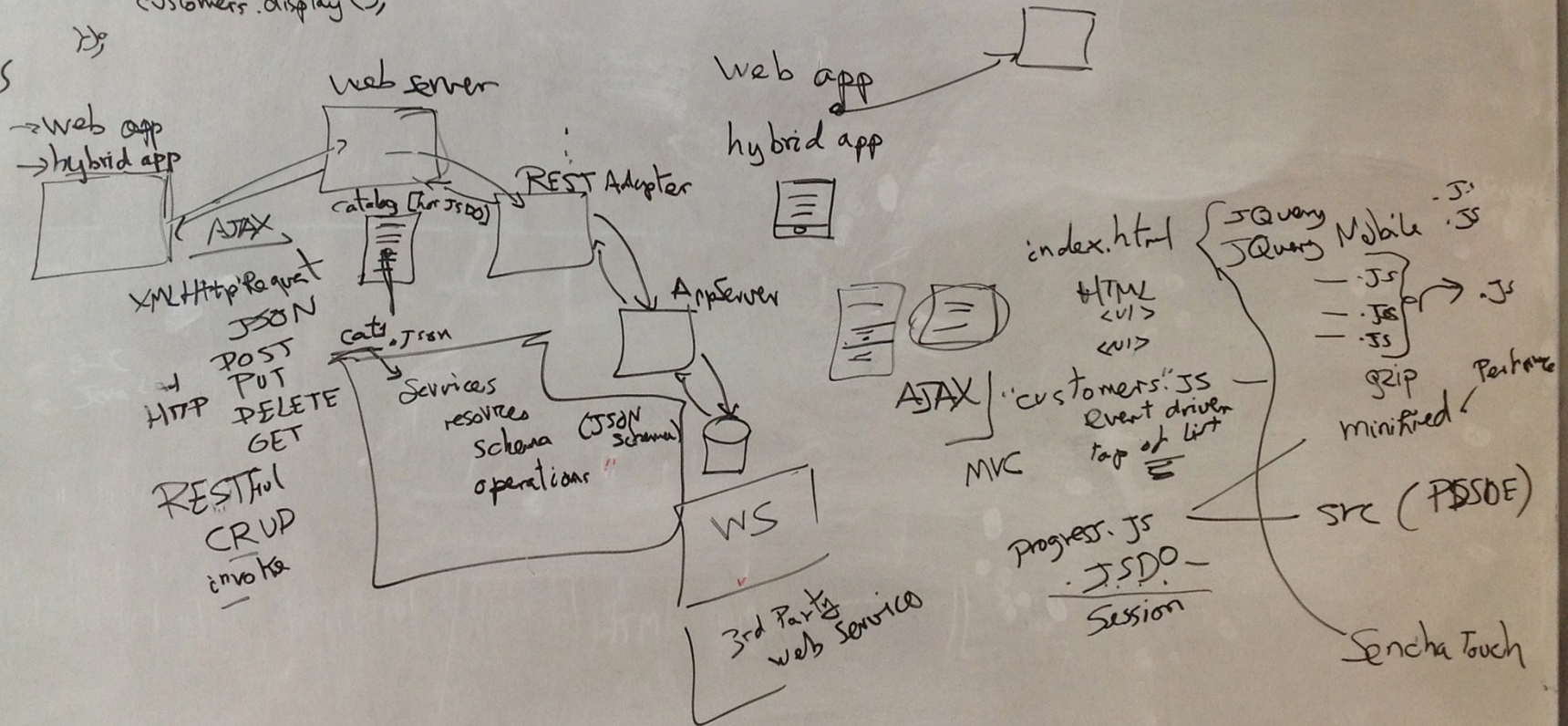
function onSuccess: () {

customers. for each function (customer)

customers.display();

۲۷

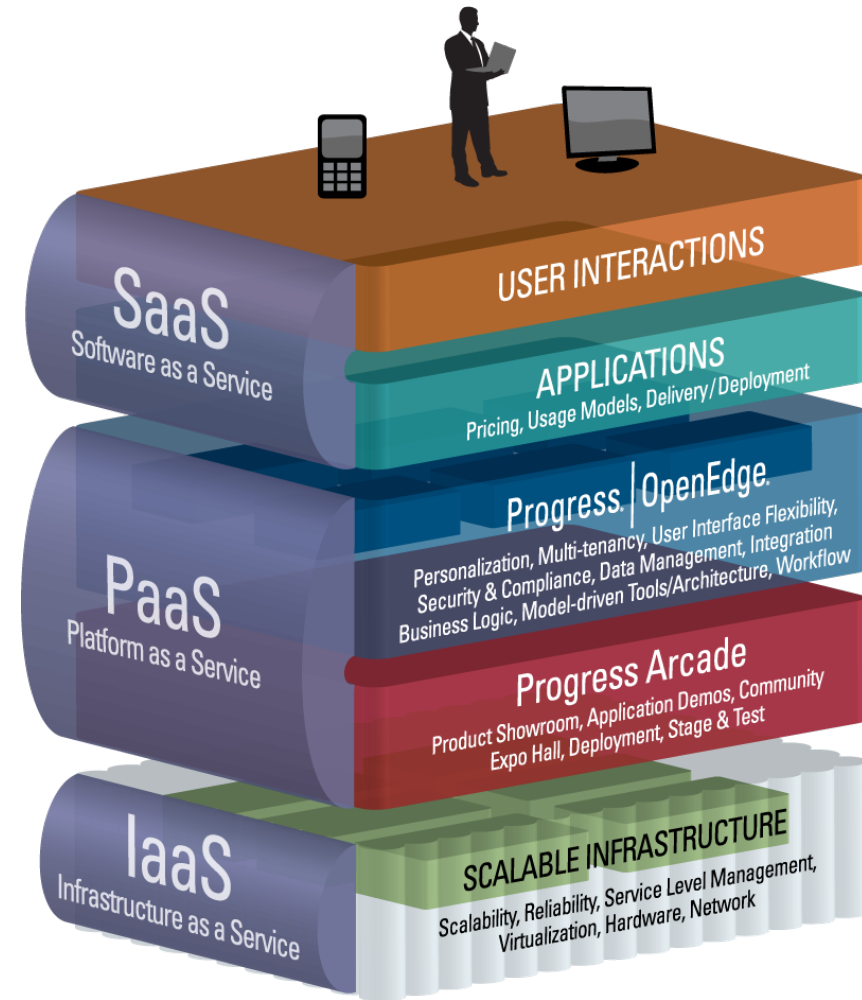
5



Questions

email: gus@progress.com

The complete development platform to build dynamic, business process-enabled applications for secure deployment across any platform, **any mobile device**, and any Cloud.





Mobility

- High productivity tools to create phone and tablet apps that run on any device
- Provide open accessibility and helper libraries to facilitate DIY approaches
- One-click app creator to create form-based apps from schemas



Productivity

- Incorporate a tightly integrate BPM to provide application workflows
- Leverage Corticon business rules to create dynamic, easily-customized logic
- Language and platform components to accelerate development and minimize operational costs



Scalability

- Horizontal table partitioning to allow applications to handle Big Data volumes
- Improved performance of admin utilities to minimize downtime
- Support for on-line operations and changes

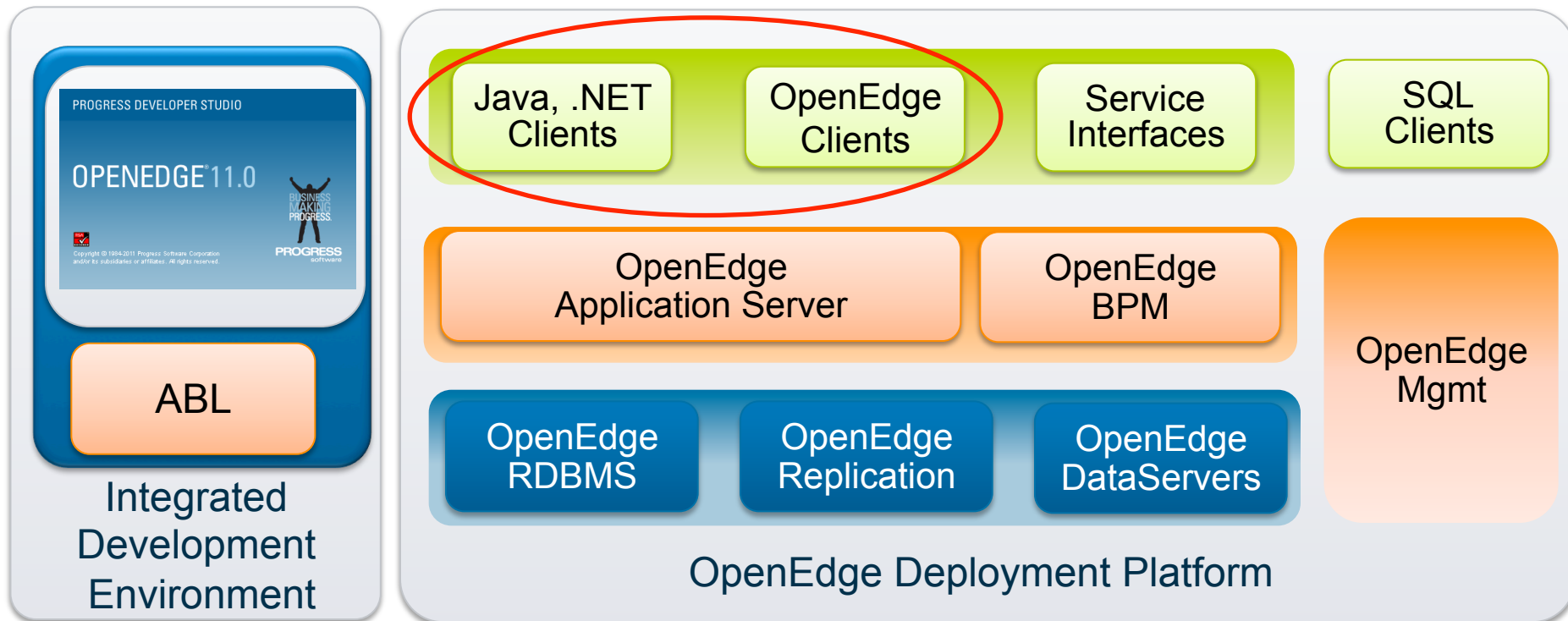


Cloud and SaaS

- New multi-tenancy capabilities that optimize SaaS operations
- Arcade support for other Amazon and non-Amazon data centers
- Arcade usability and security improvements

The OpenEdge Platform

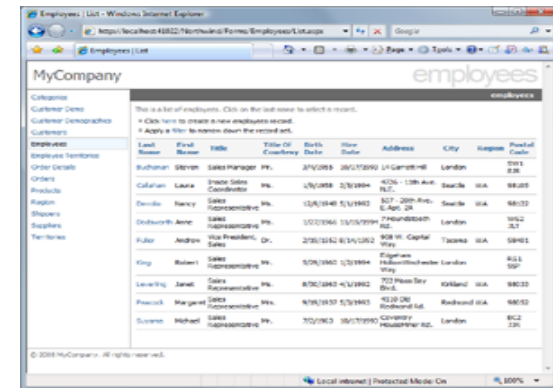
*Flexible and open support for industry leading UI technologies with
native support for the most strategic ones*



UI Types for OpenEdge Applications

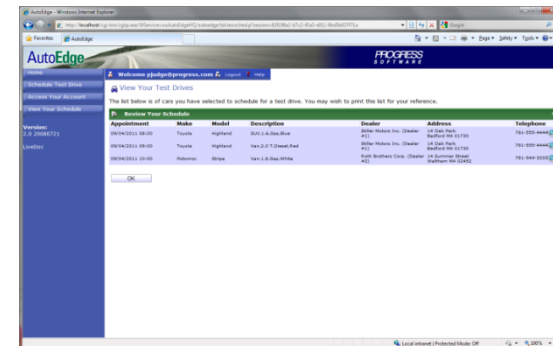
■ Desktop

- Richest user experience
- User productivity is key
- Powerful features working with complex data
- Internal/back office power users



■ Web Browser

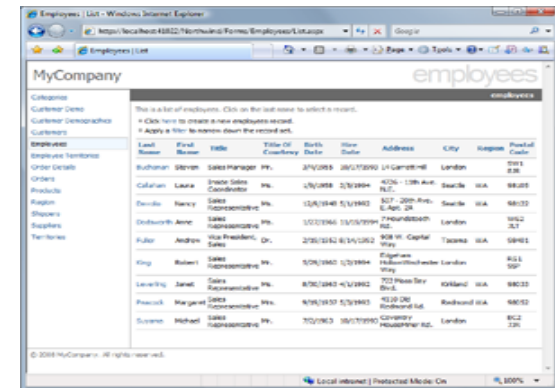
- Location and machine independence
- Target features made available
- Internal and external users



UI Types for OpenEdge Applications

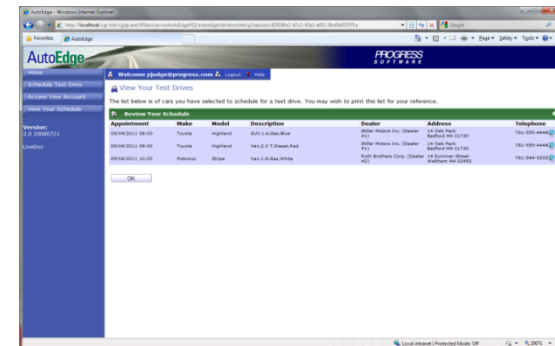
■ Desktop

- Richest user experience
- User productivity is key
- Powerful features working with complex data
- Internal/back office power users



■ Web Browser

- Location and machine independence
- Target features made available
- Internal and external users

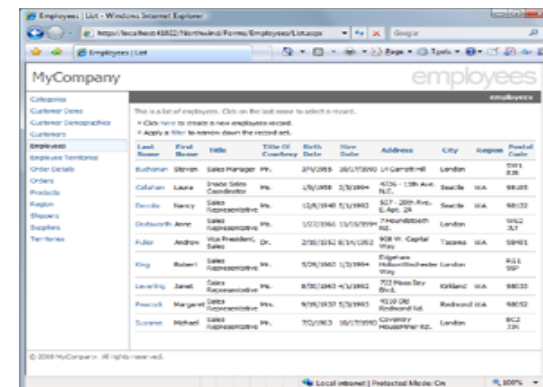


■ vi, tty

UI Types for OpenEdge Applications

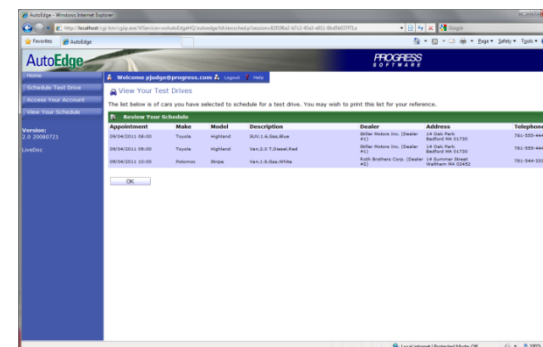
■ Desktop

- Richest user experience
- User productivity is key
- Powerful features working with complex data
- Internal/back office power users



■ Web Browser

- Location and machine independence
- Target features made available
- Internal and external users

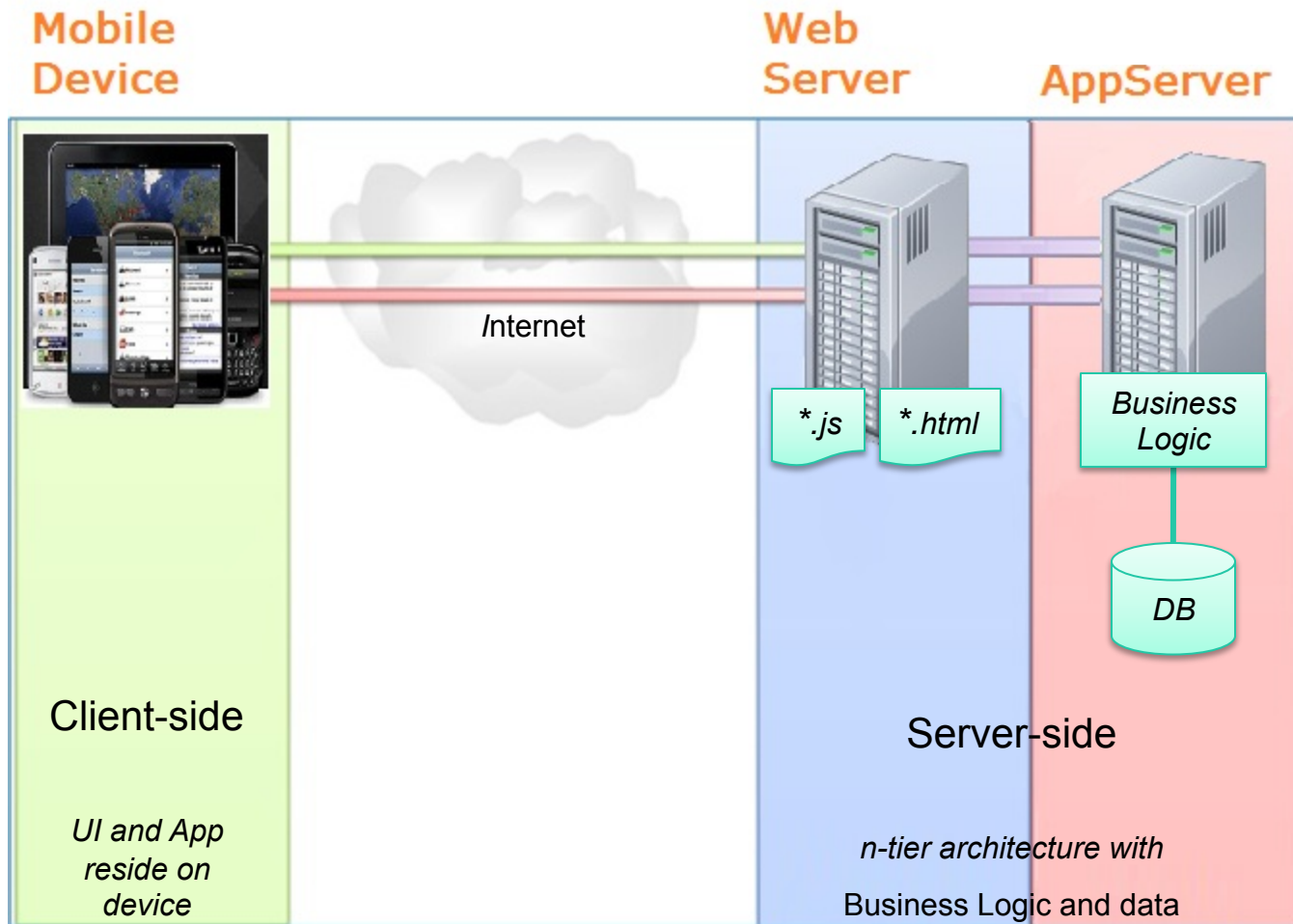


■ Mobile Web Browser / Native Device

- Access from anywhere, any device
- Limited features
- Internal and external users



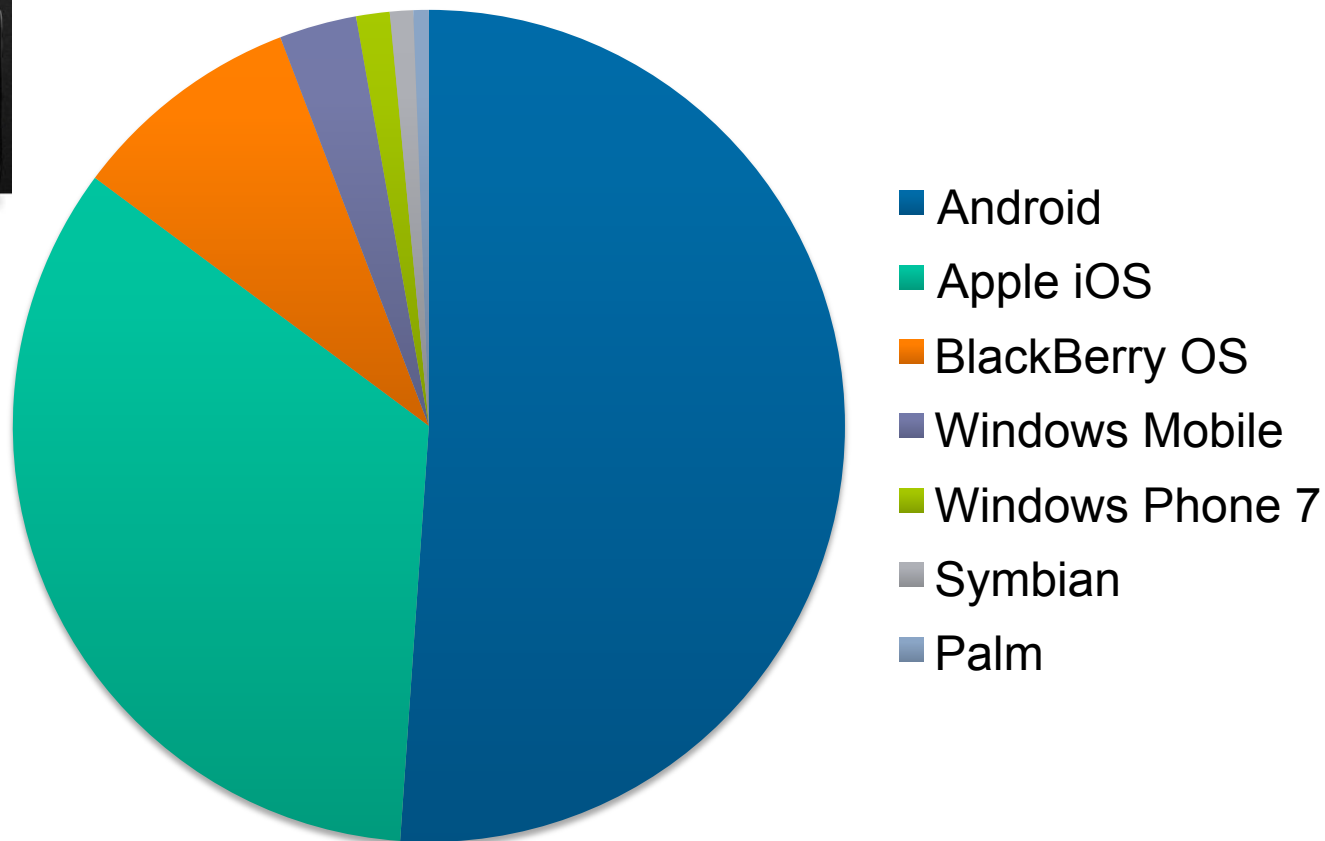
Typical Mobile Application Architecture



A Typical Mobile Device



Smartphone Operating System Market Share (Q2 2012)



Apple iOS + Android = 85 %
source: nielsen

You can have any device ...

*You can have any device ...
as long as it is an Android device or
an iOs device ...*

*You can have any device ...
as long as it is an Android device or
an iOS device ...*

Android 4 and later

iOS 5 and later

*You can have any device ...
as long as it is an Android device or
an iOS device ...*

*Sorry but the Mars Curiosity rover
is not supported.*

*Can't I Just Use My Rich Internet Application (RIA)
and Display It On a Mobile Device...?*

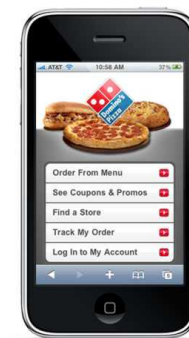
Can't I Just Use My Rich Internet Application (RIA) and Display It On a Mobile Device...?

■ Different user experience and user expectations

- Apps can be installed
- Home screen with no menus/toolbars
- Portrait and landscape display modes
- Additional user input: touch screen, compass, accelerometer, etc.

■ Limited real estate

- Focus is on maximum productivity for least effort for end-user
- Normally a subset of features of a Desktop Web App



Mobile Website

VS



Standard Website

Can't I Just Use My Rich Internet Application (RIA) and Display It On a Mobile Device...?

■ Device-specific interaction

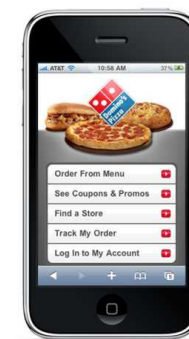
- Contact lists and calendars
- Dedicated hardware controls, like music playback or volume buttons, GPS, etc.

■ Physical resource limitations

- Fluctuations in bandwidth - need to minimize data transfer
- Unreliable connection, battery power

■ Additional security challenges

- Off-line data storage
- User access, authentication, authorization



Mobile Website

VS



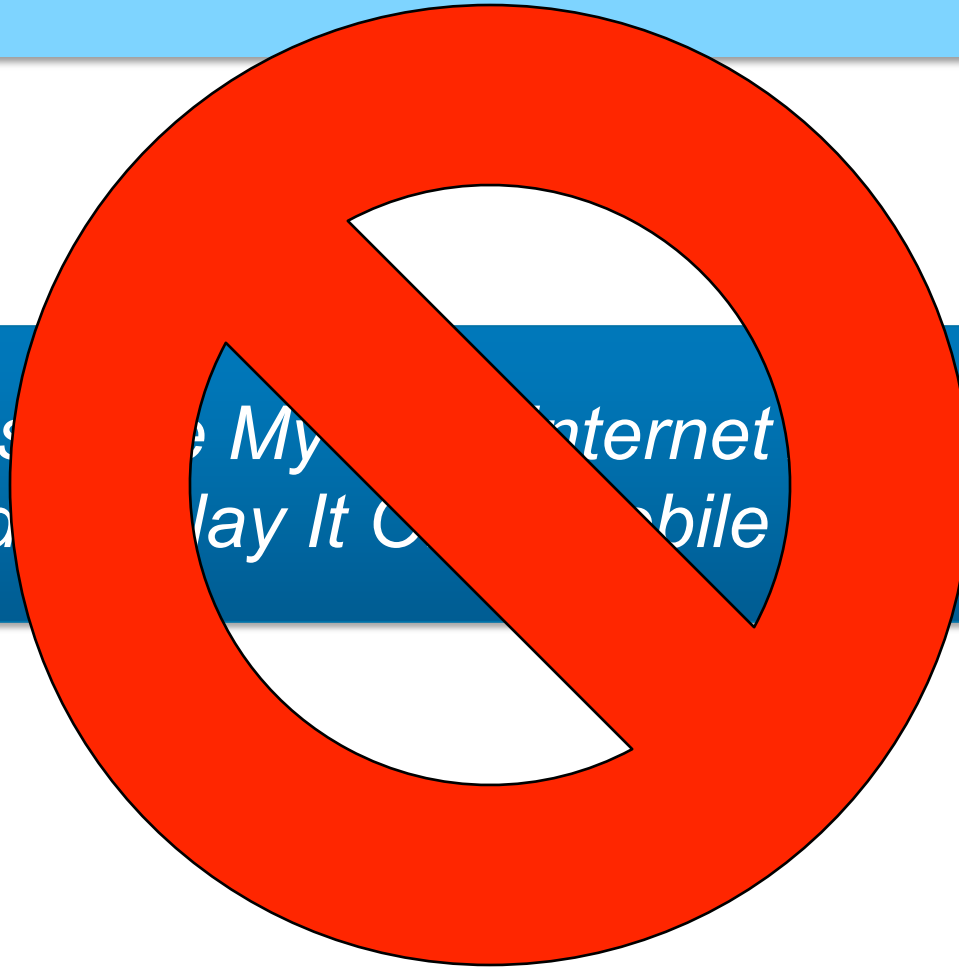
Standard Website

Device specific stuff

- Touch screen
- Accelerometer
- GPS
- Compass
- Camera
- Tilt meter
- Microphone
- Clock
- Contact list
- Calendar
- Spectrometer
- etc., etc. etc.

*Can't I Just Use My Rich Internet Application (RIA)
and Display It On a Mobile Device...?*

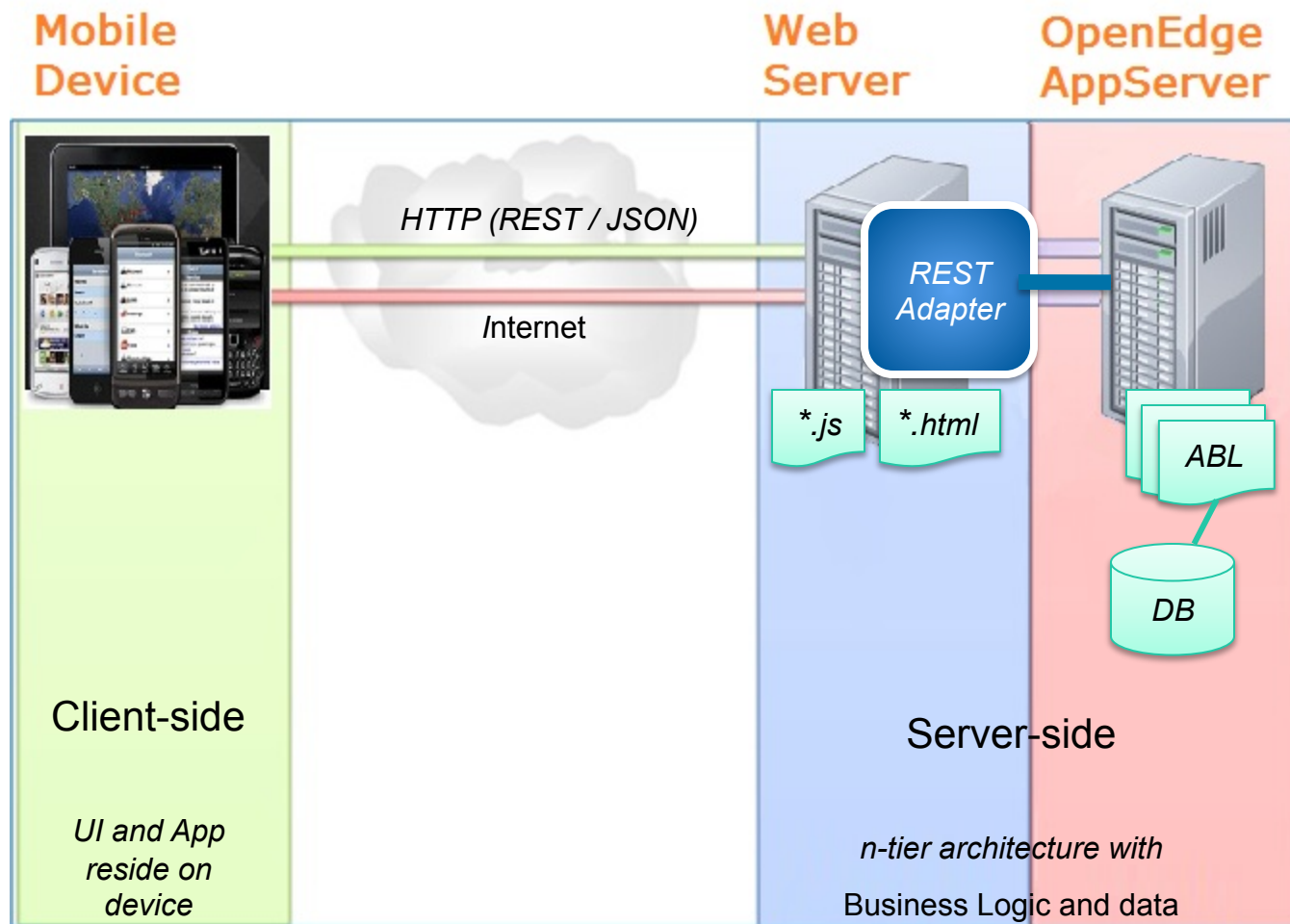
*Can't I Just Use My Internet Application (RIA)
and Play It On Mobile...?*



- OpenEdge REST Adapter / REST support in WebSpeed & AppServer
 - Loosely-coupled architecture
 - Any client
 - Define Service Interface using annotations
 - HTTP / JSON

- OpenEdge Mobile UI
 - Tightly-coupled architecture
 - Specific UI
 - Screen designer and other tools
 - Based on BusinessEntity annotations

OpenEdge REST Adapter



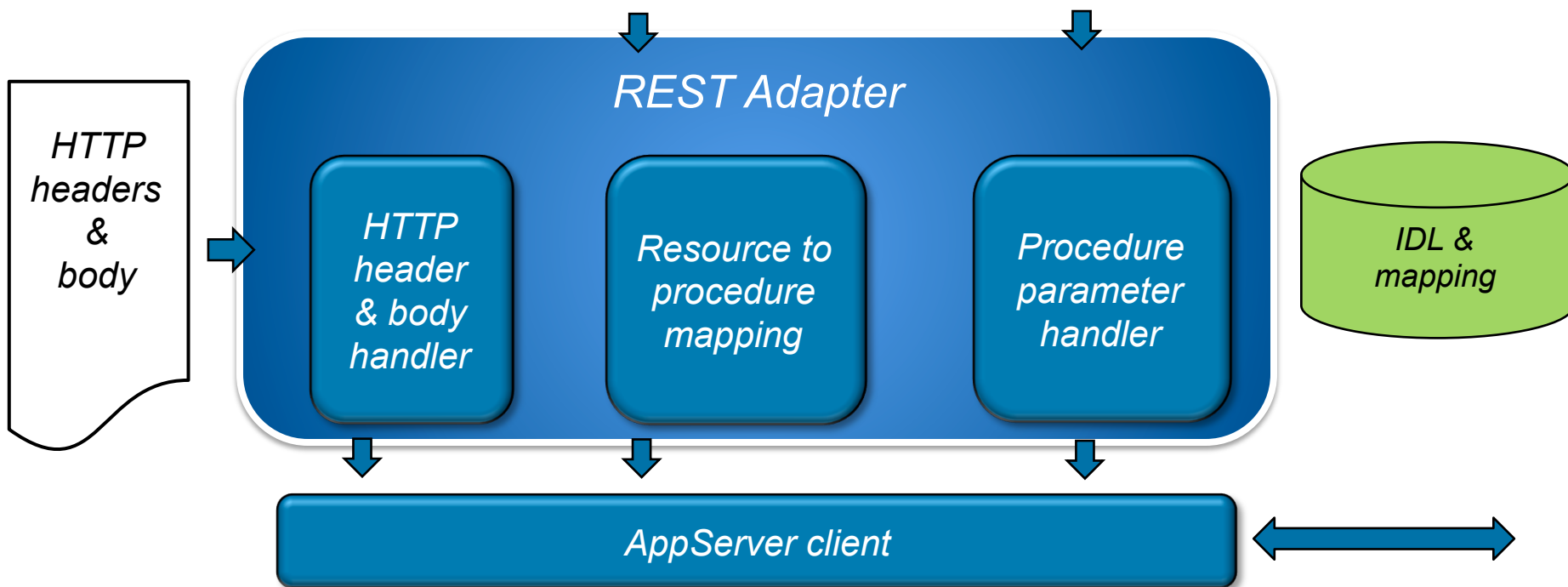
REST Adapter Handling of HTTP Requests

GET `http://acme.com/services/sales/customers?where=name%20begins%20coyote`

deployment time
end point
information

design time
REST resource
information

runtime time
client
information



Mobile UI: 3 Types of Mobile Apps

Three Types of Mobile Applications

Native App (Device specific)

- Build app for each platform one by one
- Full capabilities and integration with device
- AppStore presence, restricted distribution

Hybrid App (Device specific + HTML5/CSS/JavaScript)

- Write app code once
- Native container or code is compiled to native
- Access to *most* device capabilities
- AppStore presence, restricted distribution

Mobile Web App (HTML5/CSS/JavaScript)

- Write app once, runs on *any* device (in theory !)
- Internet connection required
- Limited access to device capabilities
- No native AppStore, unrestricted distribution



which is best ?

Openedge Mobile UI: Hybrid Application

“The Best Of Both Worlds”

PROGRESS
software



*According to Gartner:
By 2015, 80% of all
mobile web
applications
developed will be
hybrid or mobile-Web-
oriented.*

Hybrid App (Native Wrapper + Web App)

- Standards-based solution
- Write the UI code once, deploy to multiple devices
- Native container provides support for App Store and any HTML5 limitations
- Flexible deployment options of both **Web App** and **Hybrid App**
- No need to learn device-specific languages or development environments

Gartner 2012: Magic Quadrant for Mobile Consumer Application Platforms

Hybrid Mobile Applications Create a Beneficial Blend Of Native And Web Applications

- ① Standard web technologies used to build the Mobile App



- ② Hosted inside a native app container

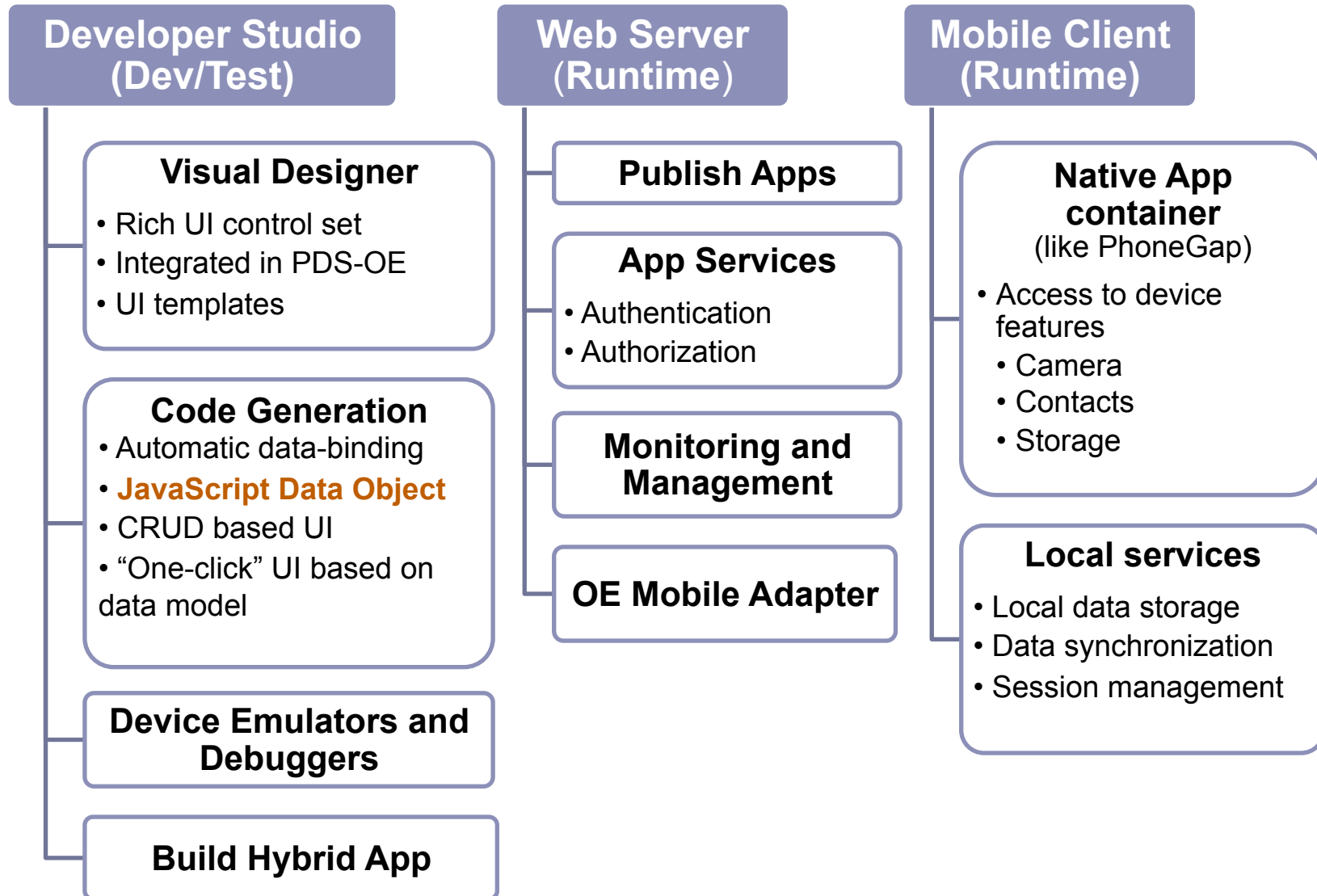


- ④ Taking advantage of features normally inaccessible by web apps

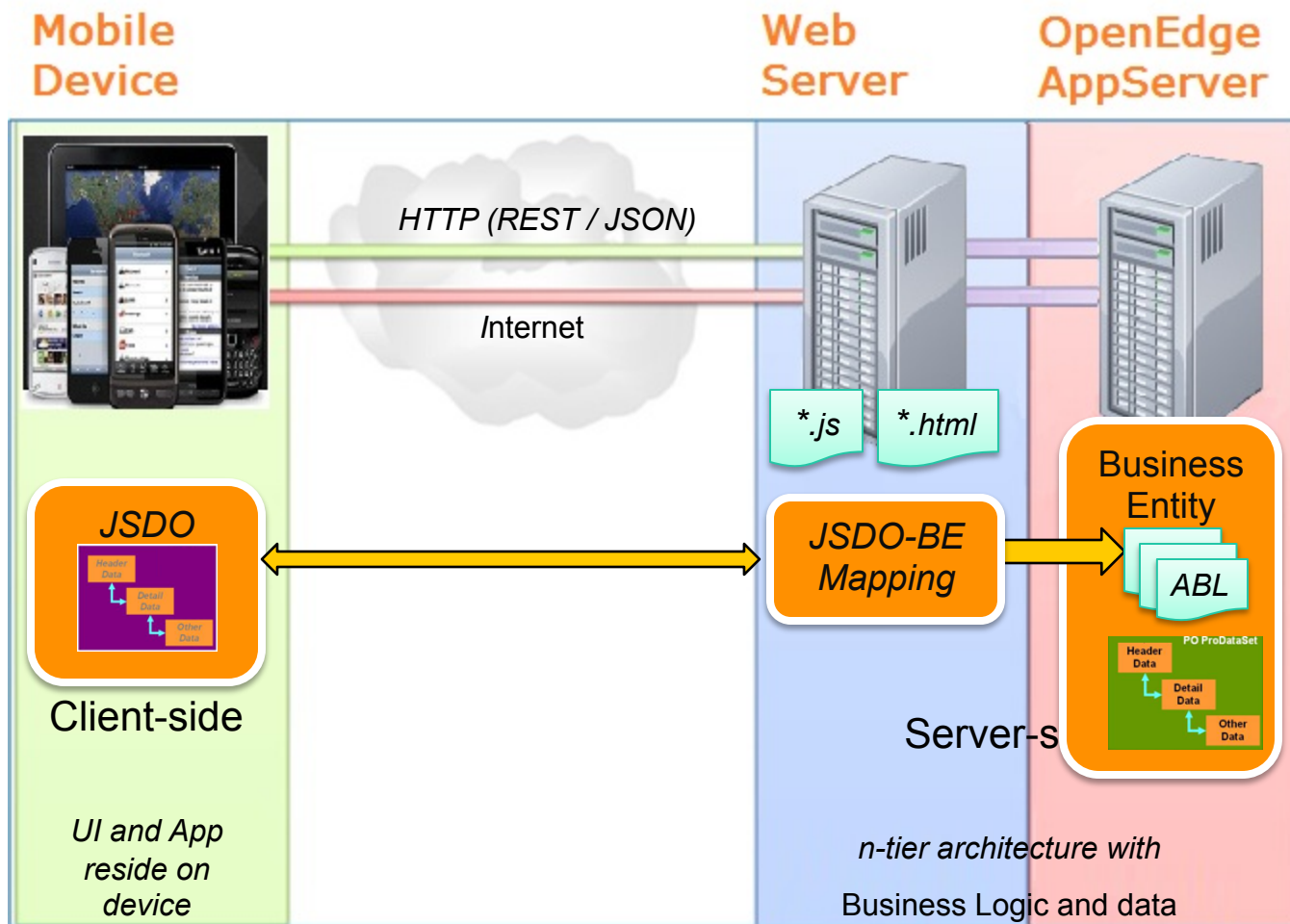
- ③ Discovered and installed from well-known app stores



Mobile UI: Complete Mobile Platform in OpenEdge



OpenEdge Mobile UI Using JSDO



Drag & Drop to Build the Mobile UI

The screenshot displays the Progress Mobile UI Builder interface. On the left, a 'COMPONENTS' panel lists various UI controls under 'Design' and 'Data' categories. The central workspace shows a mobile app design with a 'Page Title' and three buttons labeled 'Button 1', 'Button 2', and 'Button 3'. A red dashed box highlights 'Button 1'. On the right, the 'PROPERTIES - Button' panel shows settings for the selected button, including 'Text' (Button 1), 'Icon' (None), 'Position' (left), 'Swatch' (inherit), 'Tab Index' (2), 'Name' (mobilebutton1_4), 'Visible' (checked), and 'Class Name' (button). At the bottom, an 'EVENTS' panel shows a table with columns for Index, Component, Event, Action, and Editor. The table contains one row for the selected button with the event 'Click' and action 'Select Action'. A 'Show All' button is located to the right of the table, and an 'Add event' button is at the bottom right.

home / mobilecontainer1 / mobilegroupedbuttons1_3 / mobilebutton1_4

COMPONENTS

Controls

- Button
- GroupedButtons
- Input
- Textarea
- Datepicker
- Label
- Link
- Radio
- Checkbox
- Slider
- Toggle
- Select
- List
- Image
- Grid
- Navbar
- Search
- CollapsibleSet
- Collapsible
- Spacer

Page Title

Button 1

Button 2

Button 3

Push Here for More Information

PROPERTIES - Button

Custom

- Text: Button 1
- Icon: None
- Position: left
- Swatch: inherit
- Tab Index: 2

Common

- Name: mobilebutton1_4
- Visible: ☒
- Class Name: button

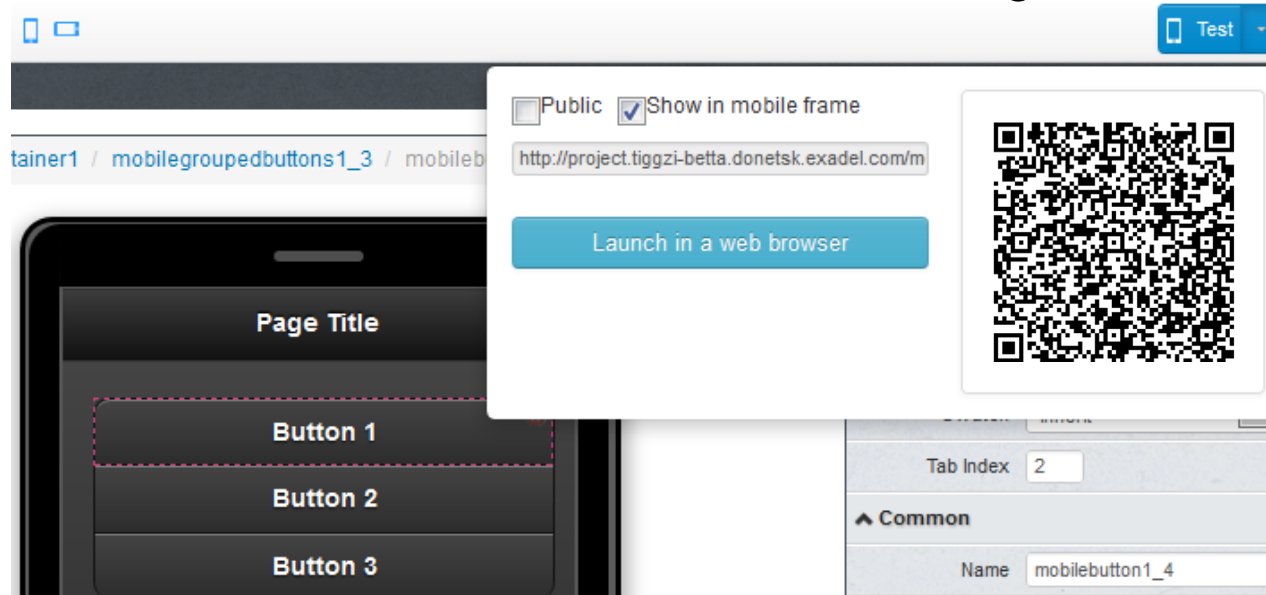
EVENTS

Index	Component	Event	Action	Editor
	mobilebutton1_4	Click	Select Action	

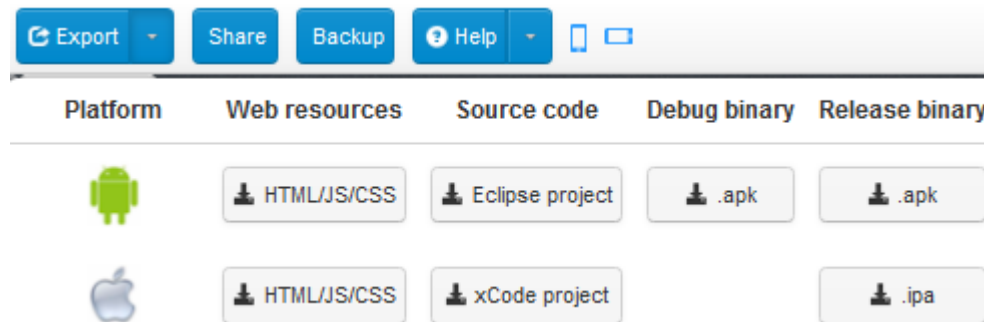
Show All

Add event

- Built-in Mobile Device Emulators for Testing



- Deploy to a Variety of Target Devices



A small example



Add Food

White, Granulated (Sugar)

Number of Servings 3 ▶

Serving Size 16434761386938900... ▶

Add Now

Nutritional Facts

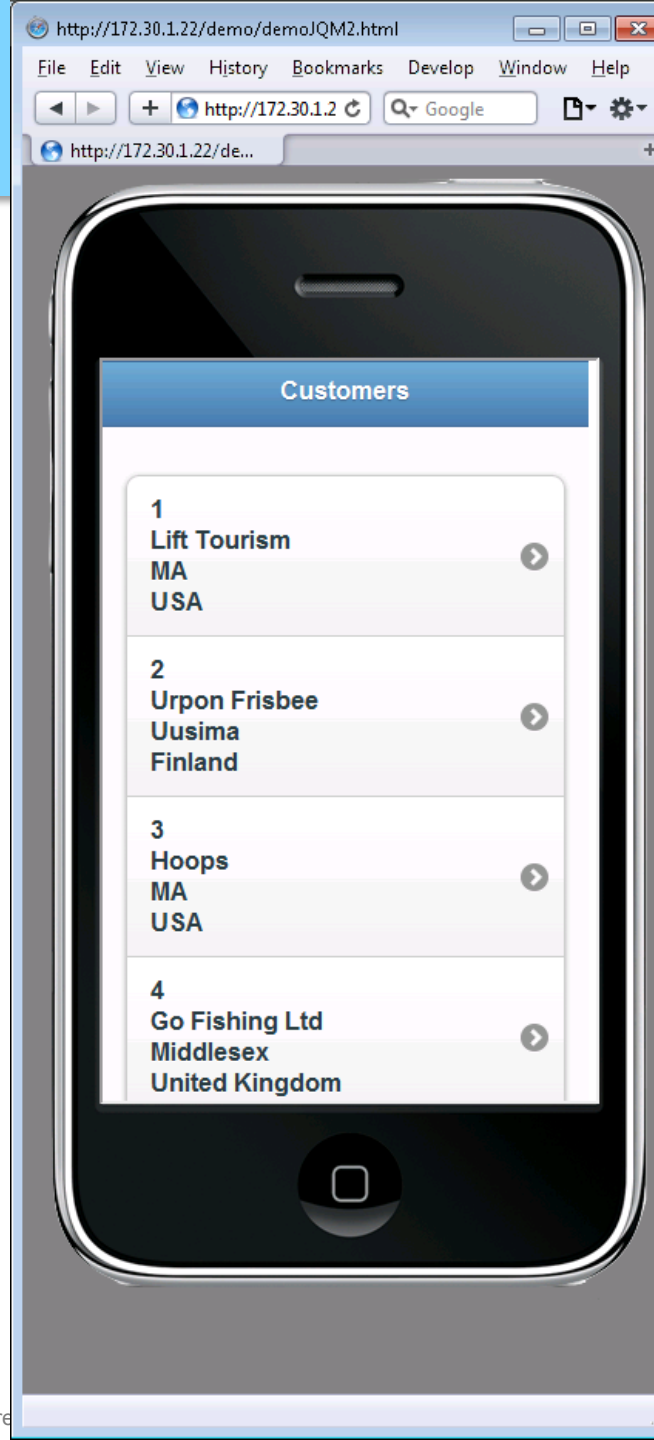
Calories 214...	Total Fat (g)	N/A
	Saturated (g)	10778762
		45418480
		00.0 g
	Polyunsaturated (g)	37168466
		68166730
		000000.0
		g
	Monounsaturated (g)	0.0 g
	Trans (g)	N/A

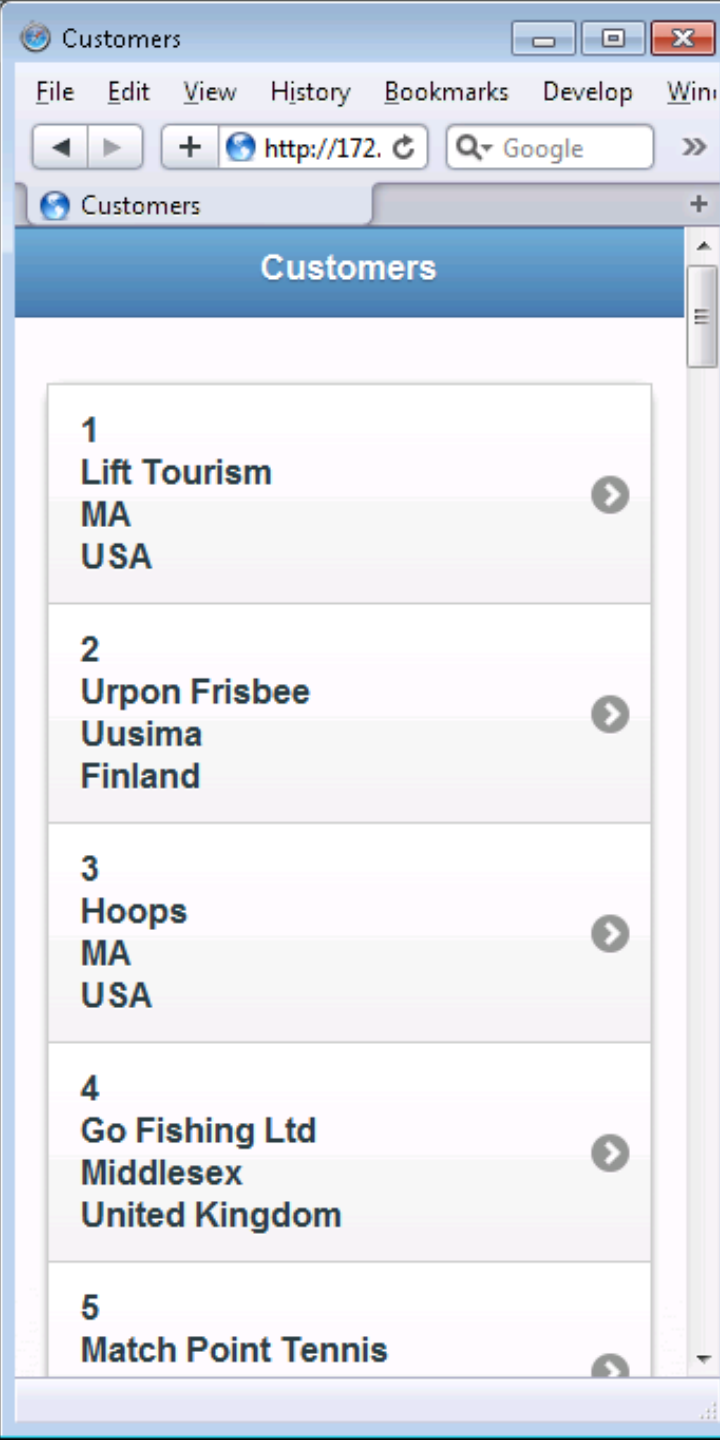


Google play



GET ROMANCE AND LOVE
STORIES FROM \$1.99





Simple Demo

Code

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Customers</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
    <link rel="stylesheet" href="my.css" />
    <style>
      /* App custom styles */
    </style>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
    <script src="progress.js"></script>
    <script src="customers.js"></script>
  </head>
  <body>
    <div data-role="page" id="custlist">
      <div data-theme="b" data-role="header">
        <h3>Customers</h3>
      </div>
      <div data-role="content">
        <ul id="listview" data-role="listview" data-divider-theme="b" data-inset="true">
        </ul>
      </div>
    </div>

    <div data-role="page" id="custdetail" data-add-back-btn="true" data-theme="b">
      <div data-role="header"><h1>Customer</h1></div>
      <div data-role="content">
        <form action="" id="customerform">

        </form>
      </div>
    </div>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Customers</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
    <link rel="stylesheet" href="my.css" />
    <style>
      /* App custom styles */
    </style>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js"></script>
    <script src="progress.js"></script>
    <script src="customers.js"></script>
  </head>
  <body>
    <div data-role="page" id="custlist">
      <div data-theme="b" data-role="header">
        <h3>Customers</h3>
      </div>
      <div data-role="content">
        <ul id="listview" data-role="listview" data-divider-theme="b" data-inset="true">
        </ul>
      </div>
    </div>

    <div data-role="page" id="custdetail" data-add-back-btn="true" data-theme="b">
      <div data-role="header"><h1>Customer</h1></div>
      <div data-role="content">
        <form action="" id="customerform">
        </form>
      </div>
    </div>
  </body>
</html>
```



```

var customers;
var forminitialized = false;

$(document).ready(function() {
    var session = new progress.data.Session( 'catalog.json' );
    session.login("", "");

    customers = new progress.data.JSDO( 'customer' );
    customers.onSuccessFn = onSuccessFillCustomers;
    customers.fill();

    customers.setDetailPage("#custdetail");
    customers.setListView(
        {   name: '#listview',
            display: '{CustNum}<br>{Name}<br>{State}<br>{Country}'
        }
    );

    // Add triggers for buttons
    $('#custdetail').live('pagecreate', function(event) {
        $('#btnAdd').bind('click', function(event) {
            var result = customers.getNextCustNum();

            var jsbuffer = customers.add({CustNum: result.nextCustNum, Balance: 10000, State: 'MA'});

            jsbuffer.display();
        });
        $('#btnSave').bind('click', function(event) {
            var jsbuffer = customers.findById($('#id').val());
            jsbuffer.assign();
            customers.saveChanges(null, onSuccessSaveCustomer);
        });
        $('#btnDelete').bind('click', function(event) {
            var jsbuffer = customers.findById($('#id').val());
            jsbuffer.remove();
            customers.saveChanges(null, onSuccessSaveCustomer);
        });
    });
});

function onSuccessSaveCustomer(xhr) {

```

```

function onSuccessSaveCustomer(xhr) {
    if (!xhr.success && (xhr.message)) {
        alert(xhr.message);
    }
    else {
        $.mobile.changePage('#custlist', { transition: 'flip' } );
        customers.fill(null, onSuccessFillCustomers);
    }
}

function onSuccessFillCustomers(xhr) {

    // Display records to listview
    customers.foreach(function(customer) {
        customer.display();
    });

    // Add trigger for 'tap' event to items
    $("li").each (
        function (index) {
            $(this).bind('tap',
                function (event,ui) {
                    var jsbuffer = customers.findById($(this).attr('data-id'));
                    jsbuffer.display();
                });
        });

    if (!forminitialized) {
        initializeForm();
    }
}

function initializeForm() {
    $('#customerform').html(
        customers.getFormFields()
        + '<input type="button" value="Add" id="btnAdd" />'
        + '<input type="button" value="Save" id="btnSave" />'
        + '<input type="button" value="Delete" id="btnDelete" />'
    );
    forminitialized = true;
}

```

```

function onSuccessSaveCustomer(xhr) {
    if (!xhr.success && (xhr.message)) {
        alert(xhr.message);
    }
    else {
        $.mobile.changePage('#custlist', { transition: 'flip' });
        customers.fill(null, onSuccessFillCustomers);
    }
}

function onSuccessFillCustomers(xhr) {

    // Display records to listview
    customers.foreach(function(customer) {
        customer.display();
    });

    // Add trigger for 'tap' event to items
    $("li").each (
        function (index) {
            $(this).bind('tap',
                function (event,ui) {
                    var jsbuffer = customers.findById($(this).attr('data-id'));
                    jsbuffer.display();
                });
        });

    if (!forminitialized) {
        initializeForm();
    }
}

function initializeForm() {
    $('#customerform').html(
        customers.getFormFields()
        + '<input type="button" value="Add" id="btnAdd" />'
        + '<input type="button" value="Save" id="btnSave" />'
        + '<input type="button" value="Delete" id="btnDelete" />'
    );
    forminitialized = true;
}

```

```
// Display records to listview  
customers.foreach(function(customer) {  
    customer.display();  
})  
);
```

```

@openapi.openedge.export FILE(type="REST", executionMode="singleton").
@progress.service.resource FILE(name="Item", schemaName="dsItem").

USING Progress.Lang.*.

BLOCK-LEVEL ON ERROR UNDO, THROW.

CLASS ItemBE:

    {dsDefs.i}
    DEFINE DATA-SOURCE data-srcItem FOR ITEM.

    CONSTRUCTOR PUBLIC ItemBE ():
        SUPER ().
    END CONSTRUCTOR.

    /* Get Items */
    /* relative REST URI = /Item
       type                = LOAD
       method name         = LoadItem */

    @openapi.openedge.export(type="REST").
    @progress.service.resourceMapping(type = "REST", methodType="FILL", URI="/Item", methodDefault="true").
    METHOD PUBLIC VOID LoadItem( OUTPUT DATASET dsItem ):

        EMPTY TEMP-TABLE ttItem.
        BUFFER ttItem:ATTACH-DATA-SOURCE(DATA-SOURCE data-srcItem:HANDLE).
        BUFFER ttItem:SET-CALLBACK("AFTER-ROW-FILL", "assignItem", THIS-OBJECT).

        DATASET dsItem:FILL().
        BUFFER ttItem:DETACH-DATA-SOURCE().

    END METHOD.

    METHOD PUBLIC VOID assignItem( INPUT DATASET dsItem ):
        DEFINE VARIABLE pagenum AS CHARACTER NO-UNDO.

        ASSIGN
            ttItem.id = STRING(ROWID(Item)).

```

```

@openapi.openedge.export(type="REST").
@progress.service.resourceMapping(type = "REST", methodType="FILL", URI="/Item", methodDefault="true").
METHOD PUBLIC VOID LoadItem( OUTPUT DATASET dsItem ):

    EMPTY TEMP-TABLE ttItem.
    BUFFER ttItem:ATTACH-DATA-SOURCE(DATA-SOURCE data-srcItem:HANDLE).
    BUFFER ttItem:SET-CALLBACK("AFTER-ROW-FILL", "assignItem", THIS-OBJECT).

    DATASET dsItem:FILL().
    BUFFER ttItem:DETACH-DATA-SOURCE().

END METHOD.

METHOD PUBLIC VOID assignItem( INPUT DATASET dsItem ):
    DEFINE VARIABLE pagenum    AS CHARACTER                NO-UNDO.

    ASSIGN
        ttItem.id = STRING(ROWID(Item)).

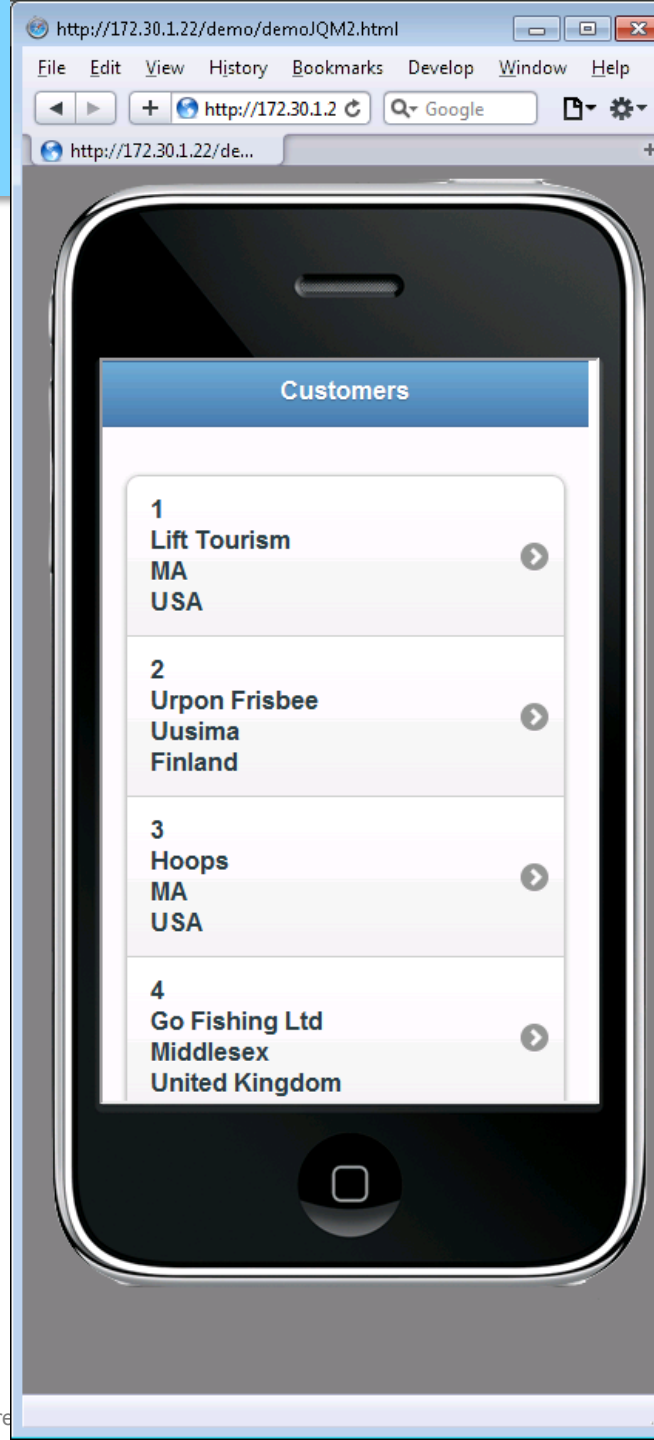
    IF Item.CatPage <> ? and Item.CatPage <> 0 THEN
        ASSIGN pagenum = STRING(Item.CatPage, "99999")
        ttItem.ItemImage = "images/cat" + pagenum + ".jpg".
    ELSE
        ASSIGN ttItem.ItemImage = "".

    RETURN.

END METHOD.

END.

```



- Mobile App Builder as a Service
 - White-labeled 3rd-party mobile app dev environment
 - Mobile capabilities continuously updated for best UI and expanding device support
- Tools to Create Mobile Applications
 - Visual UI designer
 - Automated mobile application builder
 - Mobile device emulation for testing
- Tight Integration with Progress Developer Studio for OpenEdge
 - Run ABL on AppServer from mobile device (using new REST adapter)
 - Ties to data in the OpenEdge database (using new JavaScript Data Objects)



- OpenEdge Mobile Provides A Complete End-to-end Solution For Supporting Mobile Devices With OpenEdge Applications
 - Tightly integrated with the premier OpenEdge development environment, Progress Developer Studio for OpenEdge
- Full Support For Creating Mobile Apps
- Ease-of-use Features To Simplify Interaction With OpenEdge Applications
 - JavaScript Data Objects to easily connect UI screens to the supporting data
 - REST Adapter to make it easy to execute OpenEdge business logic on the AppServer
- Available NOW! on Progress ESD

*Now,
what should i do while waiting to get
OpenEdge 11.2 ?*

*Have a coffee while you download.
Play with the demos at
<http://oemobiledemos.progress.com>*

Questions

email: gus@progress.com

PROGRESS
software