Using JSON to Simplify Mobile Applications with CICS

Andy Armstrong,
CICS System Test – Senior Inventor
IBM Hursley - UK

TAC-2622

Please Note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

What is JSON?

JavaScript Object Notation

- Lightweight, Human readable, *text based format for data*
- Simple structure
- Native JavaScript support
- Widely adopted by the industry

An alternative to XML

- Heavyweight structure
- Support usually provided by XML specific parsers (Eg. DOM/SAX)

```
var invocationData = {
            adapter: “GENAPPInquire”,
            procedure: “getCustomerDetails”,
            parameters: [otherUsername]
};
```

JavaScript Example

Bit more about JSON…

JavaScript Object Notation

- JSON represents data as name-value pairs (Eg. “name” : “Andy”)
- *Ideal format for data interchange*

Objects

```
var personObject = {
            "name":"John Johnson",
            "street":"Oslo West 555”,
            "age":33,
            "phone":"555 1234567”
};
var personAge = personObject.age;
```

Simple data access

Arrays

```
var employeesArray = [
            { "firstName":"John”, “lastName”:"Doe" },
            { "firstName":"Anna”, “lastName":"Smith" },
            { "firstName":"Peter”, "lastName": "Jones" }
];
var employeeName = employeesArray[1].lastName;
```

How does XML compare to JSON?

XML

```
<employees>
        <employee>
                <firstName>John</firstName>
                <lastName>Doe</lastName>
        </employee>
        <employee>
                <firstName>Anna</firstName>
                <lastName>Smith</lastName>
        </employee>
        <employee>
                <firstName>Peter</firstName>
                <lastName>Jones</lastName>
        </employee>
</employees>
```

300 Bytes Approx.

50,000 Example customer records:

XML: ~14 MB
JSON: ~7 MB

JSON

```
var employeesArray = [
  { "firstName":"John" , "lastName":"Doe" },
  { "firstName":"Anna" , "lastName":"Smith" },
  { "firstName":"Peter" , "lastName": "Jones" }
];
```

150 Bytes Approx.

It’s the same data, *but 50% smaller!*

JSON brings lots of benefits

The benefits of JSON data

- Describes the same data with less meta-information (tags)
- Processing is simpler, ideal for mobile devices
- Natively supported in Javascript - easy for application developersE
- Easy for Human’s to read – helps with application development
- Support is available in many languages ***(but not COBOL)***

This is all very well, but I have a question…

So why don’t we just use XML?

So why don’t we just use XML?

JSON is the mobile format of choice

- The growth in mobile helped boost the popularity of JSON
- The lightweight data format is ideally suited to mobile data transfer
- As a result numerous tools and frameworks now support JSON…

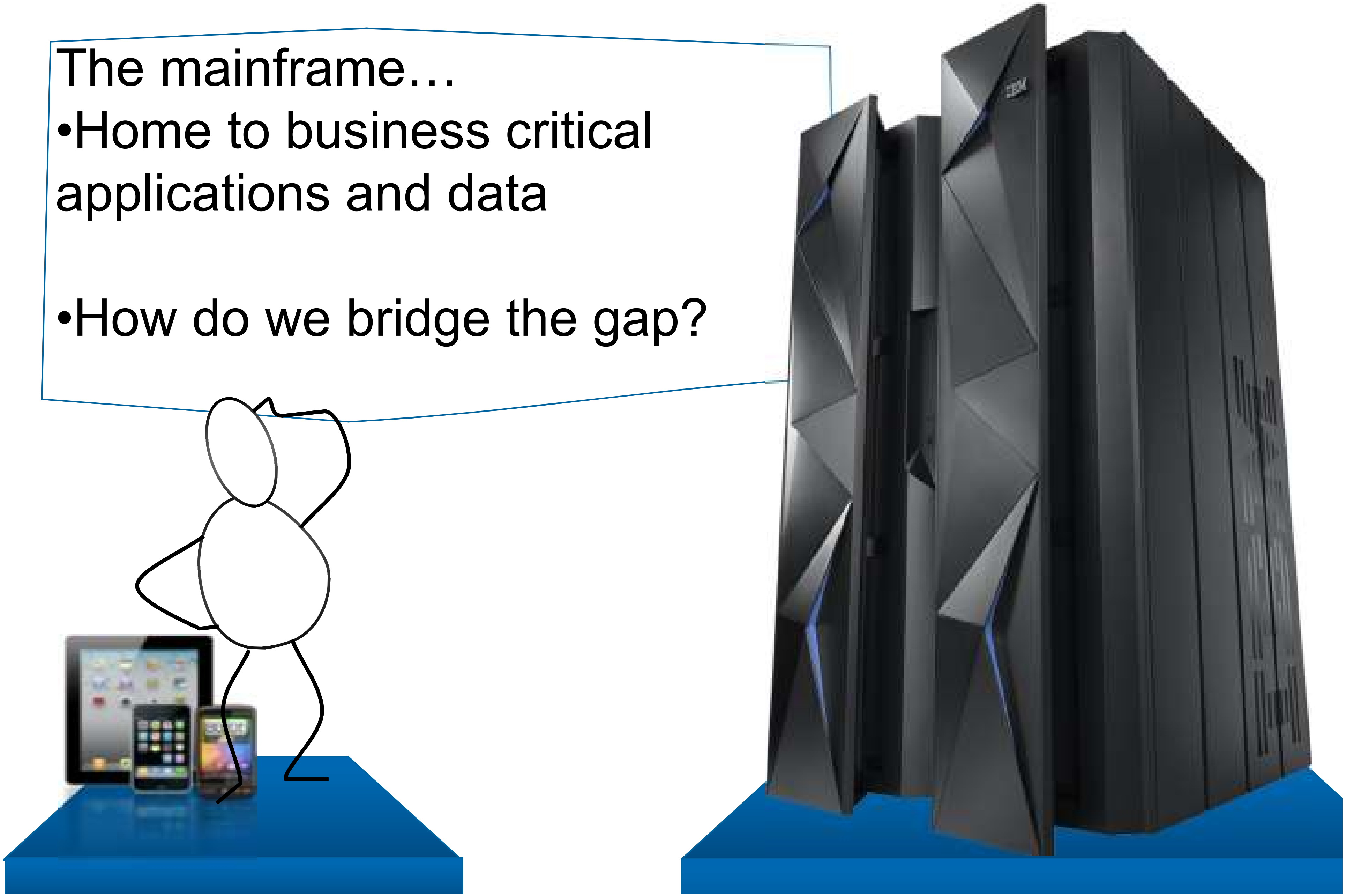
IBM Worklight uses JSON:

- For communication between a mobile application and the Worklight Server
- Provides a JSON Store for offline storage of data
- Automatically converts Webservice SOAP replies into JSON

Numerous other frameworks depend upon JSON data…

Eg. jQuery & dojo use JSON:
•JSON Store
•Ajax calls
•All data interchange

What about the mainframe?



The mainframe…

- Home to business critical applications and data

- How do we bridge the gap?

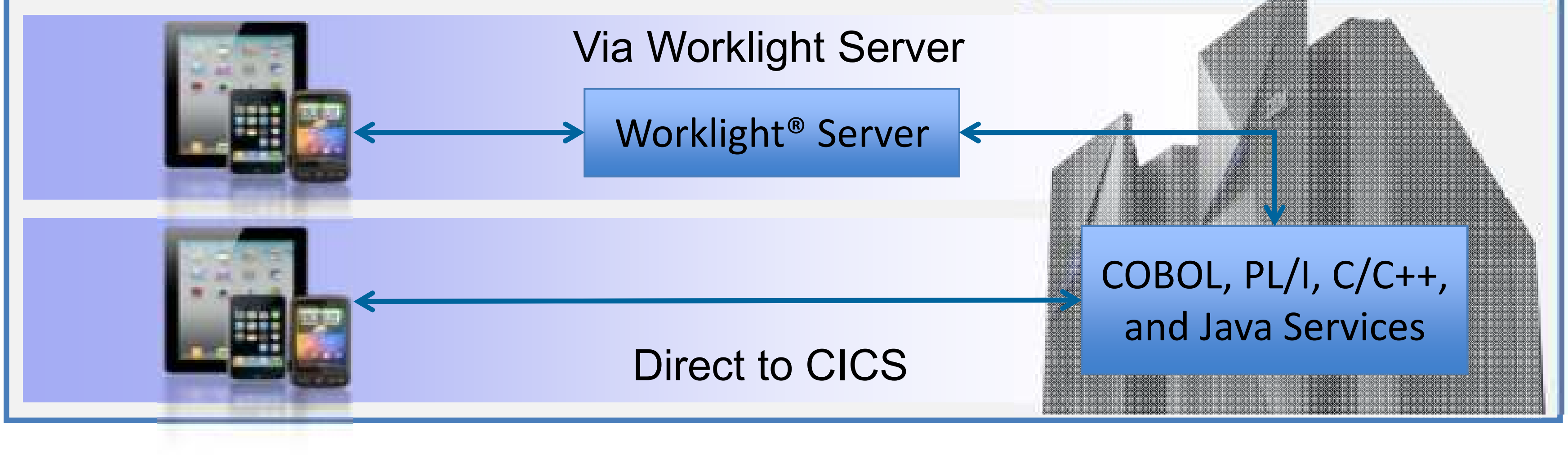
Bridging the gap between mobile and CICS

Available 2Q 2013 on CICS TS V4.2 & V5.1

Mobile extensions – simplified integration with mobile devices

CICS Transaction Server Feature Pack for Mobile Extensions

Agile integration	*Service enablement*	*Scalable platform*
JSON data and RESTful interface makes integration with mobile devices simple and efficient	*COBOL, C/C++, PL/I and Java programs can be RESTful service providers*	*CICS TS V5.1's advanced scalability makes it the ideal platform for managing mobile workloads*



What is REST?

The basics of RESTful services

- REST is an architectural style, unlike SOAP which is a protocol
- Typically use JSON to describe the data for the request
- Uses the HTTP verbs (below) to indicate the nature of an operation
- Uses the URI to determine the resource(s) to operate upon

Create	**Read**	**Update**	**Delete**
HTTP **Post**	HTTP **Get**	HTTP **Put**	HTTP **Delete**

Example URIs

A collection

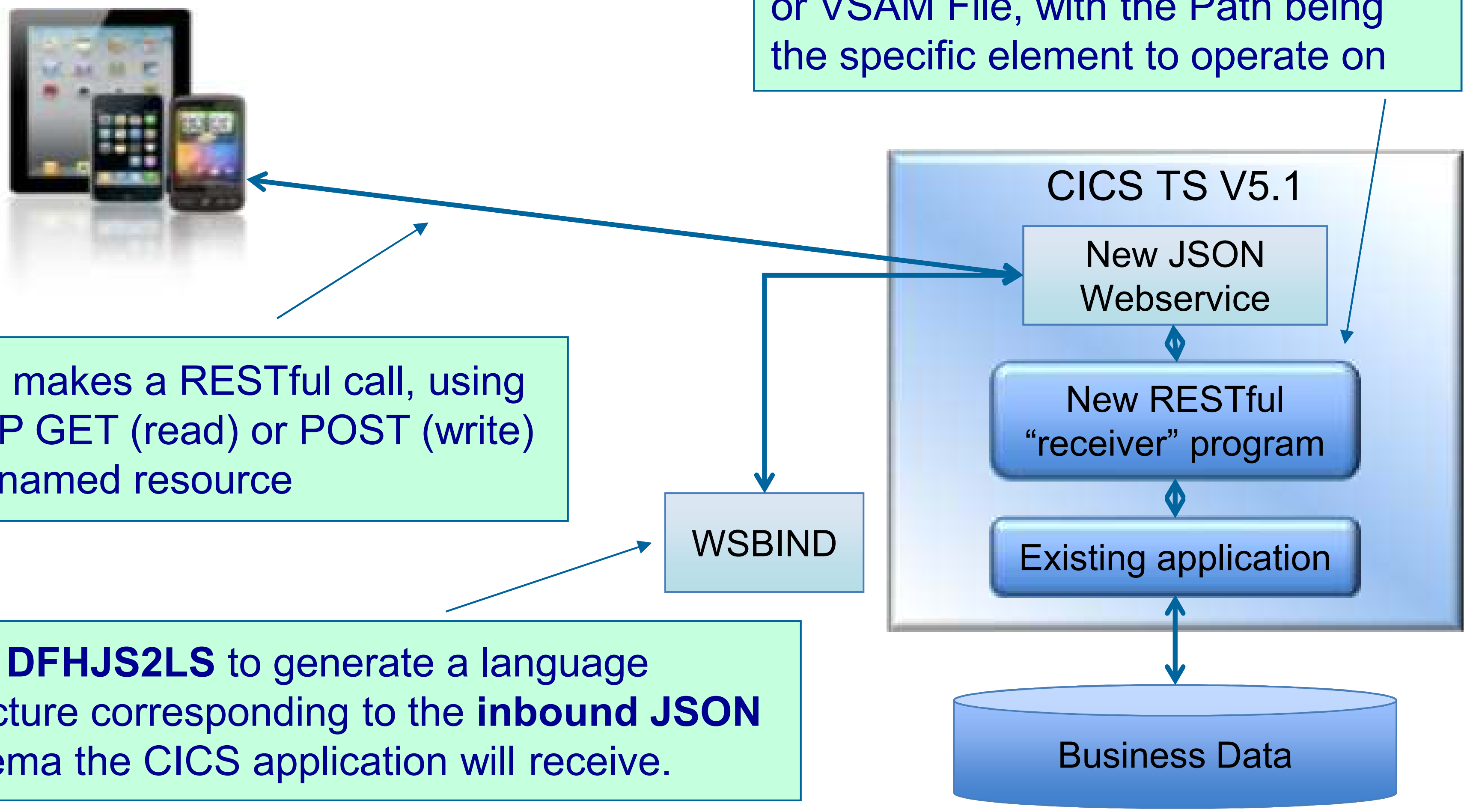
http://my.cics.example/resources/

http://my.cics.example/resources/specific_resource

A Top down scenario… Starting with existing JSON

Connect a mobile application to CICS using the RESTful JSON interface

Wrapper application in CICS uses the URL to identify the resource, DB or VSAM File, with the Path being the specific element to operate on



App. makes a RESTful call, using HTTP GET (read) or POST (write) to a named resource

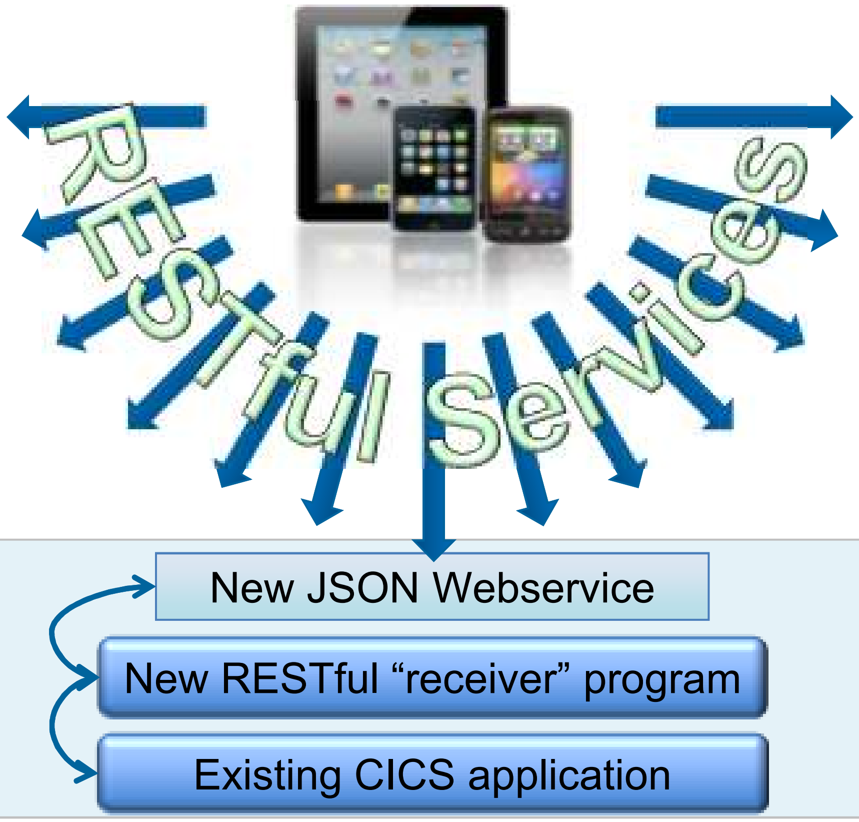
Use DFHJS2LS to generate a language structure corresponding to the **inbound JSON** schema the CICS application will receive.

Top down scenario

Goal:
Integrate mainframe into existing mobile application

"To help us deliver world class services we will standardize our service delivery on a RESTful architecture, promoting a uniform interface to all our consumers." – An IT Architect might mandate

Current environment:
Deployed mobile Apps have standardized the JSON service request interface, but the mainframe has been unable to consume it



New JSON Webservice

New RESTful "receiver" program

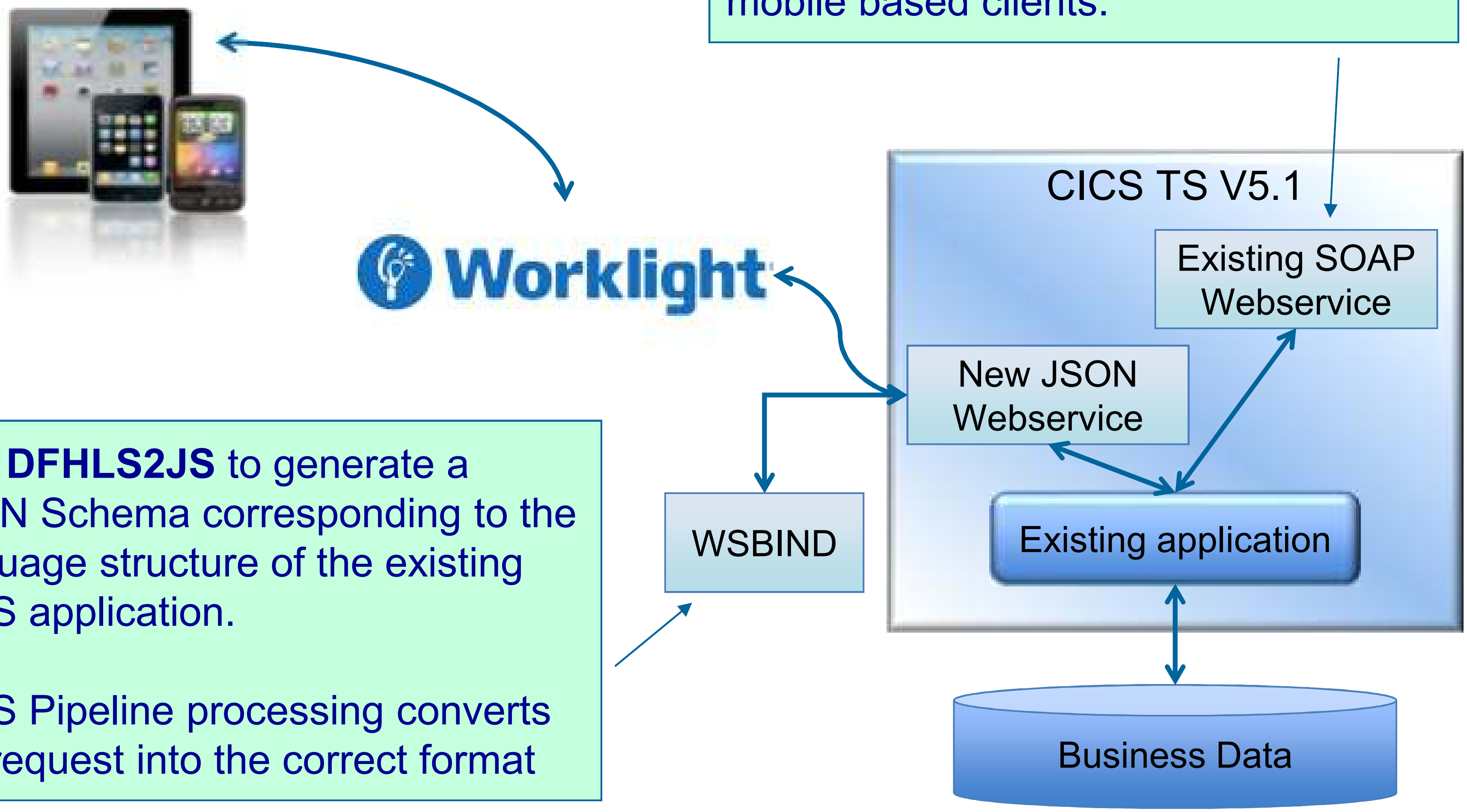
Existing CICS application

Solution:
Use top-down approach to consume standardized JSON

Bottom up scenario - start with language structure

Exposing an existing CICS application as a JSON callable service

Existing SOAP Webservices remain unaffected by the introduction of new mobile based clients.



Use DFHLS2JS to generate a JSON Schema corresponding to the language structure of the existing CICS application.

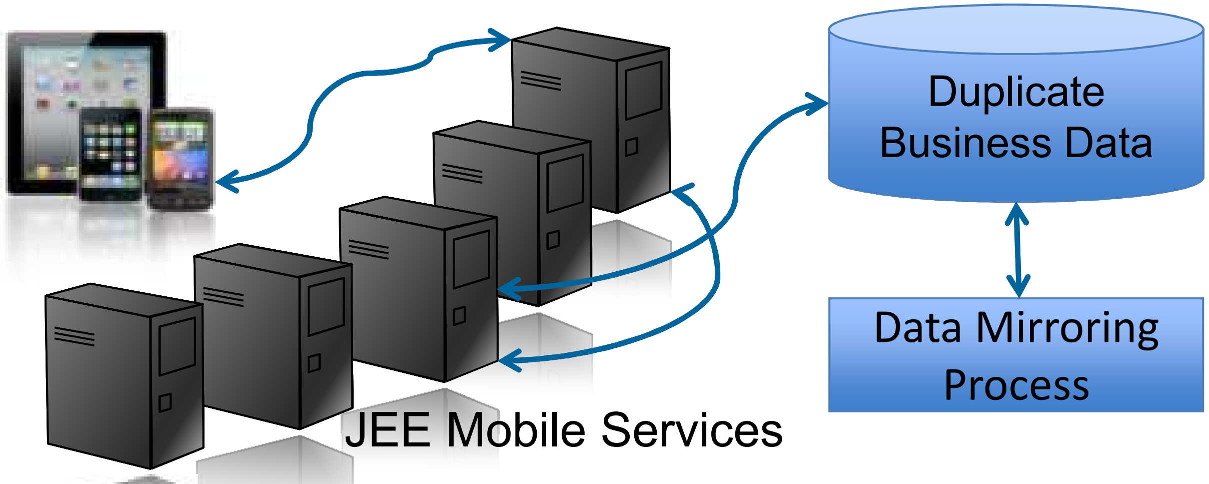
CICS Pipeline processing converts the request into the correct format

Bottom up scenario

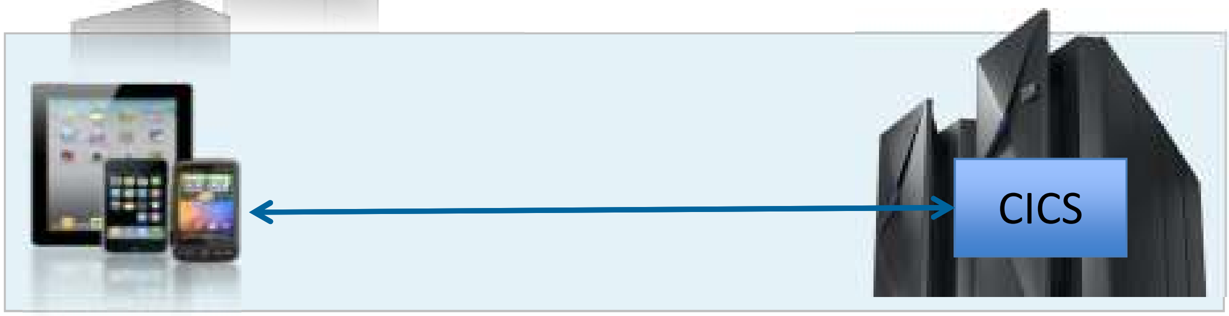
Goal:
Deliver mobile solution and remove data and service duplication

"The quick solution we implemented in order to get to market faster creates a headache for us in terms of duplicate data and services.
We must transform our mobile offering into a well architected solution." – An IT Architect might mandate

Current environment:
Mobile solution exists in parallel and duplicates data and services to support the mobile offering



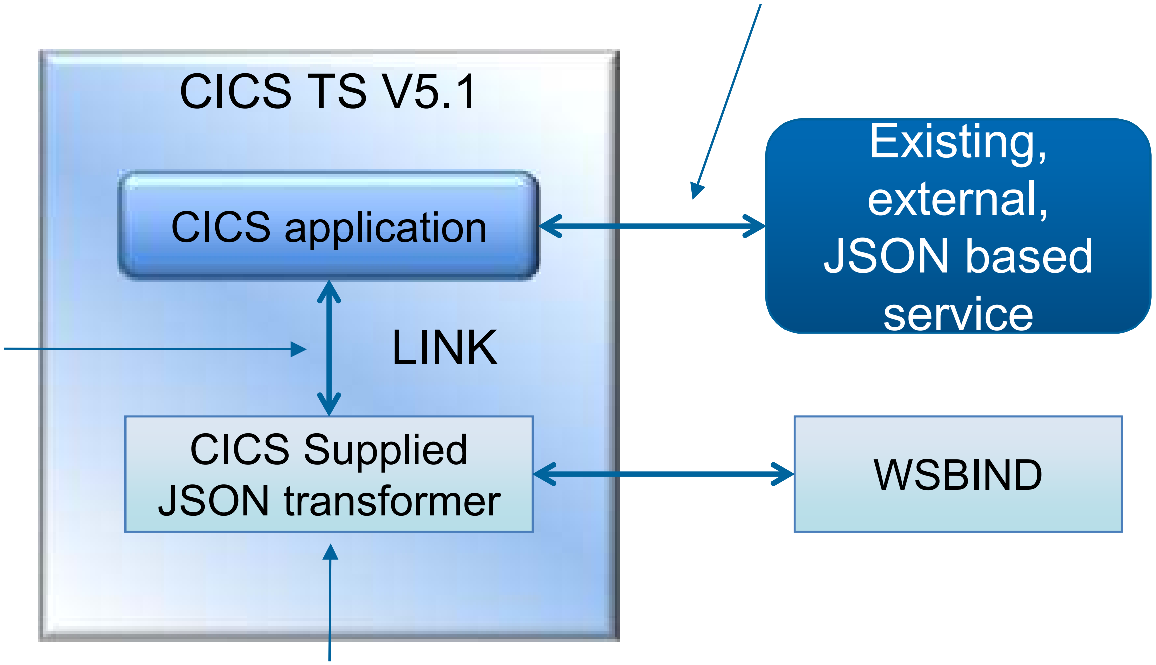
Solution:
Consolidate mobile solution onto the mainframe



Call a service to convert data into JSON format

A CICS application wants to convert data into JSON format to call an external service

Use a EXEC CICS WEB OPEN call to invoke the target external service passing the transformed JSON data


CICS TS V5.1
CICS application
LINK
CICS Supplied JSON transformer
Existing, external, JSON based service
WSBIND

Analogous to using the **XML Transform API**, the CICS application LINKs to the JSON transformer to convert binary data into the desired JSON format

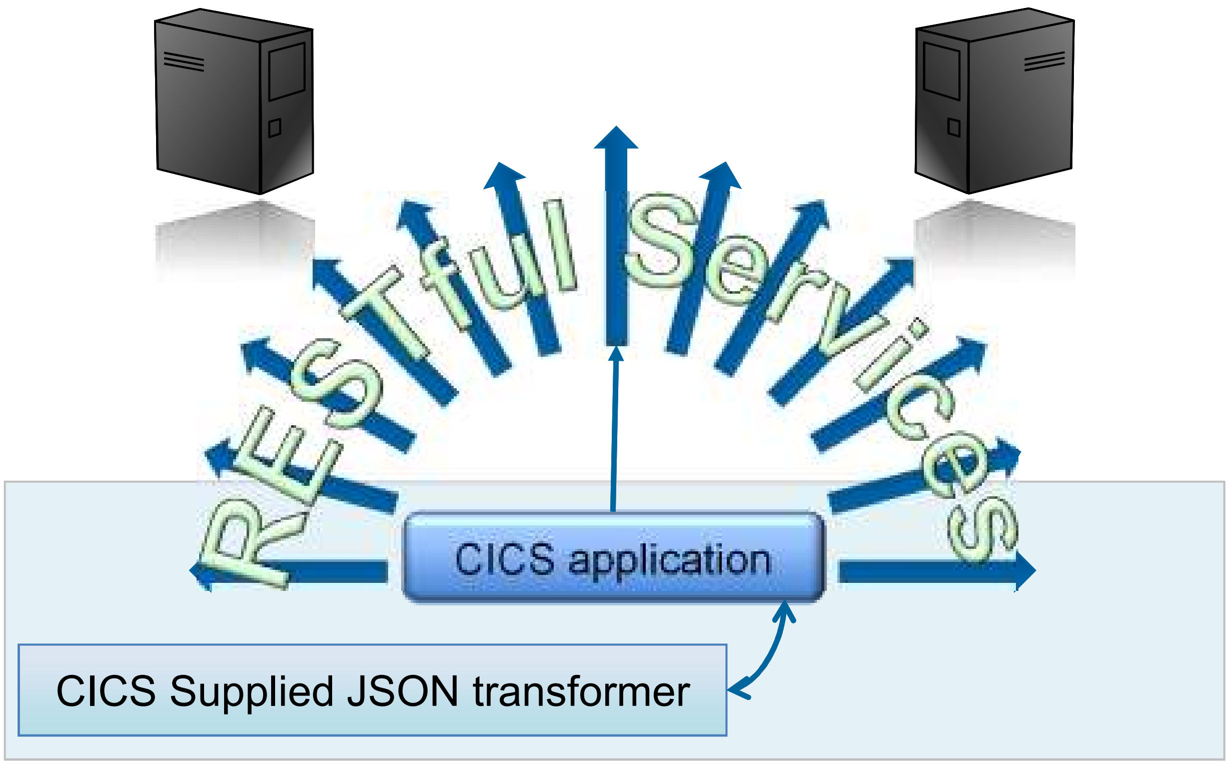
The JSON transformer works both ways…
Pass in **JSON to** generate **binary** data.
Pass in **binary** data **to** generate **JSON** data.

Service call scenario

Goal:
Exploit internal services that are only available through a RESTful interface

"We've standardized our internal service delivery model on RESTful interfaces and need to phase out our old and difficult to maintain alternatives. Key to our success is integrating our mainframe into this solution." – An IT Architect might mandate

Current environment:
RESTful services are deployed throughout the organisation. Mainframe applications need to exploit these services



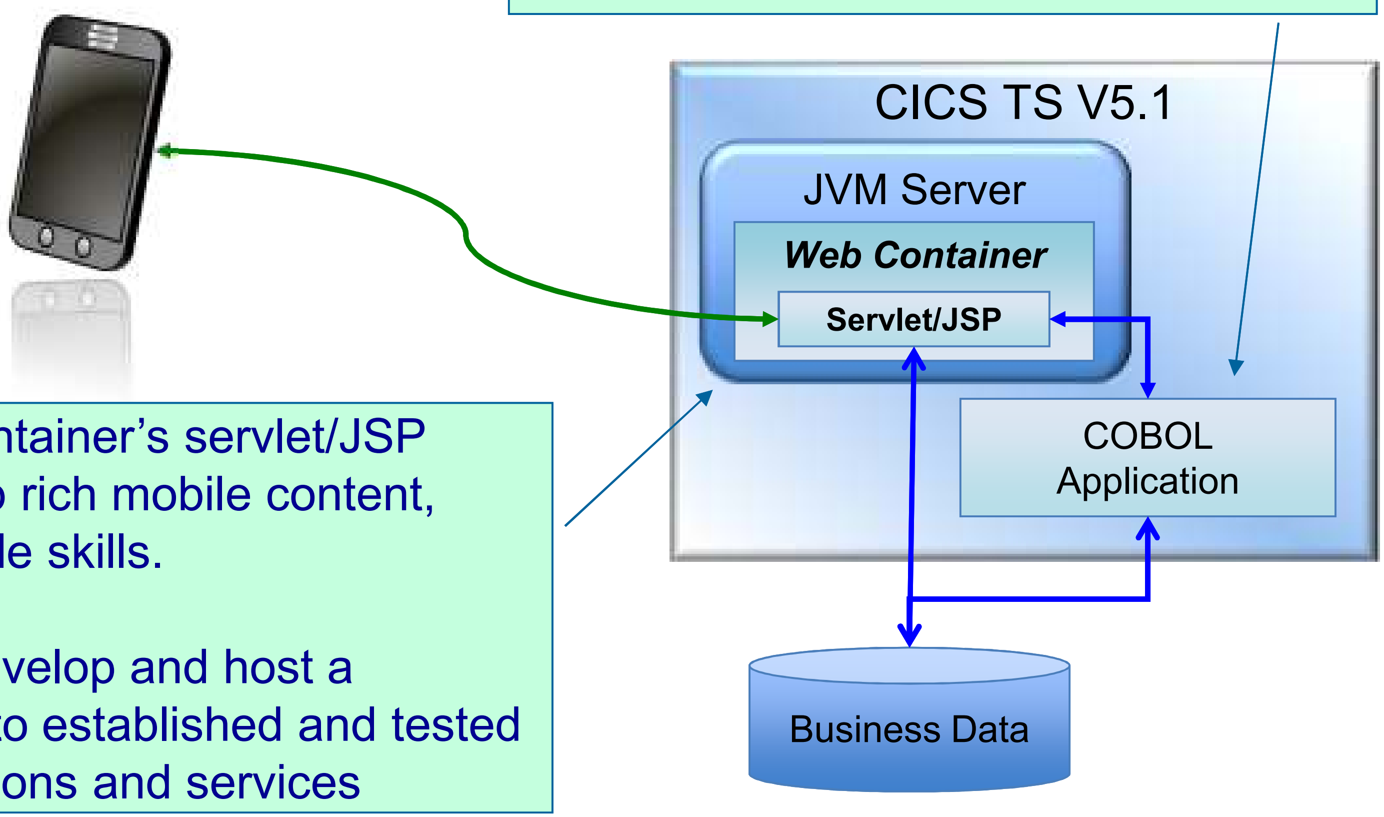
Solution:

Use JSON transformer to consume services

Doing it all Java style – Liberty Profile

RESTful services can now be hosted within the CICS Web Container, with new support for the JAX-RS API

Link to existing C/C++, COBOL, PL/I, and Java applications to exploit existing enterprise applications and services



Exploit the web container's servlet/JSP features to develop rich mobile content, building on available skills.

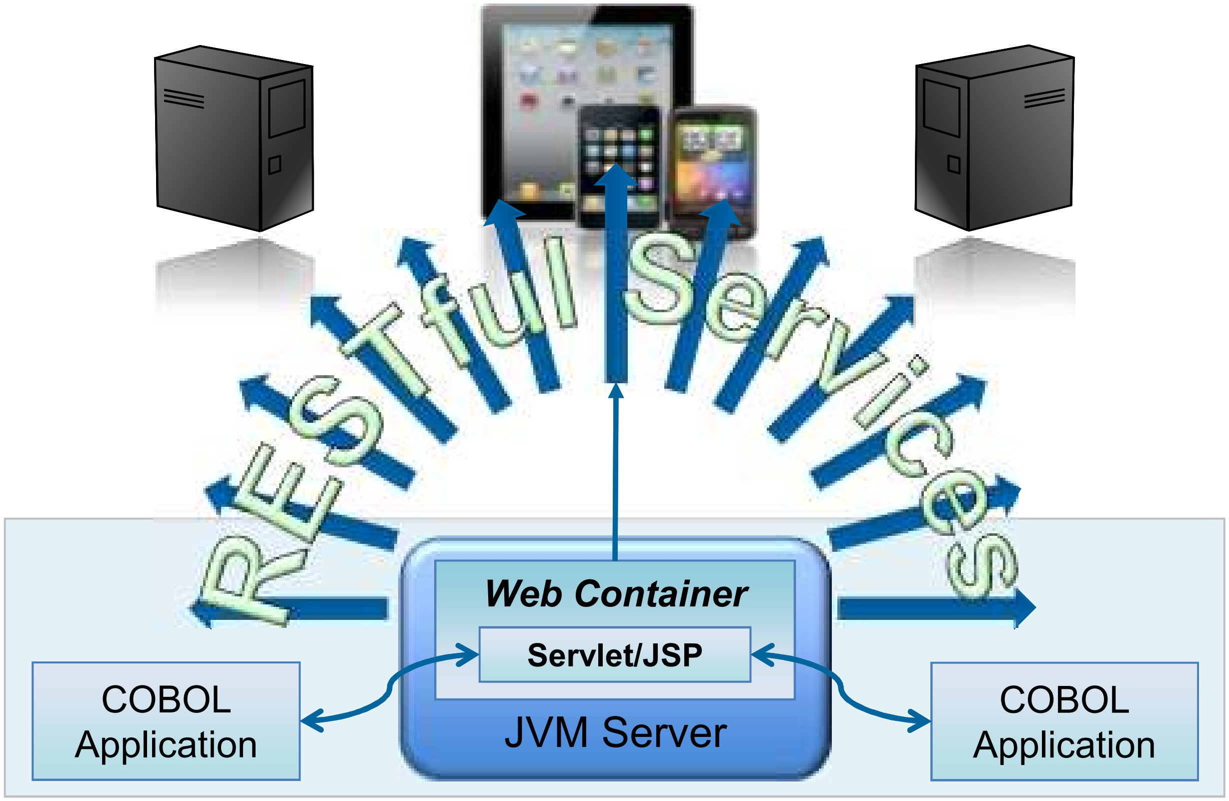
Ideal location to develop and host a RESTful interface to established and tested enterprise applications and services

Web Container Scenario

Goal:
Create a RESTful interface layer to existing mainframe services

"We've standardized our internal service delivery model on RESTful interfaces and need to phase out our old and difficult to maintain alternatives. Key to our success is integrating our mainframe into this solution." – An IT Architect might mandate

Current environment:
Enterprise services have been developed piecemeal and do not have a standardized interface



Solution:
Create standard RESTful interface to CICS services

Why CICS and Mobile makes sense

CICS is the ideal environment for hosting mobile applications

- CICS hosts key business applications
- CICS already provides service enablement of applications
- CICS is the broker of key business data

Agile

CICS' Service Agility features help your mobile applications keep pace with the market

Dynamic

CICS Event Processing lets you augment applications without code changes

Scalable

CICS provides phenomenal workload management capable of dealing with increased mobile workload

Mobile extensions – simplified integration with mobile devices

CICS Transaction Server Feature Pack for Mobile Extensions

Scalable platform

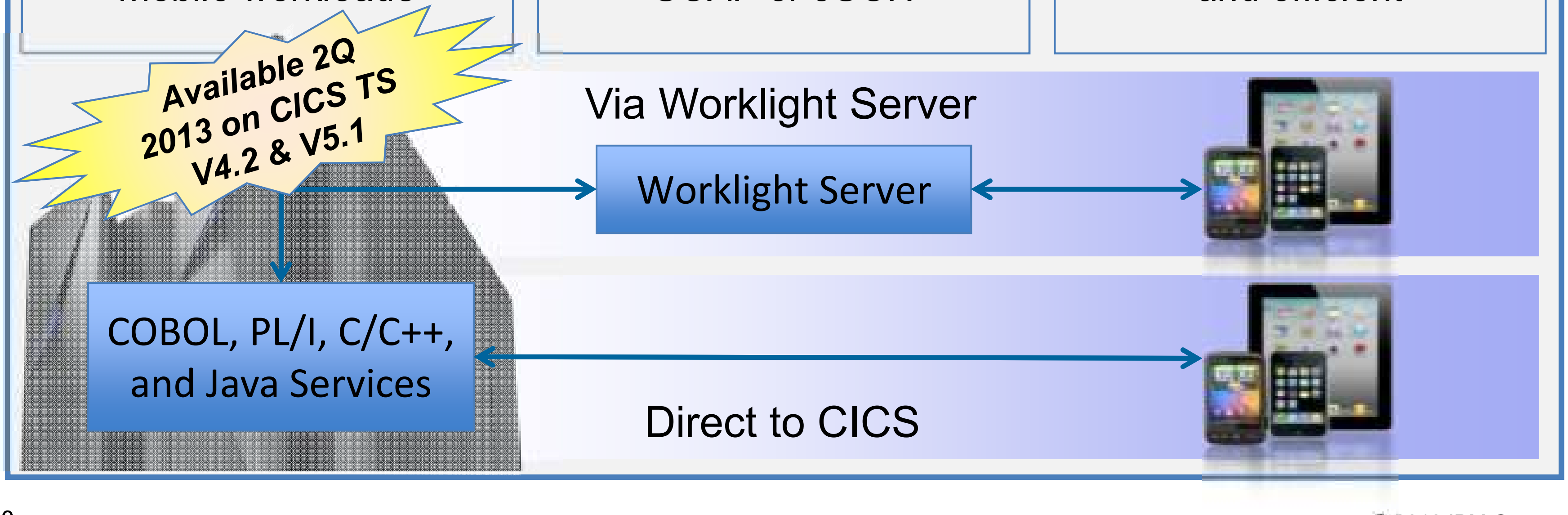
CICS TS V5.1's advanced scalability makes it the ideal platform for managing mobile workloads

Service enablement

Service enable existing COBOL, C/C++, PL/I and Java programs using either SOAP or JSON

Agile integration

JSON data and RESTful interface makes integration with mobile devices simple and efficient



Available 2Q 2013 on CICS TS V4.2 & V5.1

Via Worklight Server

Worklight Server

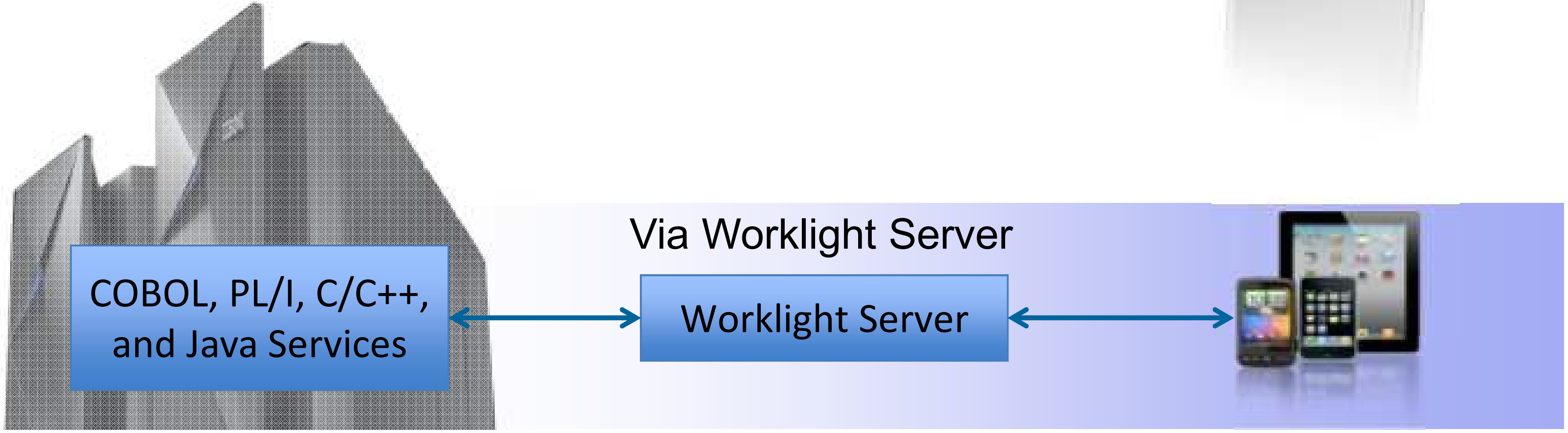
COBOL, PL/I, C/C++, and Java Services

Direct to CICS

CICS Transaction Server

Let’s see an example of how easy it is to connect a mobile device to CICS TS V5.1, via Worklight Server.

Mobile DEMO

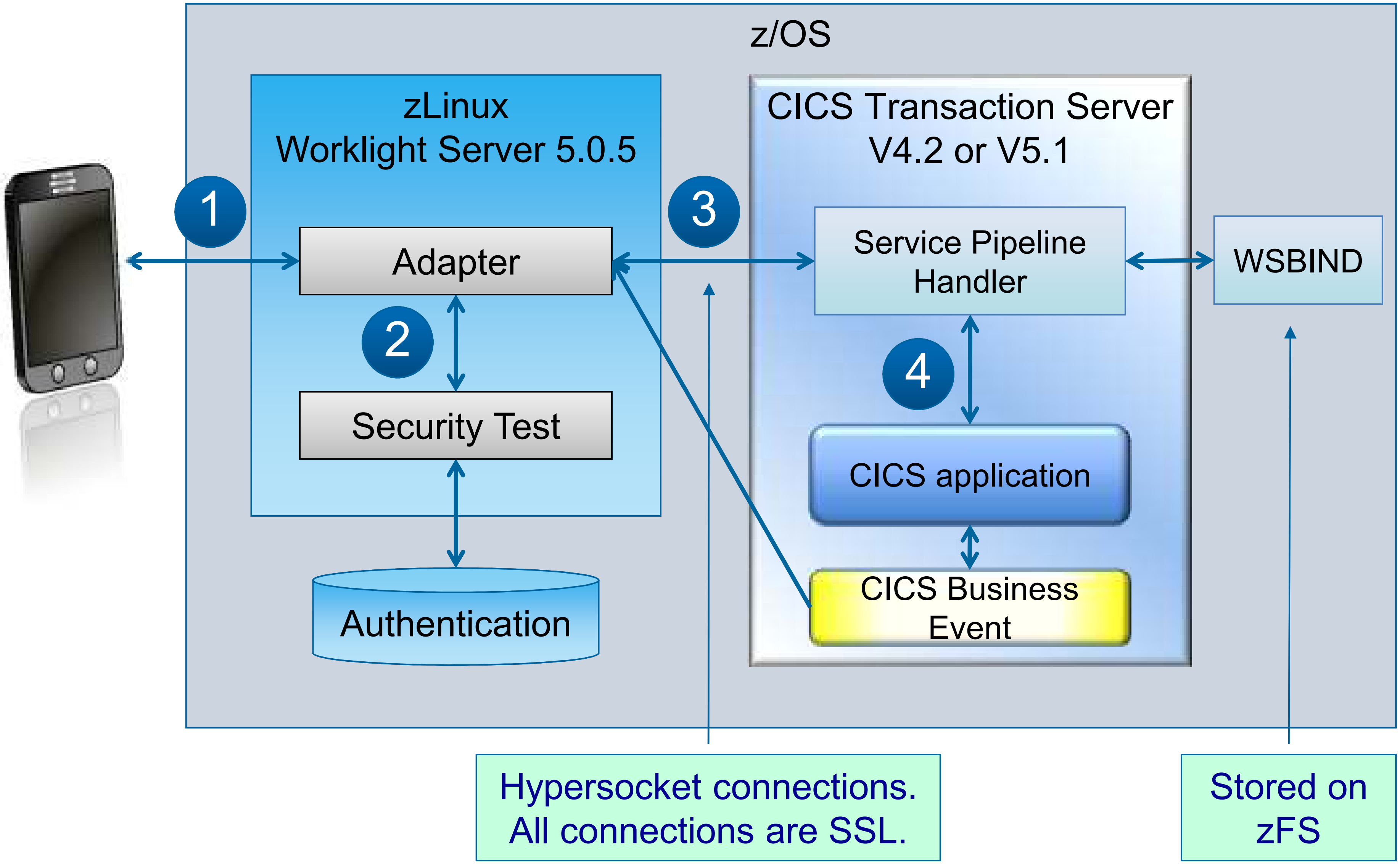


Via Worklight Server

COBOL, PL/I, C/C++, and Java Services

Worklight Server

Mobile Demo – Architecture and basic call flow

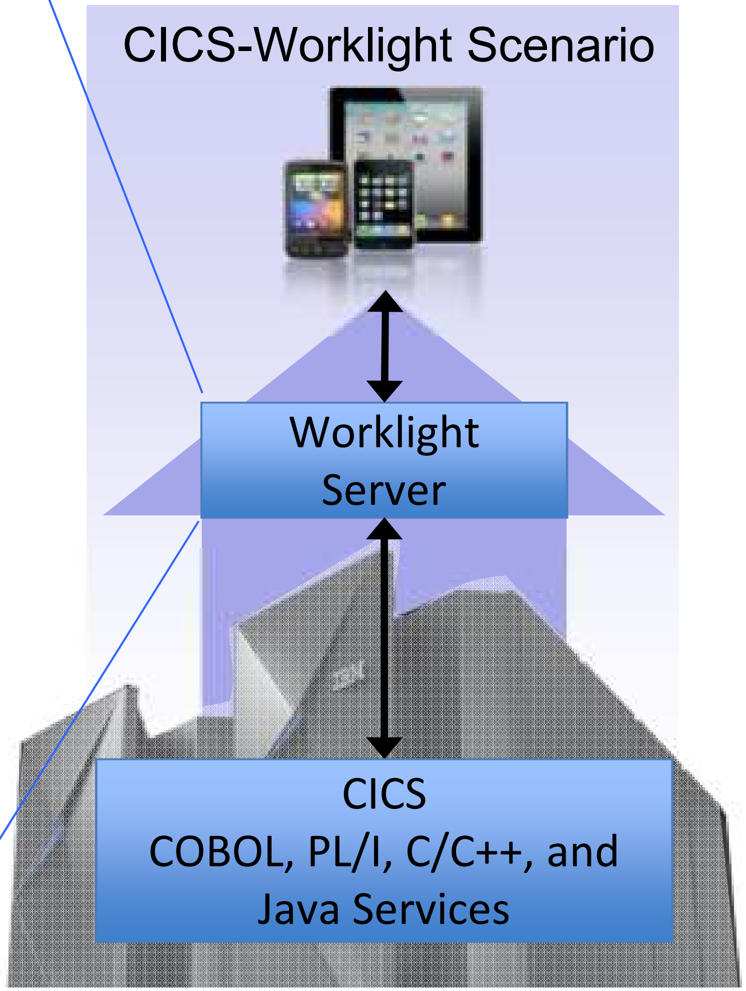


Hypersocket connections.
All connections are SSL.

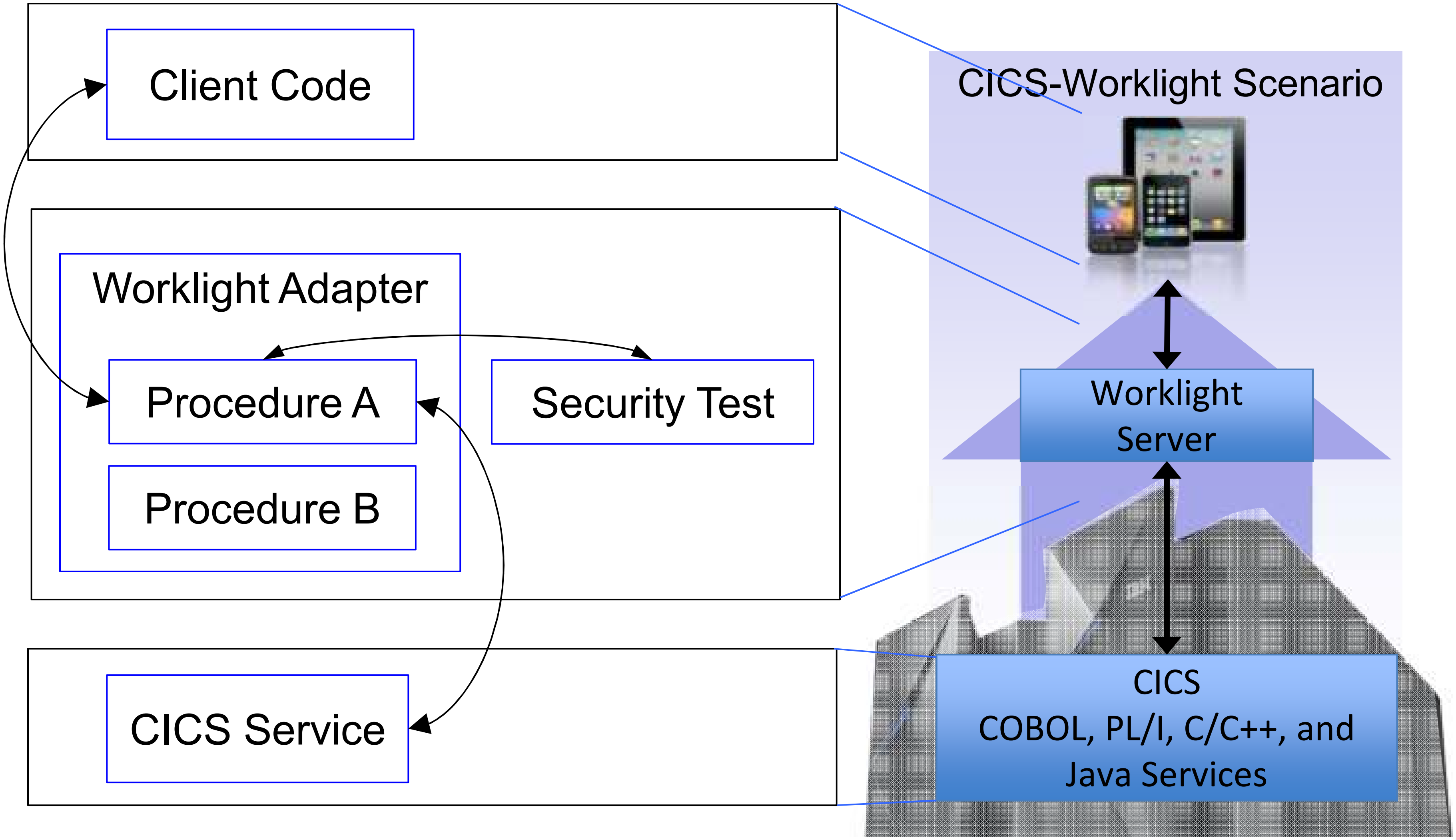
Stored on
zFS

IBM Worklight - Server

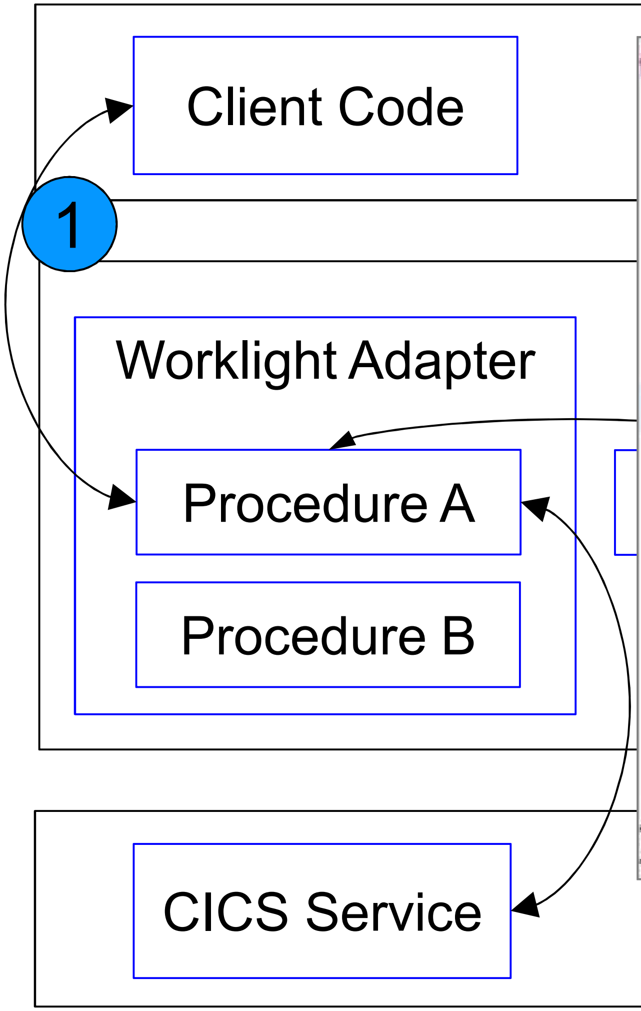
- Supports major mobile platforms
- Production ready private app store
- Define complex mashups from multiple data sources to reduce overall traffic
- Push notifications to client device
- Security-rich architecture enabled by server managed security challenges
- 2 tier architecture zLinux + CICS

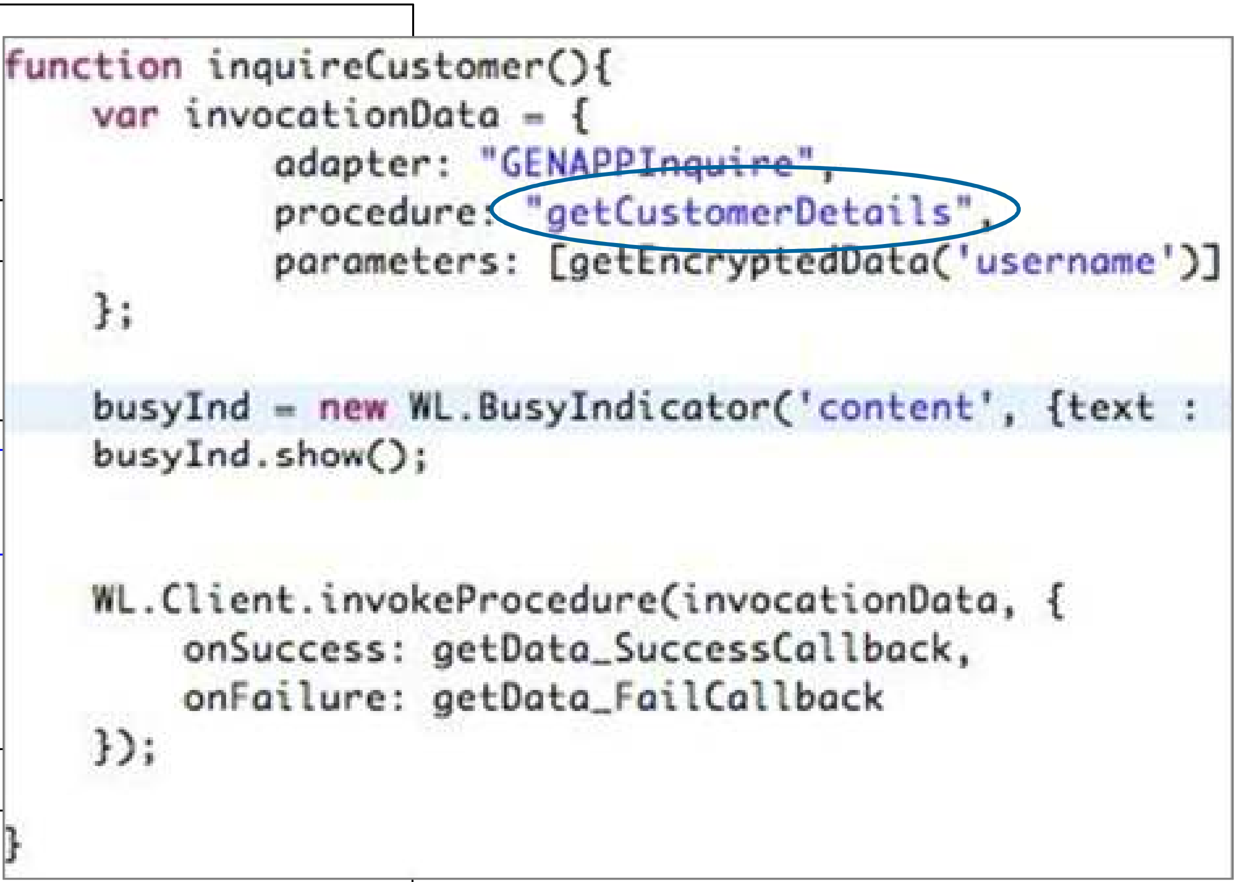


Worklight Components – basic flow



Worklight Components – Client call





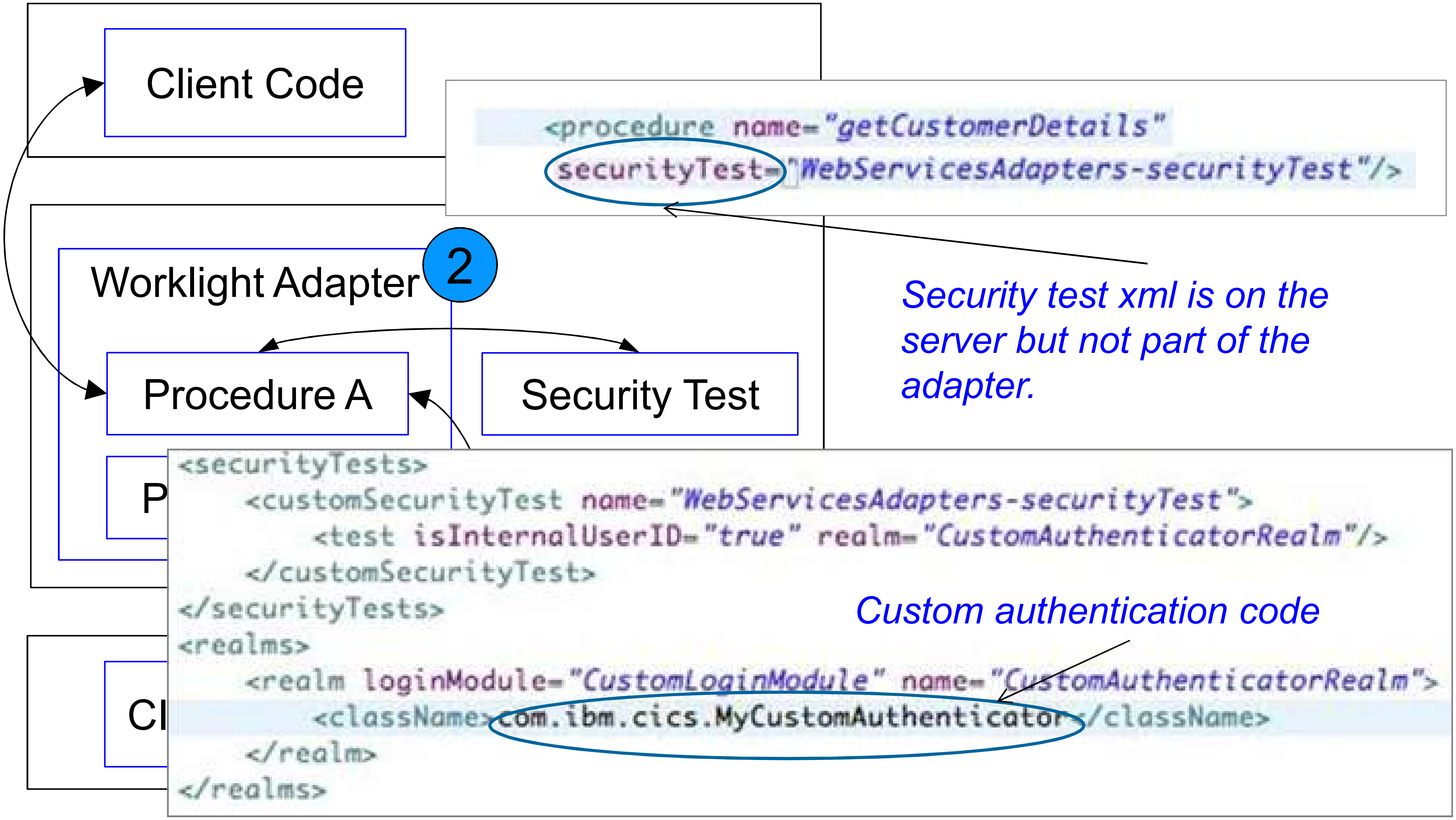
```
function inquireCustomer(){
    var invocationData = {
            adapter: "GENAPPInquire",
            procedure: "getCustomerDetails",
            parameters: [getEncryptedData('username')]
    };

    busyInd = new WL.BusyIndicator('content', {text :
    busyInd.show();


    WL.Client.invokeProcedure(invocationData, {
        onSuccess: getData_SuccessCallback,
        onFailure: getData_FailCallback
    });

}
```

Worklight Components – security check



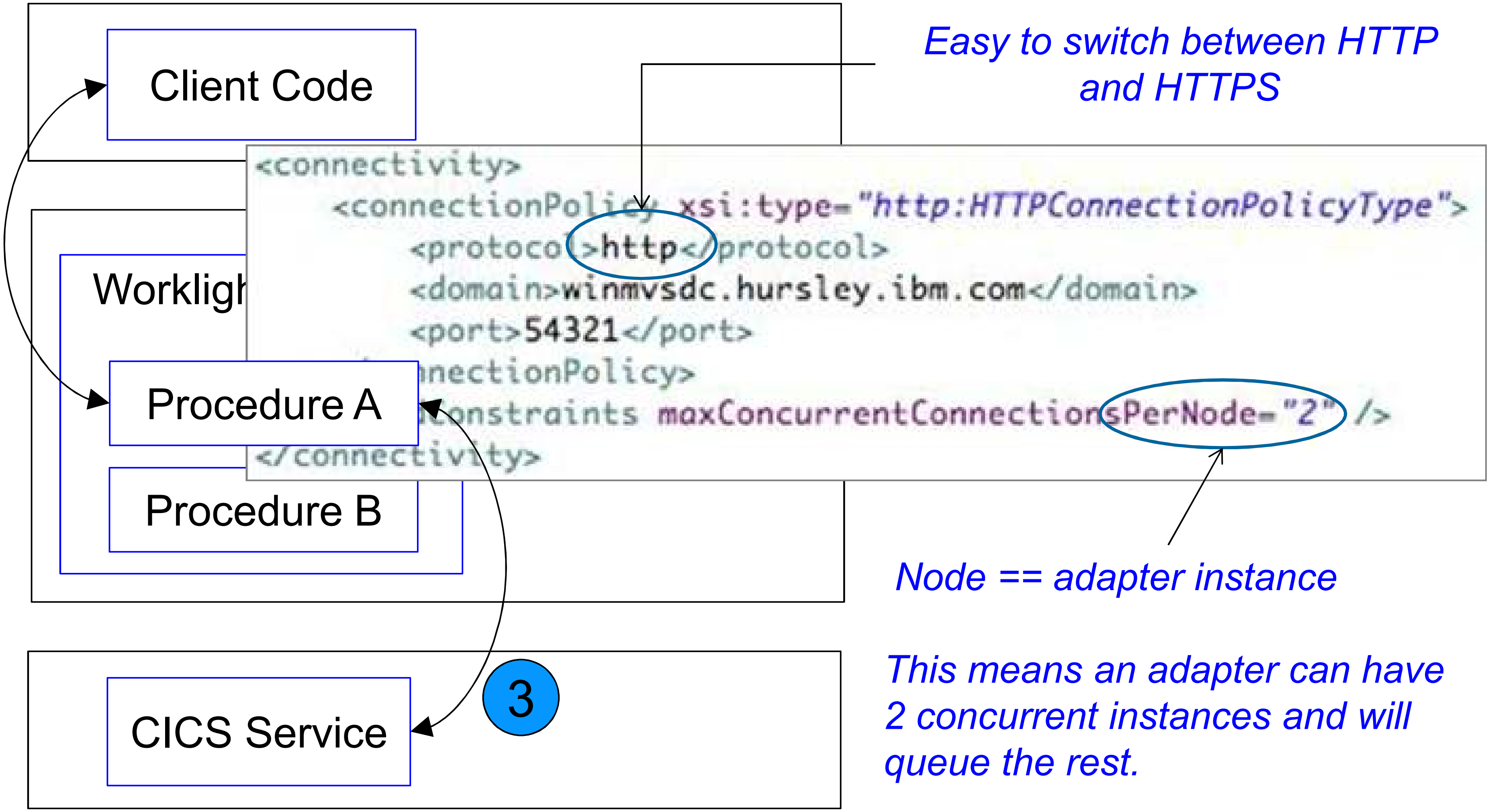
```
<procedure name="getCustomerDetails"
    securityTest="WebServicesAdapters-securityTest"/>
```

Security test xml is on the server but not part of the adapter.

```
<securityTests>
    <customSecurityTest name="WebServicesAdapters-securityTest">
        <test isInternalUserID="true" realm="CustomAuthenticatorRealm"/>
    </customSecurityTest>
</securityTests>
<realms>
    <realm loginModule="CustomLoginModule" name="CustomAuthenticatorRealm">
        <className>com.ibm.cics.MyCustomAuthenticator</className>
    </realm>
</realms>
```

Custom authentication code

Worklight Components – basic flow



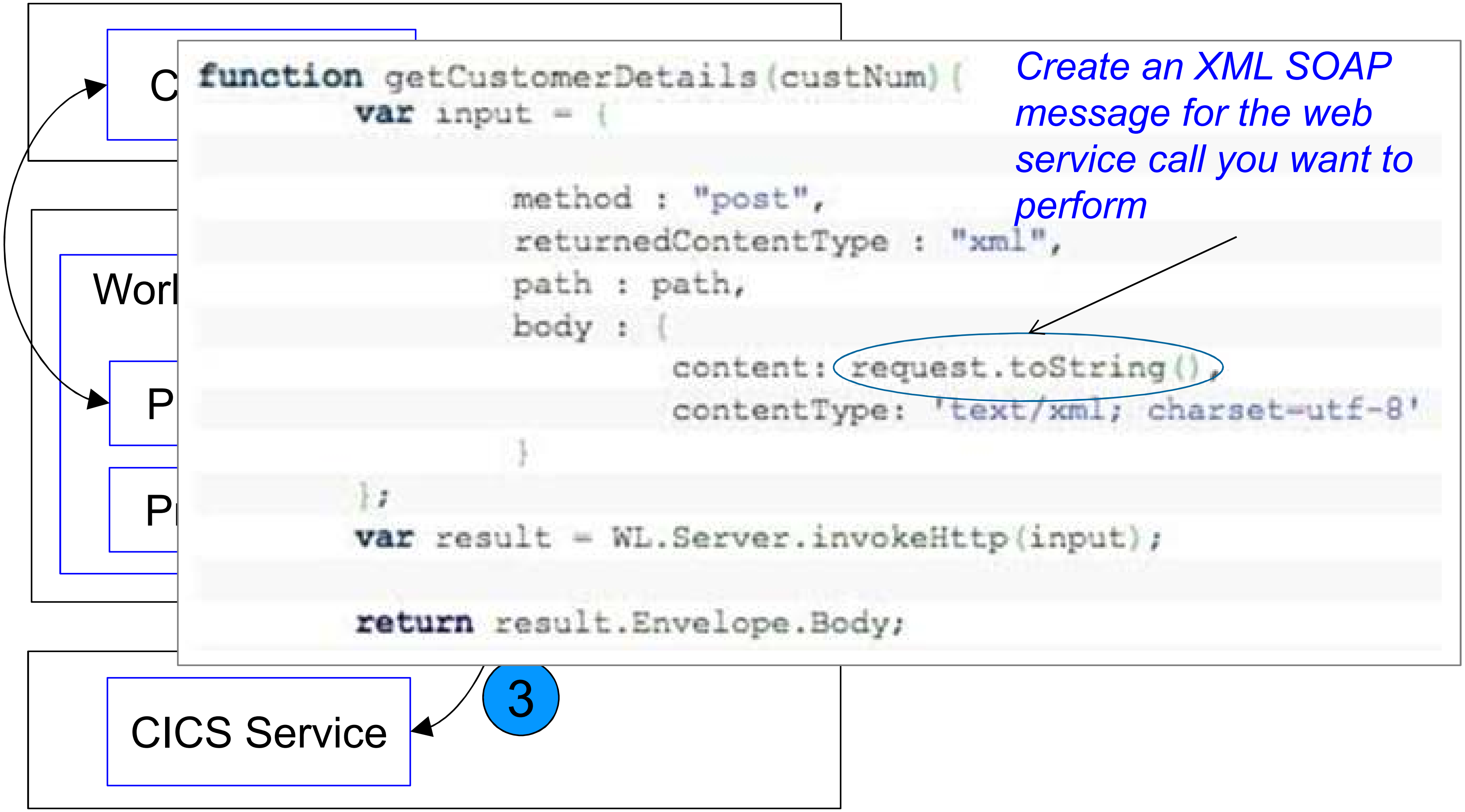
```
<connectivity>
    <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
        <protocol>http</protocol>
        <domain>winmvsdc.hursley.ibm.com</domain>
        <port>54321</port>
    </connectionPolicy>
    <loadConstraints maxConcurrentConnectionsPerNode="2" />
</connectivity>
```

Easy to switch between HTTP and HTTPS

Node == adapter instance

This means an adapter can have 2 concurrent instances and will queue the rest.

Worklight Components – basic flow



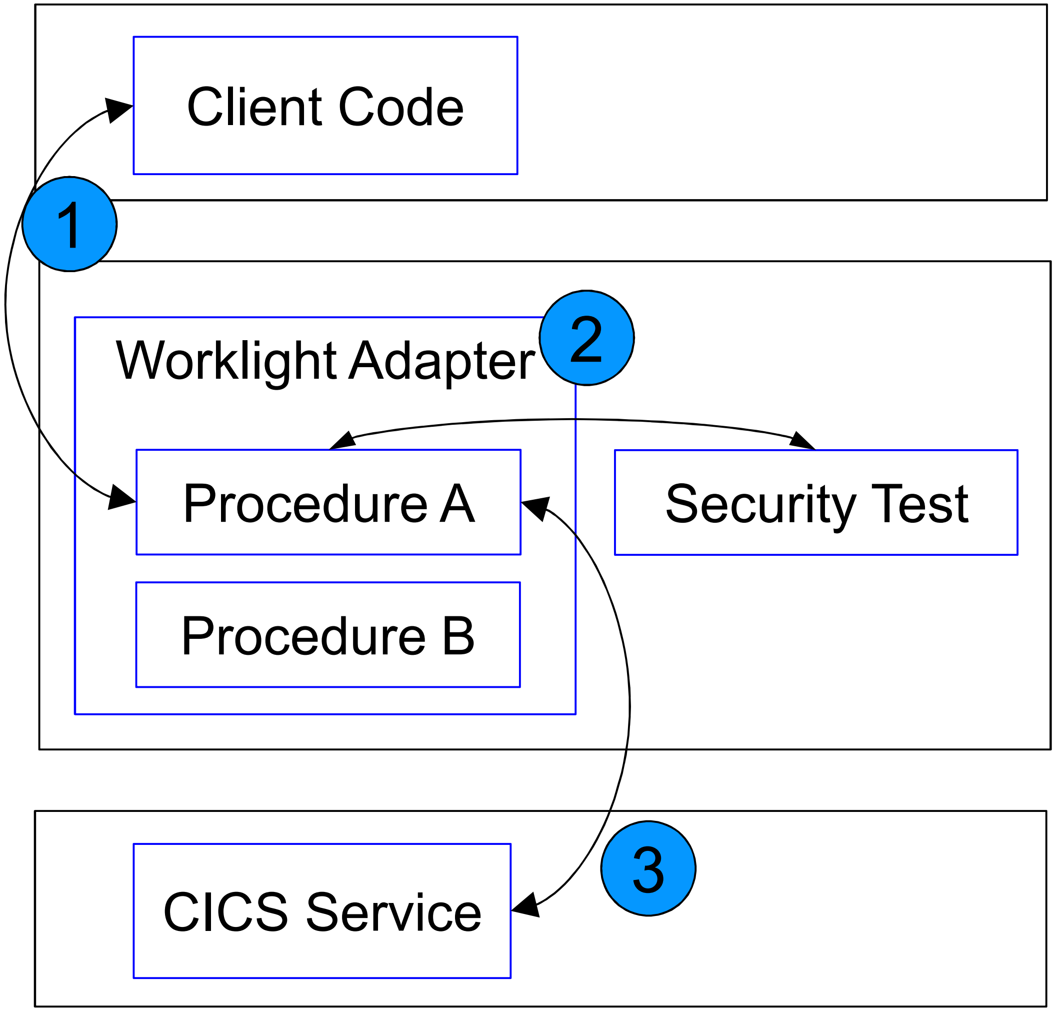
```
function getCustomerDetails(custNum){
        var input = {

                method : "post",
                returnedContentType : "xml",
                path : path,
                body : {
                        content: request.toString(),
                        contentType: 'text/xml; charset=utf-8'
                }
        };
        var result = WL.Server.invokeHttp(input);

        return result.Envelope.Body;
```

Create an XML SOAP message for the web service call you want to perform

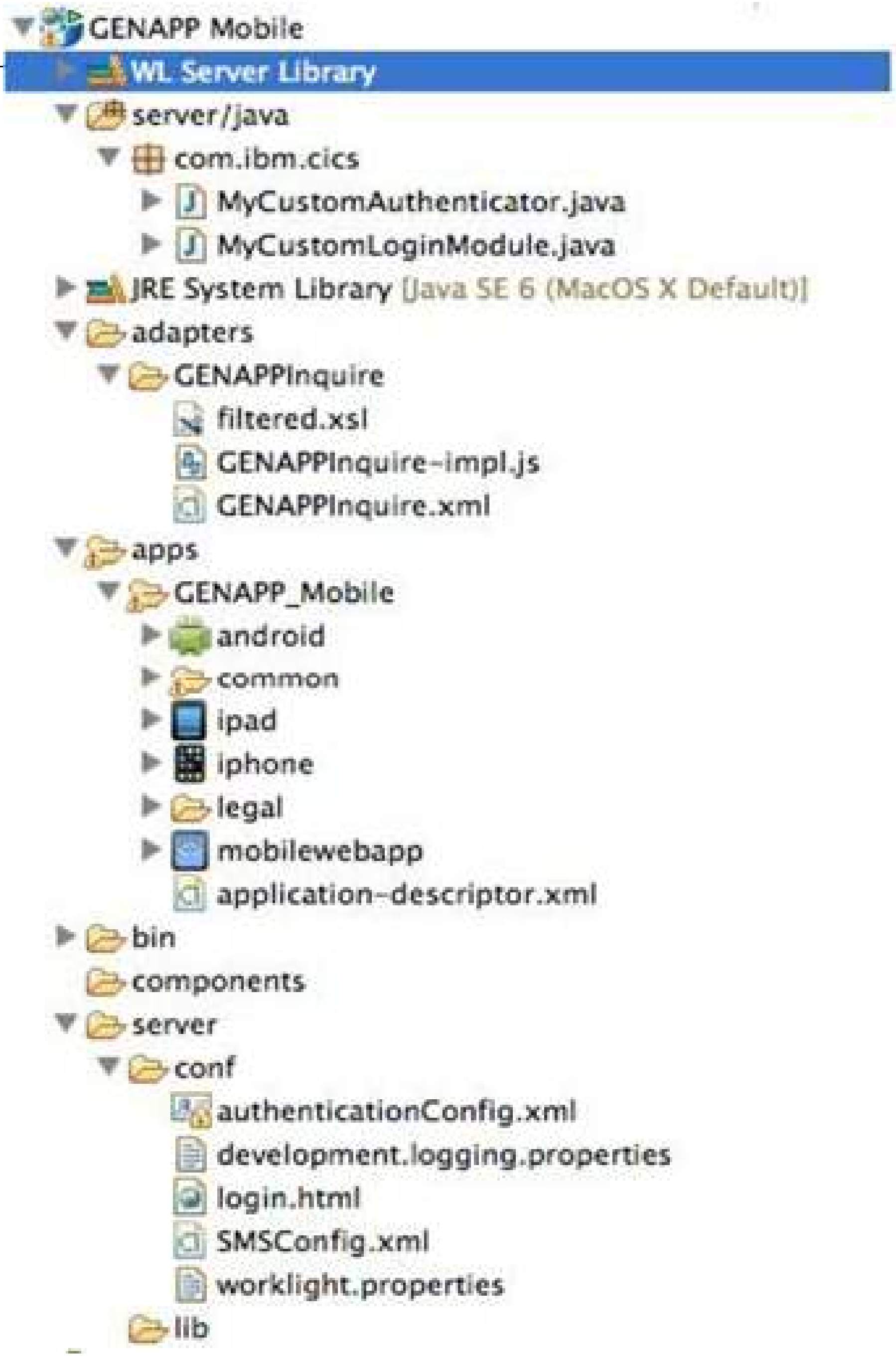
Worklight Components – basic flow



Worklight server provides separation, security, and a variety of mobile services, helping control the impact mobile has on CICS Transaction Server

Worklight Resources

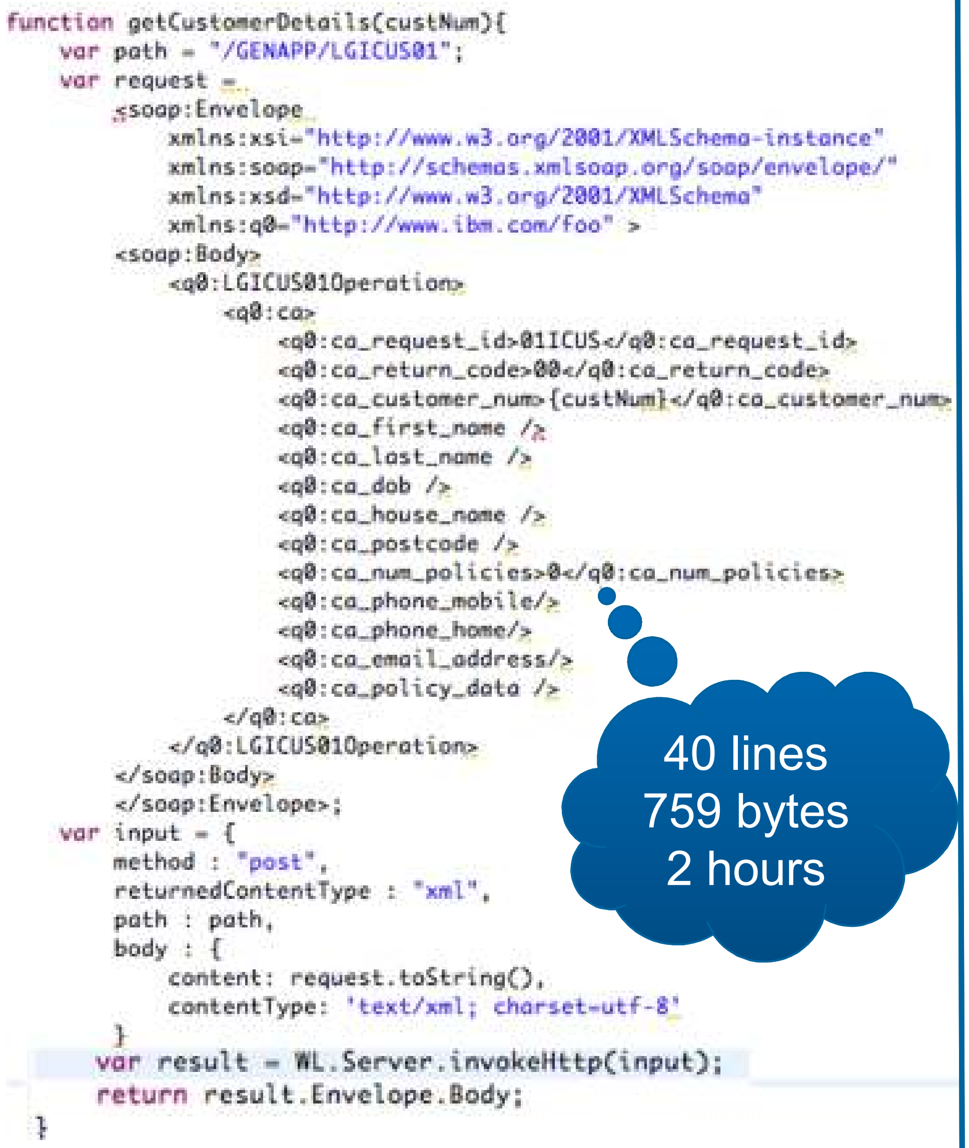
- Worklight development environment helps manage all the resources for your mobile project



```
GENAPP Mobile
  WL Server Library
  server/java
    com.ibm.cics
      MyCustomAuthenticator.java
      MyCustomLoginModule.java
  JRE System Library [Java SE 6 (MacOS X Default)]
  adapters
    GENAPPInquire
      filtered.xsl
      GENAPPInquire-impl.js
      GENAPPInquire.xml
  apps
    GENAPP_Mobile
      android
      common
      ipad
      iphone
      legal
      mobilewebapp
      application-descriptor.xml
  bin
  components
  server
    conf
      authenticationConfig.xml
      development.logging.properties
      login.html
      SMSConfig.xml
      worklight.properties
    lib
```

getCustomerDetails – a closer look

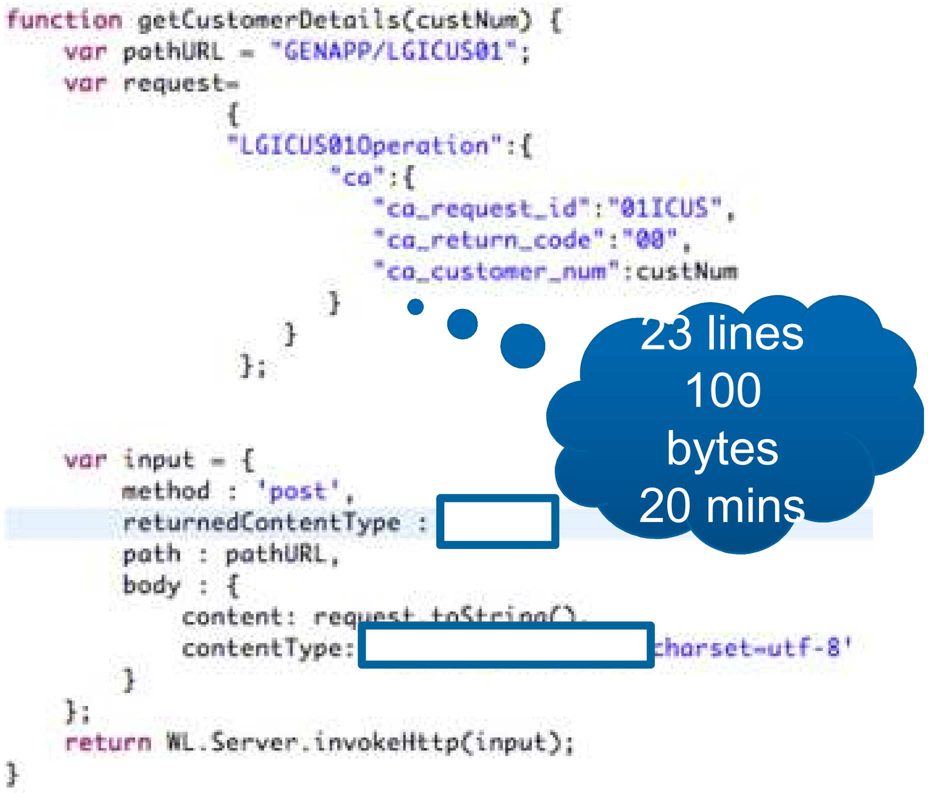
- SOAP adapter – CICS Web Service



```
function getCustomerDetails(custNum){
    var path = "/GENAPP/LGICUS01";
    var request =
        <soap:Envelope
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:q0="http://www.ibm.com/foo" >
        <soap:Body>
            <q0:LGICUS01Operation>
                <q0:ca>
                    <q0:ca_request_id>01ICUS</q0:ca_request_id>
                    <q0:ca_return_code>00</q0:ca_return_code>
                    <q0:ca_customer_num>{custNum}</q0:ca_customer_num>
                    <q0:ca_first_name />
                    <q0:ca_last_name />
                    <q0:ca_dob />
                    <q0:ca_house_name />
                    <q0:ca_postcode />
                    <q0:ca_num_policies>0</q0:ca_num_policies>
                    <q0:ca_phone_mobile/>
                    <q0:ca_phone_home/>
                    <q0:ca_email_address/>
                    <q0:ca_policy_data />
                </q0:ca>
            </q0:LGICUS01Operation>
        </soap:Body>
        </soap:Envelope>;
    var input = {
        method : "post",
        returnedContentType : "xml",
        path : path,
        body : {
            content: request.toString(),
            contentType: 'text/xml; charset=utf-8'
        }
    var result = WL.Server.invokeHttp(input);
    return result.Envelope.Body;
 }
```

40 lines
759 bytes
2 hours

CICS JSON support



```
function getCustomerDetails(custNum) {
    var pathURL = "GENAPP/LGICUS01";
    var request=
            {
            "LGICUS01Operation":{
                "ca":{
                    "ca_request_id":"01ICUS",
                    "ca_return_code":"00",
                    "ca_customer_num":custNum
                }
            }
        };


    var input = {
        method : 'post',
        returnedContentType :
        path : pathURL,
        body : {
            content: request.toString(),
            contentType:                    charset=utf-8'
        }
    };
    return WL.Server.invokeHttp(input);
}
```

23 lines
100
bytes
20 mins

Questions…

We love your Feedback!

Impact2013

Don’t forget to submit your Impact session and speaker feedback!

•Your feedback is very important to us – we use it to improve next year’s conference

•Go to the Impact 2013 SmartSite (http://impactsmartsite.com):

- Use the session ID number to locate the session
- Click the “Take Survey” link
- Submit your feedback

IBM®

Legal Disclaimer

- © IBM Corporation 2013. All Rights Reserved.
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:
  Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:
  All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to http://www.ibm.com/legal/copytrade.shtml for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:
  Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:
  Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:
  Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:
  Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:
  UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:
  Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.