z/OS Connect: Opening up z/OS Assets to the Cloud and Mobile Worlds

Session 17368
August 11th, 2015
Gary Picher

WebSphere Application Server

Session	Title	Time	Room
17363	Debug 101-Using ISA Tools for Apps in WebSphere Application Server z/OS	Monday 11:15	Europe 11
17367	WebSphere Liberty on Windows and z/OS (Among Other Things) Hands-On Lab	Tuesday 10:00	Asia 5
17361	ABCs of WAS	Tuesday 1:45	Oceanic 7
17368	z/OS Connect: Opening up z/OS Assets to the Cloud and Mobile Worlds	Tuesday 3:15	Oceanic 7
17362	Configuring Timeouts for WebSphere Application Server on z/OS	Wednesday 8:30	Oceanic 7
17366	WebSphere Liberty and WebSphere Application Server Classic - What's New?	Wednesday 11:15	Oceanic 7
17364	IBM Installation Manager for z/OS System Programmers: Web-based Installs, Fix Packs, and How iFixes Really Work	Thursday 4:30	Oceanic 7
17365	JSR 352 - The Future of Java Batch and WebSphere Compute Grid	Friday 10:00	Oceanic 6

Agenda

The API Economy – the bigger picture

What are the drivers and how will z/OS integrate?

z/OS Connect

Overview / Problem statement

Perspectives –

The ‘front end’ – users of z/OS Connect ‘systems of engagement’

What’s happening inside z/OS Connect?

Service providers

Interceptors

Data transformers

The ‘back-end’ – interacting with business assets ‘systems of record’

What's New

The API Economy

A view from on high

APIs - the building blocks for apps

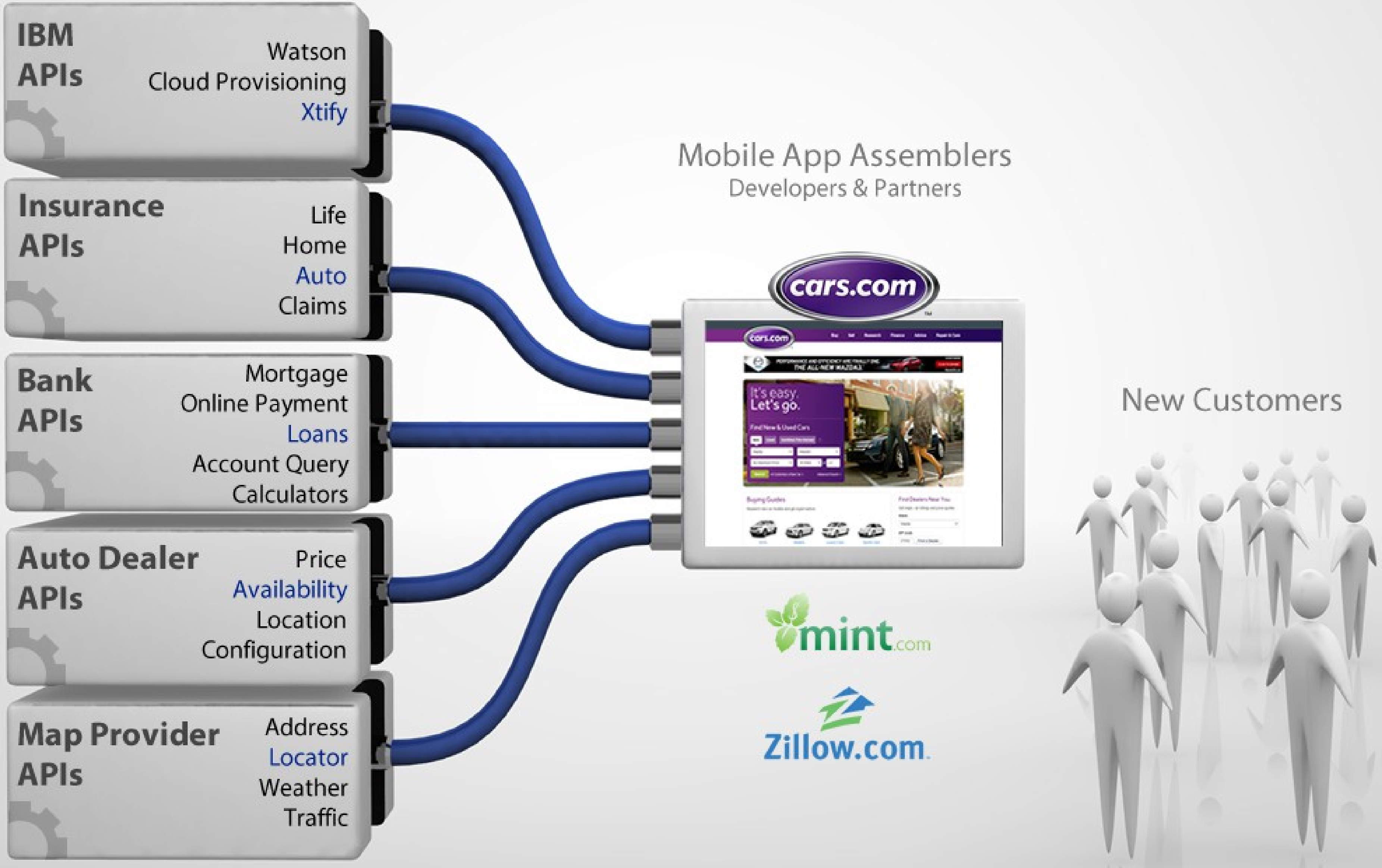
Application Programming Interface

The **"API economy"** has changed how developers think about building apps, and how organizations deploy software in the cloud

Bank Externalized Services
Bank APIs
Mortgage
Online Payment
Loans
Account Query
Calculators
Developers

API Economy

Mobile App Assembly



IBM APIs

Watson
Cloud Provisioning
Xtify

Insurance APIs

Life
Home
Auto
Claims

Bank APIs

Mortgage
Online Payment
Loans
Account Query
Calculators

Auto Dealer APIs

Price
Availability
Location
Configuration

Map Provider APIs

Address
Locator
Weather
Traffic

Mobile App Assemblers

Developers & Partners

cars.com

mint.com

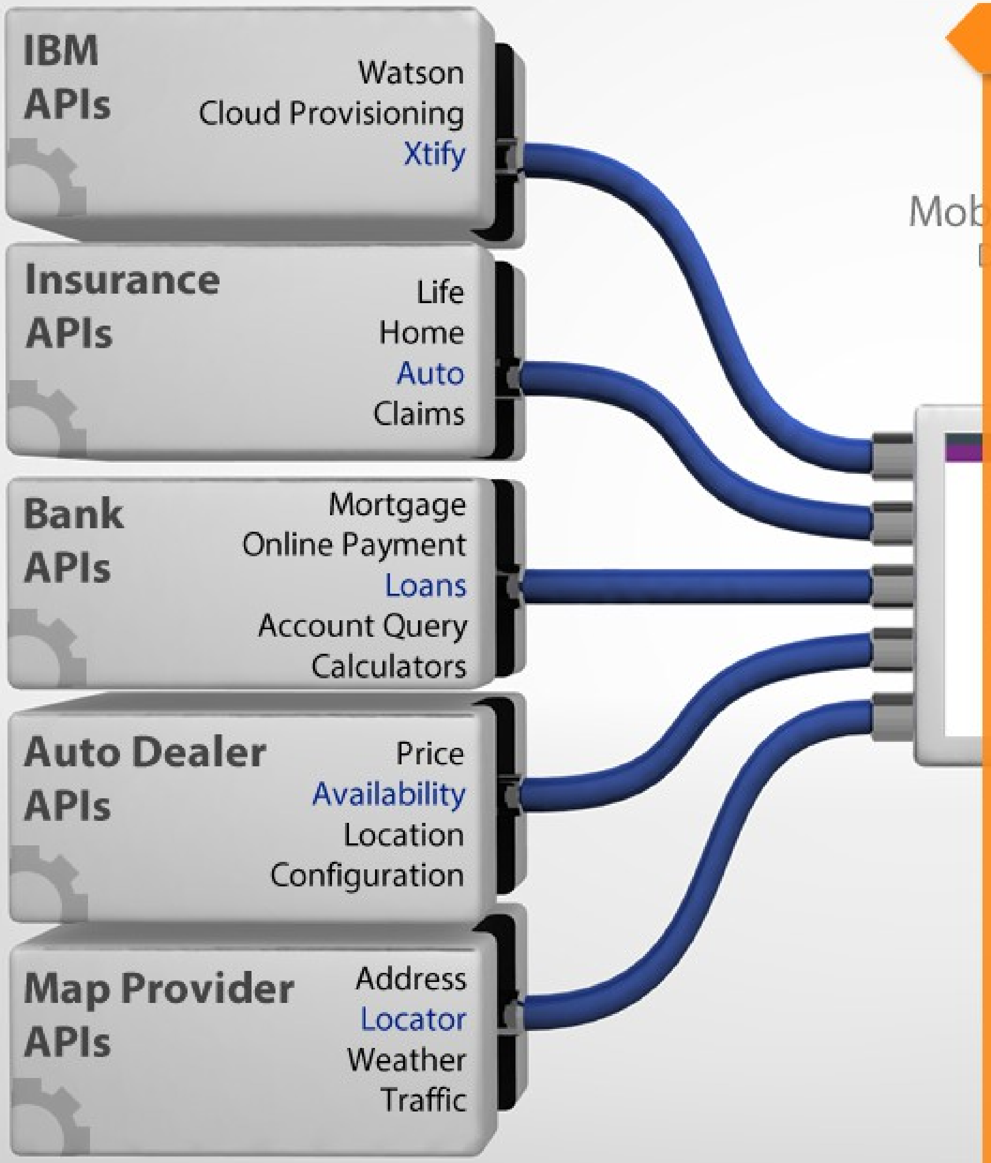
Zillow.com

New Customers

Providers **Consumers**

API Economy

Mobile App Assembly



Consumers

Benefits for Application Assemblers

- Quick time to Market by using existing assets and services to decrease costs and speed up product and service development
- Driving innovation by capitalizing on the merging of capabilities provided by different APIs
- Attracting clients to your products and services by joining other (established) API ecosystems
- Potentially having freedom of choice in terms of available APIs (avoiding vendor lock-in)
- Focus on core competence and value add rather than reinventing common capabilities
- Create new business opportunity from APIs
- Innovate around business models and reach new customers from the provider's ecosystem

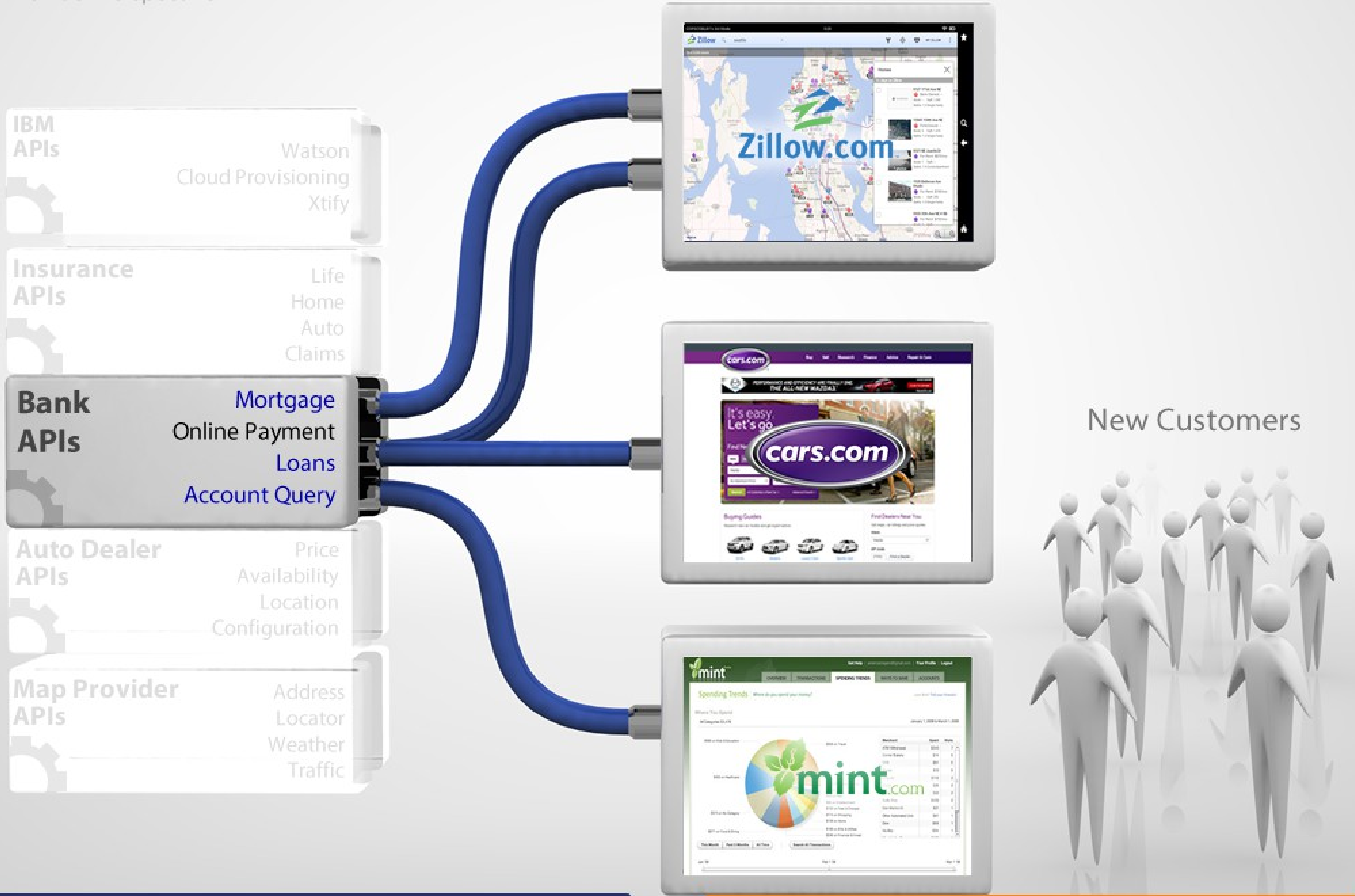
Providers

API Economy

Provider Perspective

Mobile App Assemblers

Developers & Partners



IBM APIs

- Watson
- Cloud Provisioning
- Xtify

Insurance APIs

- Life
- Home
- Auto
- Claims

Bank APIs

- Mortgage
- Online Payment
- Loans
- Account Query

Auto Dealer APIs

- Price
- Availability
- Location
- Configuration

Map Provider APIs

- Address
- Locator
- Weather
- Traffic

New Customers

Providers

Consumers

API Economy

Provider Perspective

Providers

Benefits

- Expanding into new customer bases and niches that you normally wouldn't be able to reach on your own
- Focusing on your core value
- Expanding your brand and brand loyalty
- Easier possibilities in establishing new partnerships and capitalizing on them
- Benefitting from open innovations of crowdsourcing and expert sourcing, which are implicitly unlocked through the opening of APIs
- Realize profit from new business models
- Decreased development costs and time through software (e.g., mobile apps) produced by third parties
- Keeping up with application demands (e.g. new apps, new features)
- Wider and quicker coverage of different platforms and devices

Mobile App Assemblers

Developers & Partners

New Customers

Consumers

Overview

Setting Context Before Going into Details

Problem Statement(s)

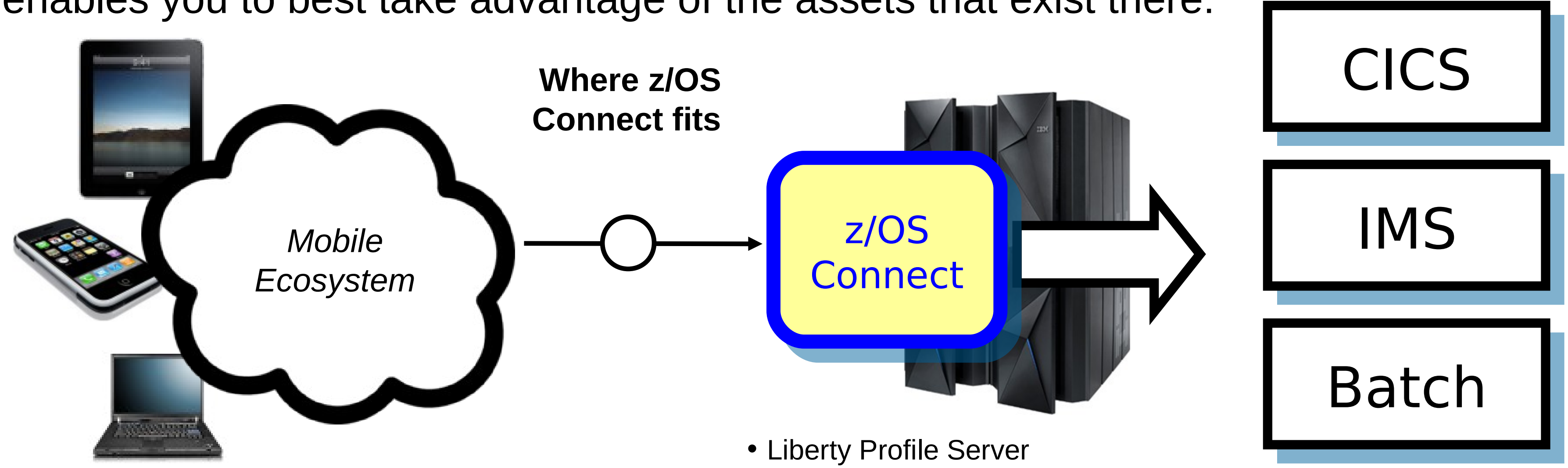
1) Customers using z/OS today are increasingly expressing **concerns about their ability to track and handle large spikes of new requests** originating from any number of almost instantly available clients and systems that have a need for the business assets available there. Mobile is one of the larger reasons for this.

2) Cloud Infrastructure providers (like IBM's **Bluemix)** and mobile services registries (like IBM's **API Management**) benefit from a uniform way to interact with z-based middleware for discovery, provisioning, data transformation, and service invocation.

3) Customers have expressed an interest in **a common way to interact with all z/OS business and infrastructure assets** using **REST and JSON** technology that is closely integrated with z/OS.

z/OS Connect – What is it?

It's about getting REST and JSON into your mainframe environment in a way that enables you to best take advantage of the assets that exist there:



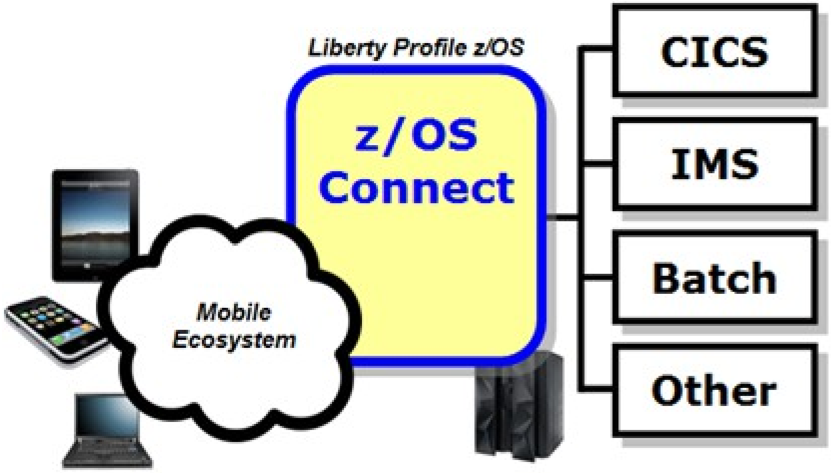
- Liberty Profile Server
- Function IBM wrote to run in Liberty Profile
- No charge function provided with license entitlement to WAS z/OS, CICS or IMS

REST – Representational State Transfer … the use of HTTP URLs that map to a 'service', such as 'query account' or 'update data'

JSON – JavaScript Object Notation … a standard of representing data as a set of name/value pairs. This is passed back and forth along with REST request/responses

Why z/OS Connect?

This represents another component to configure and maintain in your environment. So what value does it bring?



You *could* enable Mobile access without z/OS Connect

z/OS Connect simplifies and makes the environment more consistent and manageable

- Provides a common and consistent entry point for mobile access to one or many backend systems
- Java, so runs on specialty engines
- Shields backend systems from requiring awareness of RESTful URIs and JSON data formatting
- Provides point for authorization of user to invoke backend service
- Provides point for capturing usage information using SMF
- Simplifies front-end functions by allowing them to pass RESTful and JSON rather than be aware of or involved in data transformation

Different *Delivery* Approaches

This is delivered with WAS z/OS, CICS and IMS … objective is to provide different approach paths depending on what you have:

WAS z/OS

Delivered as function that runs inside Liberty Profile z/OS. Initially will use WOLA (WebSphere Optimized Local Adapters) to access backend.

CICS

Delivered as part of Liberty Profile that runs inside of CICS region. Will use JCICS interface to access CICS functions

IMS

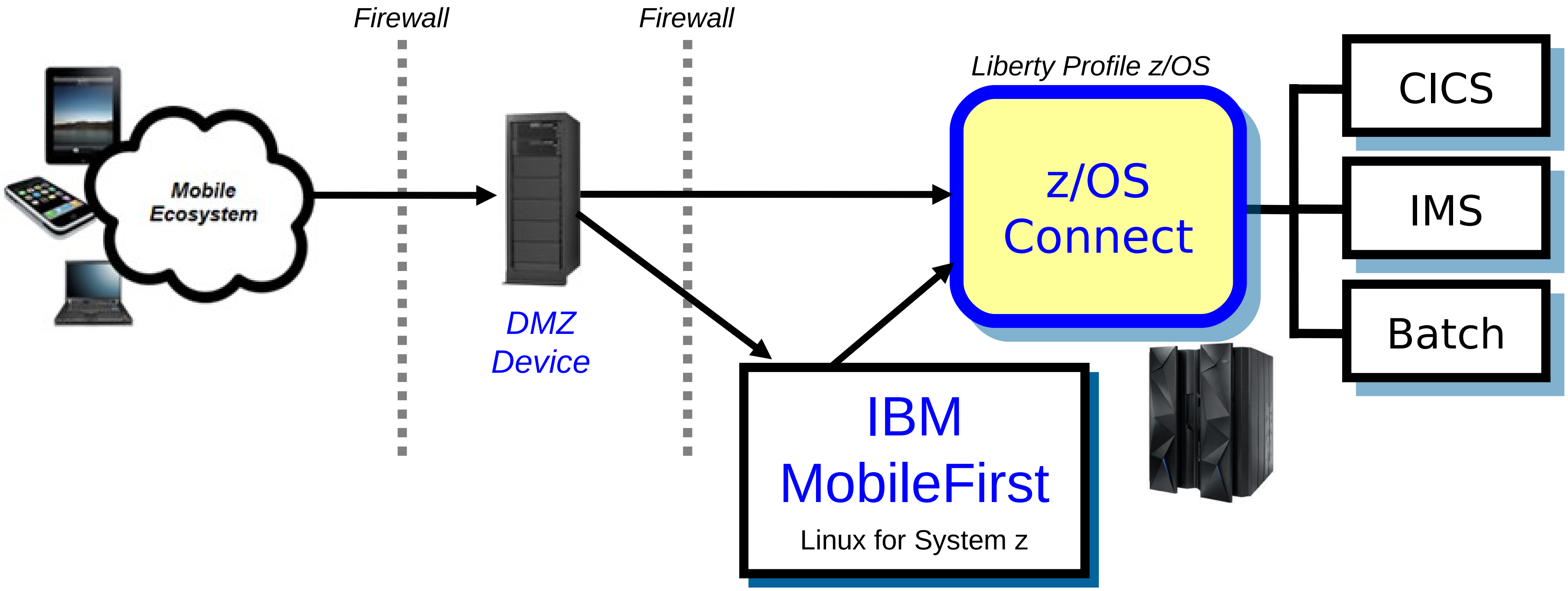
Initially this ends up looking just like the WAS z/OS approach: that is, Liberty Profile z/OS with z/OS Connect inside. Difference is this: IMS z/OS Connect uses JCA* to talk to IMS Connect to get access into IMS.

These different delivery mechanisms tend to obscure the main story of what it is and how it works, so for now let's stipulate IBM offers several ways to get this and now focus on some details

* A supplied IMS JCA resource adapter, as opposed to the local adapter support

Context Within Overall Mobile Architecture

The message here is that z/OS Connect is a *piece* of the Mobile architecture, but in most cases will not be the only component:



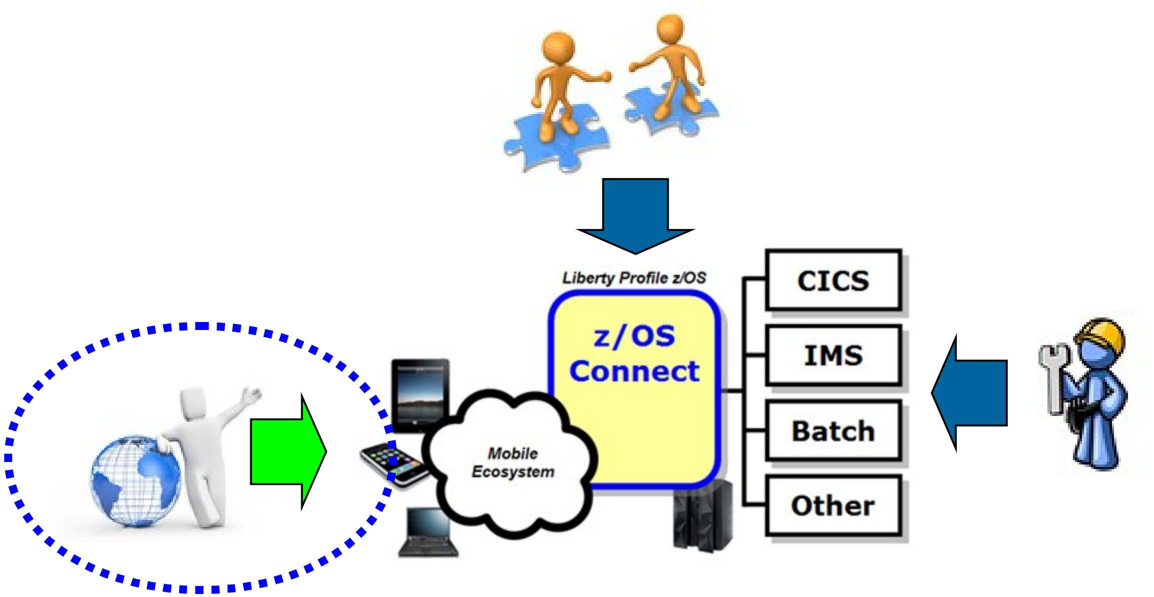
Users of z/OS Connect would access through normal corporate firewall infrastructure

IBM MobileFirst to provide application management, security and operational governance for mobile applications

z/OS Connect would be behind the secure firewall, and on LPARs along with backend systems

Front-End Perspective

Looking at z/OS Connect from a perspective of users of the function



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

RESTful Services

Stands for Representational State Transfer … this is a protocol built on HTTP, using HTTP verbs*, where the URI indicates the service requested:

URI = Uniform resource identifier

```
https://mysite.com/CustomerApp/getCustomer?cn=1234
```

There's no magic to this … if the URI is understood by the receiving server, then the implied action is taken. What that action is depends on how the server is configured.

Knowing what URIs the server supports is important, which is why z/OS Connect has a discovery function that can be used to query for the configured services and details on those services.

RESTful services are growing in popularity because it's easier to implement than other web service protocols such as SOAP, which involves XML and WSDL and parsing …

* For example, GET, PUT, POST, DELETE

JSON – JavaScript Object Notation

It is a way of passing data back and forth as a series of name/value pairs.

URI = Uniform resource identifier

```
https://mysite.com/CustomerApp/update?cn=1234
```

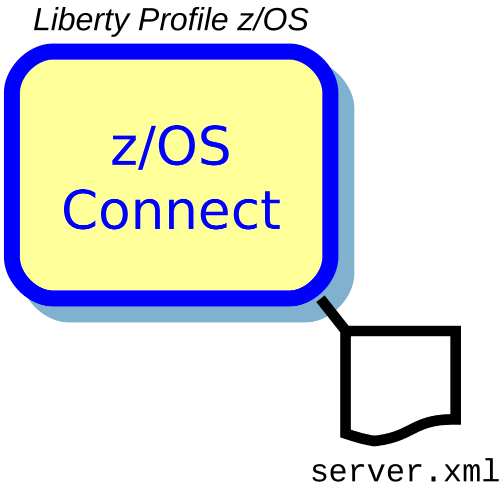
```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": “1234 Main Street",
        "city": “Anytown",
        "state": "NY",
        "postalCode": "10021-1234"
    },
}
```

The data being passed in is appended to the URL and passed in to the server

JSON can be passed back to the client as well.

A Peek Inside z/OS Connect Configuration

We're about to show you the discovery function, which returns a list of configured services. Let's first look inside z/OS Connect …



```
<feature>zosConnect-1.0</feature>

<zosConnectService serviceName="serv1" ... />
<zosConnectService serviceName="serv2" ... />
<zosConnectService serviceName="serv3" ... />
<zosConnectService serviceName="serv4" ... />
```

- For z/OS Connect to understand what URIs to handle and how to handle them, you need to configure that information into `server.xml`

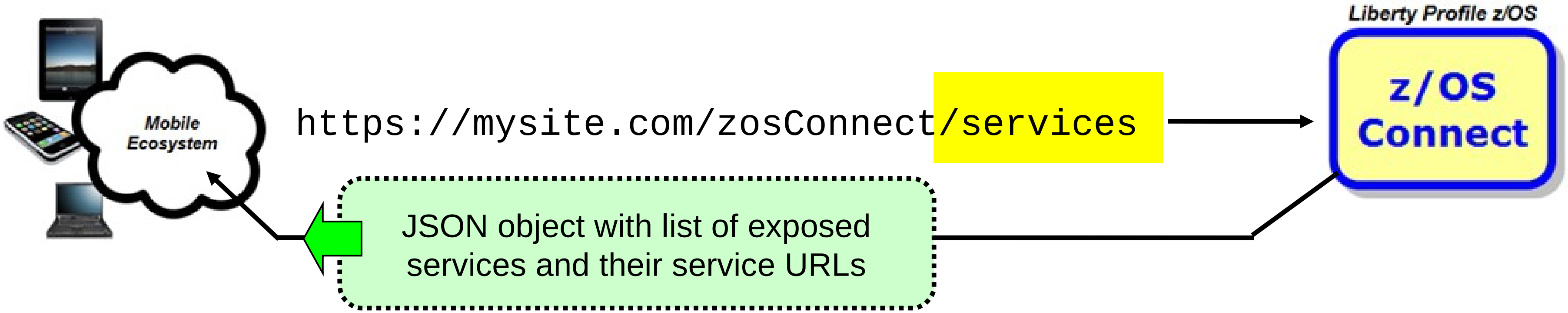
  More information on server.xml configuration coming up

- The sample above is a simplified representation of the configuration for multiple services
- z/OS Connect understands its environment based on this configuration data … *and it can provide information back using a discovery function*

z/OS Connect Discovery Function

A discovery function is provided to allow developers to query for a list of configured services, and drill down for details on a given configured service:

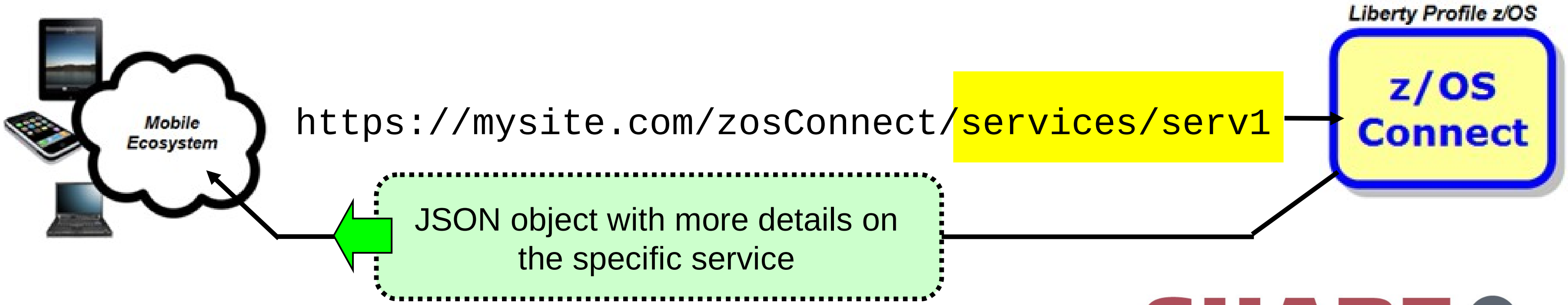
Query for configured services



`https://mysite.com/zosConnect/services`

JSON object with list of exposed services and their service URLs

Query for details on a given configured service

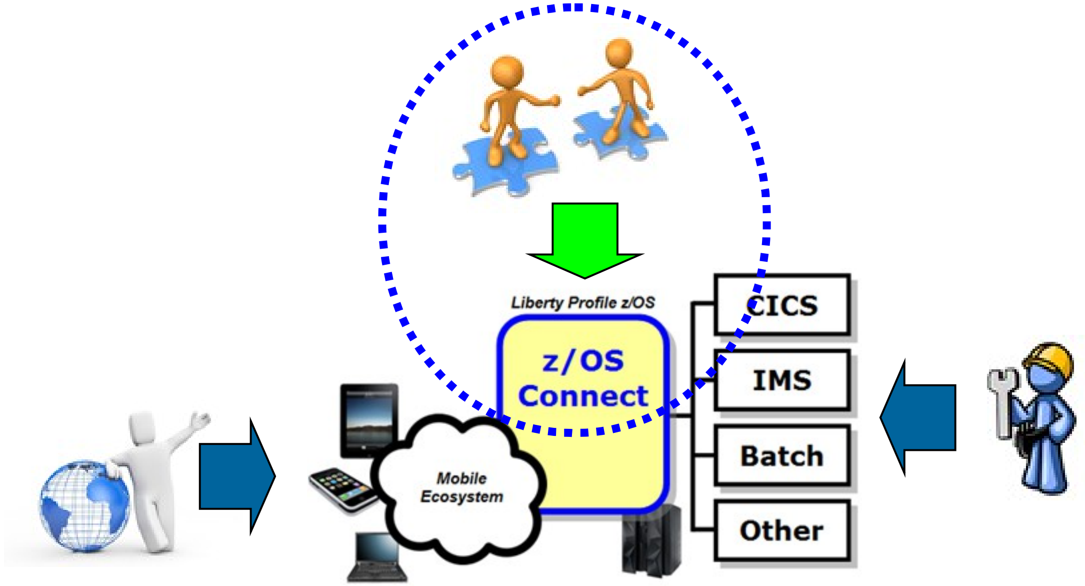


`https://mysite.com/zosConnect/services/serv1`

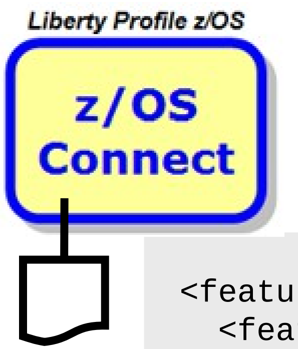
JSON object with more details on the specific service

Inside Perspective

Looking at how z/OS Connect works inside the function itself



Overview of server.xml updates



z/OS Connect behavior is defined by updates to the server.xml of the Liberty Profile in which z/OS Connect operates:

https://myhost.com/zosConnect/services/HelloWorld

```
<featureManager>
  <feature>zosConnect-1.0</feature>
  <feature>zosLocalAdapters-1.0</feature>
</featureManager>

<zosConnectService id="serv1"
  serviceName="HelloWorld"
  serviceRef="wolaHelloService"
</zosConnectService>

<authData id="cauth1" user="user1" password="{xor}LDo8Ki02KyY="/>
<connectionFactory id="wolaCF" jndiName="eis/ola" containerAuthDataRef="cauth1" >

  <properties.ola/>
</connectionFactory>

<localAdaptersConnectService id="wolaHelloService"
  wolaRegisterName="CICSREGA"
  wolaServiceName="PROGRAM1"
  connectionFactoryRef=“wolaCF"
</localAdaptersConnectService>
```

z/OS Connect is defined as a feature in the Feature Manager section, along with WOLA used to get to the backend

This defines a service – HelloWorld – that will be exposed to Mobile users. It maps to a service provider that defines how the backend CICS will be accessed.

This defines the service provider – WOLA in this case – that provides the connectivity to the backend CICS region

Multiple Configured Services

You provide definitions for each service you wish to expose using z/OS Connect. For example, this shows Update, Create and Delete:

```
https://myhost.com/zosConnect/services/ContactCreate

https://myhost.com/zosConnect/services/ContactUpdate

https://myhost.com/zosConnect/services/ContactDelete
```

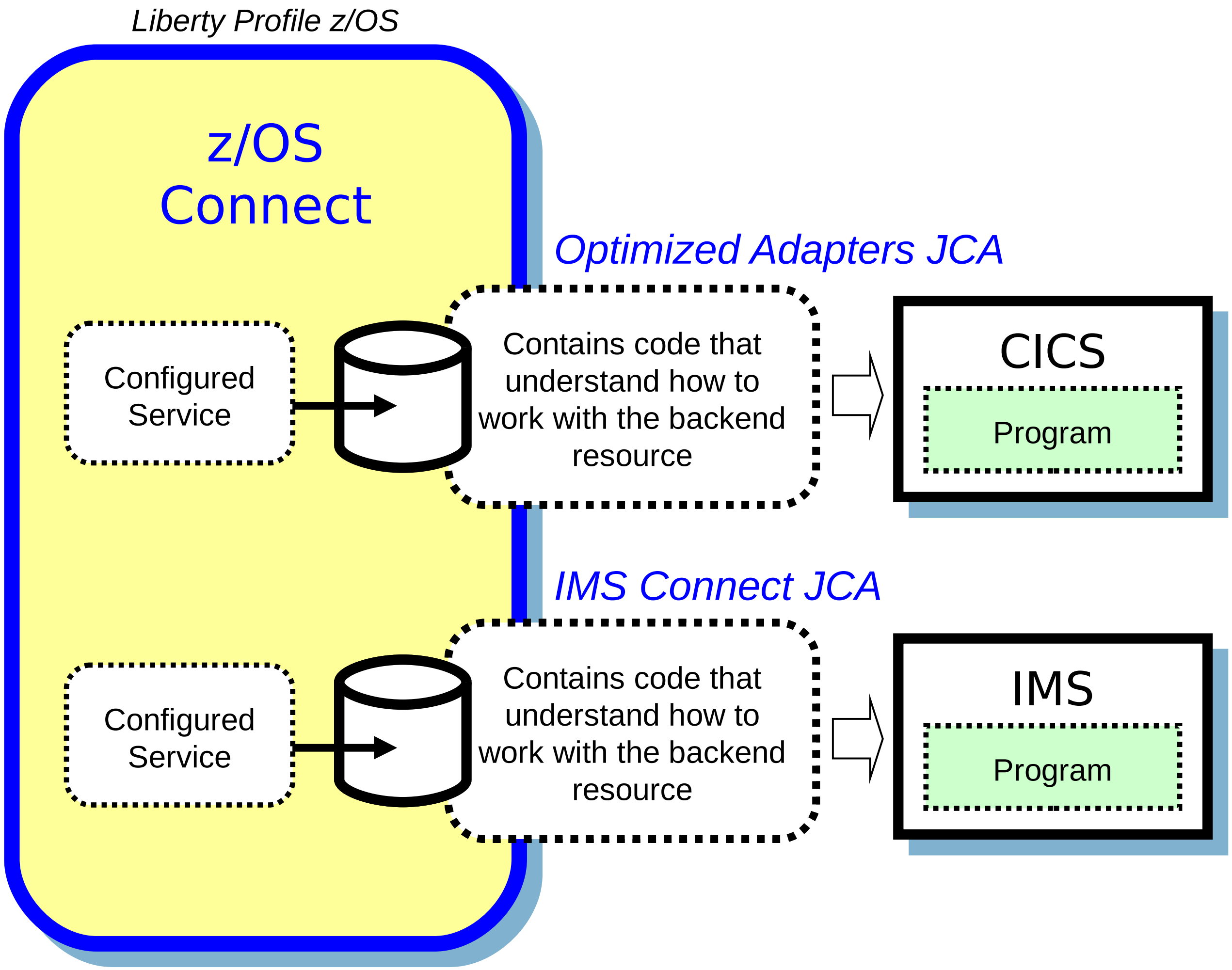
```
<zosConnectService id="zcs3" serviceName="ContactCreate"
     serviceDescription=“Create"
     serviceRef="wolaKIXPHONE"
     dataXformRef="xformJSON2Byte">
</zosConnectService>

<zosConnectService id="zcs4" serviceName="ContactUpdate"
     serviceDescription=“Update"
     serviceRef="wolaKIXPHONE"
     dataXformRef="xformJSON2Byte">
</zosConnectService>

<zosConnectService id="zcs5" serviceName="ContactDelete"
     serviceDescription="Delete"
     serviceRef="wolaKIXPHONE"
     dataXformRef="xformJSON2Byte">
</zosConnectService>
```

Pointer to section in XML that defines the “Service Provider,” which provides connectivity to the backend system

Pointer to section in XML that provides information on how to do data transformation

z/OS Connect Service Providers

A ‘service provider’ is what provides connectivity to a specific backend resource. You may have one or more service providers configured:


Liberty Profile z/OS
z/OS Connect
Optimized Adapters JCA
Configured Service
Contains code that understand how to work with the backend resource
CICS
Program
IMS Connect JCA
Configured Service
Contains code that understand how to work with the backend resource
IMS
Program

To connect to a backend CICS region, for example, you need to specify which CICS region and what mechanism is to be used to connect

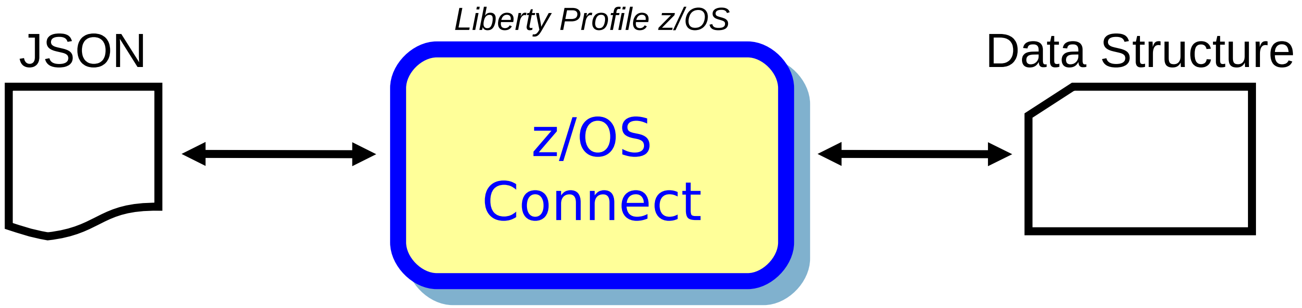
A ‘service provider’ definition provides that information to z/OS Connect

Configured services are tied to service provider definitions to complete the circuit

Multiple service providers are supported, as illustrated here

Data Transformation

z/OS Connect accepts JSON data, but then needs to convert that to the data format required by the backend program. Data conversion provides that:



```
<zosConnectDataXform id="xformJSON2Byte"
    bindFileLoc="/u/user1/bindfiles"
    bindFileSuffix=".bnd"
    requestSchemaLoc="/u/user1/schema"
    responseSchemaLoc="/u/user1/schema"
    requestSchemaSuffix=".json"
    responseSchemaSuffix=".json">
</zosConnectDataXform>
```

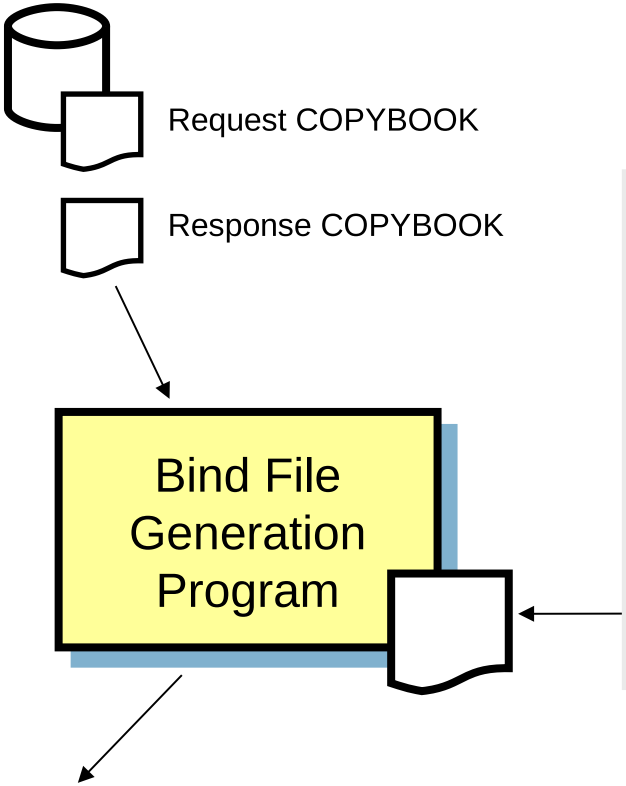
ID of transform section that was referenced in the service definition

XML that tells z/OS Connect where the bind file is located and the file suffix that is used

XML that tells z/OS Connect where the JSON schema information is located

Bind Files

Bind files are generated with a supplied utility. Bind files provide z/OS Connect with knowledge of how JSON maps to the target data structure



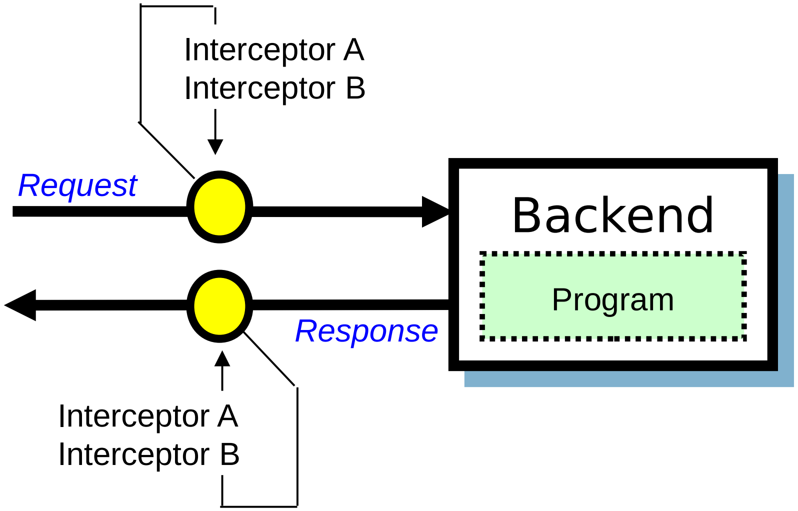
```
//JOB ...
//INPUT.SYSUT1 DD *
PDSLIB=ZCONNECT.CNTL
REQMEM=REQDATA
RESPMEM=RESPDATA
STRUCTURE=(requestData,responseData)
JSON-SCHEMA-REQUEST=/u/user1/schema/GETCUSTC_request.json
JSON-SCHEMA-RESPONSE=/u/user1/schema/GETCUSTC_response.json
PGMINT=COMMAREA
PGMNAME=GETCUSTC
WSBIND=/u/user1/bindfiles/GETCUSTC.bnd
/*
```

- Bind File
- JSON Schema

The bind file provides the JSON-to-COPYBOOK mapping; the JSON schema files are used for the `getRequestSchema` and `getResponseSchema` methods

Overview of Interceptors

The interceptor framework provides a way to call code to do pre-invoke work and then again to do post-invoke work:



In `server.xml` you can:

Define ‘global interceptors,’ which apply to all configured services

Define interceptors specific to a given configured service

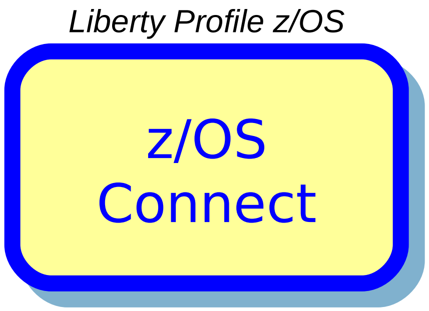
Have services ‘opt out’ of global interceptors

z/OS Connect comes with a security interceptor (for authorization) and an audit interceptor (for SMF recording)

It is also possible to write your own interceptor and have it called as part of request/response processing

Audit (SMF) Interceptor

The audit interceptor writes SMF 120.11 records with the following information captured:



- System Name
- Sysplex Name
- Jobname
- Job Prefix
- Address Space Stoken

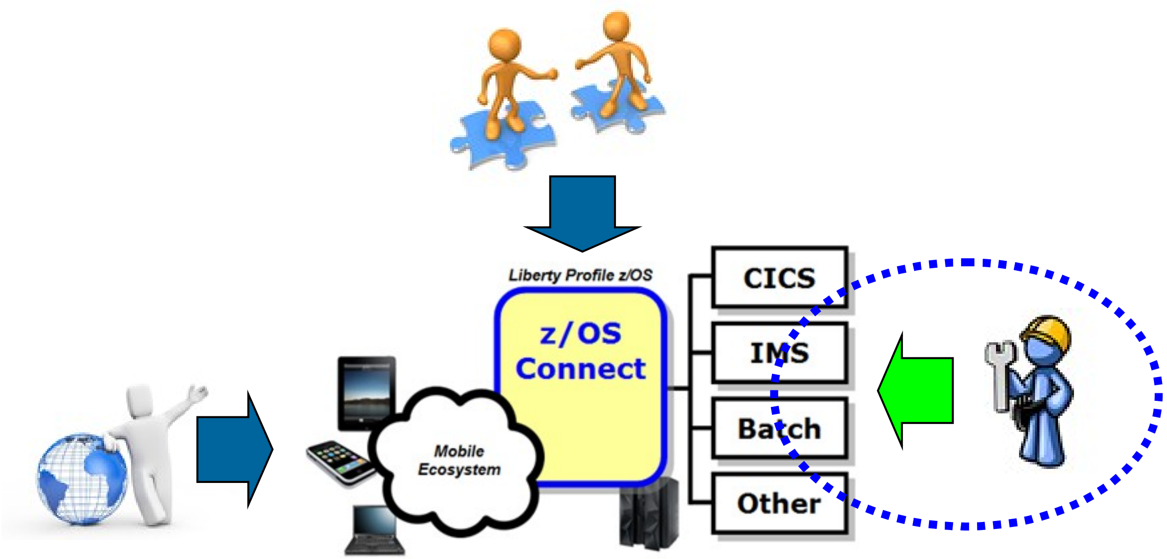
Server Identification Section

- Arrival Time
- Completion Time
- Target URI
- Input JSON Length
- Response JSON Length
- Method Name
- Service Name
- Userid

z/OS Connect User Data Section

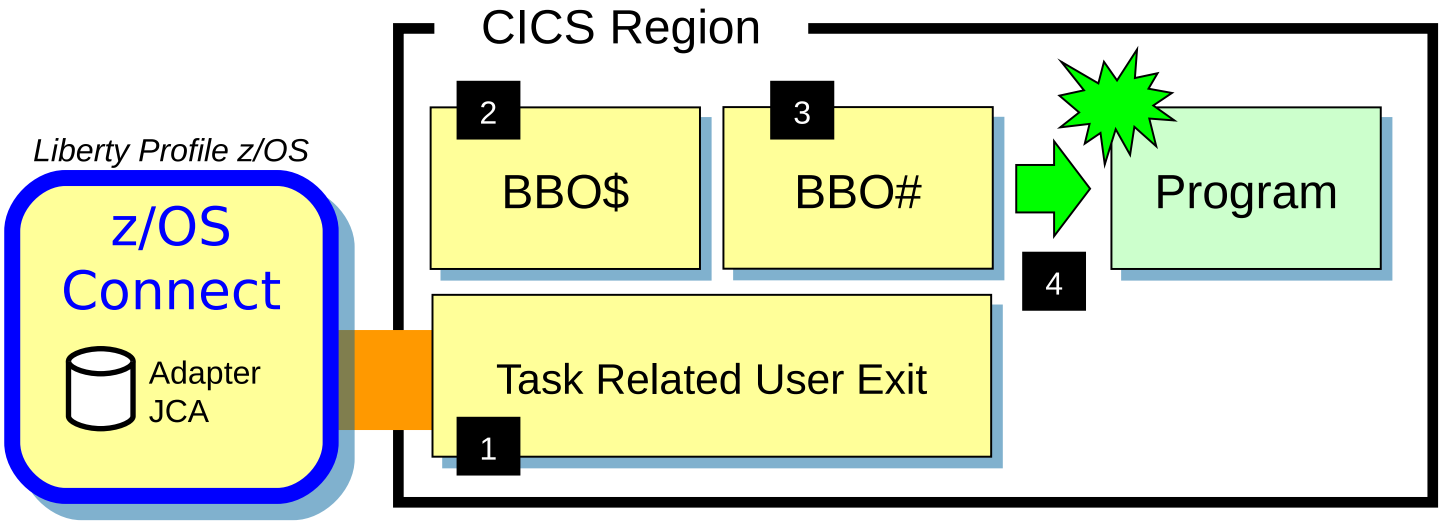
Backend Perspective

Looking at z/OS Connect from perspective of CICS, Batch and IMS



Optimized Adapters Supports in CICS

To support WOLA in CICS the following code must be installed. JCL samples provided to do the installation:



1 The WOLA TRUE provides the low-level connectivity for cross-memory communications using Optimized Adapters.

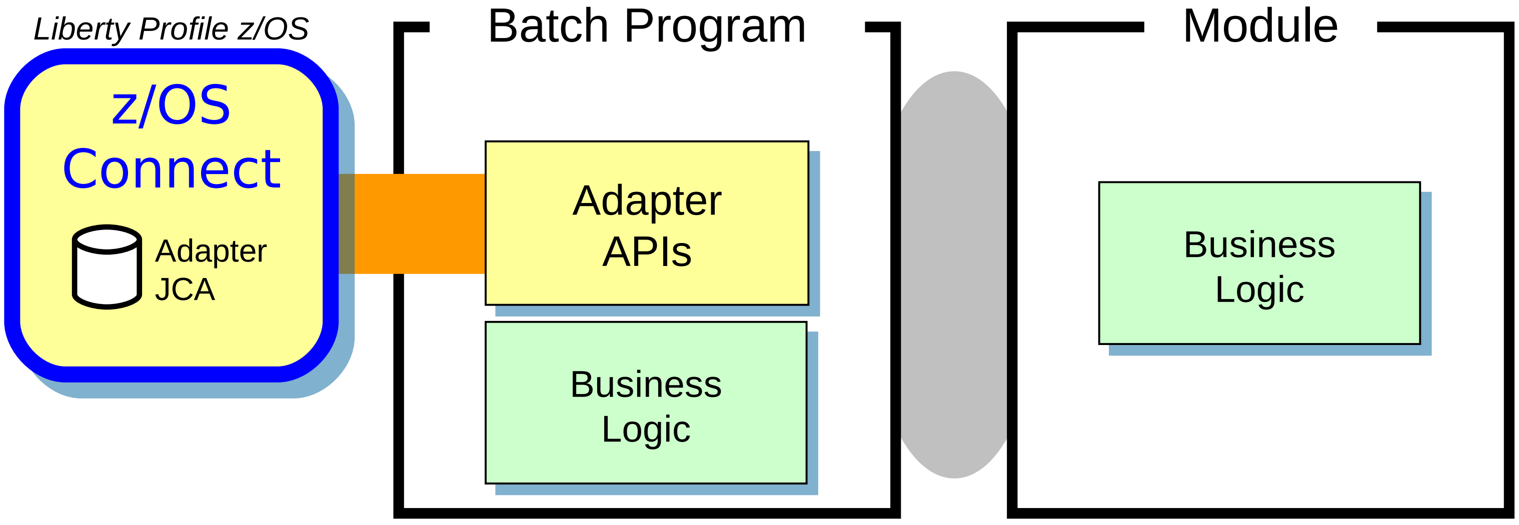
2 The BBO$ long running task handles the calls coming from z/OS Connect. The BBO$/BBO# combination shields the target program from any awareness of Optimized Adapters.

3 The BBO# invocation task is used to perform the EXEC CICS LINK to the target service.

4 Finally, an EXEC CICS LINK is performed and the target service invoked with the data from z/OS Connect passed in. The target CICS program has no knowledge of Optimized Adapters.

Optimized Adapters and Batch Programs

The Batch Program would use the APIs to “Host a Service” and wait for call from z/OS Connect:



Static or dynamic call to other modules

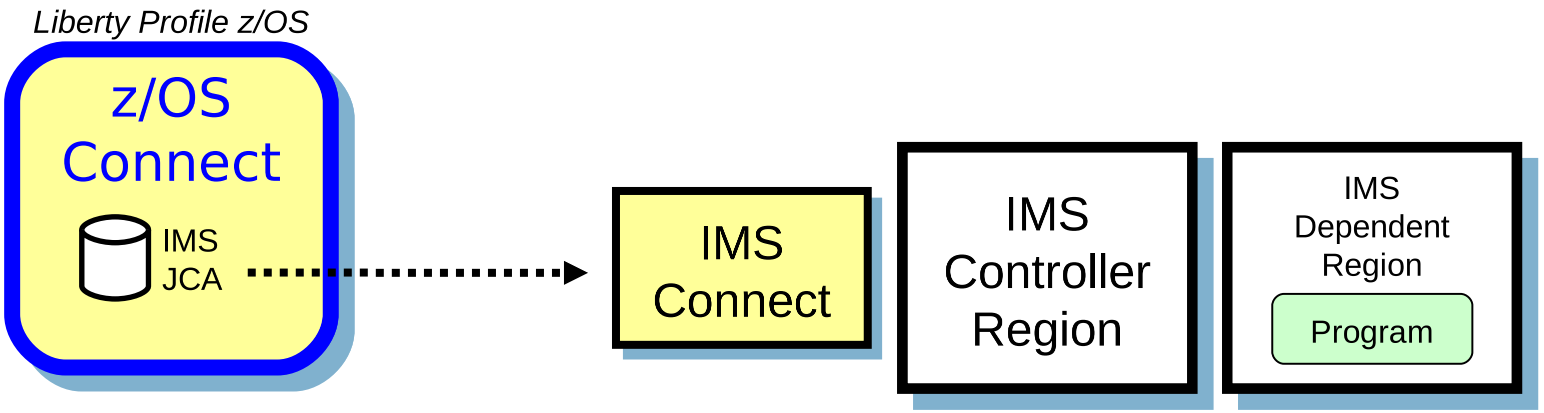
When the target of the z/OS Connect call is a Batch Program, that program will need to use WOLA APIs to “Host a Service”

BBOA1SRV, BBOA1RCA or BBOA1RCS

Business logic being called may reside in the hosting batch program, or that program may perform static or dynamic calls to another module where business logic resides

z/OS Connect and IMS

z/OS Connect interacts with IMS using function in the IMS Mobile Feature Pack with an IMS Connect service provider:



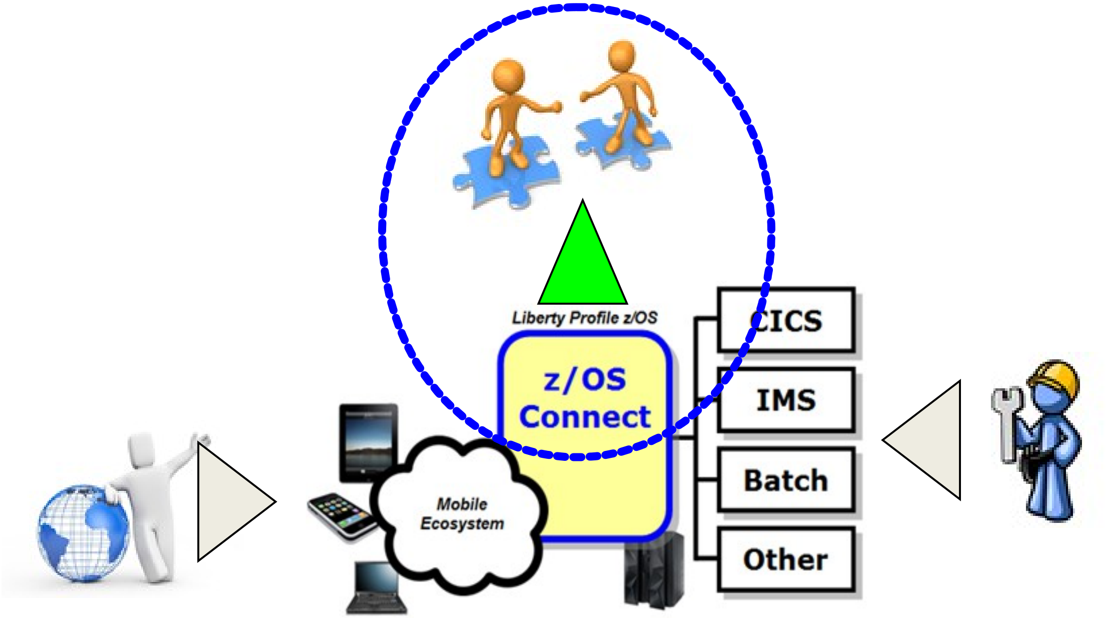
From a z/OS Connect perspective the concepts are the same as what we've shown so far. The syntax for the service provider would be different (IMS Connect vs. Optimized Adapters). And the pointer to the JCA resource adapter would specify the IMS JCA.

It is possible to use the Liberty Optimized Adapters with IMS, but only if the program uses the optimized adapter APIs to 'host a service' like described for Batch programs

ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS214-220&appname=USN

What’s new?

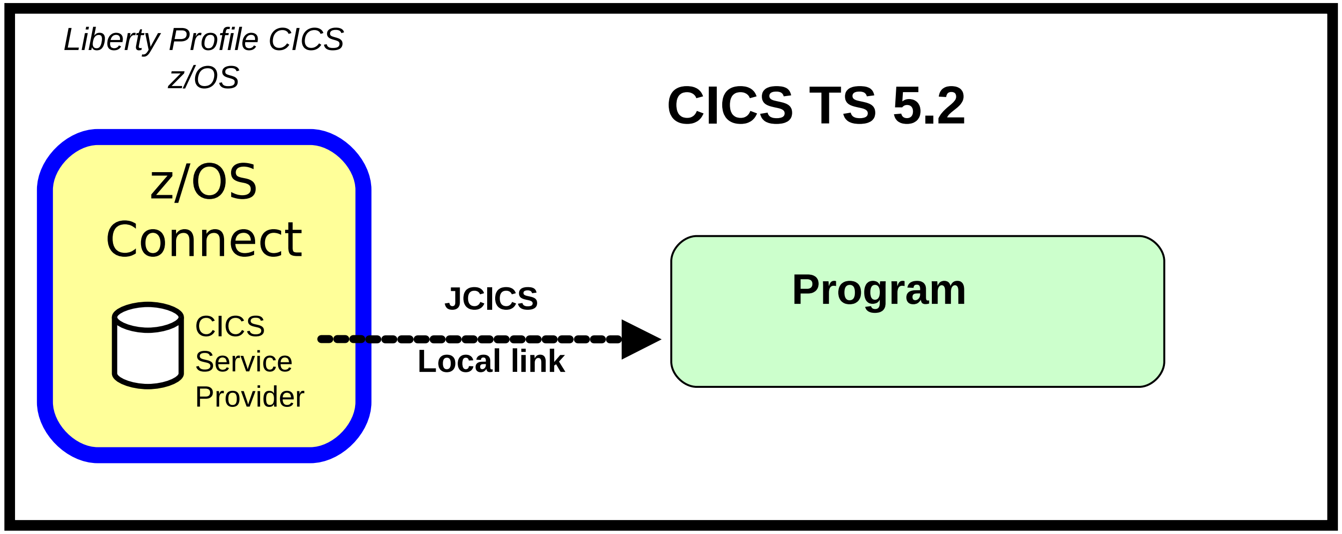
Discuss newest capabilities



z/OS Connect and CICS Liberty

NEW

z/OS Connect is supported in the CICS embedded Liberty server. The service provider used here uses JCICS to interact with CICS resources.

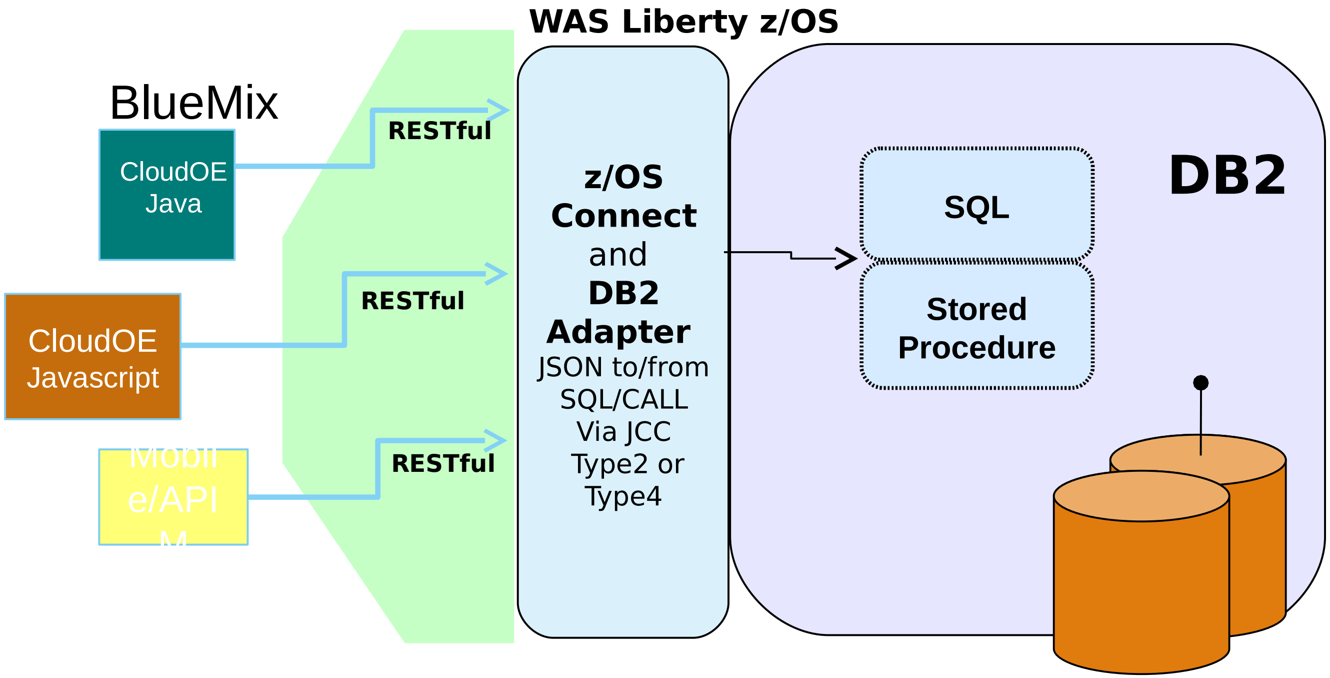


Support shipped in CICS TS 5.2 with APAR PI25503 in 4Q2014

http://www-01.ibm.com/support/docview.wss?uid=swg1PI25503

z/OS Connect and DB2 (intent announced)

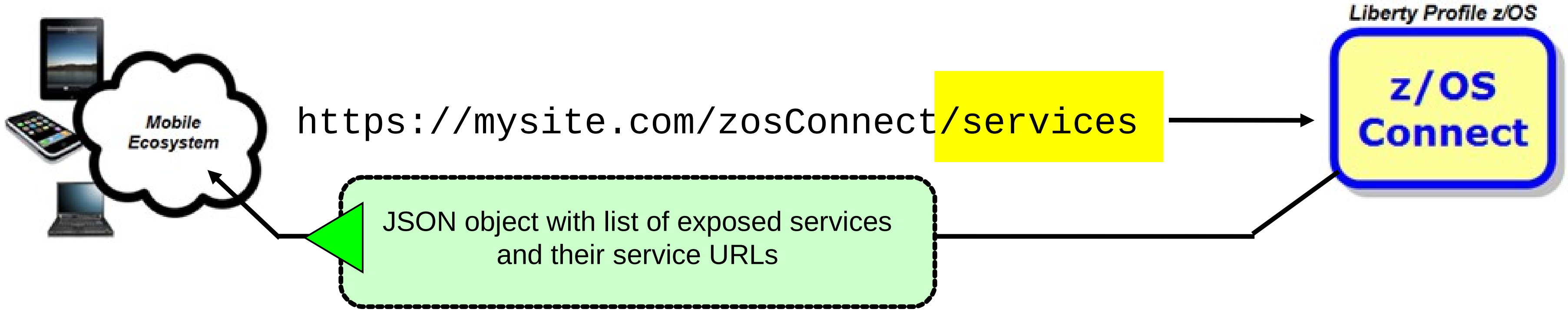
- Make existing DB2 assets – SQL and Stored Procedures – available in today’s growing mobile and cloud application ecosystem
- DB2 Adapter for z/OS Connect ships via Accessory Suites
  - Functionality of DB2 10 and DB2 11
  - Some WAS skill needed to install and configure
- Planned availability in 2H 2015



z/OS Connect Discovery Function (today)

A discovery function is provided to allow developers to query for a list of configured services, and drill down for details on a given configured service:

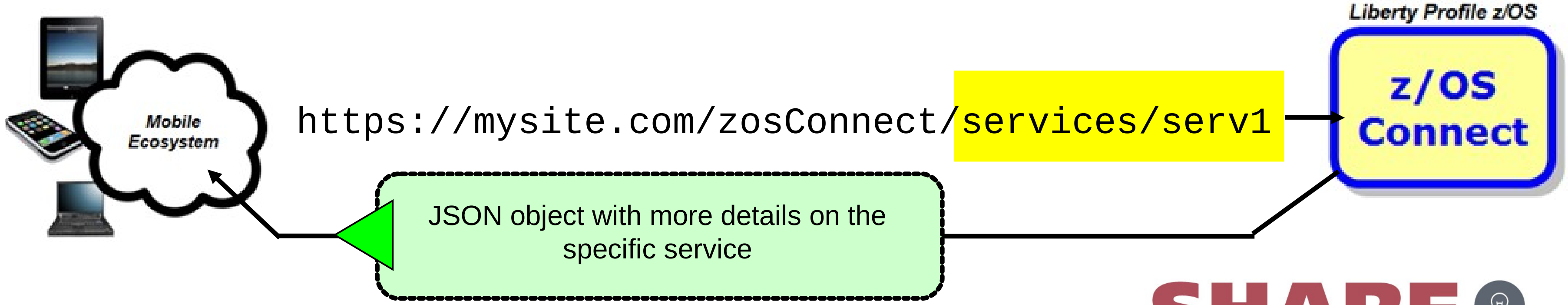
Query for configured services



https://mysite.com/zosConnect/services

JSON object with list of exposed services and their service URLs

Query for details on a given configured service



https://mysite.com/zosConnect/services/serv1

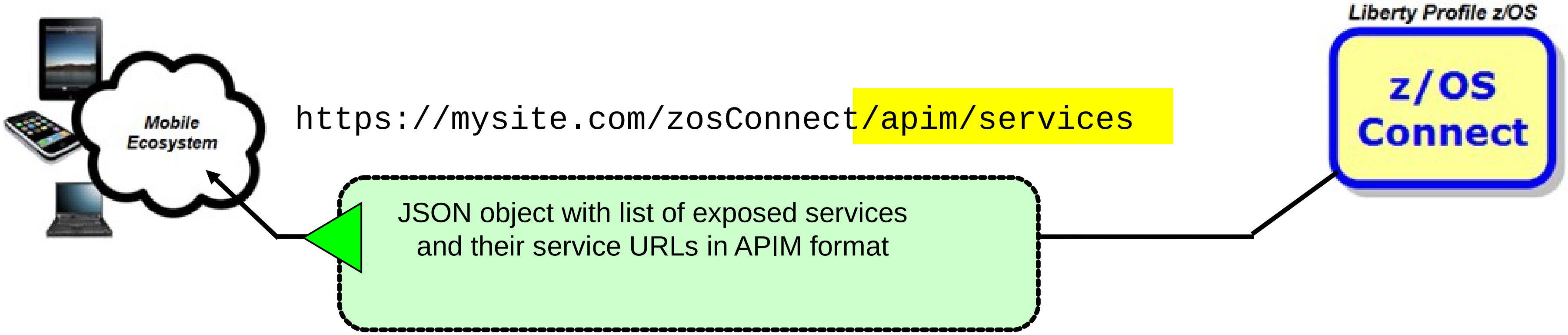
JSON object with more details on the specific service

Discover Enhanced for API Management

NEW

New support in Liberty z/OS Connect in 1Q2015 (WAS 8.5.5.5) that provides improvements for discovery, service search and APIM-compatible service configuration data.

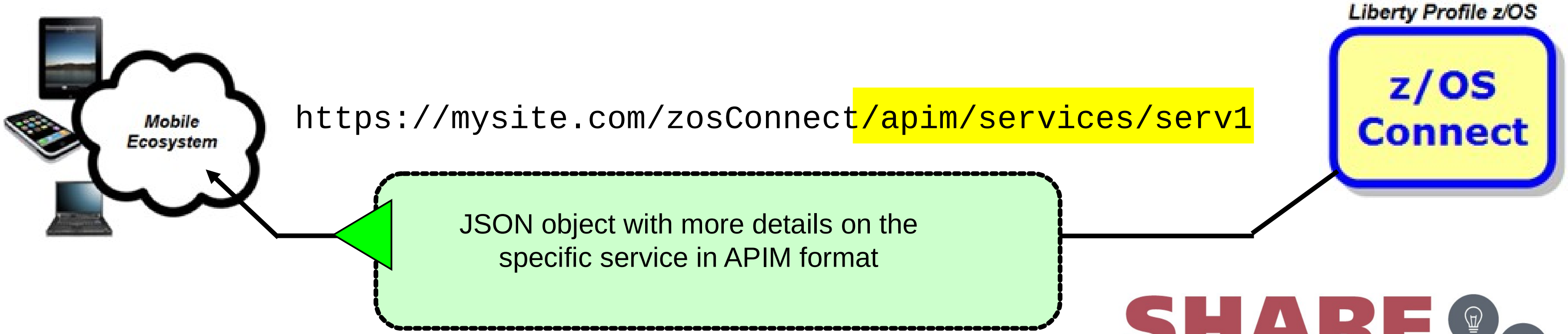
Query for configured services – API Management extensions



https://mysite.com/zosConnect/apim/services

JSON object with list of exposed services and their service URLs in APIM format

Query for details on a given configured service – API Management compatible response data



https://mysite.com/zosConnect/apim/services/serv1

JSON object with more details on the specific service in APIM format

Logger Interceptor

BETA

File Log Interceptor for z/OS Connect to log message payloads

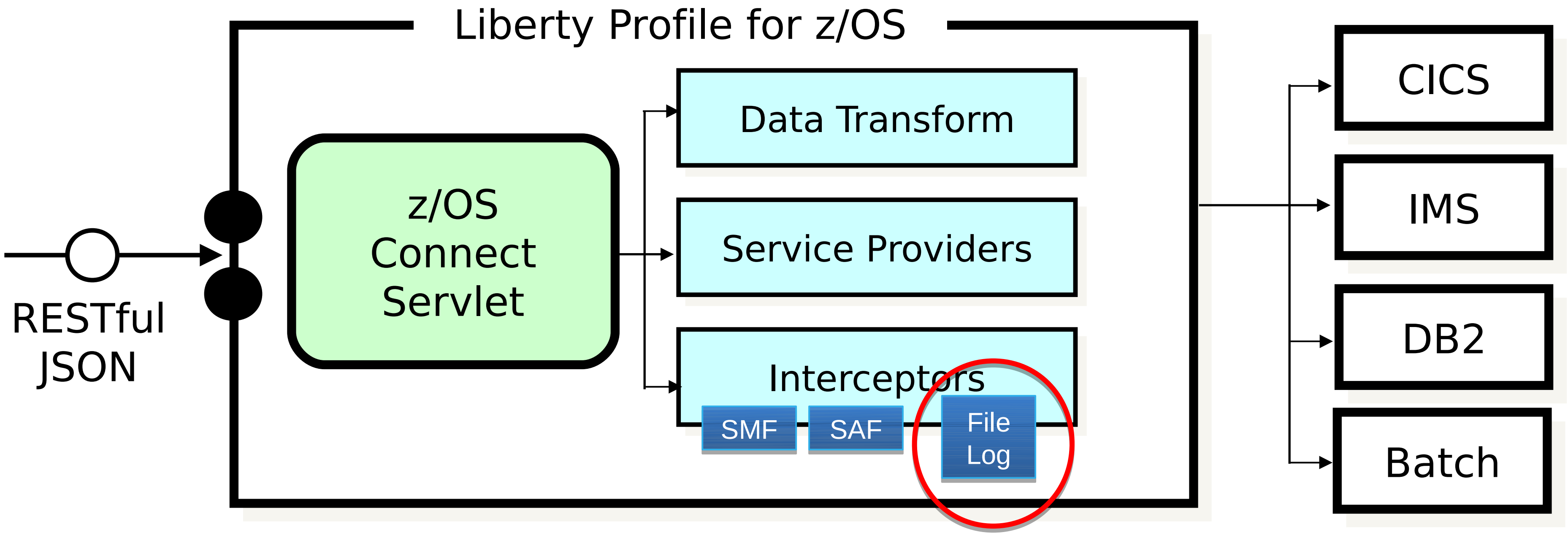
Separate interceptor for file system logging

Log options (REQUEST | RESPONSE | BOTH)

Maximum size or time for log roll over

Ability to pass variables to set log file name (ie: %serverName% and %jobName%, etc)

Context header/banner provided for each Request/Response – Timestamp, size, URL, z/OS Connect service name, etc



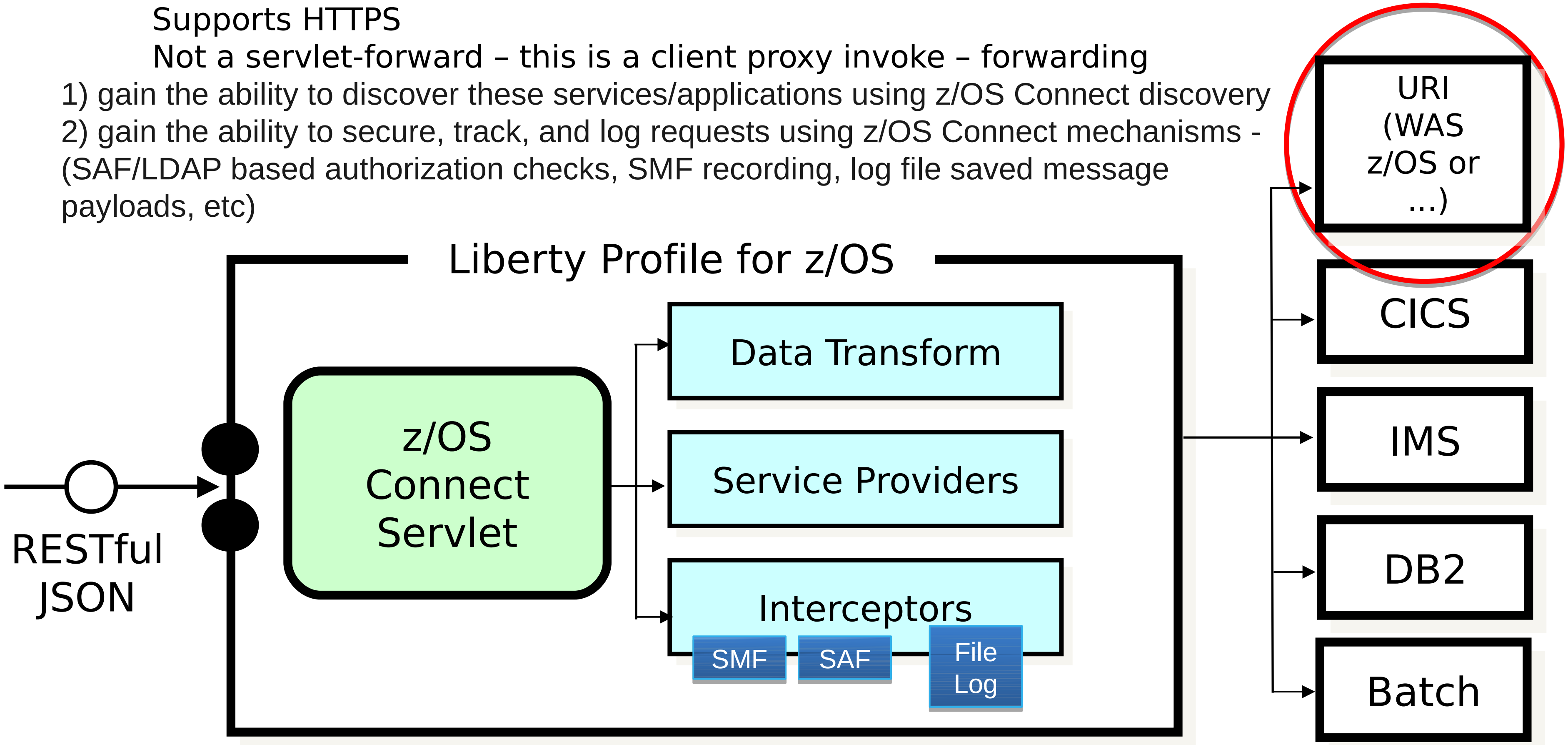
Discovery Proxy Service Provider

BETA

z/OS Connect as a discovery proxy for HTTP-based applications

A service provider for HTTP client requests
Allows specification of URL in service provider parameters
Supports HTTPS
Not a servlet-forward – this is a client proxy invoke – forwarding

1) gain the ability to discover these services/applications using z/OS Connect discovery
2) gain the ability to secure, track, and log requests using z/OS Connect mechanisms - (SAF/LDAP based authorization checks, SMF recording, log file saved message payloads, etc)



HTTP Connections

BETA

Allow HTTP connections to z/OS Connect

Currently, z/OS Connect mandates the use of HTTPS

Adds new ‘**requireSecure**’ and ‘**requireAuth**’ parameters in zosConnectManager and zosConnectService elements

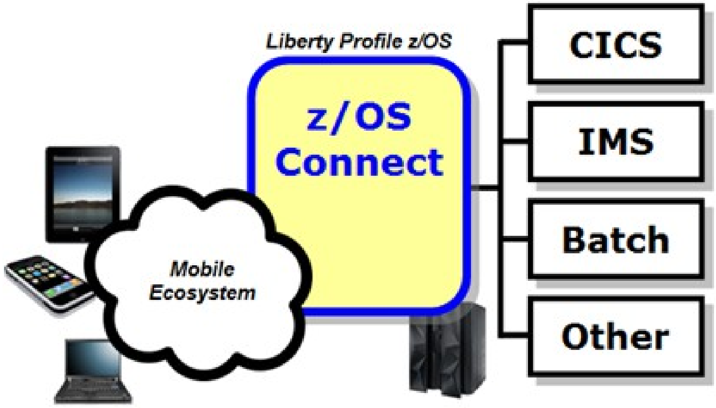
HTTPS connections with
- client or server certificate authentication
- basic authentication or no authentication

HTTP connections with
- basic authentication or no authentication

Less secure HTTP services are often rated as good-enough for intranet solutions, although this will be a narrowing marketplace in certain areas of the world where end to end IP based encryption is becoming a feature of solutions regardless of the intranet/internet topology - and with more and more critical systems management task being enabled for mobile devices.

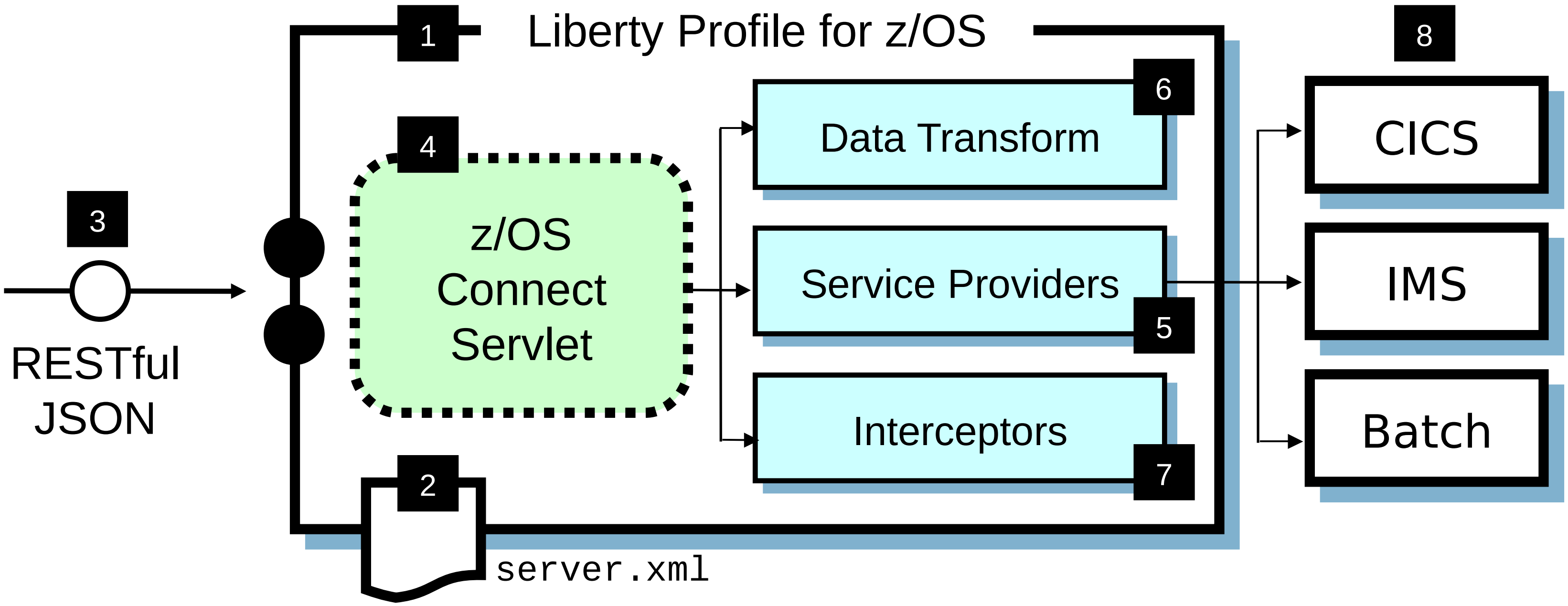
Summary

Wrap-up of Presentation



Summary in One Picture

A summary of what we covered:



1 z/OS Connect is software function that runs in Liberty Profile for z/OS.

2 z/OS Connect is described and configured in the Liberty `server.xml` file

3 z/OS Connect is designed to accept RESTful URIs with JSON data payloads

4 One part of z/OS Connect is a servlet that runs in Liberty Profile z/OS.

5 A ‘Service Provider’ is software that provides the connectivity to the backend system

6 z/OS Connect provides the ability to transform JSON to the layout required by backend

7 ‘Interceptors’ are callout points where software can be invoked to do things such as SAF authorization and SMF activity recording

8 Initially the backend systems supported will be CICS, IMS and Batch

IBM Mobile Test Drive

- **Partner with IBM resources for a complimentary Mobile Test Drive:**
  - Select an entry point such as composing a **Bluemix mobile app** connected to a system of record, assessing the benefits of **Mobile Workload Pricing**, leveraging API enablement using **API Management or z/OS Connect**, building a **mobile front end for an existing 3270 app**, and others
- **Benefits:**
  - Work with IBM mobile specialists to **review existing mobile projects, priorities and requirements**
  - Leverage **best practices and subject matter expertise** for input into your enterprise mobile infrastructure strategy and enterprise mobile roadmap
  - Learn how others **accelerate time to value and differentiate their business with mobile projects** by integrating high value enterprise data and transactions
- **Who should be interested?**
  - Clients that are looking to **leverage existing z Systems data and applications via mobile channels** to drive more value from mobile initiatives
- **What is the commitment?**
  - **1-2 days Discovery** that IBM mobile experts facilitate with your business and technical team, followed by a **deeper Mobile Test Drive**, for **up to a two weeks engagement**
  - IBM will provide **complimentary** technical expertise and access to resources during the Mobile Drive engagement

Contact: Nathan Brice (nbrice@uk.ibm.com)

http://ibm.biz/MobileTestDrive

IBM Mobile Centers of Competency

IBM mobile & z Systems integration specialists help you shape your Mobile Enterprise strategy

- Mobile CoC's located at **5 IBM Client Centers** around the world
- Leverage **experts in enterprise mobile solutions** to help you integrate mobile applications with existing systems and access data in your enterprise without compromising security or performance.

Offerings:

- **Executive briefings**
- **Mobile on z workshops**: Systems of Record (SoR) in a mobile world
- **Design workshops** for mobile enablement and SoR integration
- **Demos and access to mobile technologies**, such as MobileFirst Platform on Linux on z Systems
- **Benchmarks, proof of technology (POT) and proof of concept (PoC)**

ABK Systeme GmbH
the green company

***ABK-Systeme GmbH** worked with the Boeblingen Client Center and is now developing mobile productivity apps for the financial services industry using IBM MobileFirst Foundation running on IBM z Systems, accelerating time-to-market and efficiency.*

Beijing, China

Boeblingen, Germany

Montpellier, France

Poughkeepsie, NY, USA

Tokyo, Japan

Contact: Erich Amrehn (AMREHN@de.ibm.com)

www.ibm.com/systems/services/clientcenters/mobile/