

# B11: z/OS Connect Update: The Force Awakens

Carl Farkas  
IBM Europe zWebSphere consultant  
[farkas@fr.ibm.com](mailto:farkas@fr.ibm.com)

(thanks to Don Bagwell, Bruce Armstrong and others for much of the content)



Sharpen your competitive edge  
**2016 IMS Technical Symposium**  
March 7 – 10, 2016  
Wiesbaden, Germany

[www.ims-symposium.com](http://www.ims-symposium.com)

## Session objective

**z/OS Connect was introduced by IBM in May 2014 as a feature of WebSphere Liberty. This session will present the latest version of the technology, the z/OS Connect Enterprise Edition v2.0. You'll understand how it has matured and the position it takes in your enterprise with IMS.**

**Want more details? Go to Haley's sessions A13 or B16!**

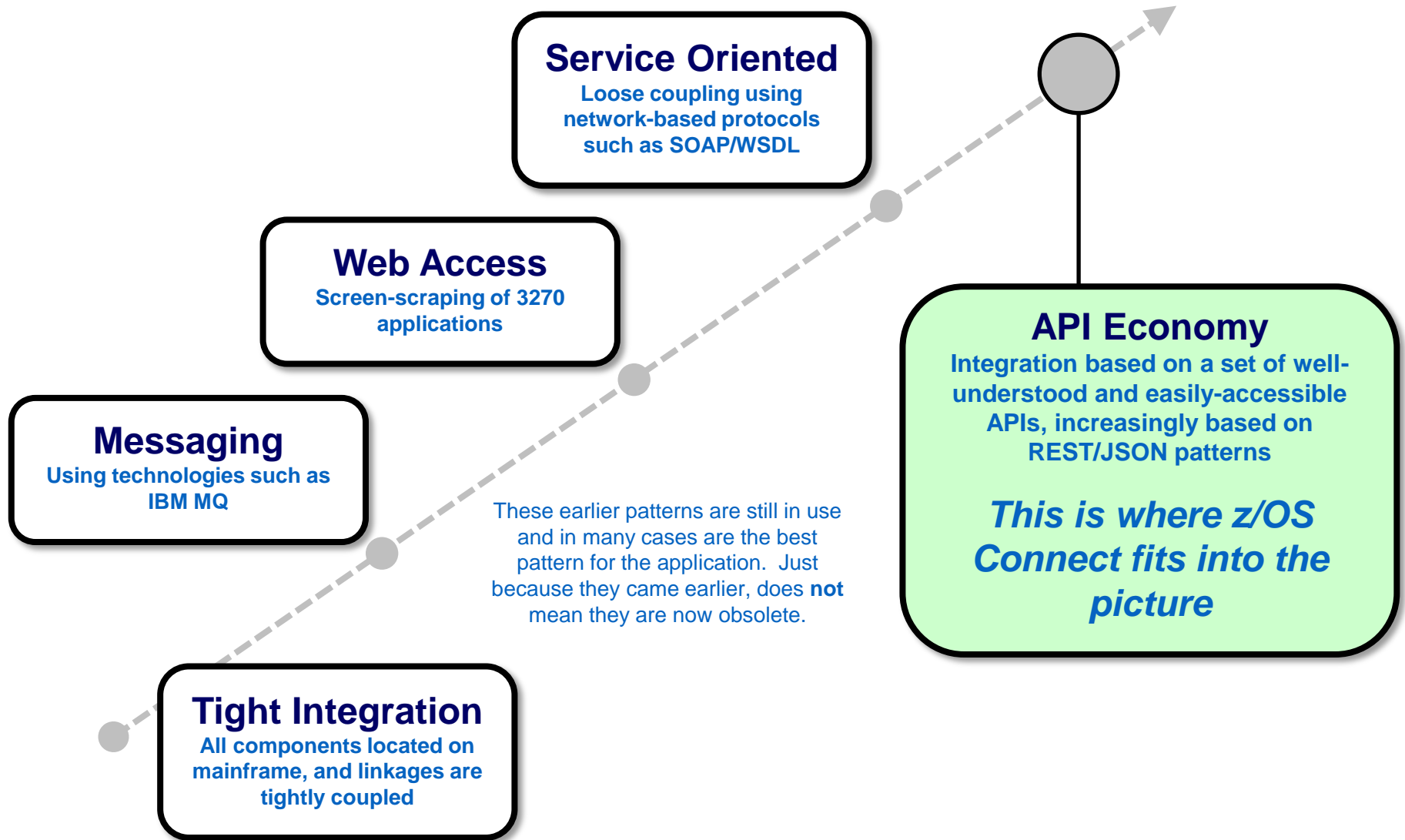
# Agenda

- Introduction z/OS Connect
- z/OS Connect Enterprise Edition v2.0

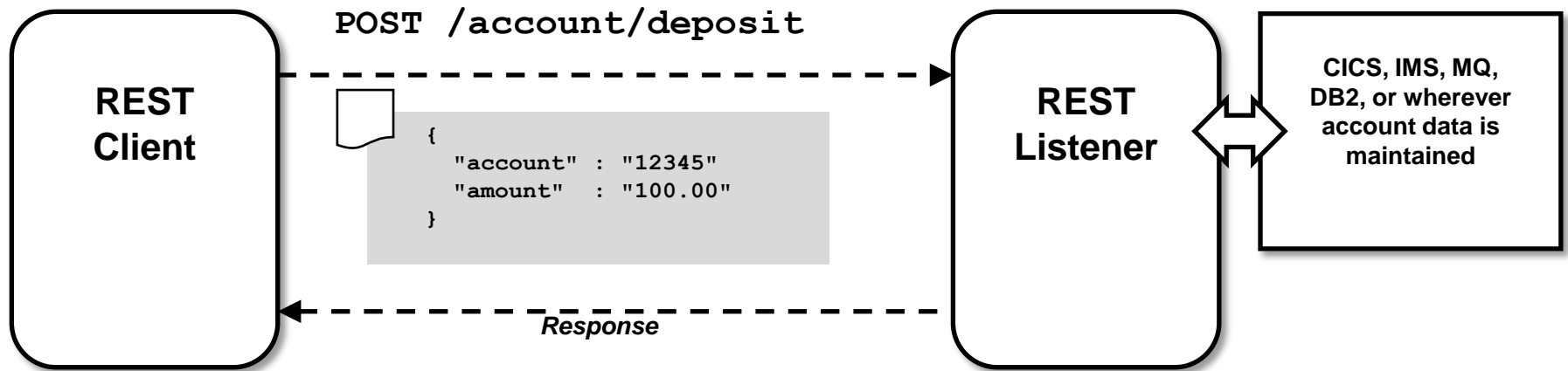


Getting a little bit older... maturing nicely!

# Evolution of Mainframe Integration Patterns



# Simplified Overview of REST/JSON



**REST** - “Representational State Transfer” ... which uses HTTP and HTTP verbs to allow a client to interact with a server over the TCP/IP network.

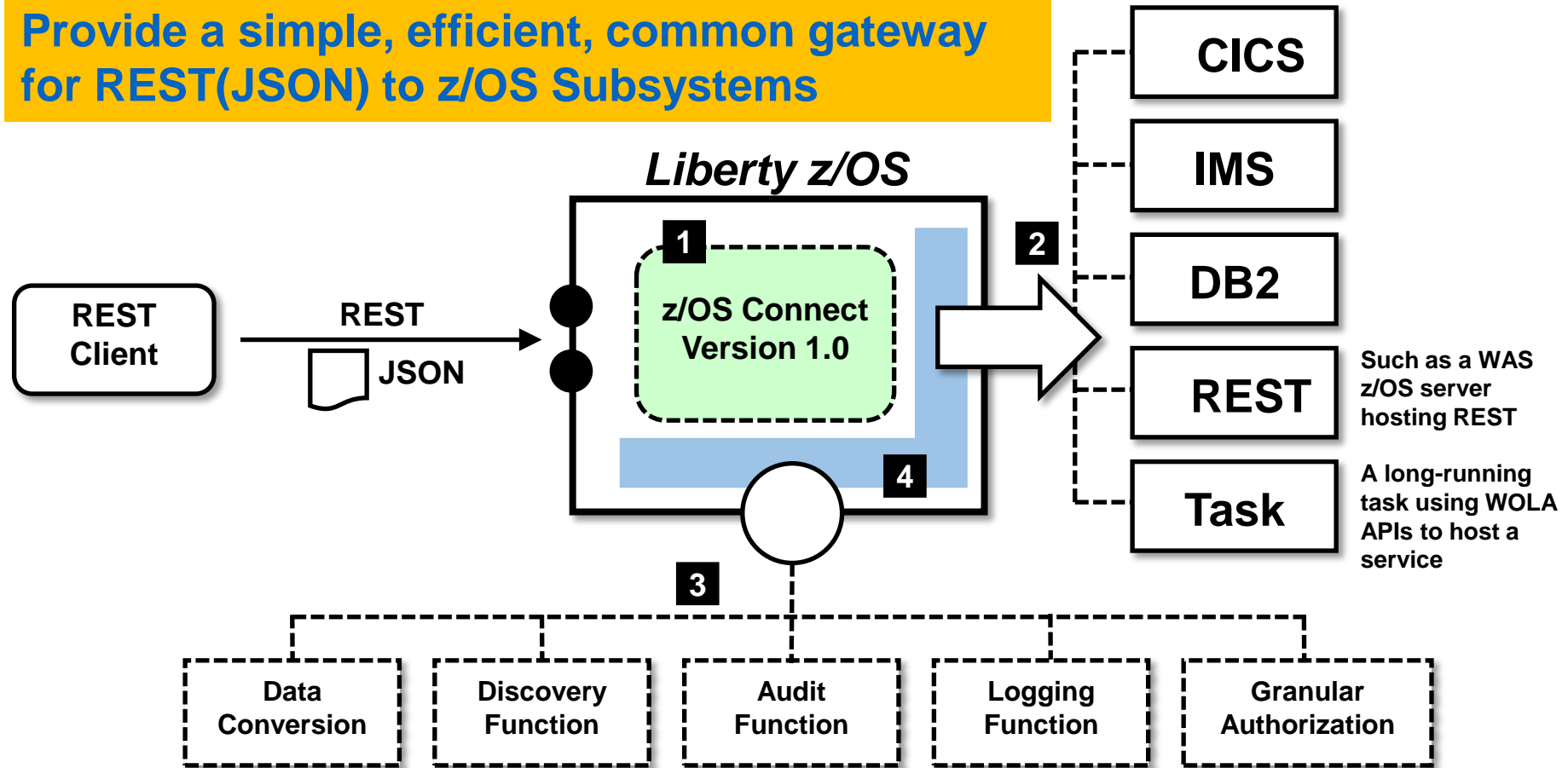
**JSON** - “JavaScript Object Notation” ... a name/value pair representation of data that is relatively lightweight and generally simpler to handle and parse than XML.

**REST is increasingly popular as an integration pattern because it is stateless, relatively lightweight, is relatively easy to program to, and operates well with discovery mechanisms such as IBM’s API Management product.**

# z/OS Connect Version 1 (Liberty Feature)

Goal of z/OS Connect:

**Provide a simple, efficient, common gateway for REST(JSON) to z/OS Subsystems**



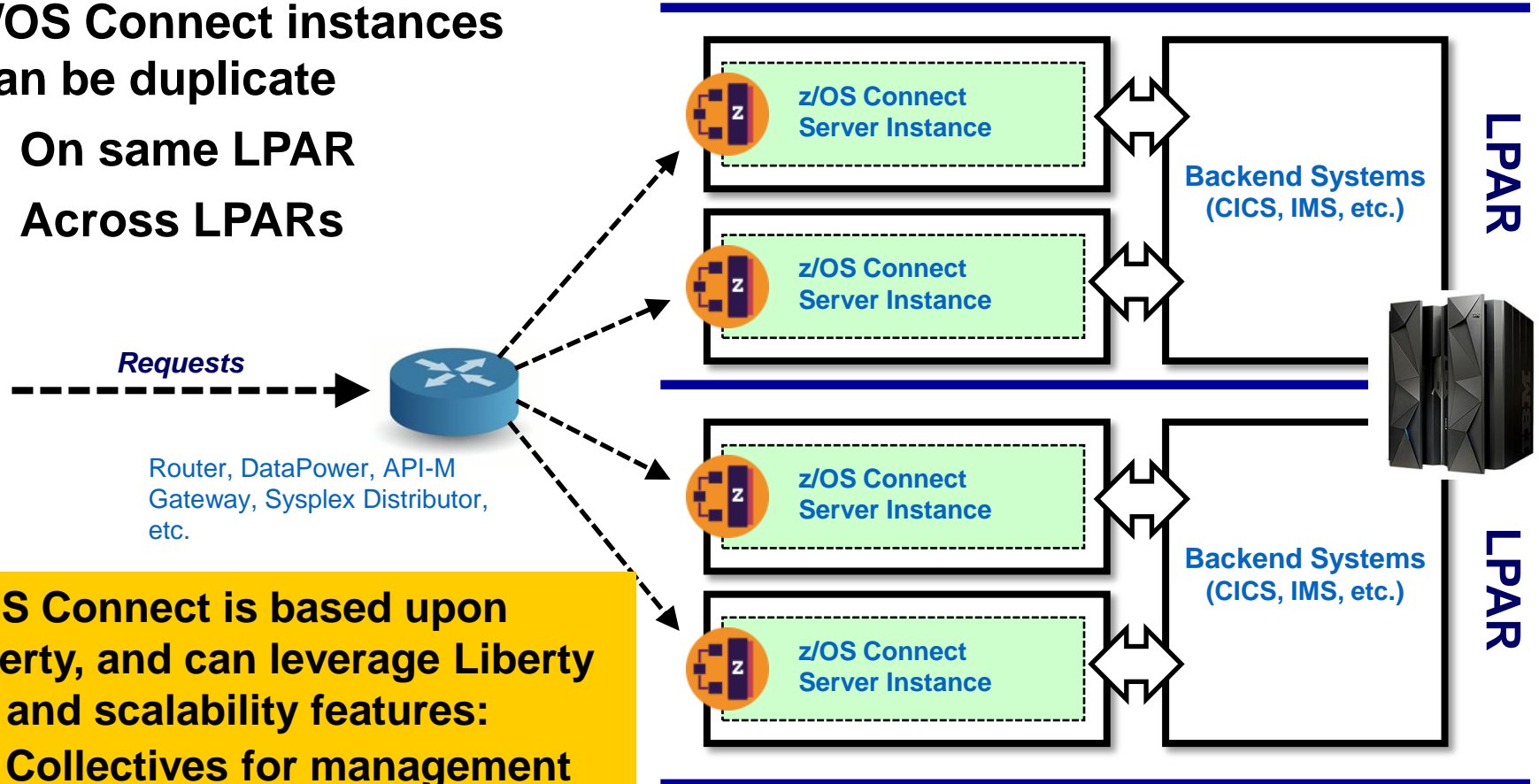
1. z/OS Connect a “feature” of Liberty
2. “Service Provider” = backend connectivity

3. “Interceptors” = configurable function
4. Extensible interface = flexibility

# Achieving a Highly Available and Scalable Environment

z/OS Connect instances can be duplicate

- On same LPAR
- Across LPARs



**z/OS Connect is based upon Liberty, and can leverage Liberty HA and scalability features:**

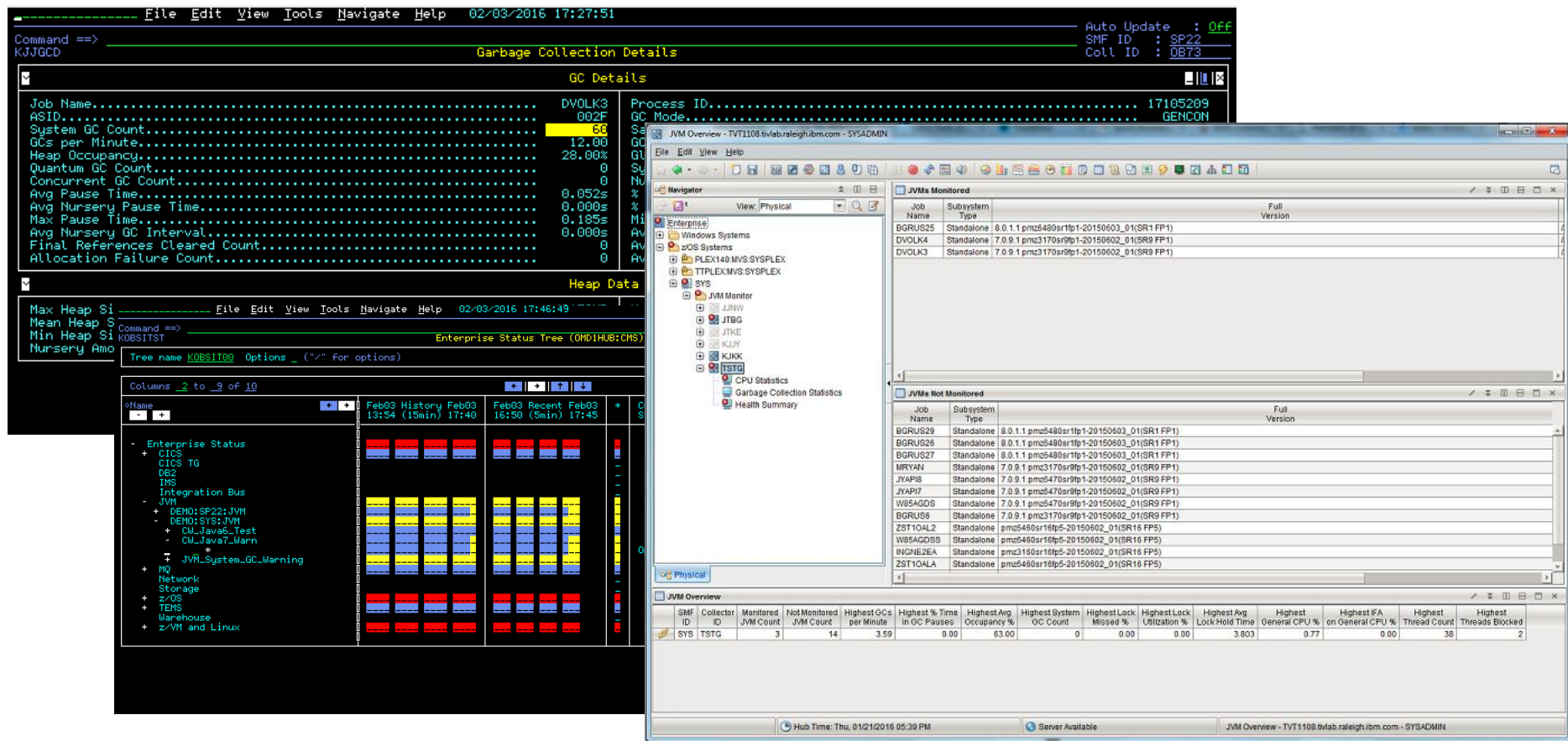
- ✓ Collectives for management
- ✓ Clusters for HA and scalability
- ✓ Dynamic routing
- ✓ Auto-scaling

Because REST is *stateless*, network routing functions can be placed in front of duplicated instances and balance traffic



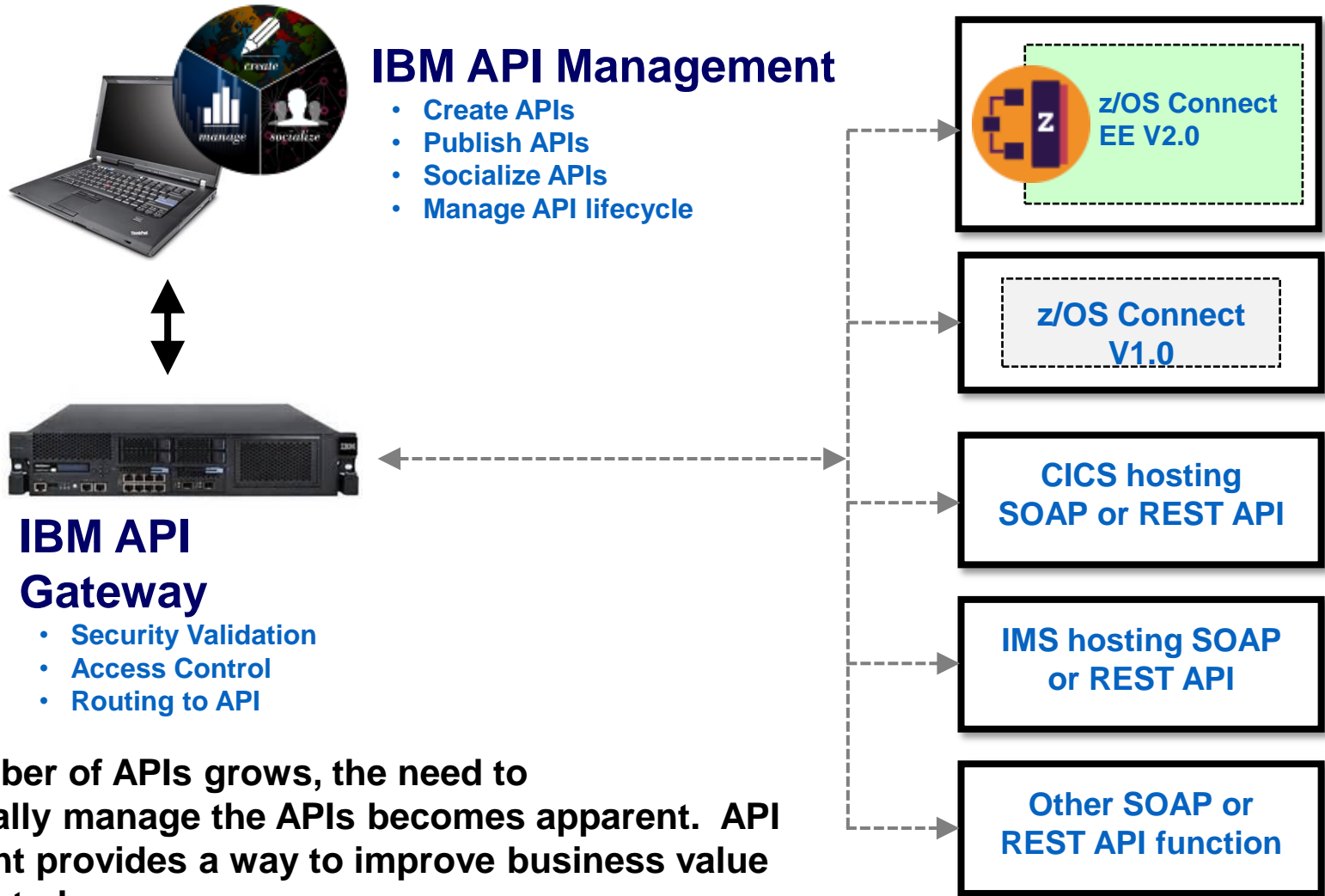
# Hot news! IBM now offers industrial-strength Liberty z/OS monitoring

- IBM announced on 16 February (see ENUS216-048) the OMEGAMON Monitoring for JVM
- Drill-down detail of your JVMs (Liberty or others)
- Coherent large view (3270 or GUI) of all of your subsystems





# High-Level of API Management



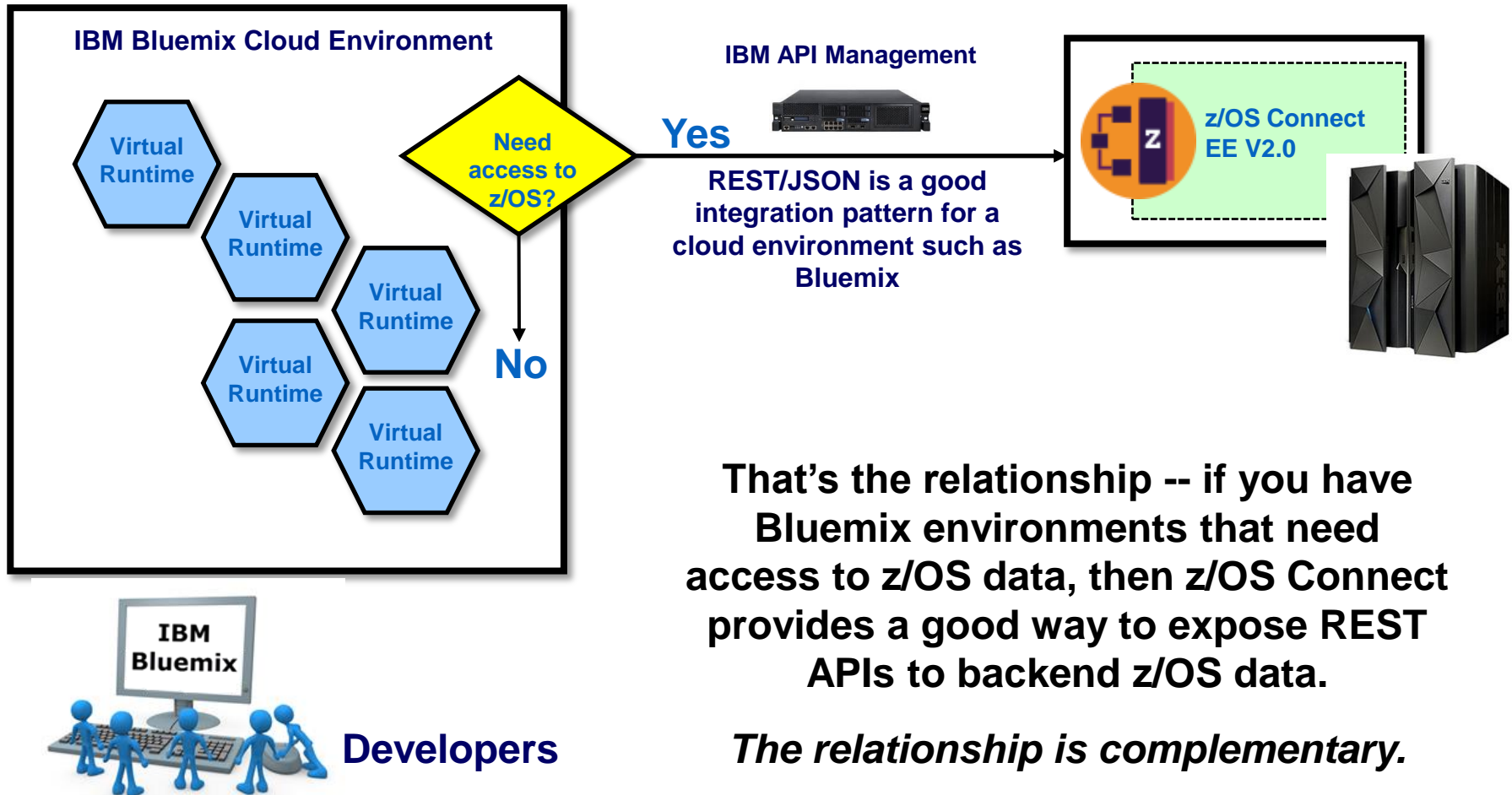
As the number of APIs grows, the need to systematically manage the APIs becomes apparent. API Management provides a way to improve business value of APIs created

The relationship is complementary -- not required by z/OS Connect or other API providers, but *very helpful* in an API environment.

# What about Bluemix and z/OS Connect EE V2?

**IBM Bluemix  
Website:**

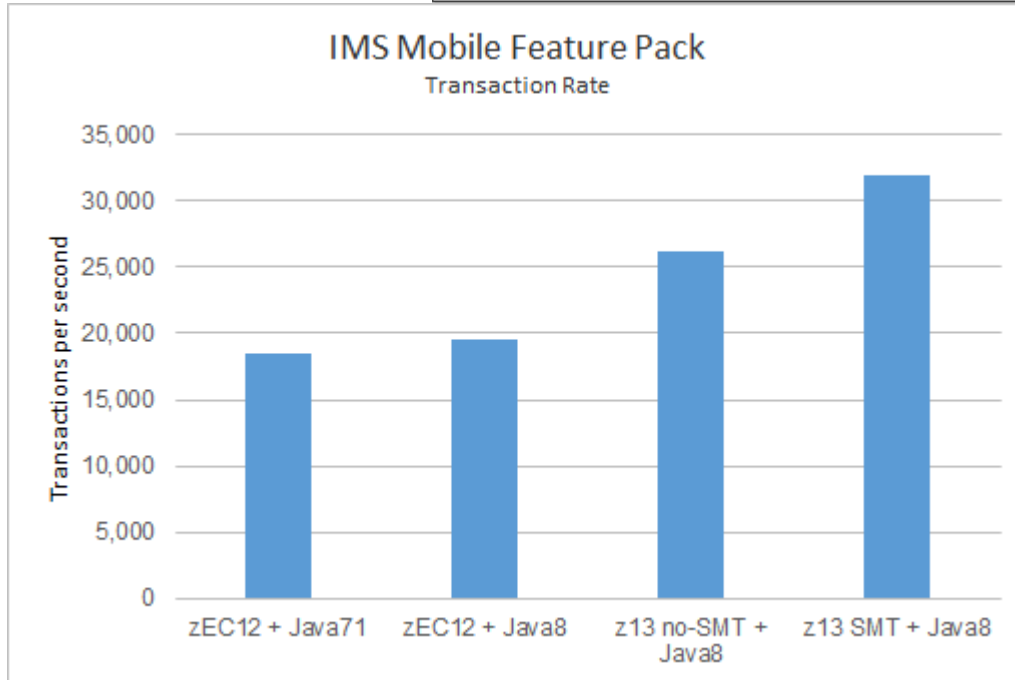
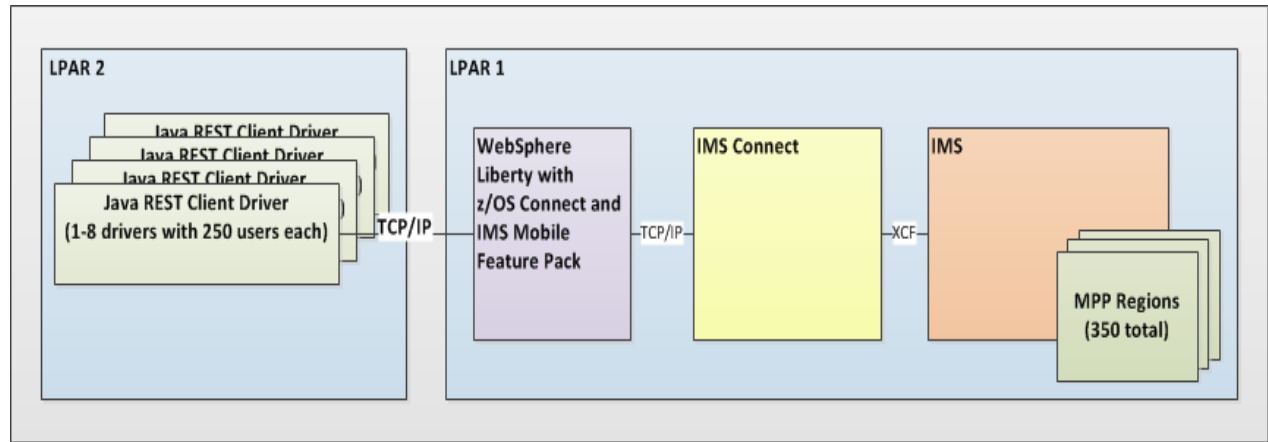
IBM Bluemix is an open-standards, cloud platform for building, running, and managing applications. With Bluemix, developers can focus on building excellent user experiences with flexible compute options, choice of DevOps tooling, and a powerful set of IBM and third-party APIs and services.



**That's the relationship -- if you have Bluemix environments that need access to z/OS data, then z/OS Connect provides a good way to expose REST APIs to backend z/OS data.**

***The relationship is complementary.***

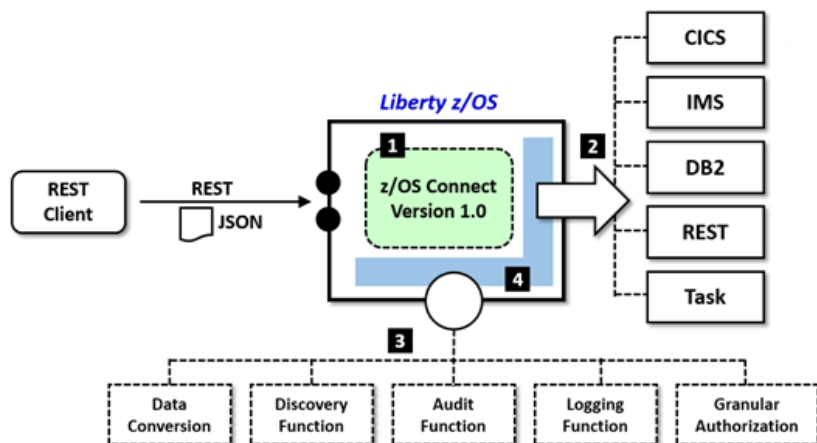
# z/OS Connect with IMS (Mobile Feature Pack)



(Controlled measurement environment, results may vary)

IMS Mobile Feature Pack **73%** aggregate improvement in throughput from z13 and IBM Java8

# V1: Good Start ... Known Areas for Improvement



## Differing Entitlement / Delivery

z/OS Connect was a no-charge feature entitled under with WAS z/OS, CICS, IMS or DB2 license. Multi-backend usage was different depending on entitlement. Acquisition and installation was different depending on entitlement.

*Summary: entitlement and delivery model created some confusion*

## Tooling differences Across Backends

The tooling support was a function of the entitlement option used to acquire z/OS Connect V1. IMS and DB2 provided an Eclipse based tooling environment. CICS employed CICS Explorer, but not to the degree IMS and DB2 integrated z/OS Connect with tooling. WAS z/OS relied on manual service definitions.

*Summary: consistent tooling preferred*

## First Generation REST

z/OS Connect V1.0 REST implementation was essentially an RPC model. Query parameters were not accessible. Header information not accessible. HTTP verb usage pattern not well-formed and consistent.

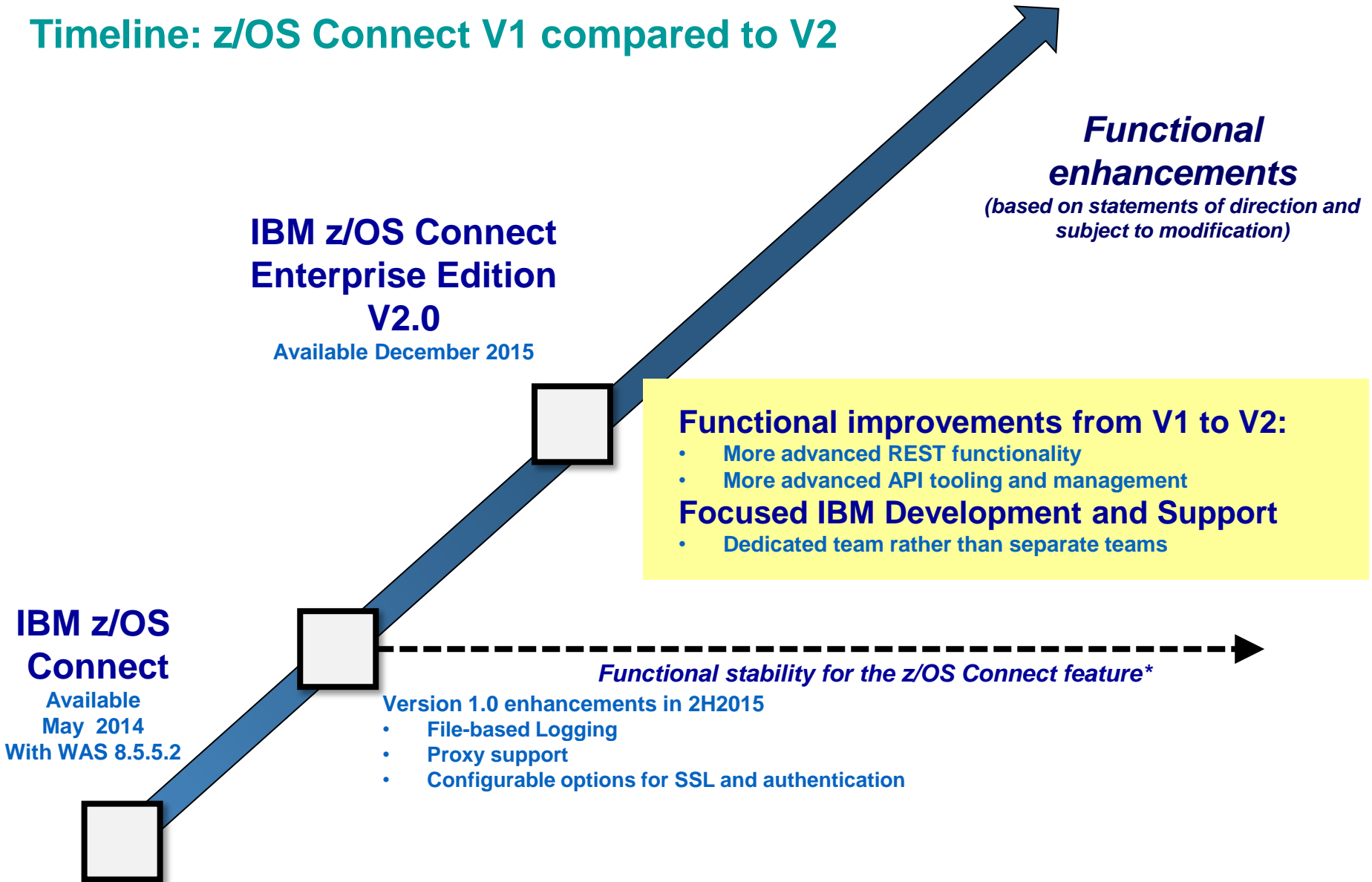
*Summary: more complete REST implementation model was needed*

# Agenda

- Introduction z/OS Connect

- **z/OS Connect Enterprise Edition v2.0**

# Timeline: z/OS Connect V1 compared to V2



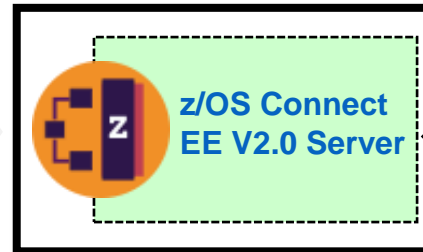
\* Except where IBM has issued a statement of direction for V1.0. Go to [this](#) page for more.

# High-Level Overview of z/OS Connect EE V2.0

## Runtime Server 1

- Runs on Liberty z/OS
- Hosts APIs you define to run in it
- Connects with backend system
- Liberty + z/OS Connect = “instance”
- You may have multiple instances

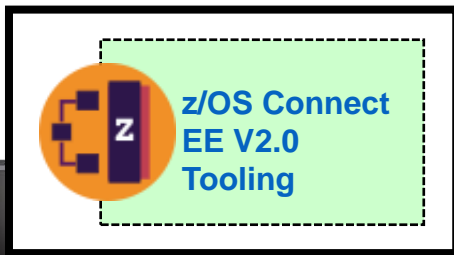
*Liberty z/OS*



Backend Systems  
(CICS, IMS, DB, etc.)

- IBM z/OS 2.1 or higher
- IBM 64-bit SDK for z/OS, Java Technology Edition V7.1.0 or V8.0.0

*Eclipse*



- IBM CICS Explorer V5.3
- IBM IMS Explorer for Development V3.2
- IBM Explorer for z/OS Aqua V3.0


## Tooling Platform 2


- Integrates with an Eclipse environment
- Define APIs
- Define data mapping
- Deploy APIs to runtime server
- Export API archive for other tools to deploy




# Comparison of REST Support V2.0 vs. V1.0

## z/OS Connect V1.0:


**POST**    /account/create    +  (JSON with account create information)

**POST**    /account/balance    +  (JSON with account number)


**POST**    /account/update    +  (JSON with account number and deposit)

This corresponds to the lower level we showed earlier. It's very basic. It may be "good enough" for some use-cases, but it falls short of what many developers seek when creating REST APIs... and Swagger 2.0 compatibility

## z/OS Connect V2.0:

**POST**    /account?name=Fred    +  (JSON with Fred's information)

**GET**    /account?number=1234

**PUT**    /account?number=1234    +  (JSON with dollar amount of deposit)

↑  
HTTP Verb conveys the method against the resources; i.e., POST is for create, GET is for balance, etc.

↑  
URI conveys the resource to be acted upon; i.e., Fred's account with number 1234

↑  
The JSON body carries the specific data for the action (verb) against the resource (URI)

**More aligned with developer requirements for REST APIs**

# Eclipse-based Tooling for z/OS Connect EE V2.0

The screenshot shows the Eclipse IDE interface for defining an API. The Project Explorer on the left shows a project named 'GoodHealth' with a sub-project 'api' containing a 'patient' folder. Inside 'patient', there is a '{patid}' folder with a 'GET' method. The main editor area shows the 'Describe your API' tab. The 'Path' field contains '/patient/{patid}?userId&zipcode'. The 'Methods' list shows 'POST', 'GET' (selected), 'PUT', and 'DELETE'. The 'GET' method is associated with the 'PatientService'. The 'Description' field contains 'CRUD api for patient details, medical threshold data.' The 'Mapping' field is empty. The 'Service' field is set to 'PatientService'. The 'Mapping' field is empty. The 'Service' field is set to 'PatientService'. The 'Mapping' field is empty.

**Eclipse project view, which is familiar to developers who have used Eclipse-tooling for other development projects**

**Access query parameters from the URI**

**Provide data mapping definitions to the service**

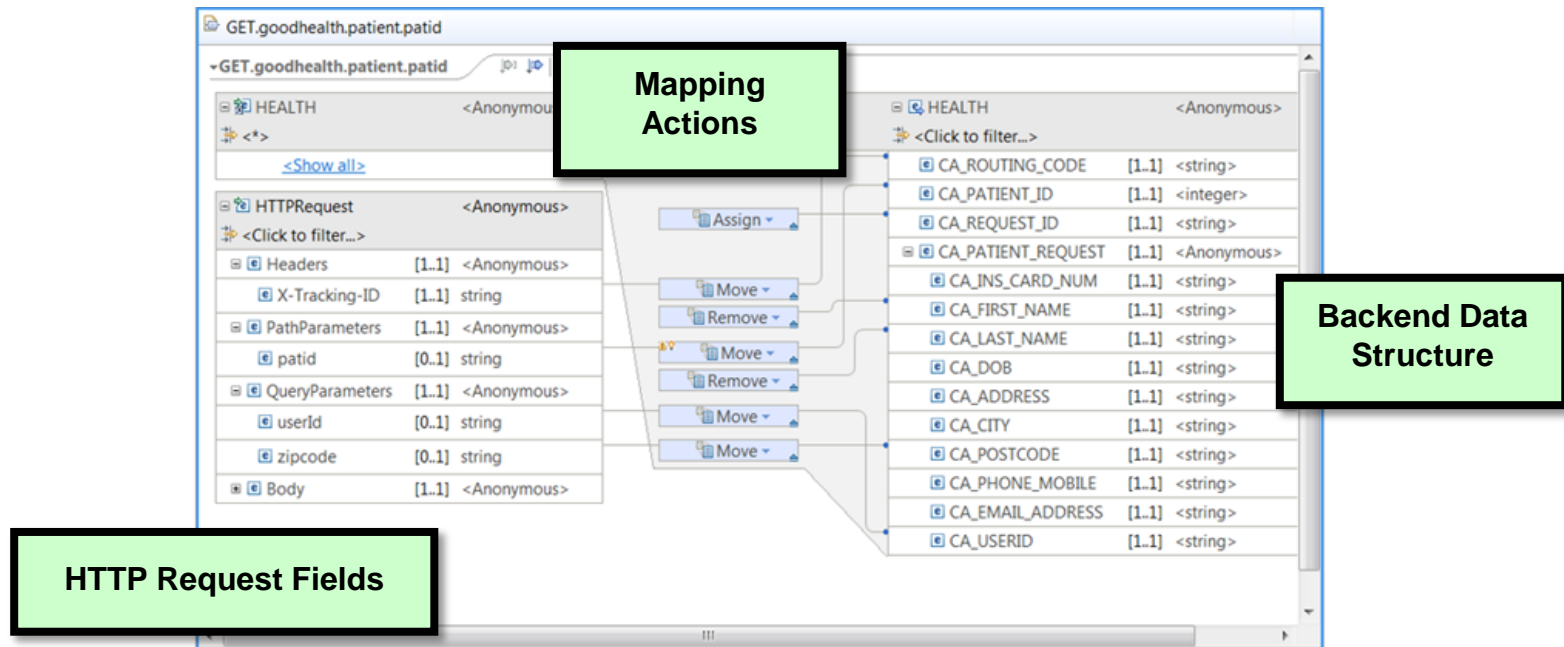
**Assign API function based on HTTP verb**

**API projects can be exported and imported for portability between developers**

**API definitions are created through the tool, which is consistent across backend systems (CICS, IMS, etc.)**

# Request Mapping Capabilities

The API mapping model adds a powerful abstraction layer between the API consumer and the underlying z/OS assets.



- Mapping of HTTP headers, path parameters (URI templates), and query parameters to the fields in the request message JSON body.
- Pass-through by default the fields in the request or response message JSON body, with possibility to eliminate fields.
- Mapping and defaulting of HTTP headers in the HTTP response message.

# And IMS Tooling?

IMS Mobile - IMS\_Explorer\_Catalog\_ReadOnly/Server Transactions/zTecBlueCFARKASzOSC/IVTNOM.trn - IMS Enterprise Suite Explorer

File Edit Navigate Search Project Run Window Help

Activity: Other Activity

Quick Access

IMS Gateway ... Project Explorer

IMS Explorer Task Launcher IVTNOM.trn

Host Connections

IMS Explorer Transaction Message Metadata Editor

Define all the input and output messages.

Trancode: IVTNOM

Input Messages

Run a Test Case Console

Run a test case

Execution time is 0.79700 seconds for the input and output message pair BrowseContacts.

Runtime configuration: RunBrowseContactService Test case name: BrowseContacts.stc

Input messages

Input and Output Message Pair	Field value	Field Length
<input checked="" type="checkbox"/> IN_NAME1	LAST1	
<input checked="" type="checkbox"/> IN_NAME2		

Output messages

View output message with: IVTNOM - OUTPUT

Input and Output Message Pair	Field value	Field Length
<input checked="" type="checkbox"/> OUT_MSG	ENTRY WAS DISPLAYED	
<input checked="" type="checkbox"/> OUT_NAME1	LAST1	
<input checked="" type="checkbox"/> OUT_NAME2	FIRST1	
<input checked="" type="checkbox"/> OUT_EXTN	8-111-1111	
<input checked="" type="checkbox"/> OUT_ZIP	D01/R01	
<input checked="" type="checkbox"/> OUT_SEGNO	0001	

Close

Lovely Eclipse tooling for centralized configuration definitions, mapping, testing. Bravo IMS!

# API Archive (AAR) -- Service Packaging



## Service definitions in server configuration file

(or in related side files)

## Discovery function returned JSON with services, but:

- Not Swagger definition
- Only service URIs, but did not contain information about connectivity to backend

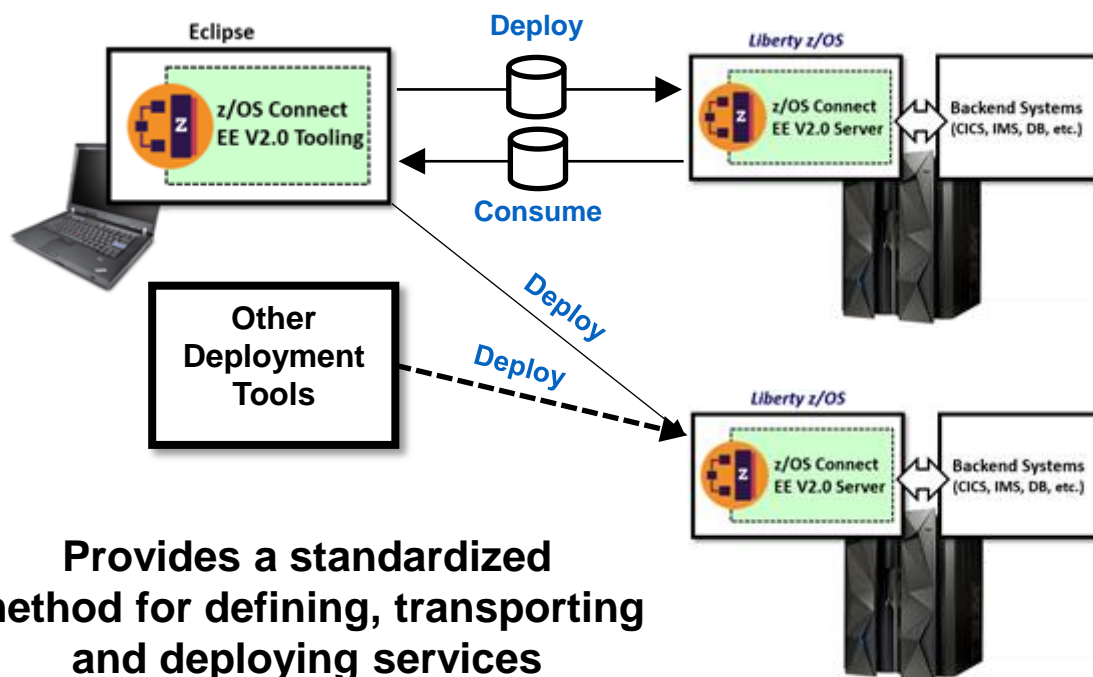
## Portability of service limited:

- No good “export” of service from a hosting server
- Import into tooling largely a manual process



## API Archive (AAR) File

- ZIP-format file
- Contains Swagger documentation of service
- Contains information about backend connectivity
- Produced by tooling
- Exportable to server runtime | Consumable by tooling
- *NOTE: manual server.xml updates are still required. for some definitions.... today*

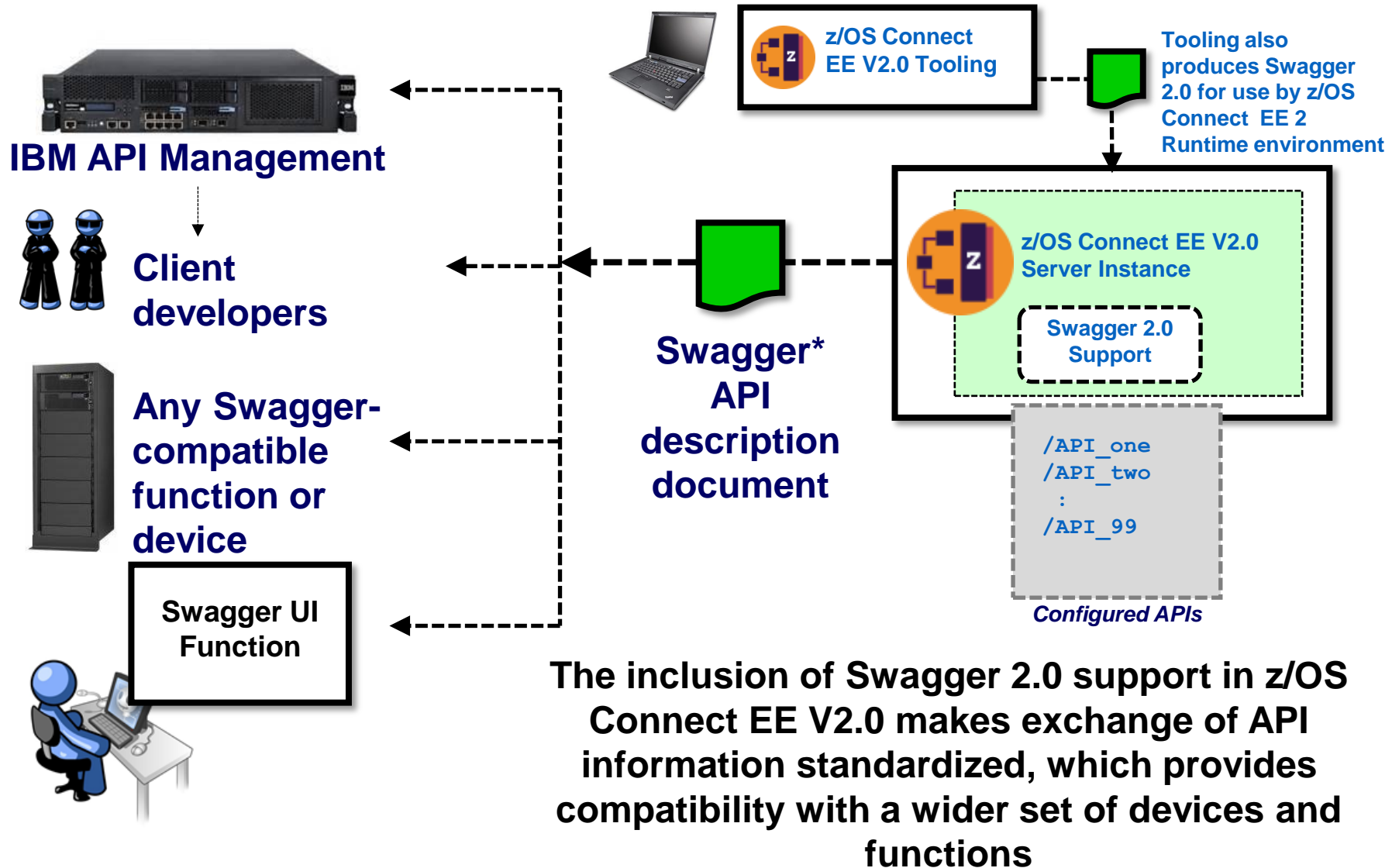


**Provides a standardized  
method for defining, transporting  
and deploying services**

More flexibility and greater productivity

Better model from V1.0's service definitions

# Discoverable APIs with Swagger 2.0



\* An emerging accepted industry standard, but not an official open standard

## Swagger 2.0 example with IMS

And more cool... from this site, I can directly generate my definition, and then I can directly generate my web API client code.

And for the icing on the cake (“la c rise sur le g teau” for H l ne), Swagger will generate my client (or server side) code. Yippee!

The image shows the Swagger Editor interface with a Swagger API definition for 'carlPhoneResource'. The definition includes a 'paths' section with a 'get' method for '/carlResource/{nomFamille}'. The 'Responses' section shows a '200' response with a schema 'carlPhone\_GET\_response\_200'. The 'Parameters' section shows a 'nomFamille' parameter of type 'string'.



# Installation & configuration overview z/OS Connect EE v2.0

1. **Install:** First install & configure a basic Liberty z/OS server as an STC with an Angel
  - See Liberty z/OS KC & Techdocs for tips.
  - We suggest using “basic security” just to start.
2. **Install:** Download and install the z/OS Connect v2.0 Workstation Toolkit
  - You’ll need an Eclipse; install and configure the best for your needs, eg. IBM Explorer, IMS Enterprise Suite Explorer for Development, etc.
  - If using IMS, install & configure IMS Mobile FP. Follow the Techdoc.
3. **Development:** determine your first target system, eg. CICS, IMS, “other”..
  - If CICS (or “other”)....
    1. Setup z/OS for WOLA (API PDS, CICS config, target app, etc.).
    2. Configure server.xml for WOLA.
    3. Setup for data transformation with Liberty updates and run BAQLS2JS. Test with service query calls.
  - If IMS...
    1. Configure server.xml for using IMSMobile, and do Ping test.
    2. Using IMS Explorer’s IMS Mobile support, configure an IMS Gateway (z/OS Connect) connection, and subsequently an IMS Connection profile, IMS interaction, a Transaction, and finally a Service. Test with built-in test tool.
4. **Development:** Compose API and mapping with z/OS Explorer Toolkit.
5. **Deploy:** Build and deploy API archive (AAR).

Tip: Follow [WP102604](#)

## IBM Statement of Direction (see ENUSZP15-0630)

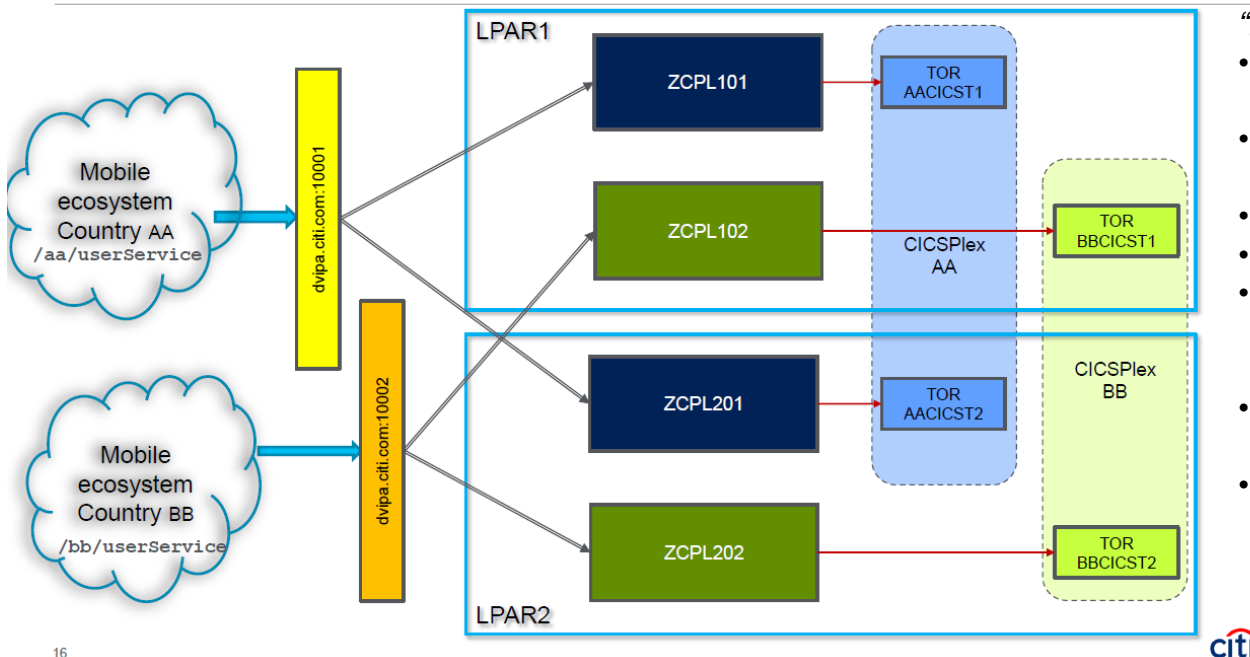
### IBM makes the following statements of general direction:

- IBM intends to deliver IBM z/OS Connect Enterprise Edition (EE) components and technologies through **continuous delivery** of new features in the coming months.
- IBM intends that a future release of IBM CICS Transaction Server for z/OS (CICS TS) will provide support for z/OS Connect EE to enable it to **execute embedded within CICS TS**.
- IBM intends that a future release of **IBM MQ** for z/OS will provide support for both z/OS Connect and z/OS Connect EE. **In beta today**
- IBM intends to update **IBM System Automation** for z/OS V3.5.0 to deliver a new sample policy to allow automated operations and restart of z/OS Connect and z/OS Connect EE.
- ~~IBM intends that a future release of **IBM IMS Enterprise Suite** will provide support for z/OS Connect EE.~~ **Done**
- IBM intends to offer IBM DB2 for z/OS Version 11, or later, with support for the external interface delivered in z/OS Connect EE V2.0, and DB2 RESTful API support that is fully integrated into the DB2 for z/OS Distributed Data Facility.

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 remain at our sole discretion.

- Citi (Large US bank) presented their z/OS Connect environment at InterConnect2016 (session 3190, Feb 2016)
- Phased rollout to production has begun since January 2016
- Reports of 99% offload to zIIP
- Measuring 4-20 ms CPU per transaction, dependent upon message size
- Running ~180 trans/second per Liberty instance

## Deployment Topology Overview

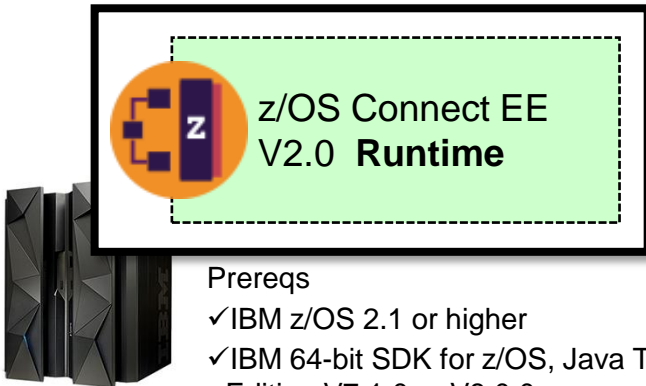


## "Benefits of z/OS Connect + APIM ...

- Rapid development of reusable services
  - Deploy into APIs to a common enterprise catalog
  - High throughput performance
  - Simplify and shorten the path to data
  - Plugs into APIM strategy – Governance, Security, Policy, Discovery
  - Common skill set across platforms and data types
  - Measureable: Chargeback & Analytics"
- Atanas Bogdanovski & Glenn Brown, CITI

# How to Order z/OS Connect EE V2.0

## *Liberty z/OS*



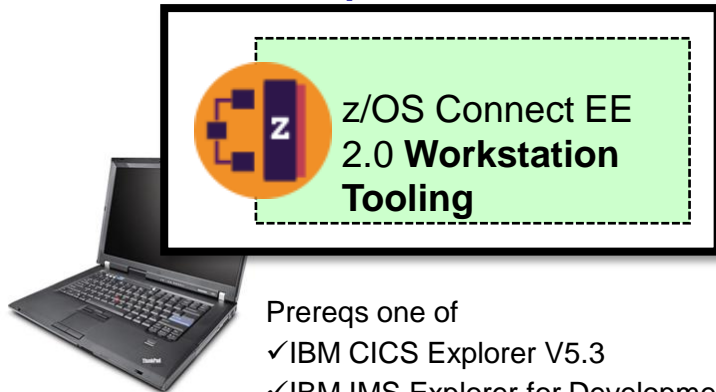
### Prereqs

- ✓ IBM z/OS 2.1 or higher
- ✓ IBM 64-bit SDK for z/OS, Java Technology Edition V7.1.0 or V8.0.0

## **z/OS Connect EE V2.0 Runtime**

- **Product Number:** 5655-CEE
- **Service and Support:** 5655-CES

## *Eclipse*



### Prereqs one of

- ✓ IBM CICS Explorer V5.3
- ✓ IBM IMS Explorer for Development V3.2
- ✓ IBM Explorer for z/OS Aqua V3.0

## **z/OS Connect EE V2.0 Workstation Tooling**

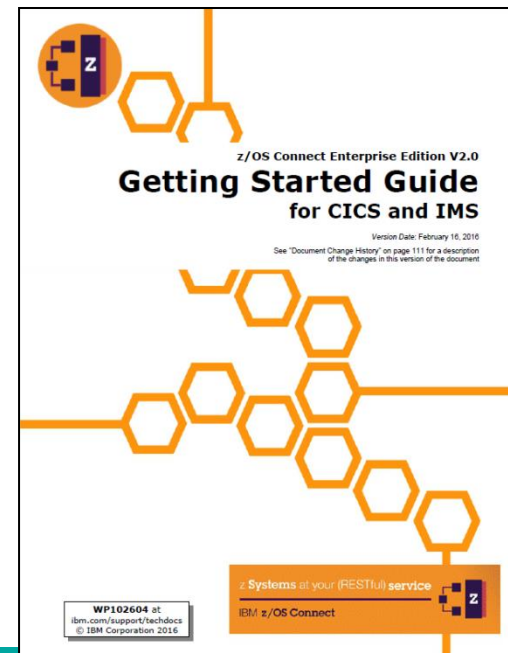
z/OS Connect EE V2.0 workstation tooling is available for download:

<https://developer.ibm.com/mainframe/>

Under the product offerings section, select "z/OS Connect".

# Bibliography

- [z/OS Connect EE V2.0 announcement letter \(ENUS215-493\)](#)
- [z/OS Connect EE V2.0 web home page](#)
- [z/OS Connect EE V2 Getting Started \(WP102604\)](#)
- [z/OS Connect 1.0 Knowledge Center](#)
- [z/OS Connect Enterprise Edition 2.0 Knowledge Center](#)
- [Red paper IBM z Systems Integration Guide for Mobile and API Economy \(REDP-5319\)](#)



## 2016 IBM Systems Technical Universities



## IBM z Systems Technical University – München, Germany June 13-17

### Enterprise IT infrastructure for cognitive business.

Reinventing IT for digital business, the new IBM z13 mainframe and z Systems are built for mobile, ready for and trusted for cloud. IBM z Systems provide the computing infrastructure for the mobile generation and the new app economy. Designed to exploit the mobile transaction explosion, z Systems apply in-transaction analytics and offer the most secure, trusted service delivery — all while transforming the efficiency and economics of IT.

### IBM z Systems lectures and labs will focus on following topics:

- The new IBM z13 and its technology innovations
- IBM z Systems Enterprise Data Compression (zEDC) and Flash Express z13 update and lessons learned, z13 and z/OS dispatching update, and SMT and SIMD
- z/OS Version 2.1 and 2.2 latest updates, migration and advanced functions
- z/OSMF Version 2.1 and 2.2 implementation and configuration
- What's new in Linux on z Systems
- z/VM new features, advanced functions and implementation updates
- What's new in z Systems software pricing on the z13
- How cloud, analytics, mobile, social (CAMS) are remaking the mainframe
- Using Hadoop to analyze z Systems data
- IBM CICS Version 5 planning and implementation ...

**Questions:** [stg\\_conferences@be.ibm.com](mailto:stg_conferences@be.ibm.com)

**Website:** [bit.ly/IBMTechU2016Munich](http://bit.ly/IBMTechU2016Munich)

**Watch. Listen. Learn more about IBM technologies.**