



IBM z/OS Connect Enterprise Edition

The Digital Transformation Roadmap for IBM Z

Al Grega
Global Product Executive
IBM Z Software

Agenda

- 8:30 – Registration
- 9:00 – Introductions
- 9:15 – The Digital Transformation Roadmap for IBM Z
- 10:30 – Tea & Coffee Service
- 12:15 – Lunch
- 13:00 – Securing z/OS Connect APIs
- 14:00 – Labs
- 14:30 – Tea & Coffee Services
- 17:00 – End of Day

/the_api_economy

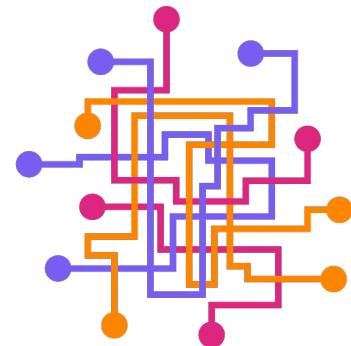
Why your business should care?

The History of APIs



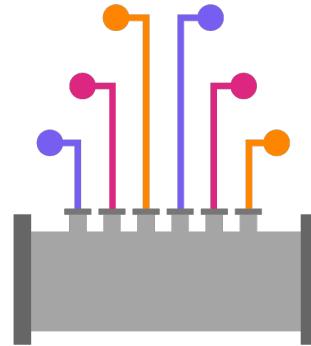
1960 - 1980

Application specific interfaces.



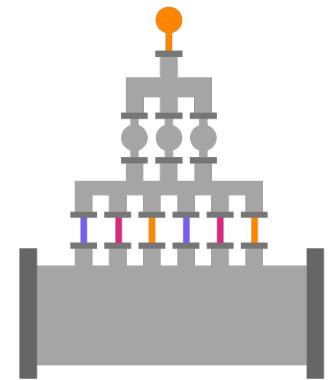
1980 - 1990

Generic interfaces called by many applications.



1990 - 2000

Focus on making it easier to provide and manage interfaces.

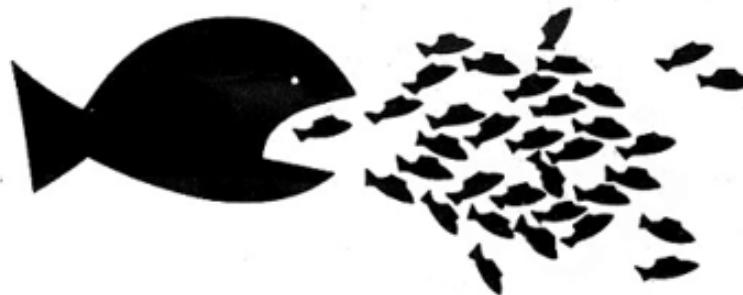


2000 - today

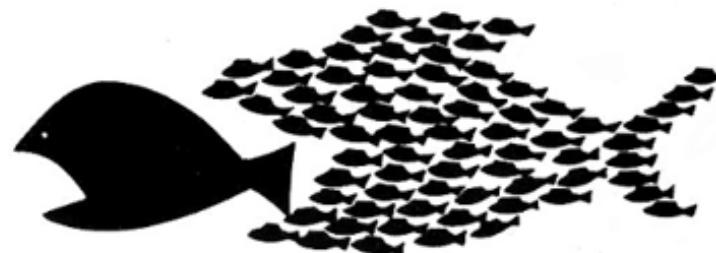
Focus on making it easier to discover, consume and combine interfaces.

Enterprises must learn to “Innovate like a Startup”

FROM.....



.....TO



“The ‘Uber syndrome’ – where a competitor with a completely different business model enters your industry and flattens you.”

Judy Lemke, CIO, Schneider, United States

“52% of the Fortune 500 firms since 2000 are gone.” – R. Ray Wang

<http://blog.softwareinsider.org/2014/02/18/research-summary-sneak-peaks-from-constellations-futurist-framework-and-2014-outlook-on-digital-disruption/>

APIs are being adopted at scale to drive digitalization

- By some estimates, there will be one million APIs before the end of the decade – including cognitive APIs
- Currently, there are 9 million developers working on private APIs. As developers see the increasing opportunities, we may see a significant shift to public API development (where there are only 1.2 million developers)
- By 2020, it is estimated that **26 billion devices** will be using **one trillion applications**

“Producing an API is the only way we are going to ... be the leader of transportation.”

– Head of Developer Relations, Uber

Public APIs are growing exponentially



13 billion
API calls/day



1.4 billion
API calls/day



5 billion
API calls/day

IBM. “Welcome to the Cognitive Era.” <http://www.slideshare.net/ibm/welcome-to-the-cognitive-era>
Murphy, Matt. “The Rise of APIs.” Techcrunch.com. May, 2016. <https://techcrunch.com/2016/05/21/the-rise-of-apis/>
DuVander, Adam. “Which APIs Are Handling Billions of Requests Per Day?” ProgrammableWeb blog. May 23, 2012. <http://www.programmableweb.com/news/which/apis/are/handling/billions/requests/day/2012/>
“Gartner Press Release.” Gartner.com. December, 2013. <http://www.gartner.com/newsroom/id/2636073>

The Transformative Mainframe



Transform revenue-supporting to revenue-generating.

Business Value Highlights

- >6:1 ratio of benefits to costs
- \$194 million per year higher/protected revenue per organization
- 19% lower mainframe cost of operations
- 64% more code releases
- 30% more efficient mainframe management
- 14% lower hardware/licensing costs
- 43% less unplanned downtime



80% of the accounts interviewed are undertaking the use of APIs

In today's economy, there are three primary types of APIs

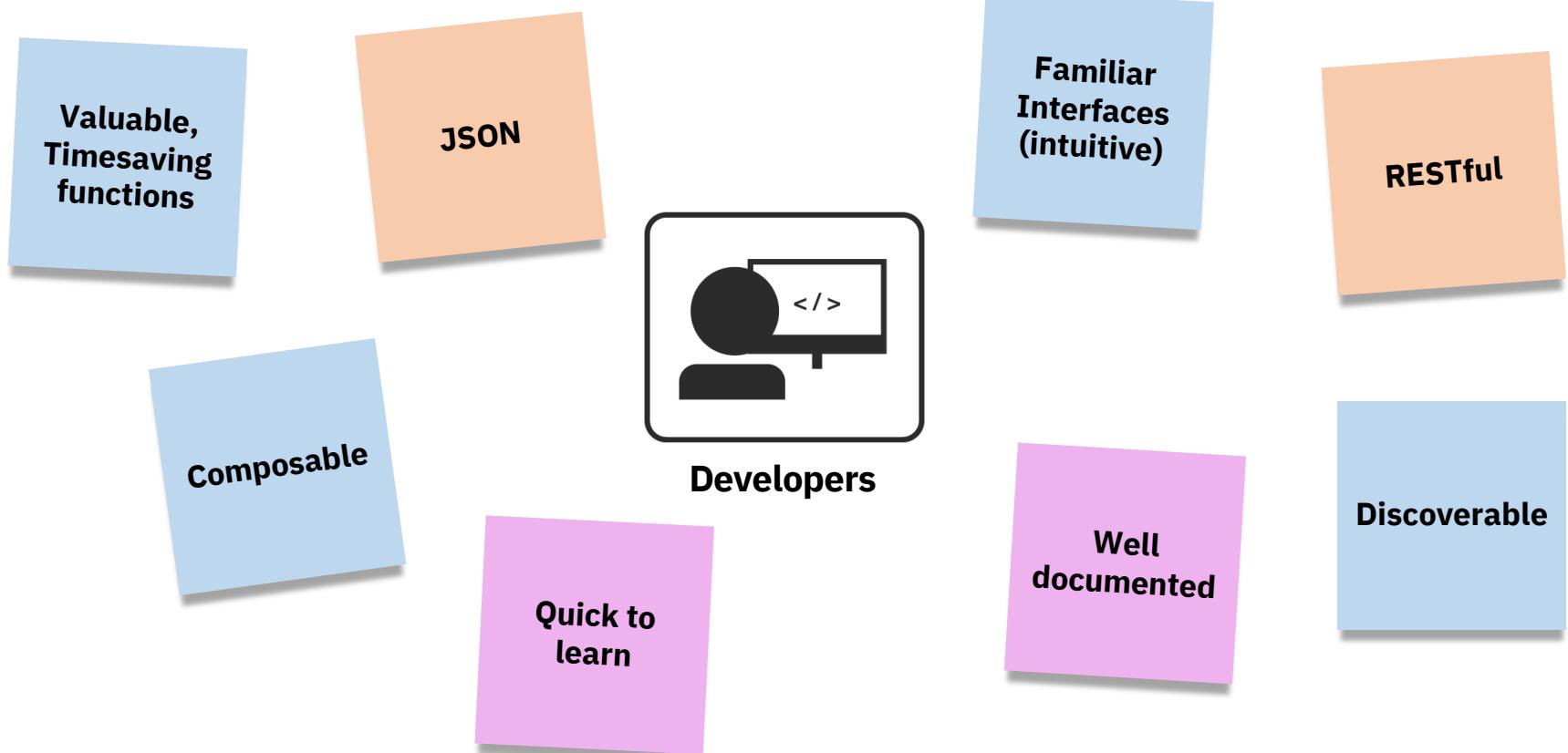
Private	Partner	Public
 APIs are exposed to only existing developers Apps are usually targeted at employees of the enterprise Business drivers: <ul style="list-style-type: none">▪ Channel consistency▪ Productivity through re-use▪ Internal innovation▪ Reduced cost	 APIs are open to select business partners Apps could be targeted at end consumers or business users Business drivers: <ul style="list-style-type: none">▪ Ability to automate processes▪ Exchange data▪ Accelerate partner on-boarding	 Public APIs open to any developer who wants to sign up Apps are more targeted towards end consumers Business drivers: <ul style="list-style-type: none">▪ Foster external innovation▪ Quickly enter new customer-facing ecosystems and tools

Developers are your new customers

APIs are your new products



z/OS Connect EE



/what_is_REST?

What makes an API “RESTful”?

REST is an Architectural Style

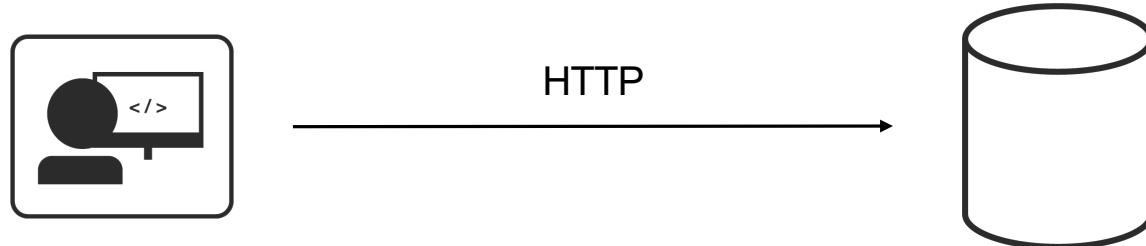


REST stands for **R**epresentational **S**tate **T**ransfer.

An architectural style for **accessing** and **updating** data.

Typically using HTTP... but not all HTTP interfaces are “RESTful”.

Simple and intuitive for the end consumer (**the developer**).



Roy Fielding, PhD, University of California Dissertation (2000):
<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

A REST Service is not the same as a RESTful API

RESTful Services:

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

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

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

It may be “good enough” for some use-cases, but it falls short of what many developers seek when creating Industry Standard REST APIs .

RESTful APIs prescribe to a set of standards established by the OPENAPI Initiative (<http://openapis.org>)

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)

XML vs. JSON

XML: Used largely by SOAP Services

```
<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

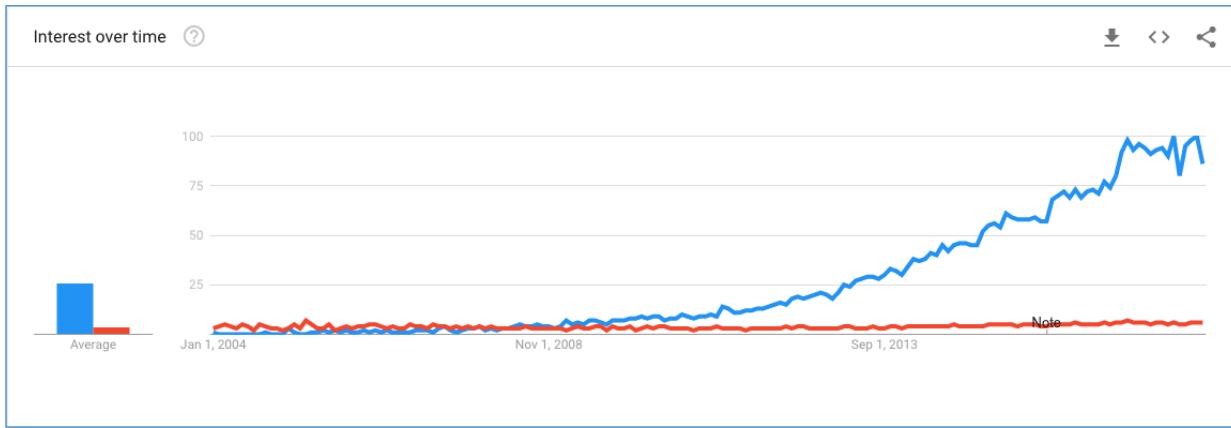
JSON: Used largely by RESTful Services and APIs

```
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!

RESTful APIs vs SOAP Services calls at Google



Blue – REST
Orange = SOAP

Source:

<https://trends.google.com/trends/explore?cat=5&date=all&q=REST%20API,SOAP%20API>



/hosting_apis

How IBM Z can give you the edge

What makes a good API host?



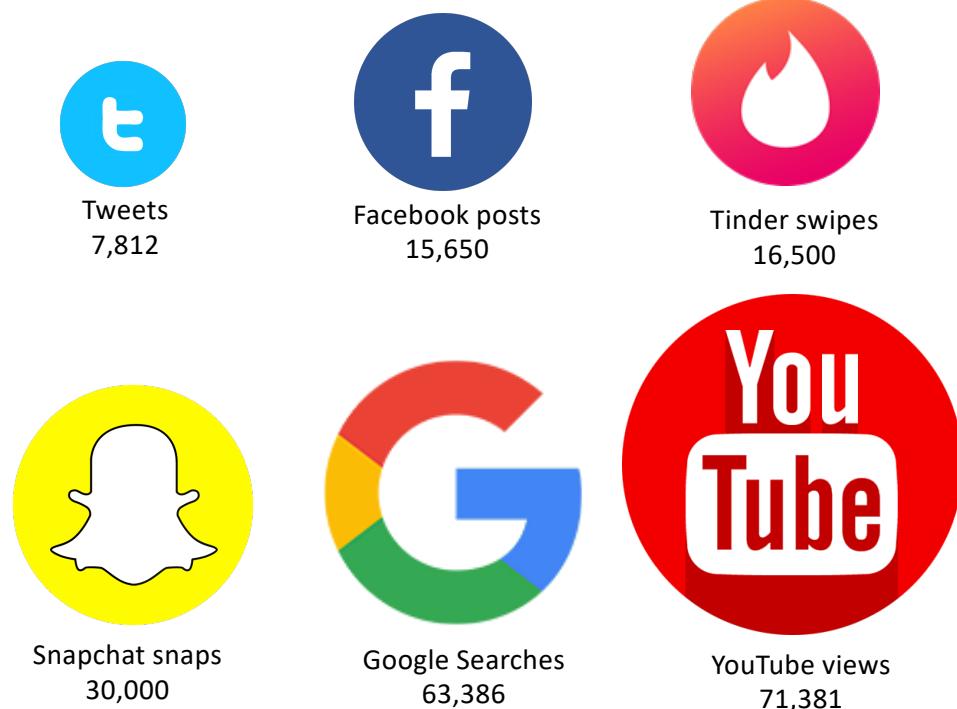
“Fast response time”.

“Extremely reliable”.

“Handles high workloads”.

“Highly available”

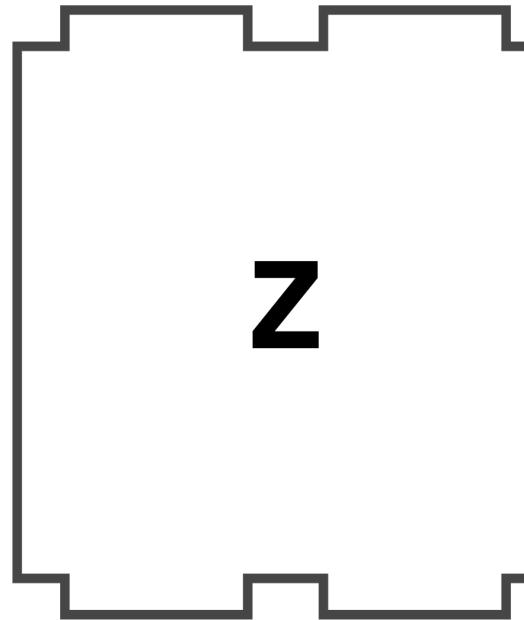
What occurs in a second?



**CICS + IMS Transactions
4,226,574**

<http://www.internetlivestats.com/one-second/#email-band>
<http://uk.businessinsider.com/everything-that-happens-in-one-minute-on-the-internet-2017-9?r=US&IR=T>

What makes a good API host?

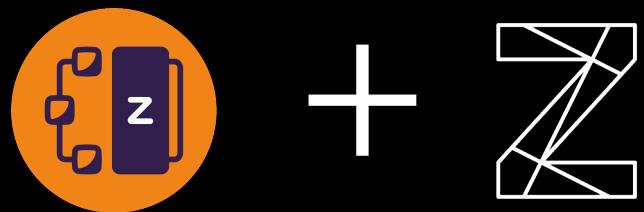


- “Fast response time”.
- “Extremely reliable”.
- “Handles high workloads”.
- “Highly available”.

Z has all of these attributes, and it's where your core assets are.

APIs on IBM Z are now mainstream!

Integration of Z assets via APIs is seen as the most effective first step on a Hybrid Journey
Market analysis states 1000s of enterprises will need integration and API enablement on Z in
the next few years



60%

of enterprises see APIs
as critical to their
business strategy of
integrating Z into their
Hybrid Architecture

1000s

of IBM Z customers will
need an API
enablement solution in
the next few years

10 of 20

of the world's top banks
already use z/OS
Connect EE, and more
are on the way

100m

API requests are driven
through z/OS Connect
EE each day by our
largest customers

A Client Story

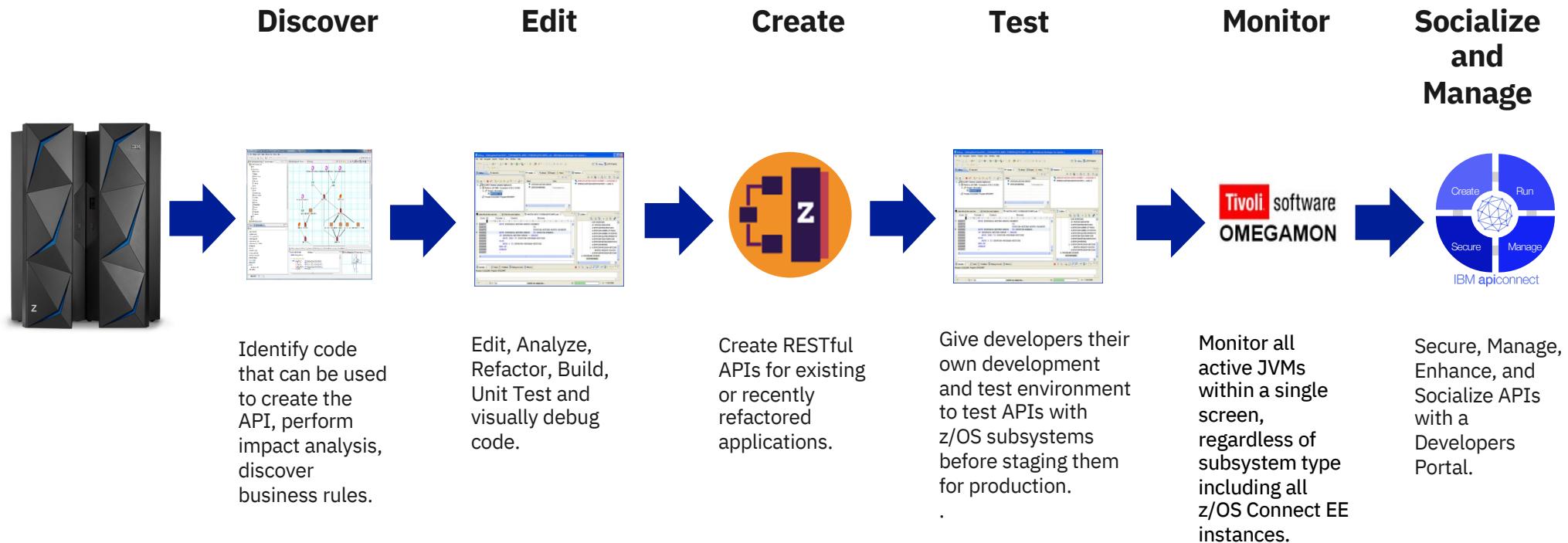
Start with something small and useful

**Start with something small and useful
AND TALK TO YOUR DEVELOPERS!**

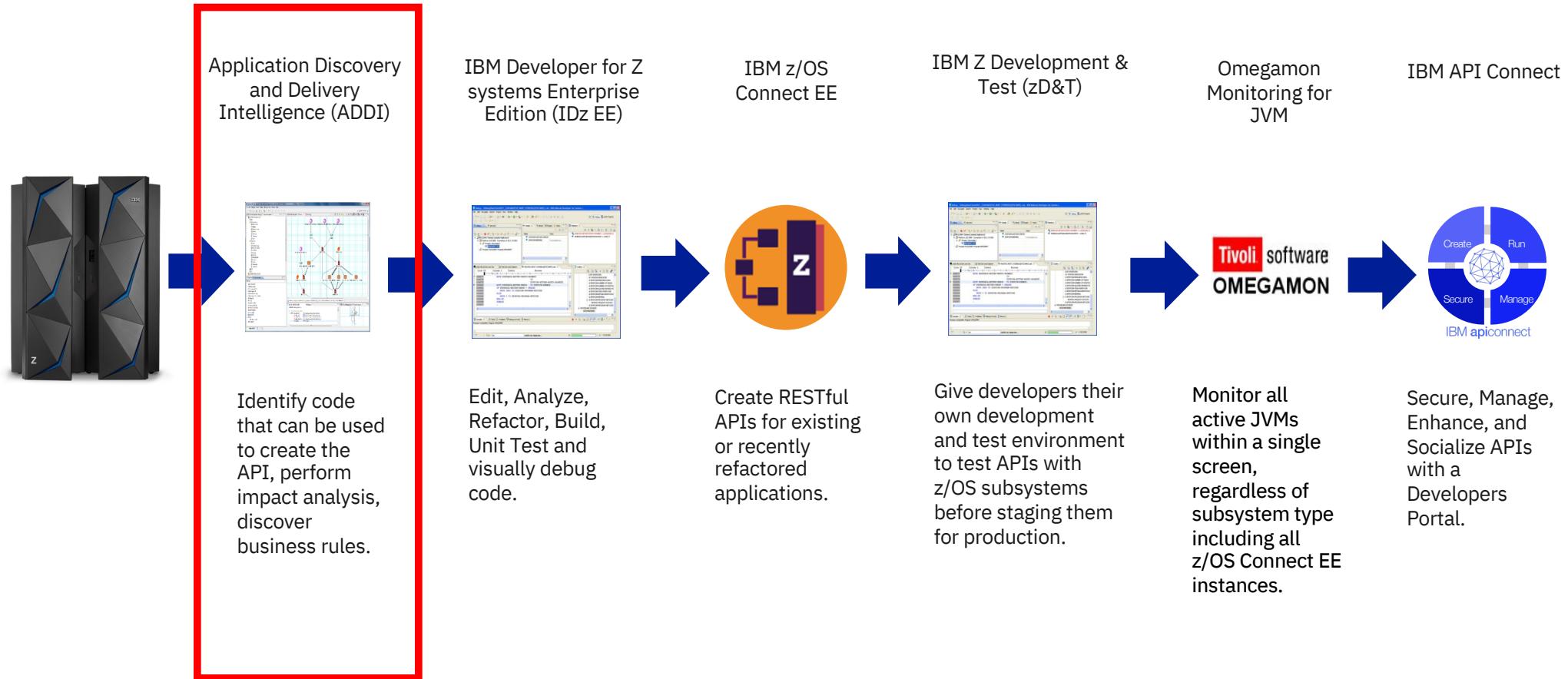
/building_the_solution

What do you need and where does it fit?

This is the Digital Transformation Roadmap for IBM Z



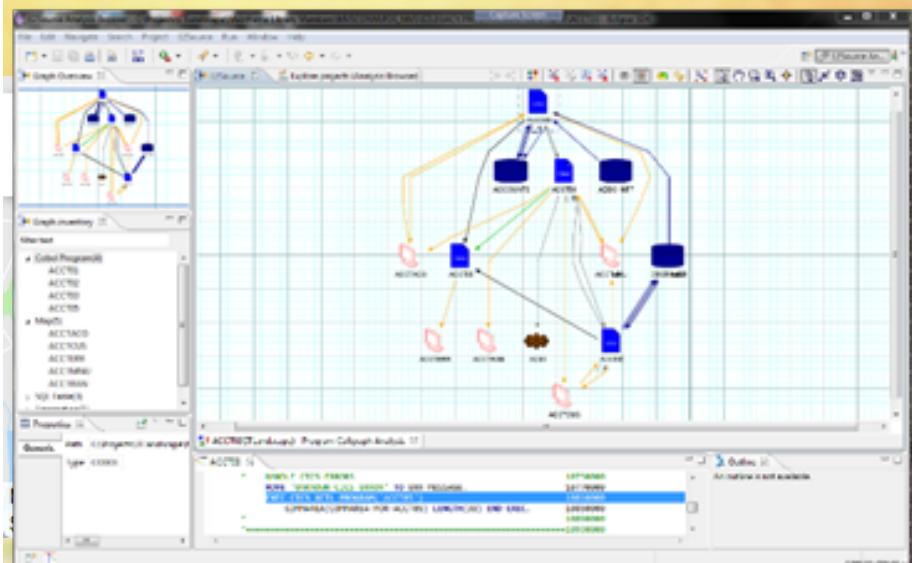
This is the Digital Transformation Roadmap for IBM Z





Trying to find a place 20 years ago...

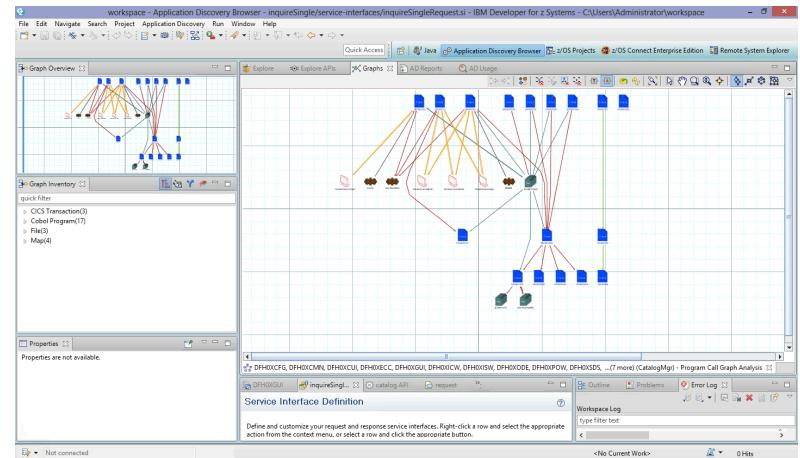
Graphical Analysis

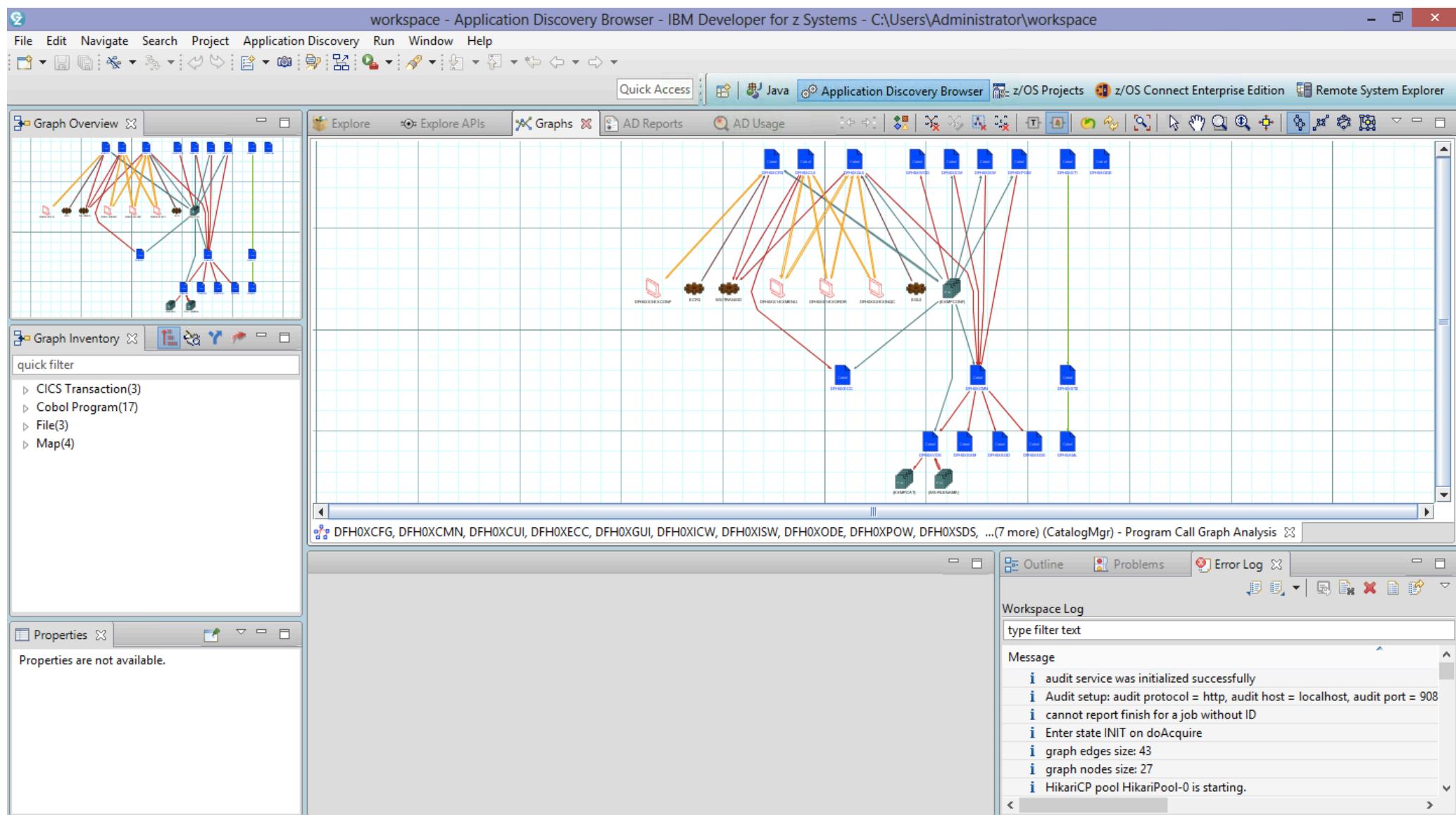


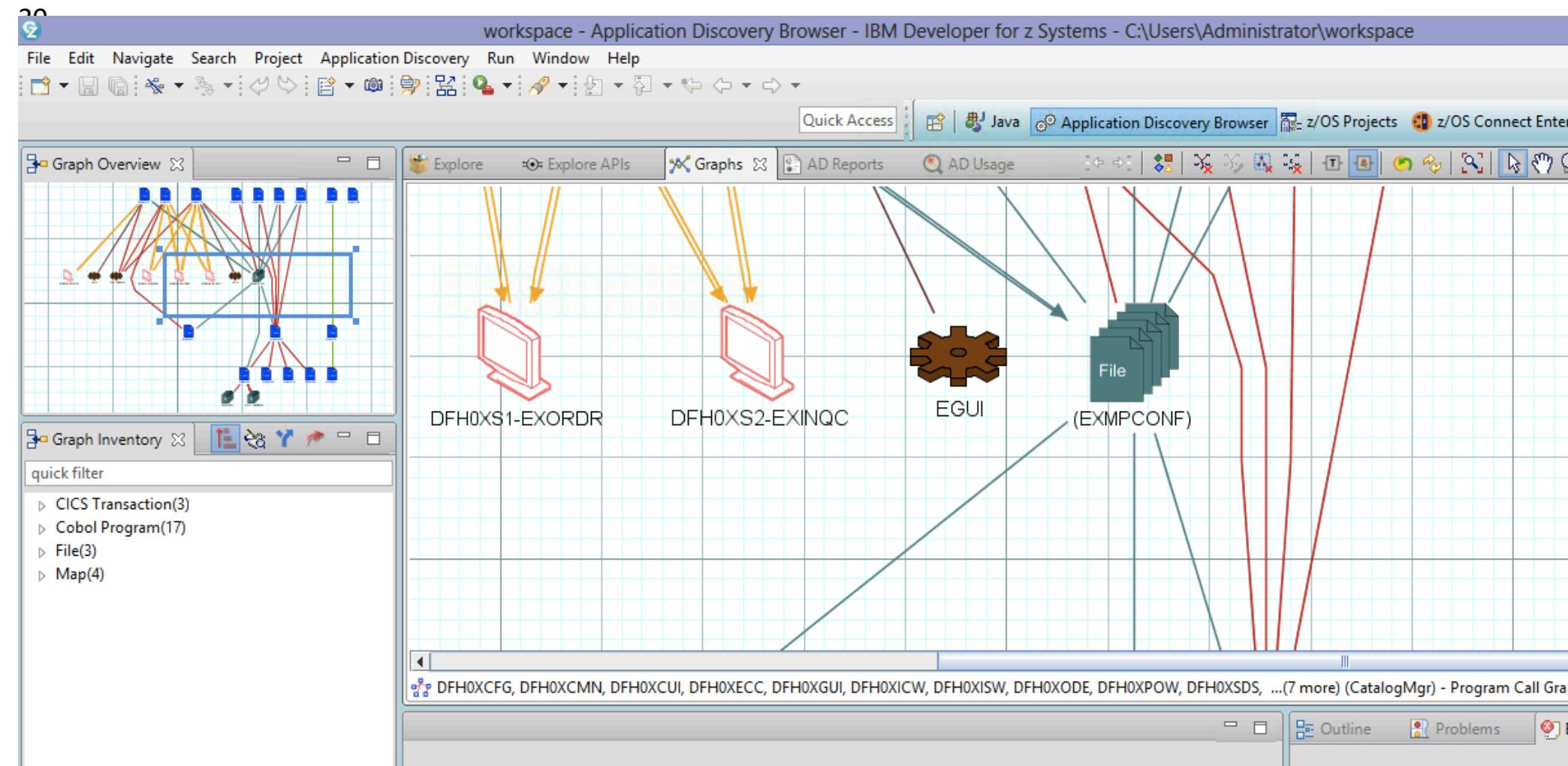
...and today! Why not Application Discovery Tools?

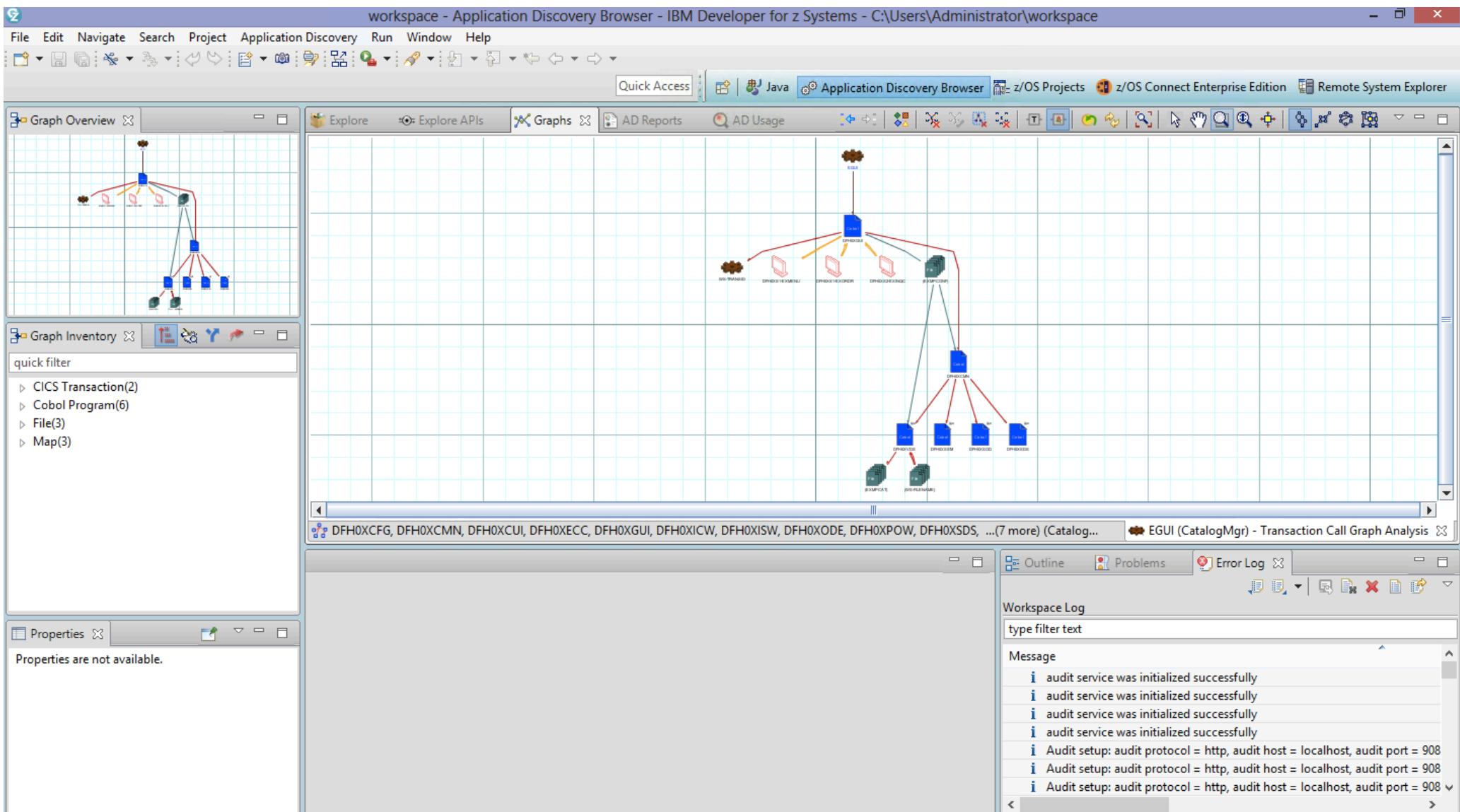
IBM Application Discovery and Delivery Intelligence helps you to....

- Automate the discovery & evaluation of existing application source code
- Perform impact analysis to avoid disruptions to core business applications and APIs
- Drive higher levels of quality through identification of areas for modernization, improvement, and componentization
- Improve the business value of applications
- Collect and correlate key metrics about the overall quality of the application portfolio for trend analysis.
- Reduce analysis time, improve application and test design, and provide enhanced build design
- Discover Business Rules in existing code









IBM Application Discover & Delivery Intelligence

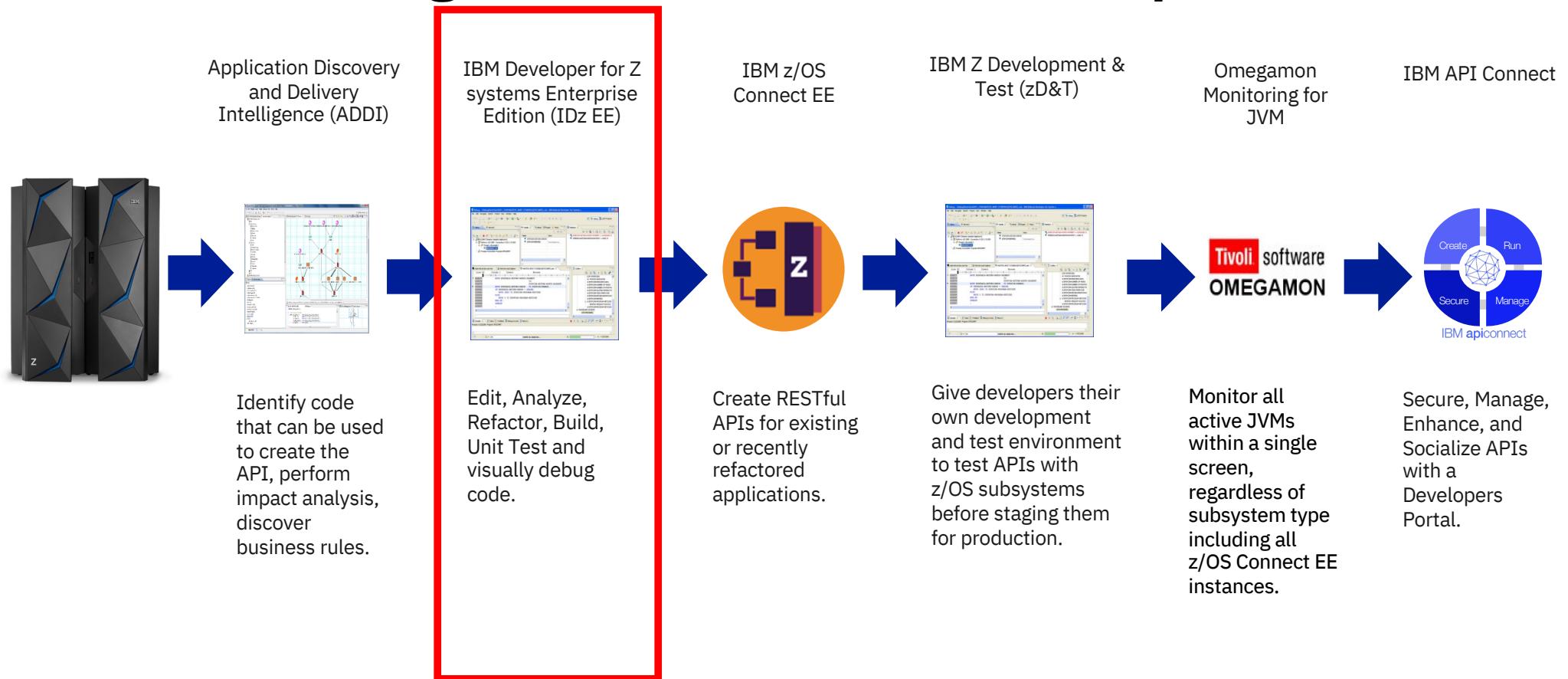


The screenshot shows three code editors side-by-side:

- Top Editor:** EXEC CICS LINK PROGRAM(WS-CATALOGMANAGER-PROG)
COMMAREA (WS-COMMAREA)
DATALENGTH(LINK-COMMAREA-LENGTH)
END-EXEC
- Middle Editor:** LINKAGE SECTION.
01 DFHCOMMAREA.
COPY DFHOXCP1.
- Bottom Editor:** * Uppercase the value passed in the Request Id field
MOVE FUNCTION UPPER-CASE(CA-REQUEST-ID) TO CA-REQUEST-
EVALUATE CA-REQUEST-ID
WHEN '01INQC'
Call routine to read catalog for inquire
PERFORM CATALOG-INQUIRE
WHEN '01INQS'
Call routine to perform for inquire for single item
PERFORM CATALOG-INQUIRE-SINGLE
WHEN '01ORDR'

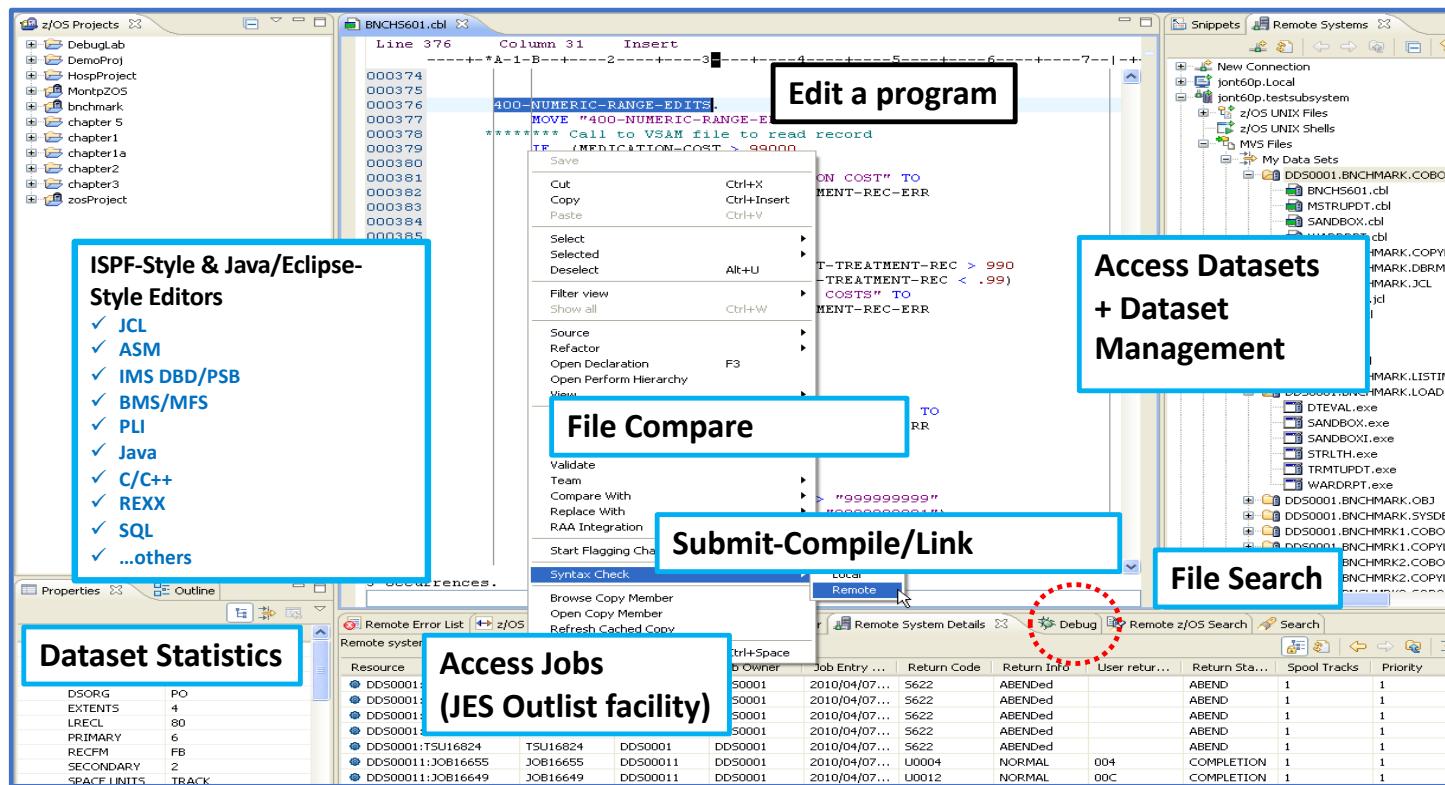
The code in the middle editor's COPY statement and the bottom editor's EVALUATE block are highlighted with red boxes.

This is the Digital Transformation Roadmap for IBM Z

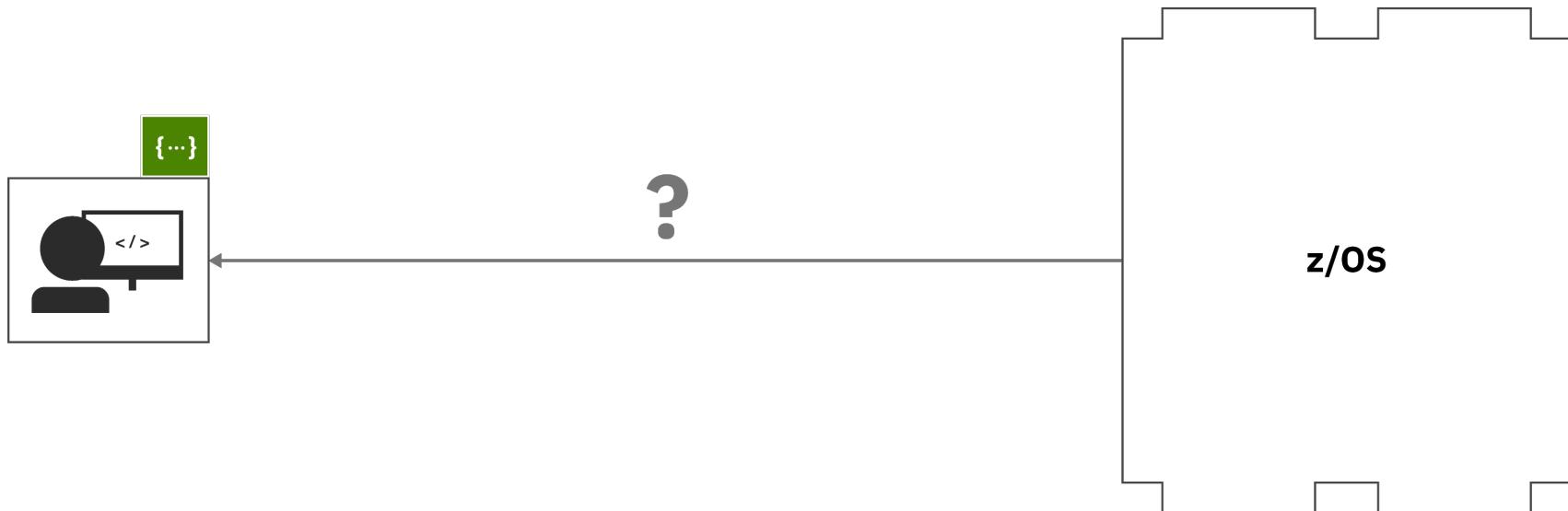


Using IDz EE for Development & Testing

Instead of maneuvering across ISPF panels and working *sequentially*, in IDz the functionality you need is always in-focus ... you work concurrently



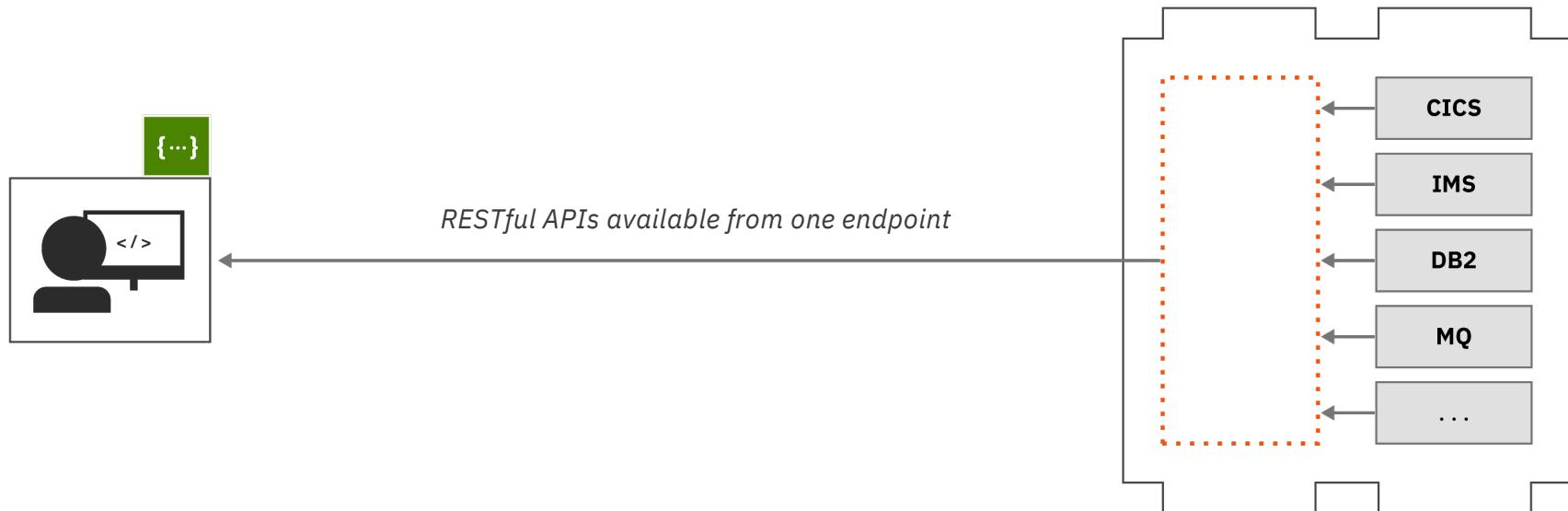
How do we expose z data and services as RESTful APIs?



You need a single entry point!

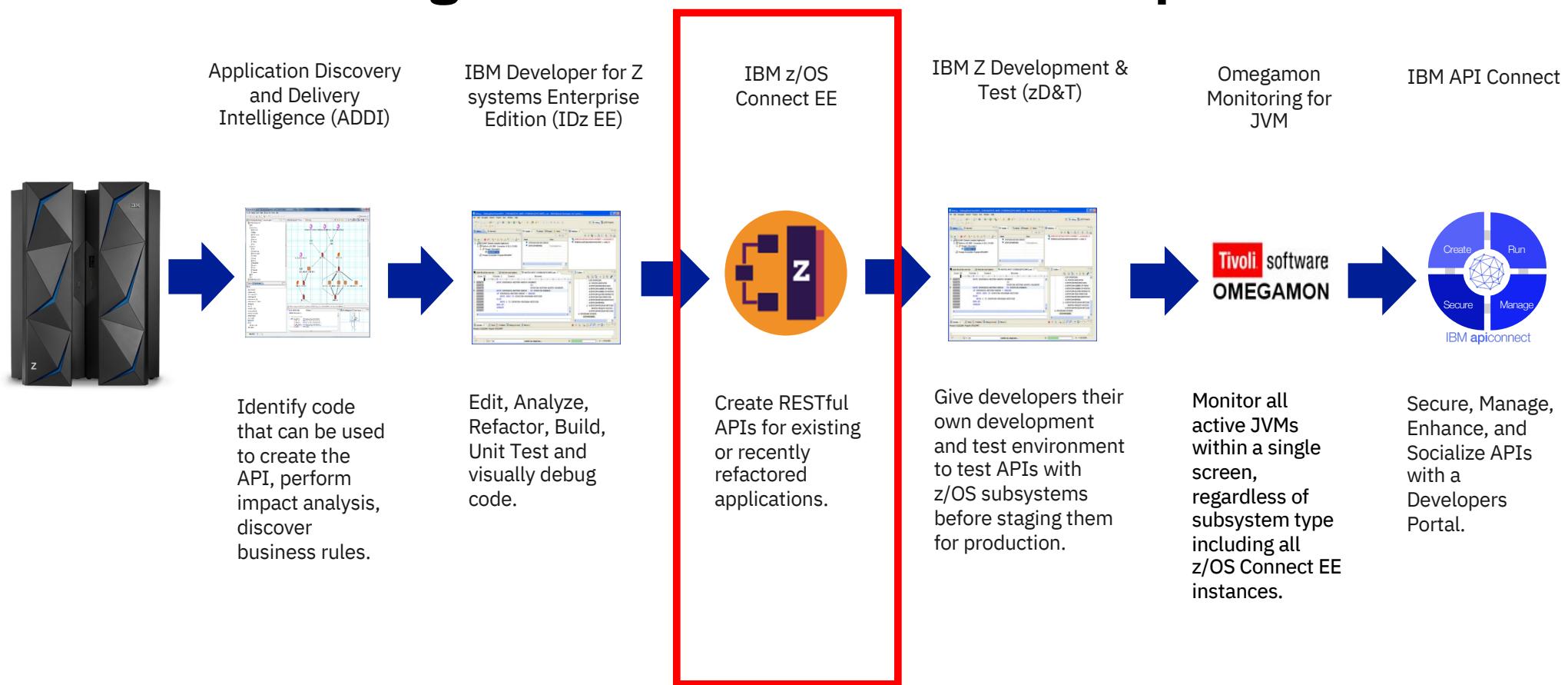


z/OS Connect EE



With sophisticated mapping of truly RESTful APIs to existing mainframe and services data.

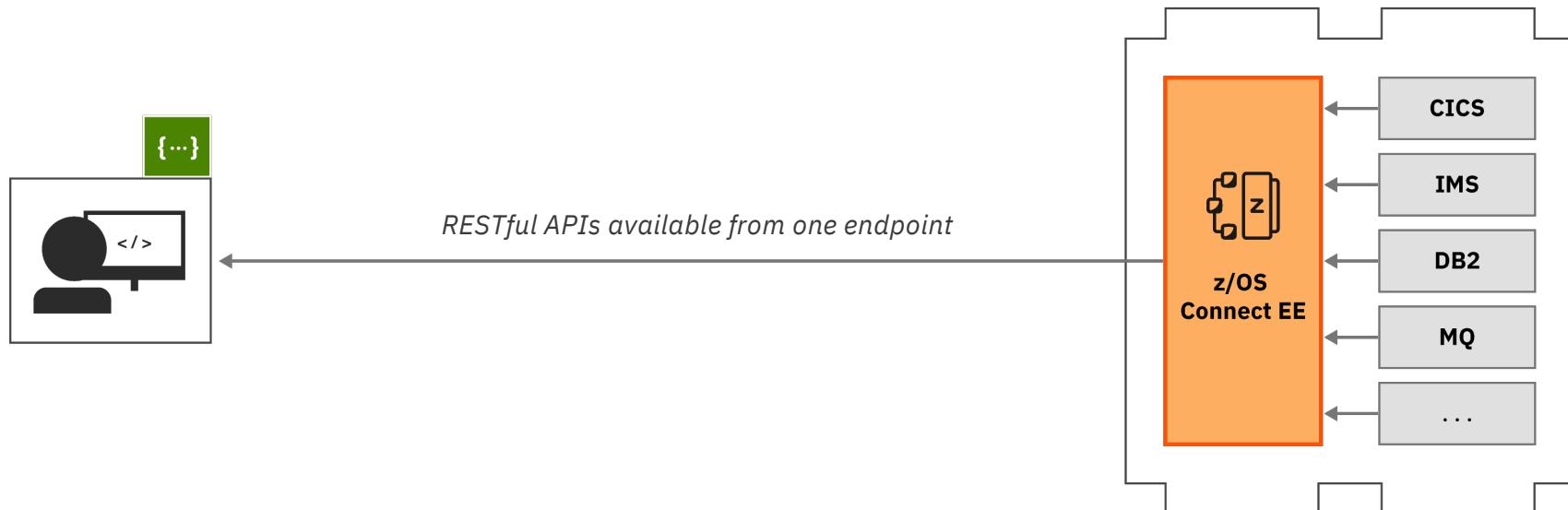
This is the Digital Transformation Roadmap for IBM Z



You need a single entry point!



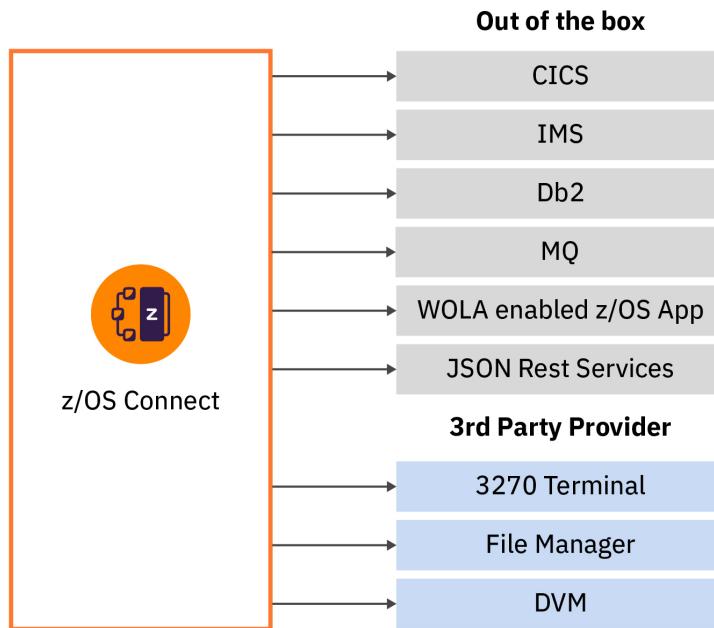
z/OS Connect EE



z/OS Connect EE runs as an address space on z/OS

With sophisticated mapping of truly RESTful APIs to existing mainframe and services data.

What other assets can z/OS Connect EE map to?



z/OS Connect EE is **pluggable** and **extensible**.
It provides API access to a wide range of z/OS assets.



/zos_connect_ee

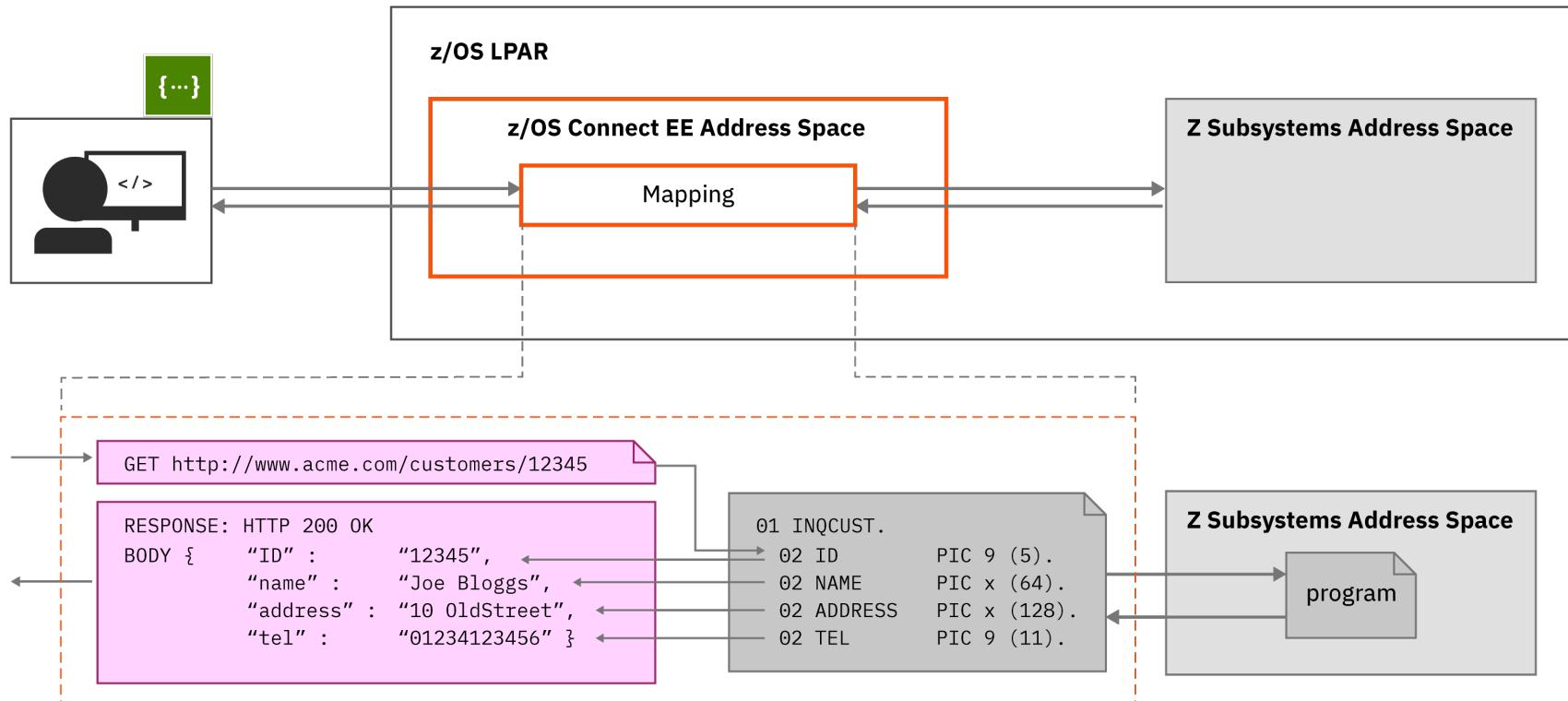
Truly RESTful APIs to and from your mainframe.

Data mapping



z/OS Connect EE

A closer look

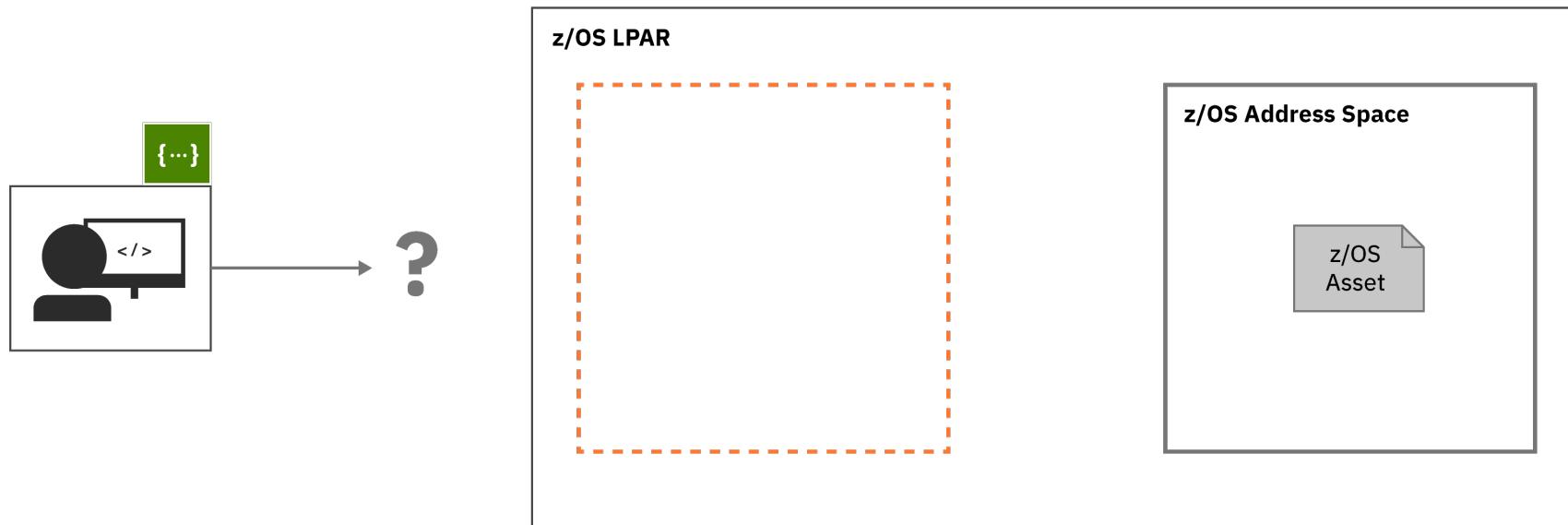


Six Steps to expose a z/OS Asset



z/OS Connect EE

Starting point

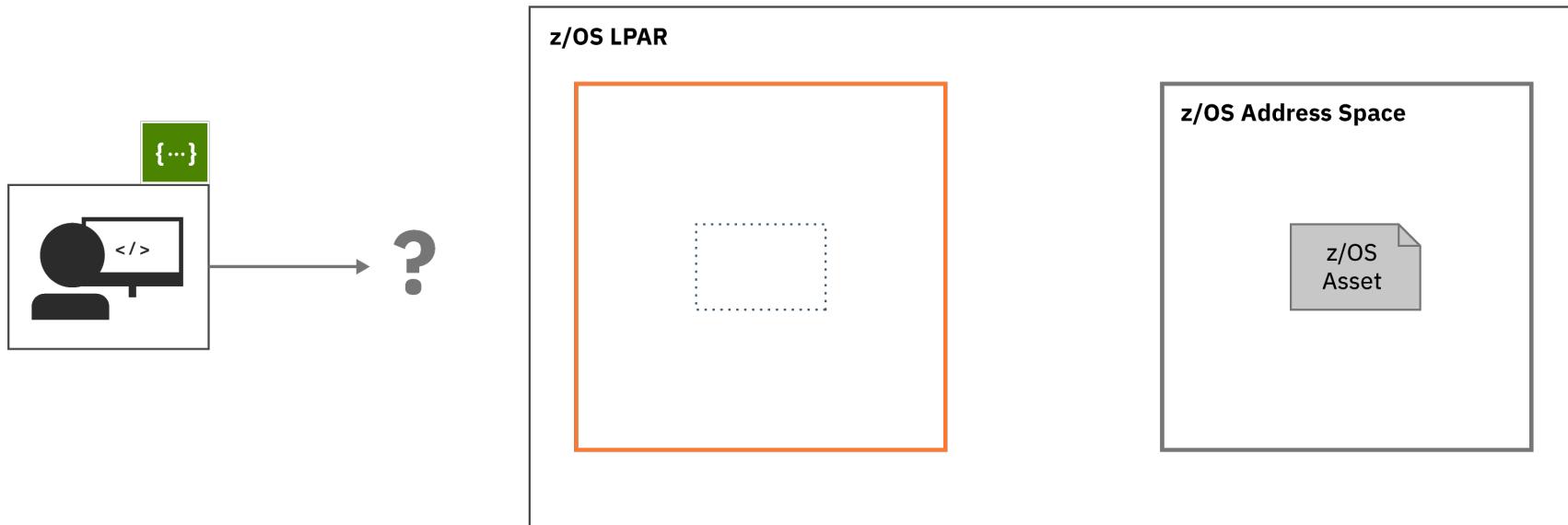


You've chosen a z/OS asset you want to expose as a RESTful endpoint.

Six Steps to expose a z/OS Asset



1. Install z/OS Connect EE (one time setup)



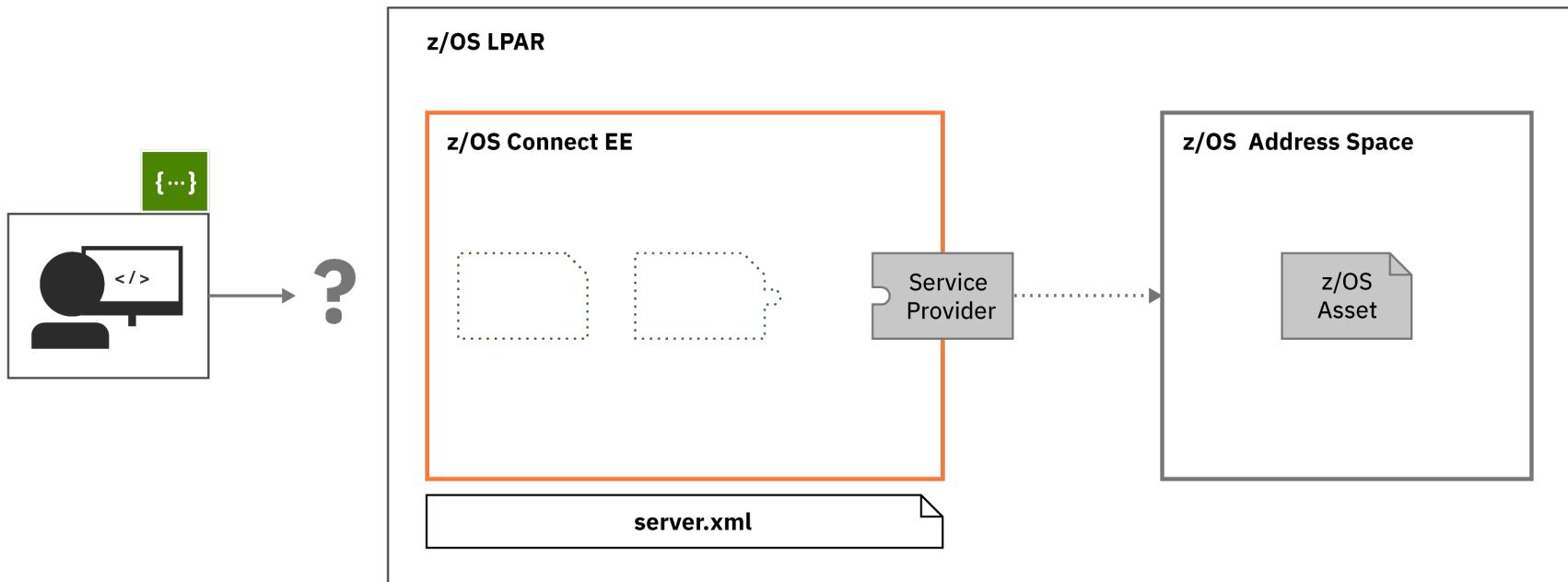
SMP/E Install, set up, and start your new z/OS Connect EE Server.

Six Steps to expose a z/OS asset



z/OS Connect EE

2. Configure your service provider (once per address space)



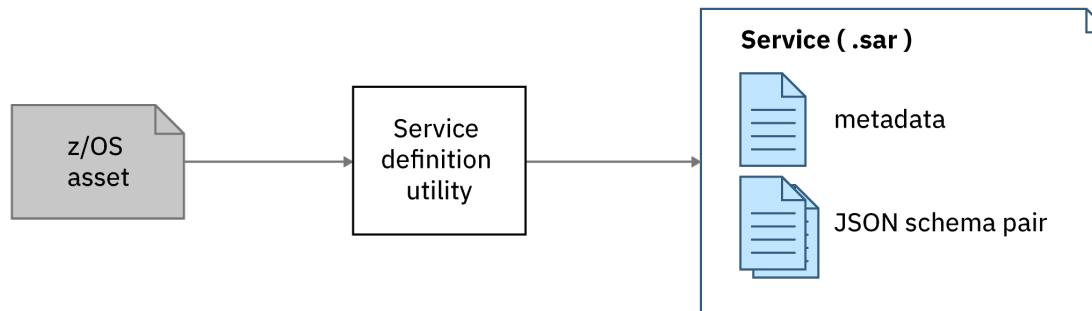
Configure the system-appropriate service provider to connect to your backend system in your `server.xml`.

Six Steps to expose a z/OS application



3. Create your service definition

To start mapping an API, z/OS Connect EE needs a representation of the underlying z/OS application: a Service Archive file (.sar).



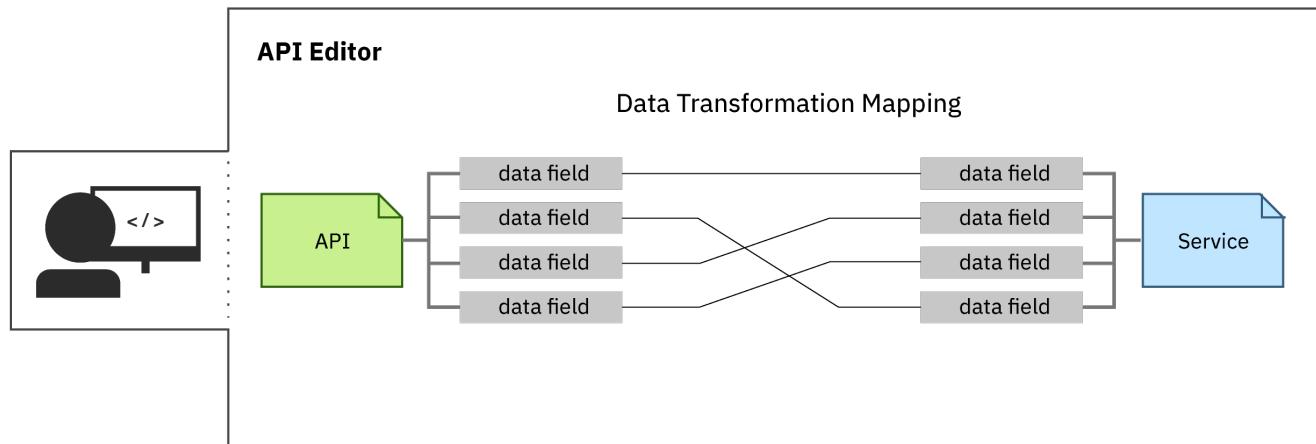
Use a system-appropriate utility to generate a .sar file for the z/OS application.

Six Steps to expose a z/OS asset



z/OS Connect EE

4. Create your API



Import your `.sar` file into the API toolkit, and start designing your API.

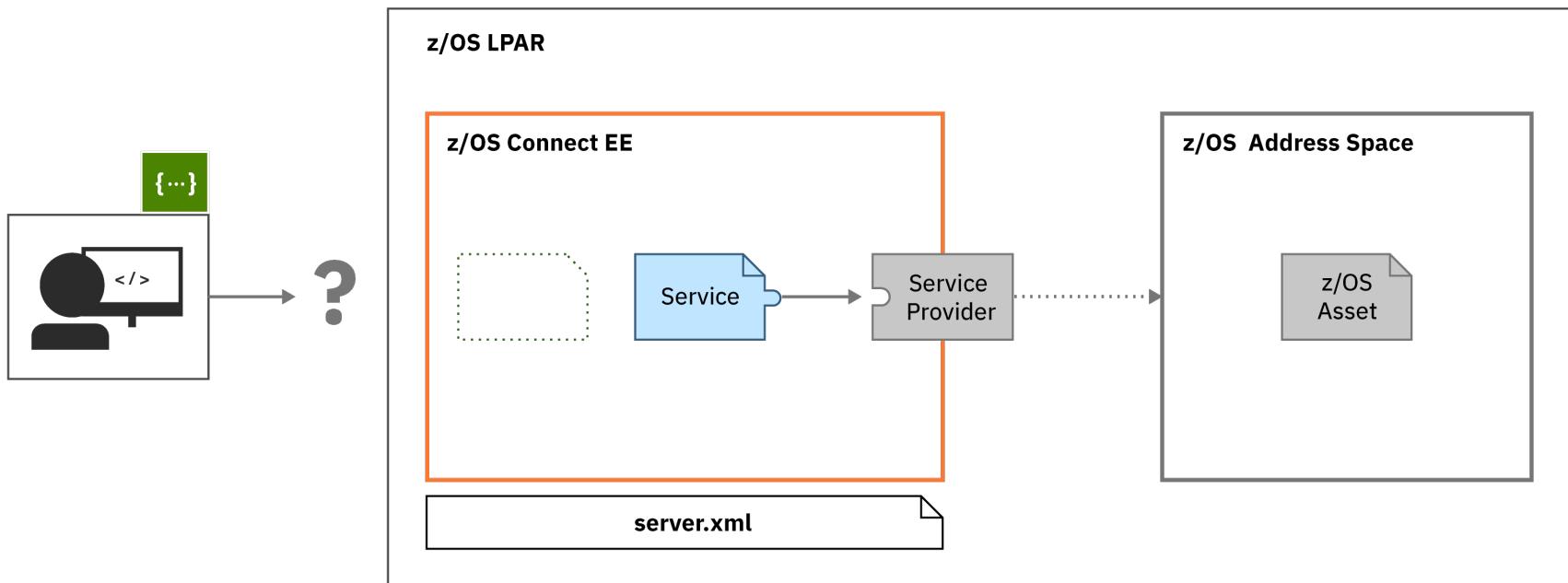
From the editor, create an API Archive file (`.aar`), which describes your API and how it maps to underlying services.

Six Steps to expose a z/OS asset



z/OS Connect EE

5. Deploy your service



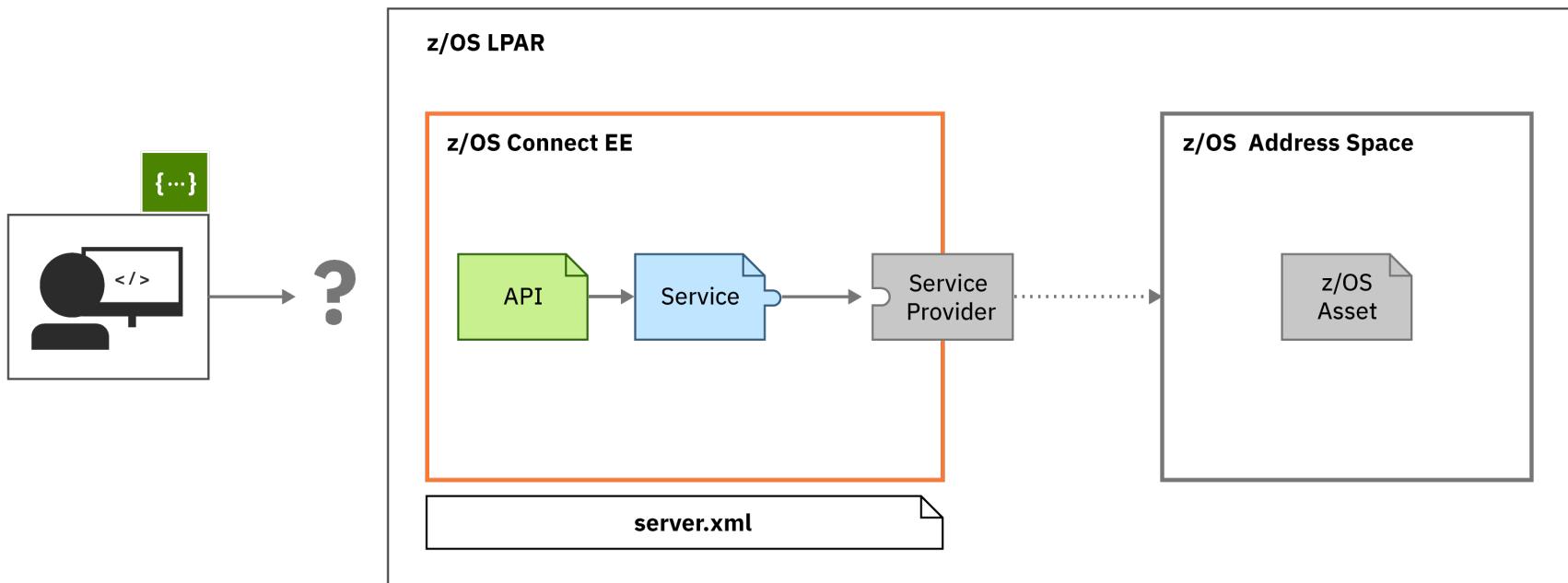
Deploy the `.sar` file generated by the service definition utility by copying the `.sar` file to the services directory. (This step uses the `.sar` file generated in Step 2.)

Six Steps to expose a z/OS asset



z/OS Connect EE

6. Deploy your API

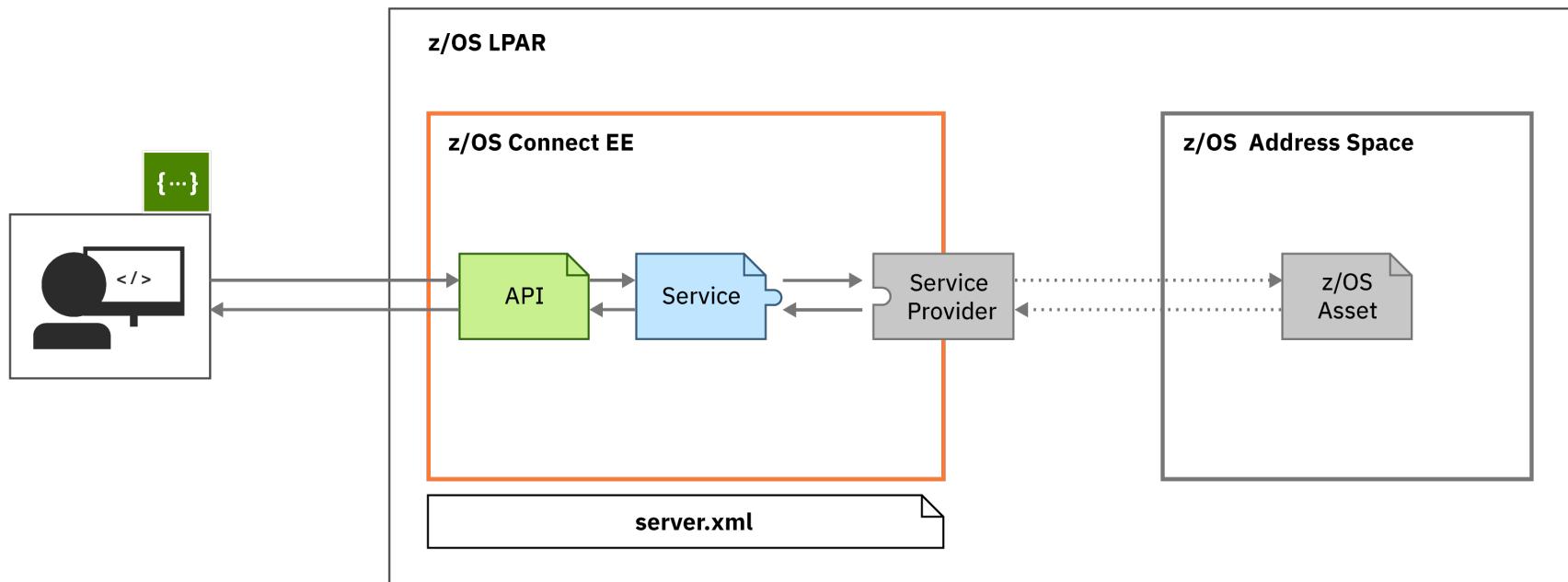


Deploy your API using the right-click deploy in the API toolkit,
or by copying the `.aar` file to the `apis` directory.

Six Steps to expose a z/OS asset



Done



Your API is ready to be consumed: go tell your developers!

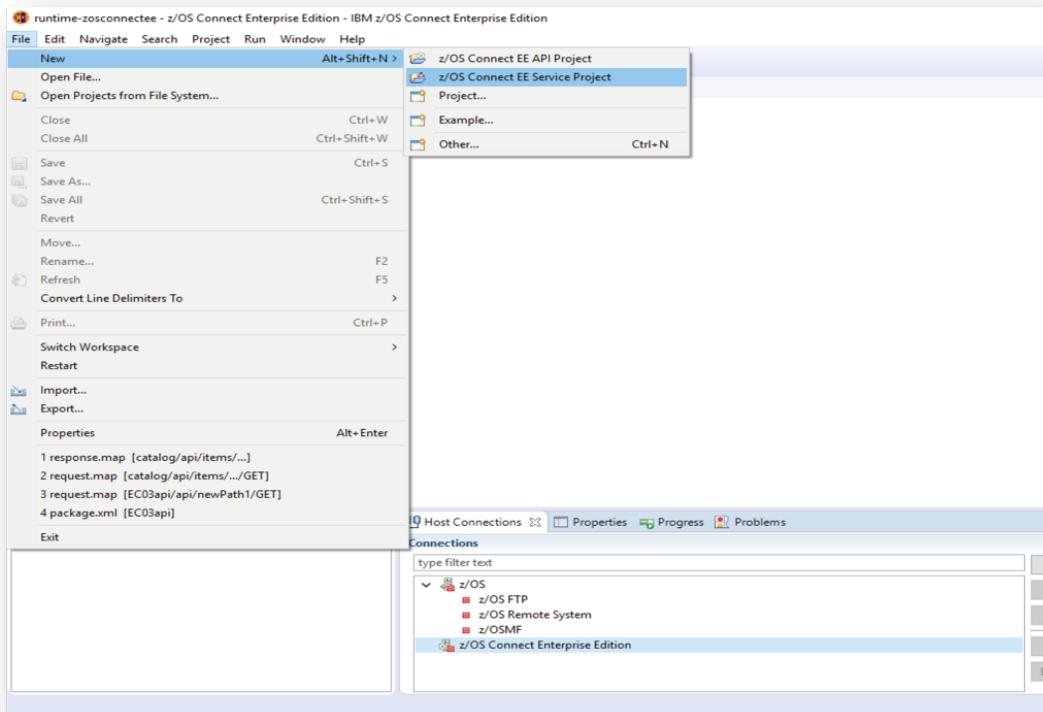


/api_toolkit/part1

Simple **service creation.**

API toolkit

Creating a service

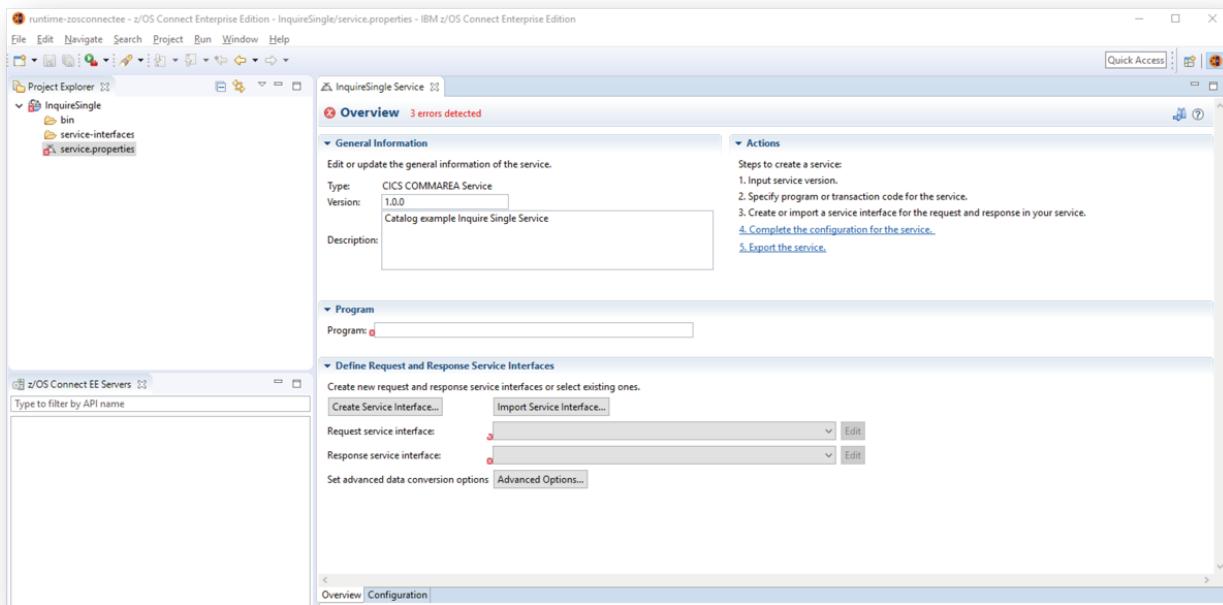


Use the API toolkit to create services through Eclipse-based tooling.

Services are described as Projects, so can be easily managed in source control.

API toolkit

Creating a service



The service project overview guides you through the steps for creating a service.

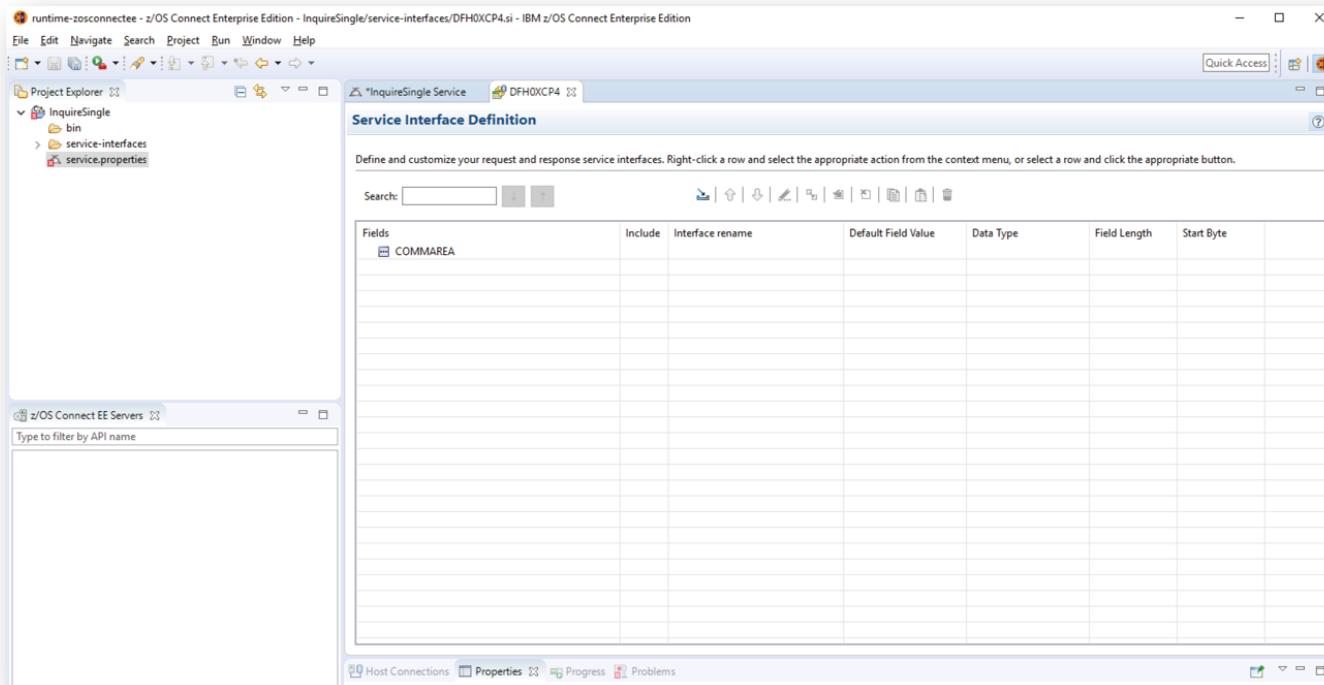
Here, you specify your target asset, such as a program, and assign service interfaces to describe its request and response.

API toolkit

Creating a service



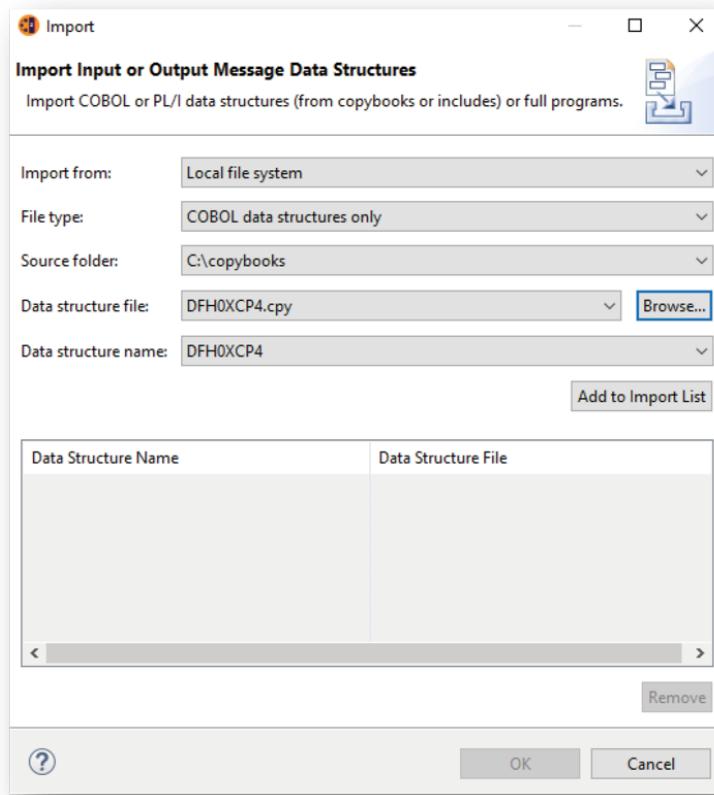
z/OS Connect EE



A service interface allows you to simplify what is exposed from your service to the API developer.

API toolkit

Creating a service



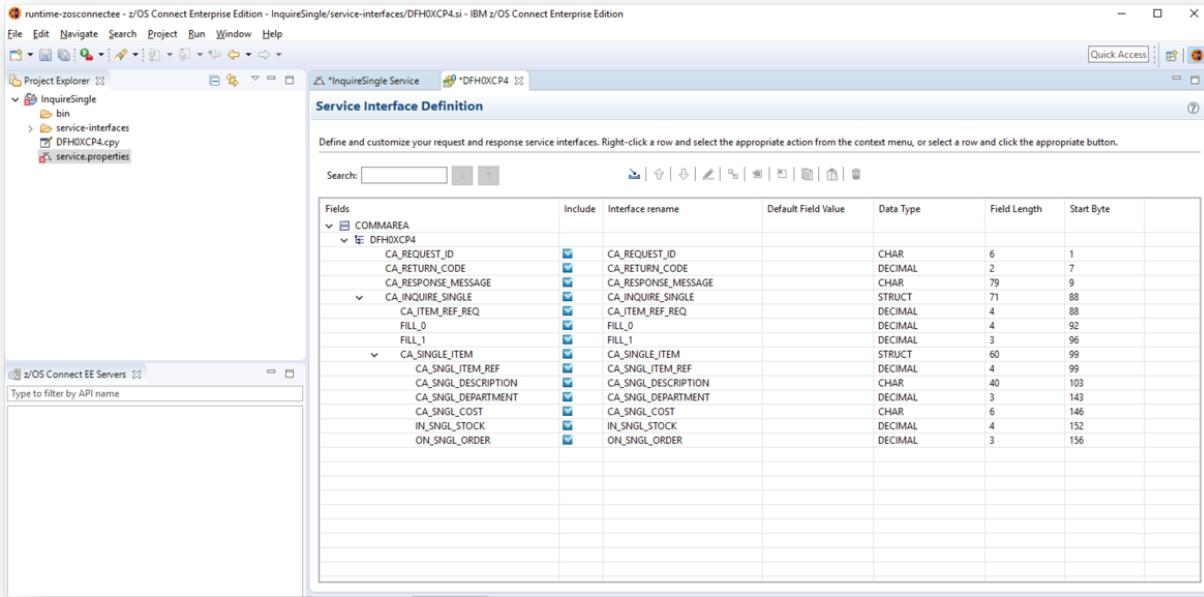
You start by importing data structures into the service interface from the local file system or the workspace.

Several restrictions apply to the supported data structures and the import function, see [here](#) for details.



API toolkit

Creating a service



z/OS Connect EE

You can then see the imported data structure and can redact fields, rename fields, and add descriptions to fields to make the service more consumable for an API developer.

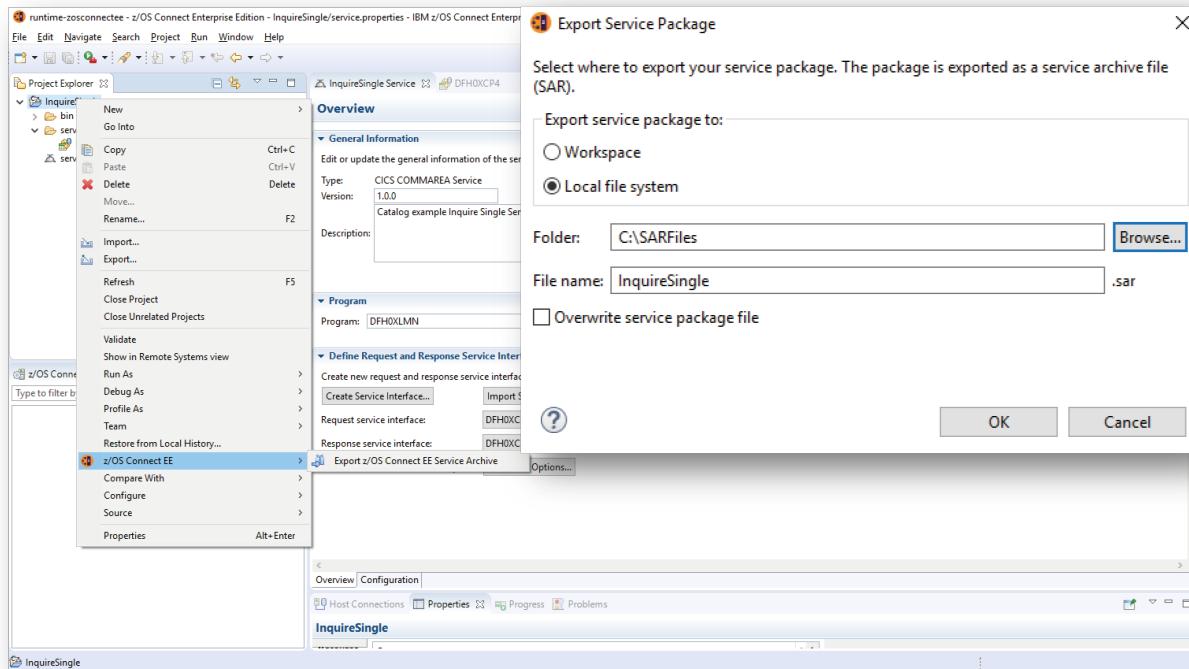
The service interface supports complex data structures, including OCCURS DEPENDING ON and REDEFINES clauses.

API toolkit

Creating a service



z/OS Connect EE



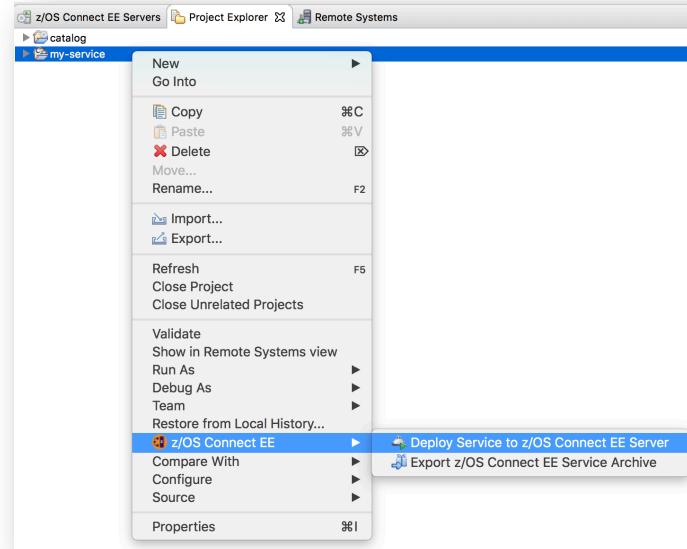
Finally, you can export the service project as a Service Archive file (.sar).

API toolkit

Server connection and Service deployment

Manage z/OS Connect EE server connections in the Host Connections view.

Right-click deploy to server enables developers to quickly deploy their services.



z/OS Connect EE Servers view allows you to start, stop, and update services on a running server.



/api_toolkit/part2

Quick and easy **API mapping**.

API toolkit

API definition

The screenshot shows the z/OS Connect EE API Editor interface. At the top, there's a header bar with the title 'catalog API' and a close button. Below it, the main title is 'z/OS Connect EE API Editor'. A section titled 'Describe your API' contains fields for 'Name' (catalog), 'Base path' (/catalogManager), and 'Version' (1.0.0). The 'Description' field contains the text 'APIs for browsing, inquiring and ordering items from a catalog'. The main workspace is divided into two sections: 'Path' and 'Methods'. The 'Path' section for the root path '/items?startItem' shows two methods: 'GET inquireCatalog' and 'POST placeOrder'. Both methods have 'Service...' and 'Mapping...' buttons followed by up/down arrows and a delete icon. The 'Methods' section for the path '/items/{itemID}' shows three methods: 'GET inquireSingle', 'PUT inquireSingle', and 'DELETE inquireSingle'. These also have 'Service...', 'Mapping...', and up/down arrow buttons.

The API toolkit is designed to encourage RESTful API design.

Once you define your API, you can map backend services to each request.

Your services are represented by .sar files, which you import into the API toolkit.

API toolkit

API definition with multiple response codes

The screenshot shows the API toolkit interface for defining API operations. At the top, there's a header with a z/OS Connect EE logo and the text "z/OS Connect EE". Below the header, the main interface shows a list of methods for an "IVTNOService": POST, GET, PUT, and DELETE. The "GET" method is expanded to show its details. The "Operation id:" is set to "getContact". Under "Responses (3)", three response codes are defined: 200 OK, 404 Not Found, and 400 Bad Request. The 400 Bad Request entry is selected, and a modal dialog titled "Edit Response 404" is open. In the dialog, the "Response code" is set to "404 - Not Found" and the "Description" is "Not Found". Below these fields, there's a section for defining rules. Two rules are listed: "Rule 1" (response/status starts with SPECIFIED PERSON) and "Rule 2" (response/status ends with NOT FOUND). A logical operator "AND" is selected between the two rules. At the bottom of the dialog are "OK" and "Cancel" buttons.

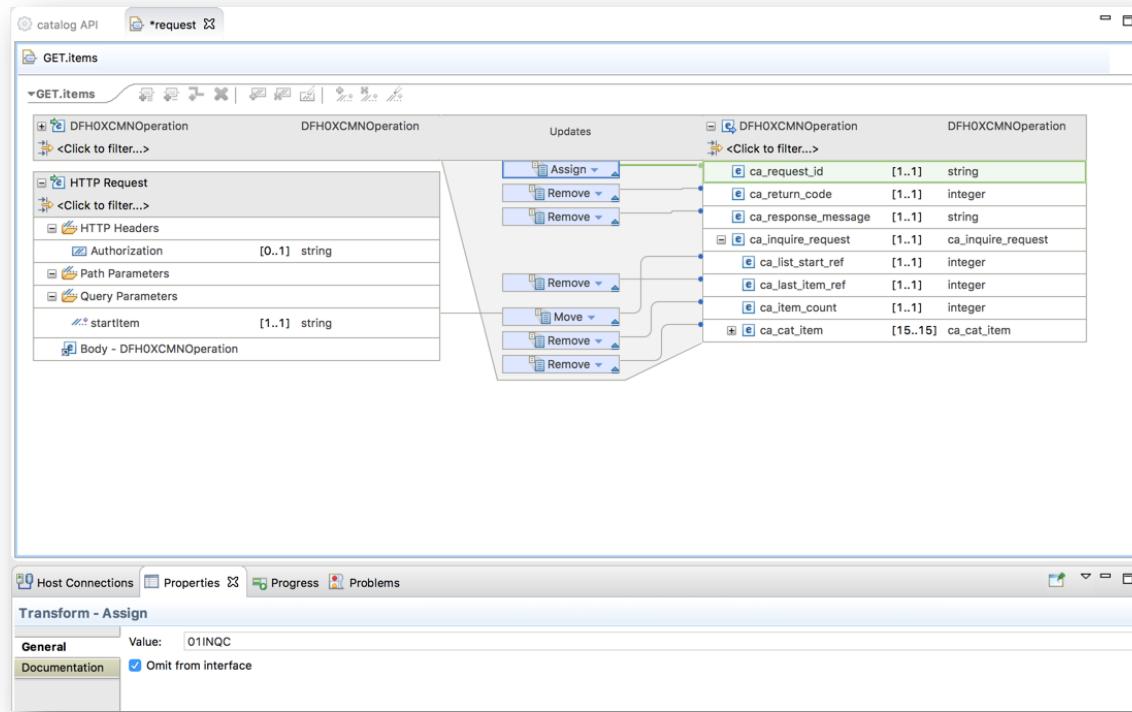
The API toolkit supports defining multiple response codes per API operation.

Separate mappings can be defined for each response code.

You can define rules based on fields in the service's return interface to tell z/OS Connect EE which response code to return

API toolkit

API mapping: Point-and-click interface



z/OS Connect EE

Map both the request and response for each API.

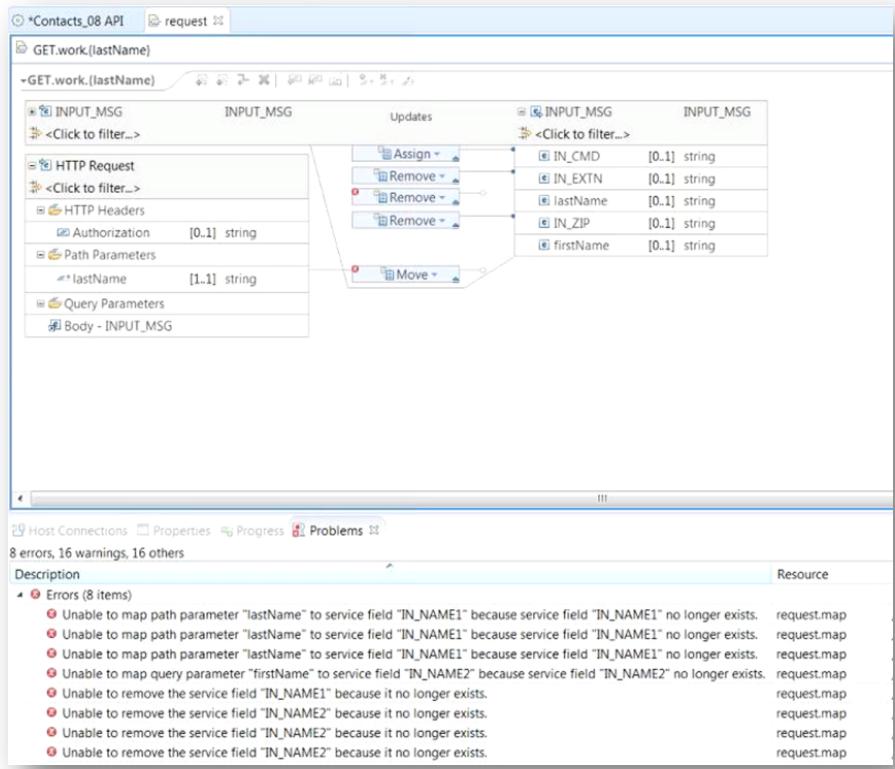
Map path and query parameters to native data structures.

Assign static values to fields, useful for Op codes.

Remove unwanted fields to simplify the API.

API toolkit

Change management



Mappings and Response Code Rules are preserved when .sar files change (for example, when a copybook is updated).

Impact analysis shows you what will break before you import the new .sar file.

Broken mappings and Response Code Rules are highlighted in the Eclipse Problems view and in the mapping editor.

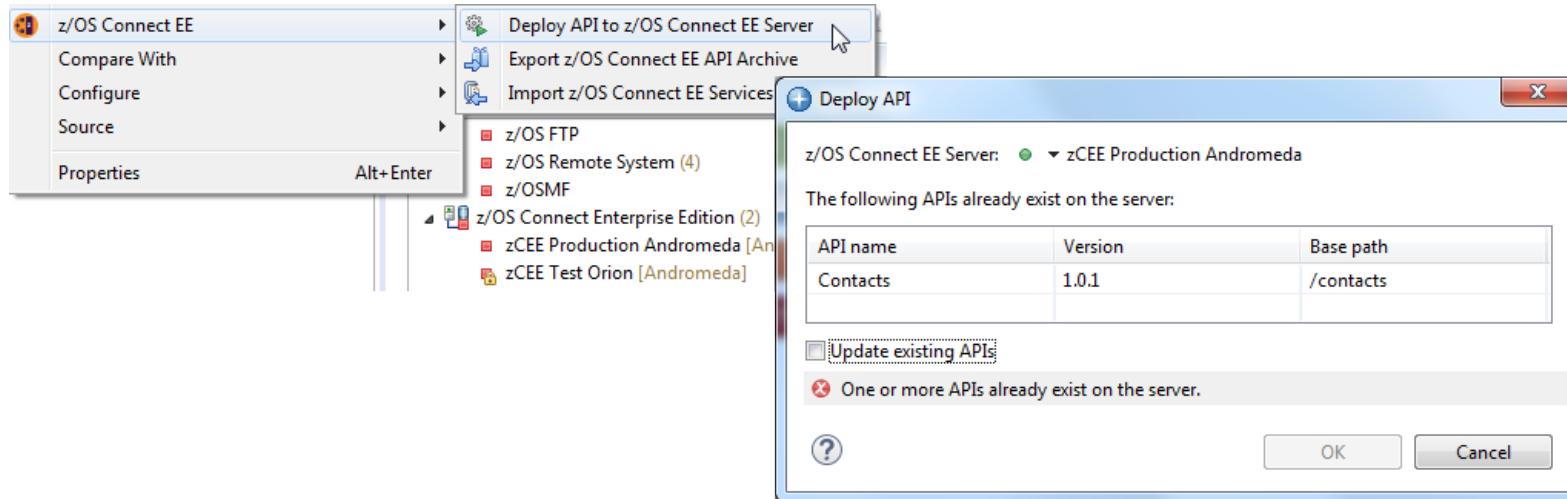


API toolkit



Server connection and API deployment

Manage z/OS Connect EE server connections in the Host Connections view:

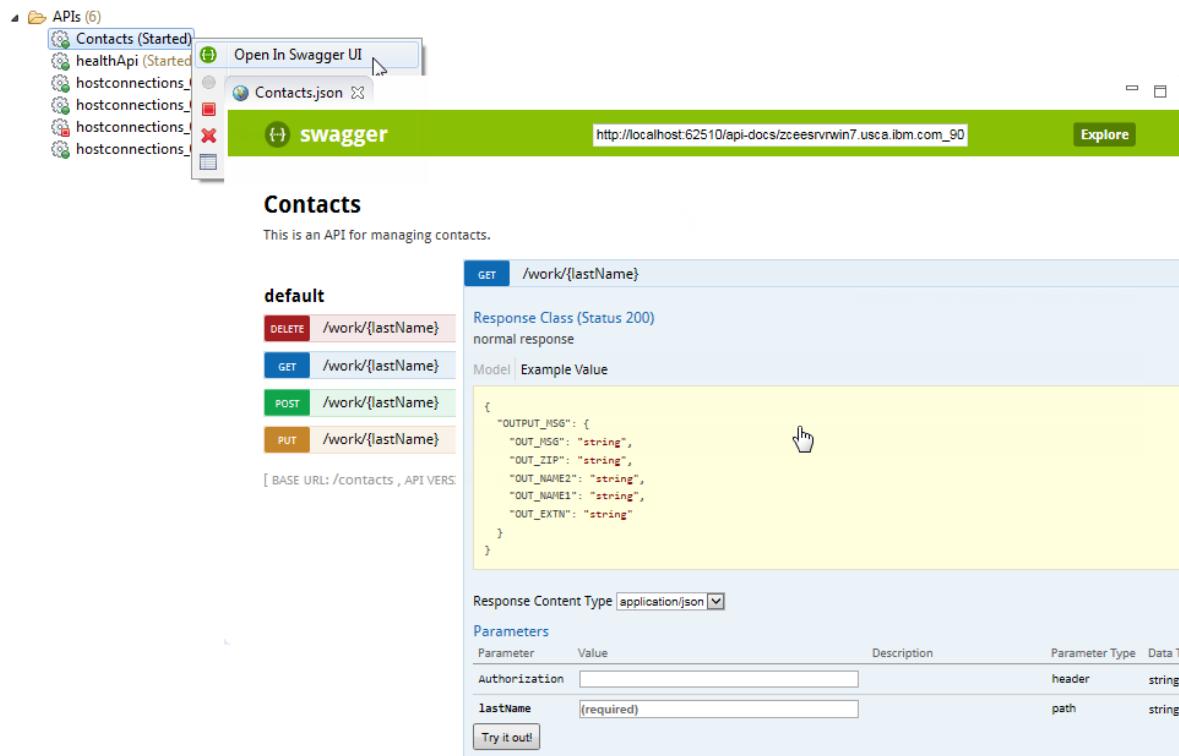


Right-click deploy to server enables developers to quickly deploy, test, and iterate on their APIs.

z/OS Connect EE Servers view allows you to start, stop, and remove APIs from a running server.

API toolkit

Testing with Swagger UI



The screenshot shows the API toolkit's main interface. On the left, a sidebar lists several APIs: 'APIs (6)' with 'Contacts (Started)', 'healthApi (Started)', and five 'hostconnections' entries. A context menu is open over the 'Contacts' entry, with 'Open In Swagger UI' highlighted. The main area is titled 'swagger' and displays the 'Contacts' API documentation. It includes a sidebar for 'default' operations: 'DELETE /work/{lastName}', 'GET /work/{lastName}' (selected), 'POST /work/{lastName}', and 'PUT /work/{lastName}'. Below this is a note: '[BASE URL: /contacts , API VERS:]'. The main content area shows a 'GET /work/{lastName}' operation with a 'Response Class (Status 200) normal response' section. It contains a JSON schema for the response:

```
{  
    "OUT_OUTPUT": {  
        "OUT_MSG": "string",  
        "OUT_ZIP": "string",  
        "OUT_NAME2": "string",  
        "OUT_NAME1": "string",  
        "OUT_EXTN": "string"  
    }  
}
```

Below the schema are sections for 'Response Content Type' (set to 'application/json') and 'Parameters'. The 'Parameters' table has two rows:

Parameter	Value	Description	Parameter Type	Data Type
Authorization	<input type="text"/>		header	string
lastName	<input type="text"/> (required)		path	string

A 'Try it out!' button is at the bottom.



z/OS Connect EE

Test your deployed APIs directly with Swagger UI inside the editor.

No need to export the Swagger doc to a separate tool.



/swagger/open_api

The industry standard framework for describing RESTful APIs.

Why Swagger?

It is more than just an API framework



z/OS Connect EE



There are a number of tools available to aid consumption:

Write Swagger

Swagger Editor allows API developers to design their swagger documents.

```
swagger: '2.0'
info:
  title: 'API for managing contacts'
  description: 'This is an API for managing contacts.'
  version: '1.0'
host: 'base-venus.demos.ibm.com:33820'
basePath: '/api-docs'
schemes:
  - https
  - http
consumes:
  - application/json
produces:
  - application/json
paths:
  /work/{lastname}:
    get:
      summary: 'Get contact by last name'
      parameters:
        - name: 'lastname'
          type: 'string'
          required: true
          description: 'Type: string'
          format: 'string'
          example: 'John'
        - name: 'header'
          type: 'header'
          required: false
          description: 'Type: header'
    post:
      summary: 'Create contact'
      parameters:
        - name: 'lastname'
          type: 'string'
          required: true
          description: 'Type: string'
          format: 'string'
          example: 'John'
        - name: 'header'
          type: 'header'
          required: false
          description: 'Type: header'
    put:
      summary: 'Update contact'
      parameters:
        - name: 'lastname'
          type: 'string'
          required: true
          description: 'Type: string'
          format: 'string'
          example: 'John'
        - name: 'header'
          type: 'header'
          required: false
          description: 'Type: header'
    delete:
      summary: 'Delete contact'
      parameters:
        - name: 'lastname'
          type: 'string'
          required: true
          description: 'Type: string'
          format: 'string'
          example: 'John'
        - name: 'header'
          type: 'header'
          required: false
          description: 'Type: header'
```

Read Swagger

Swagger UI allows API consumers to easily browse and try APIs based on Swagger Doc.

The screenshot shows the Swagger UI interface for a 'contacts' API. It displays a list of operations under the 'default' path:

- GET /work/{lastname}
- POST /work/{lastname}
- PUT /work/{lastname}
- DELETE /work/{lastname}

Below the operations, there is a section for 'Parameters'.

Consume Swagger

Swagger Codegen create stub code to consume APIs from various languages

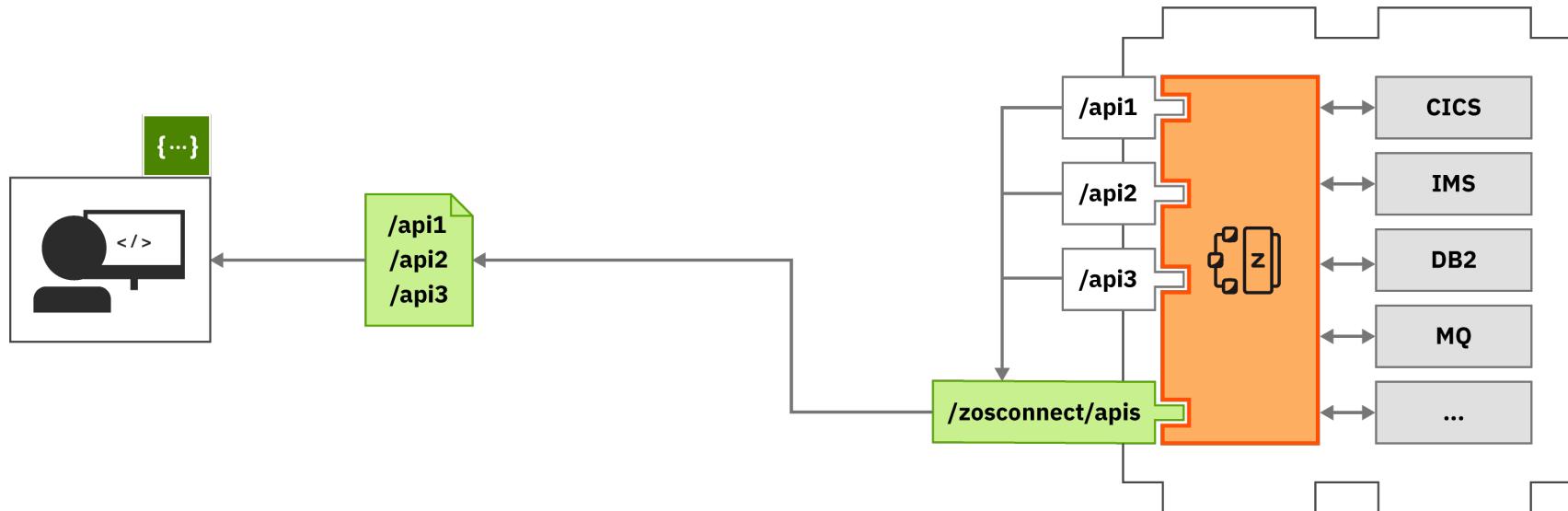


API Documentation



z/OS Connect EE

Get your Swagger on



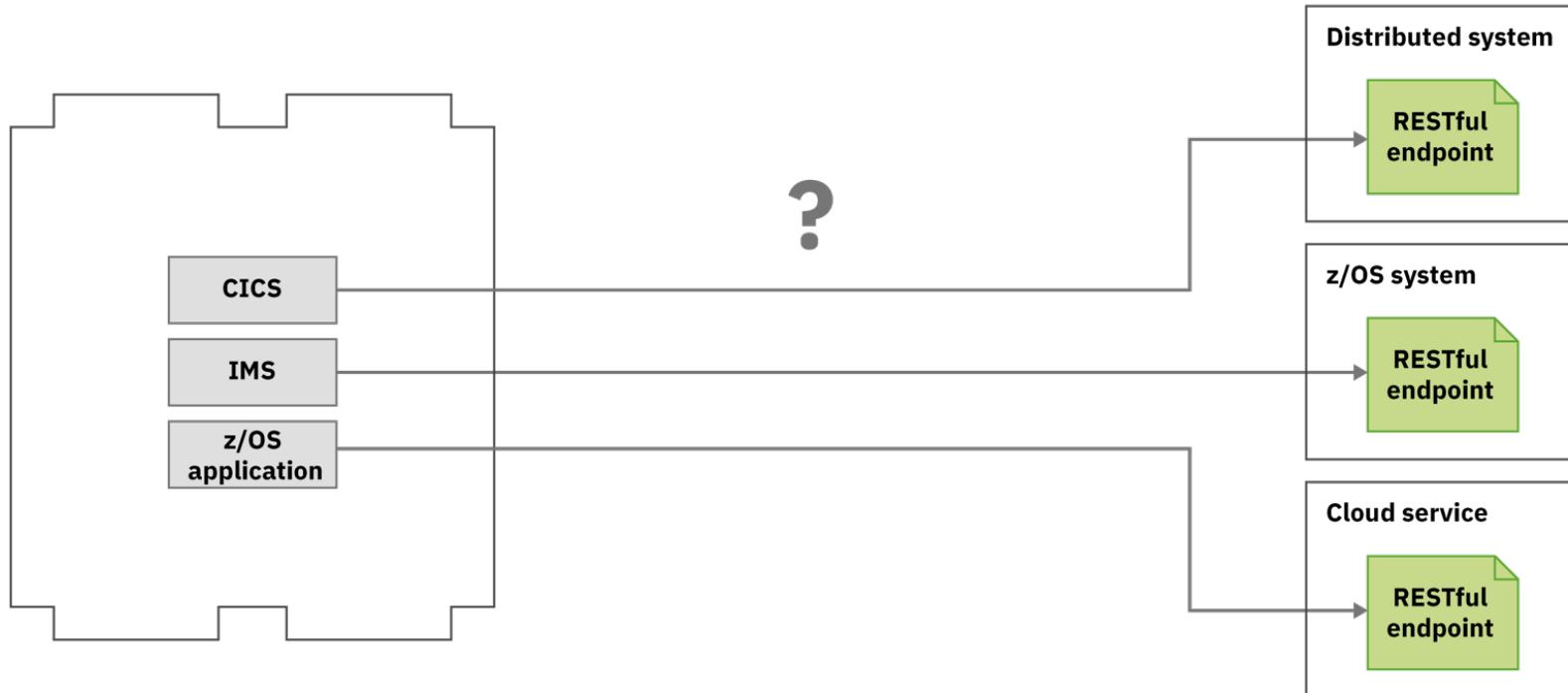
APIs are discoverable via Swagger docs served from z/OS Connect EE.

URIs of Swagger docs can be consumed by API management products such as API Connect.

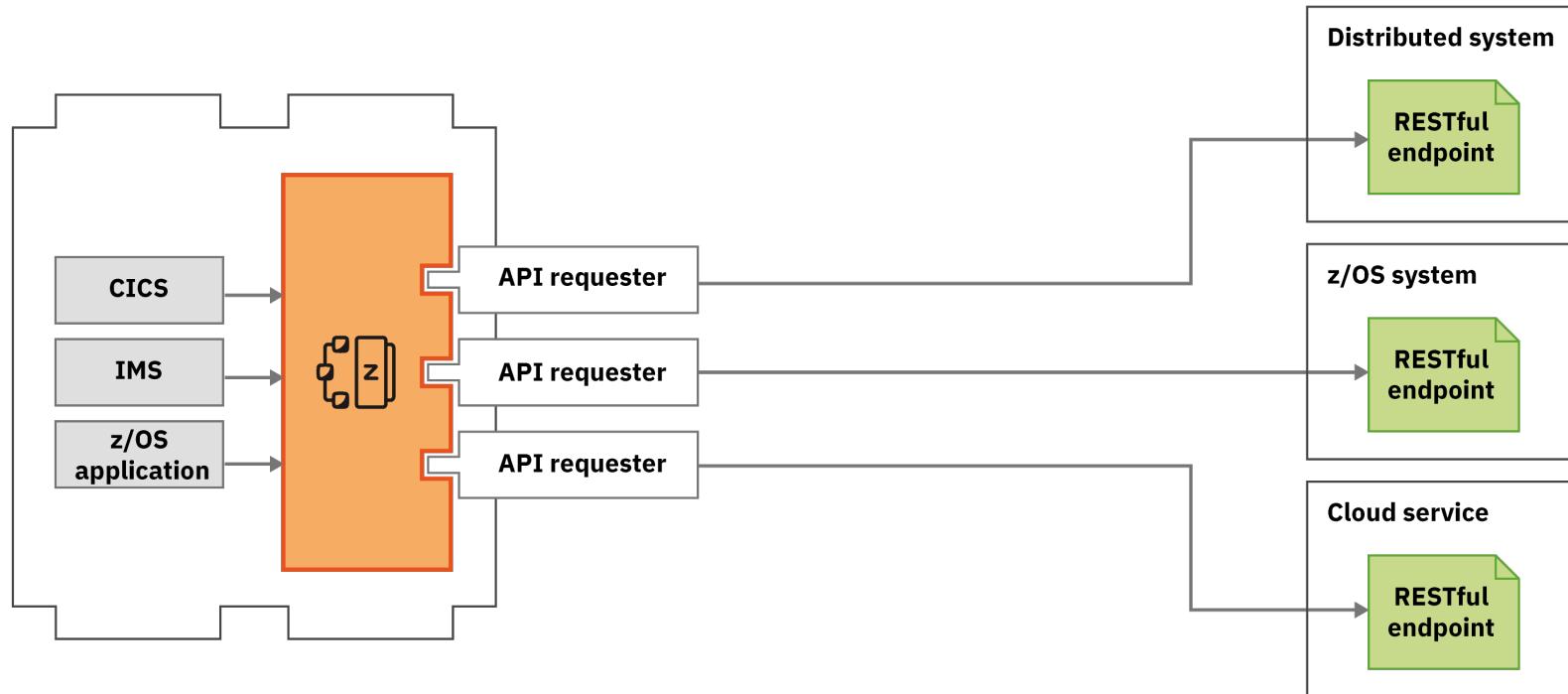
What about calling external APIs from my z/OS assets?



z/OS Connect EE



Use API requester to call external APIs from z/OS assets

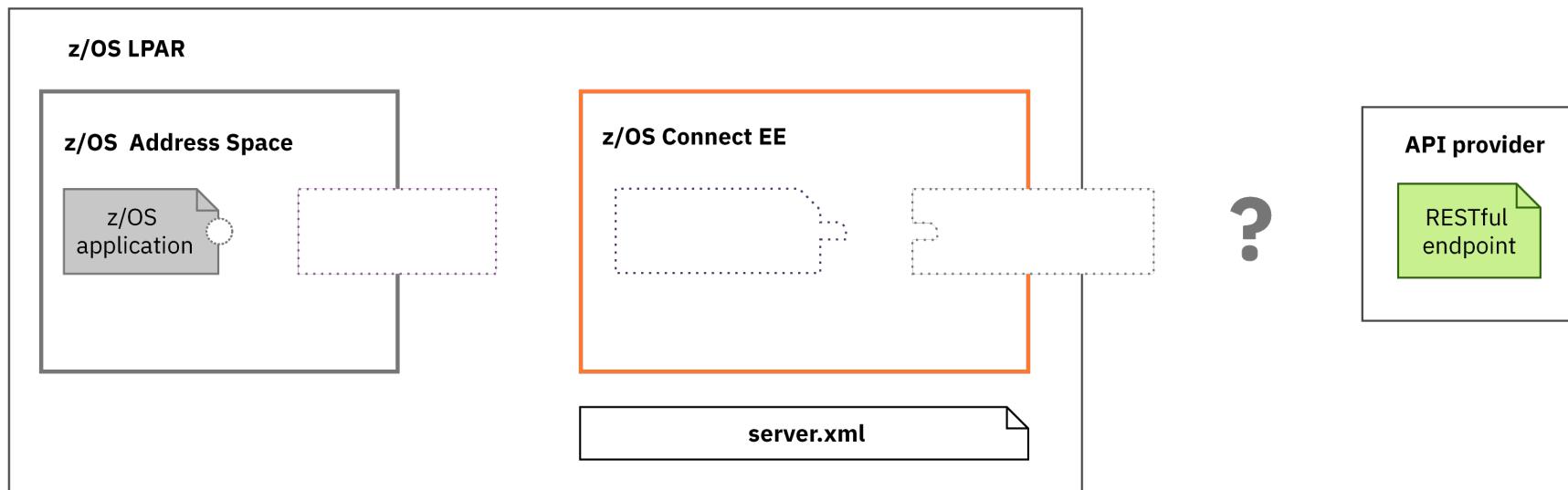


Five steps to calling an external API



z/OS Connect EE

Starting point

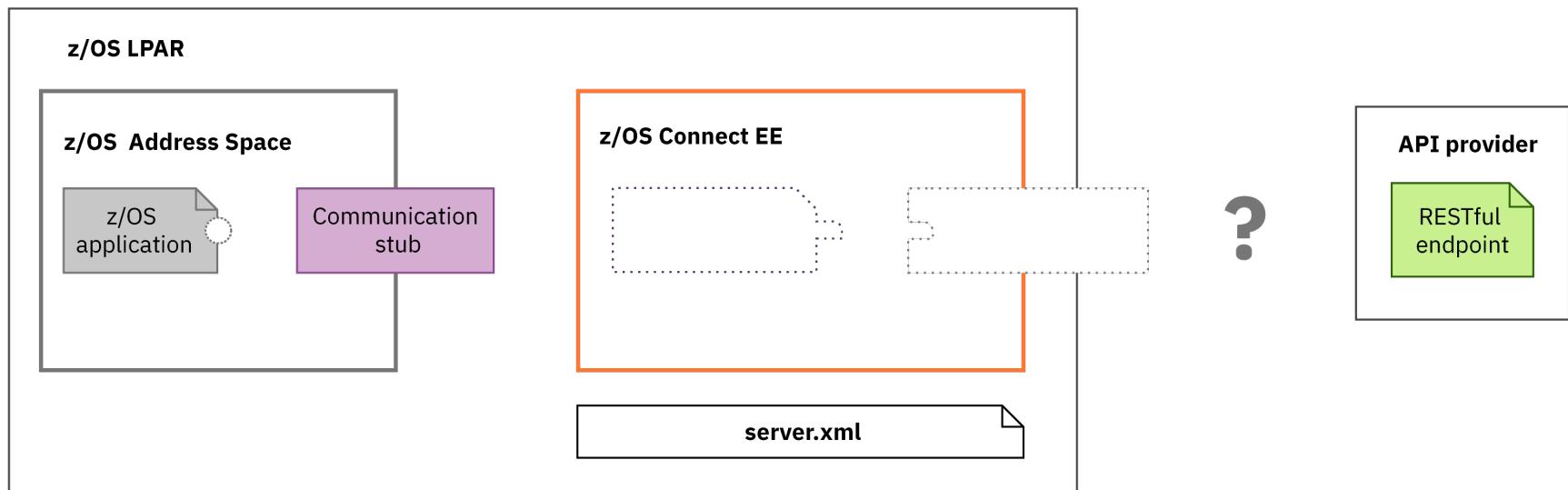


Five steps to calling an external API



z/OS Connect EE

Step 1. Configure communication stub (once per address space)

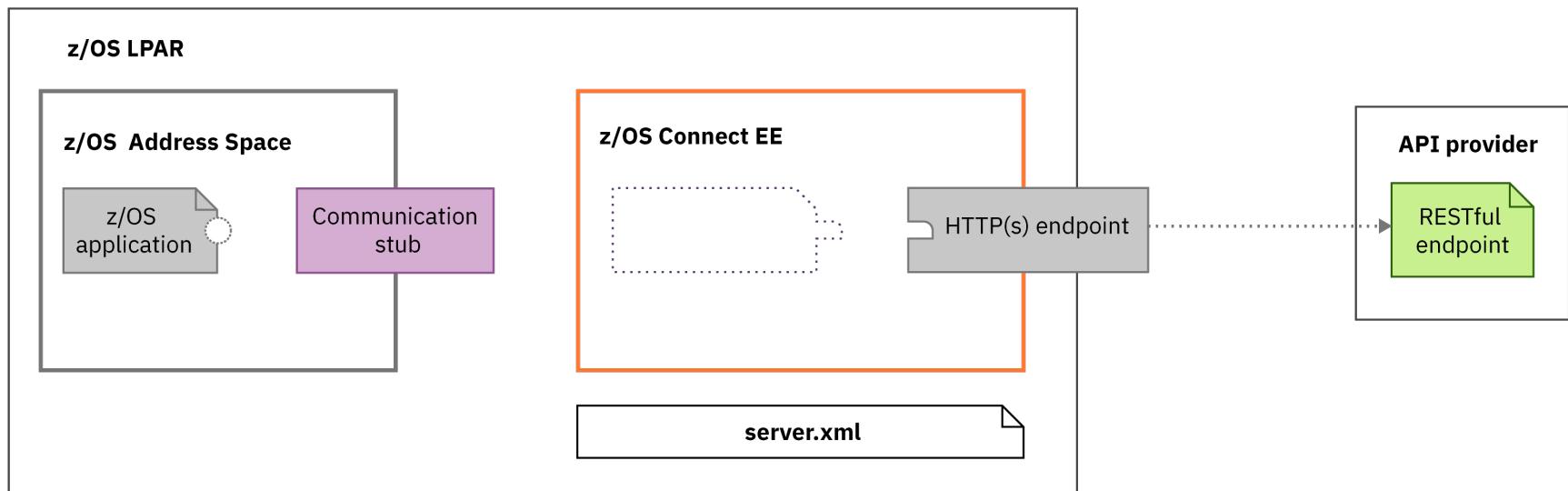


Configure a communication stub. You only need to do this once per address space.

Five steps to calling an external API



Step 2. Configure HTTP(S) endpoint (once per API Provider)



Configure the connection between z/OS Connect EE and the external API.

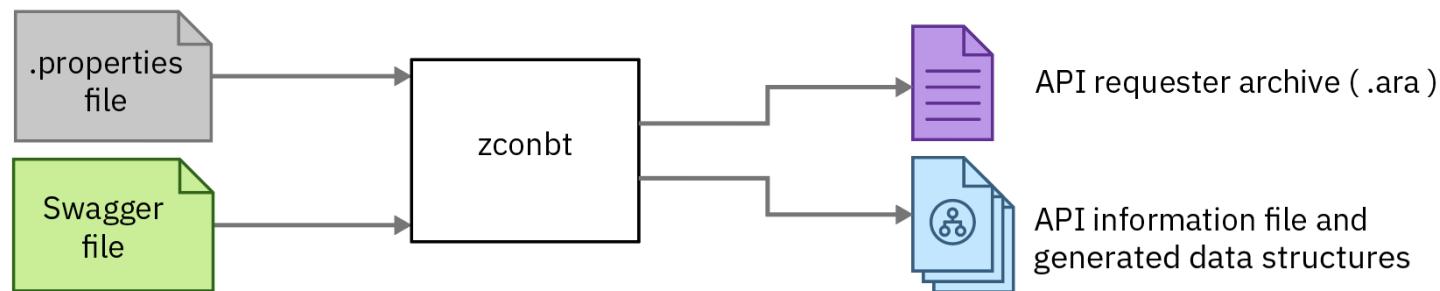
HTTP endpoints can be shared by many .ara files

Typically you only need a few HTTP Endpoints eg many APIs can be provided by one API provider or gateway

Five steps to calling an external API



Step 3. Generate API requester archive from Swagger



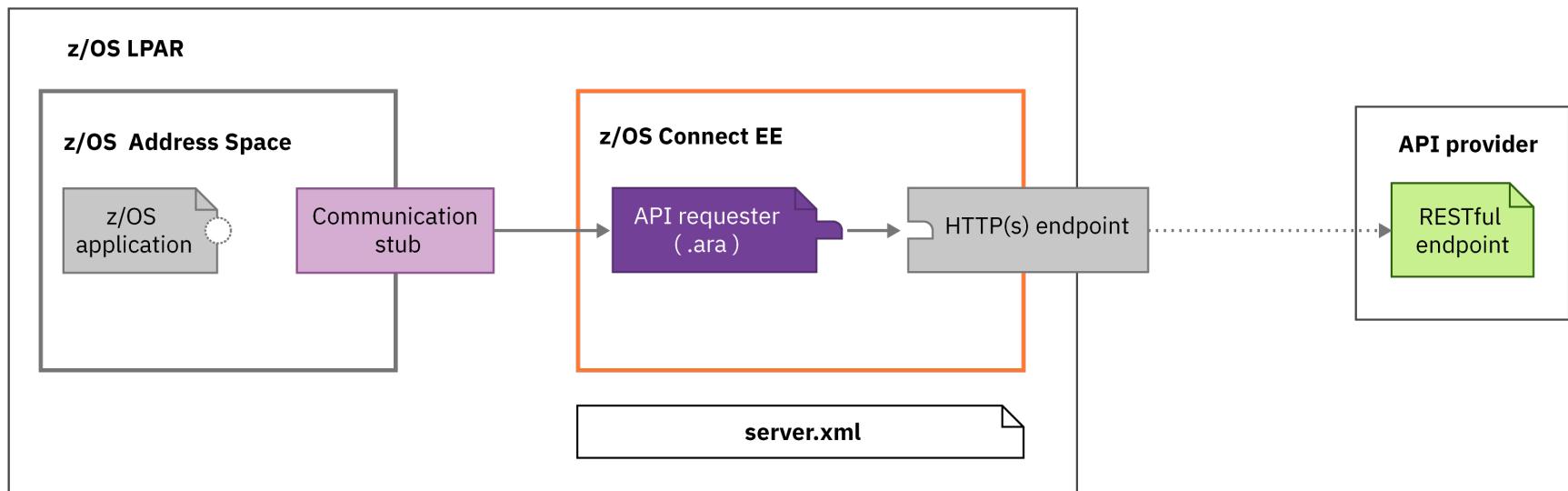
Generate your `.ara` file, API information file, and generated data structures.

Five steps to calling an external API



z/OS Connect EE

Step 4. Deploy API requester (.ara) archive



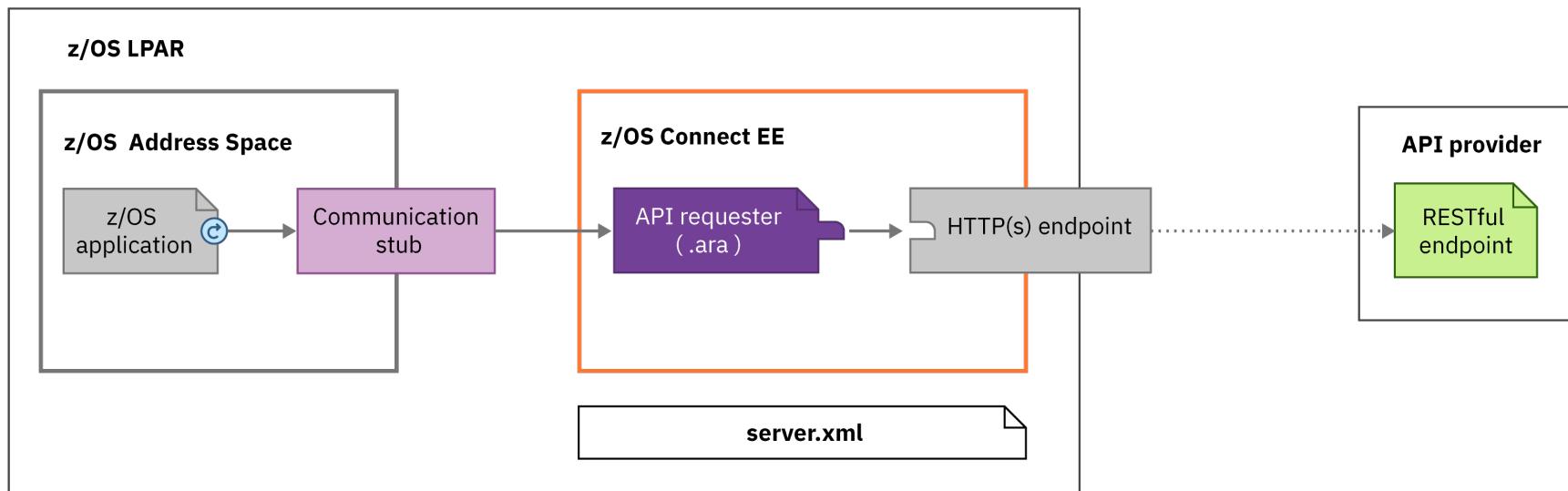
Deploy your API requester archive to the `apiRequester` directory.

Five steps to calling an external API



z/OS Connect EE

Step 5. Update z/OS application



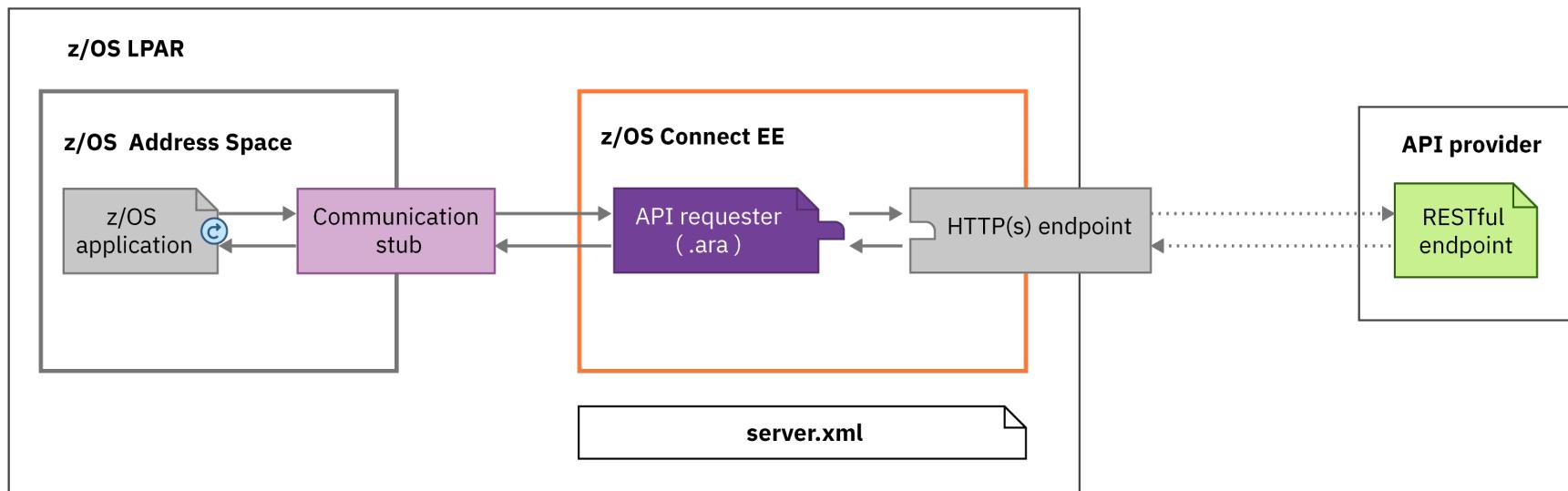
Finally, add the generated data structures to your existing application and use them to make the external API call.

Five steps to calling an external API

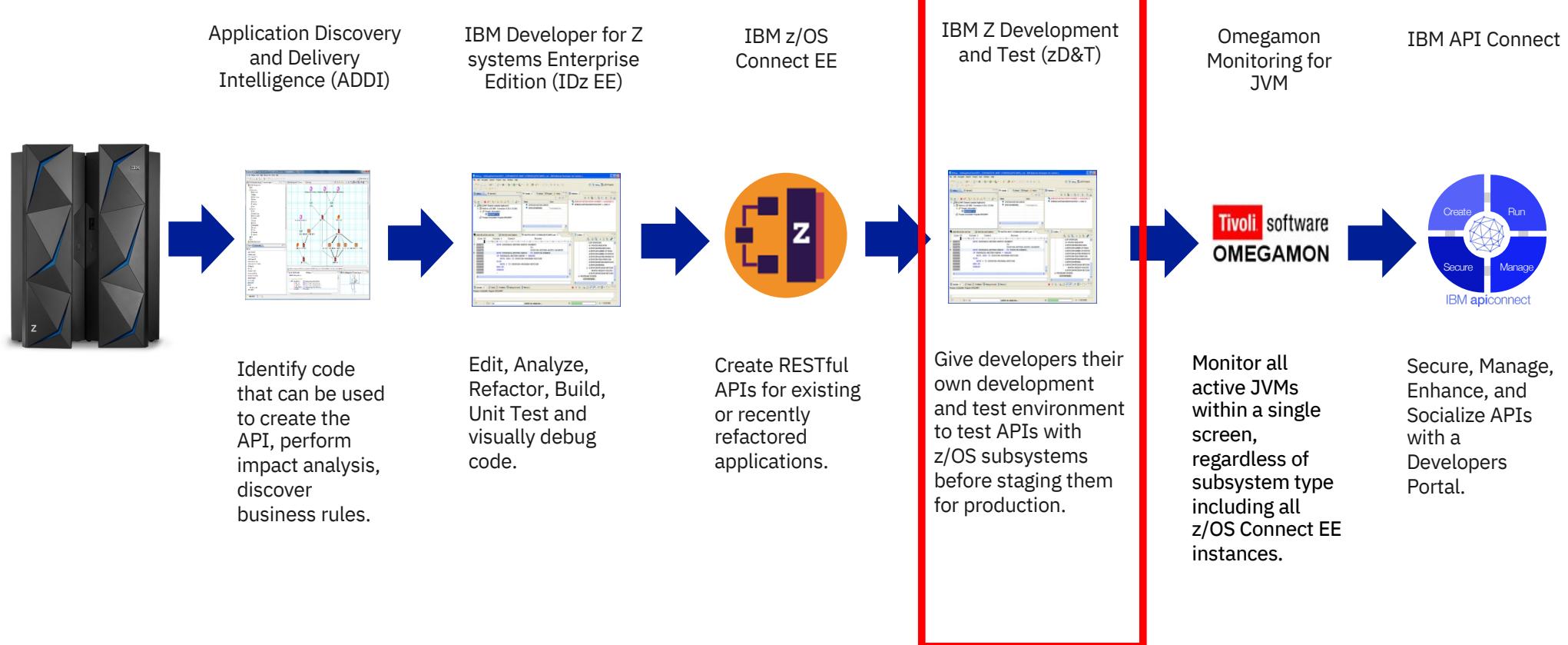


z/OS Connect EE

Done



This is the Digital Transformation Roadmap for IBM Z



ZD&T → Dev/Test Environment

Affordable off host development and test environment

IBM Z Development and Test Environment (ZD&T)

COBOL, PL/I, C++, Java, EGL, Batch, Assembler

RTC, UCD, IMS, WAS, z/OS, ZD&T, X86 PC running Linux

IDz, RIT, DT, DB2, MQ

zTrial

Do you know [zTrial](#) is powered by zD&T?

- Develop and test z/OS applications anywhere, anytime
- Free up mainframe development MIPS for production workload
- Eliminate costly delays by reducing burden on existing IT operations staff
- Reduce time to value and minimize ongoing administration and capital expense with zD&T Cloud Managed DevOps
- Exploit new hardware capability, **including pervasive encryption**
- Comprehensive z/OS 2.3 software distribution:
 - z/OS plus major subsystems
 - Underpinned by the z/OS components of DevOps for the Enterprise development, test, and deployment tooling
 - **CICS subsystem pre-provisioned**

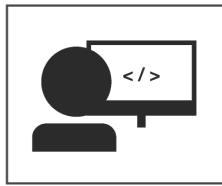
New in V12: Cloud Integration & Deployment Automation

“ZD&T improved our development and testing timeline and provided stability and quality” Developer, Large Enterprise Computer Services Company

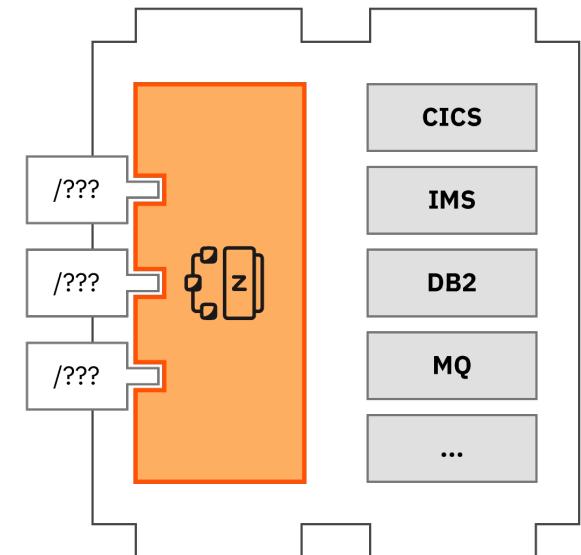
<https://www.techvalidate.com/tvid/C99-3E2-1ED>

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

How do developers know what APIs are available?

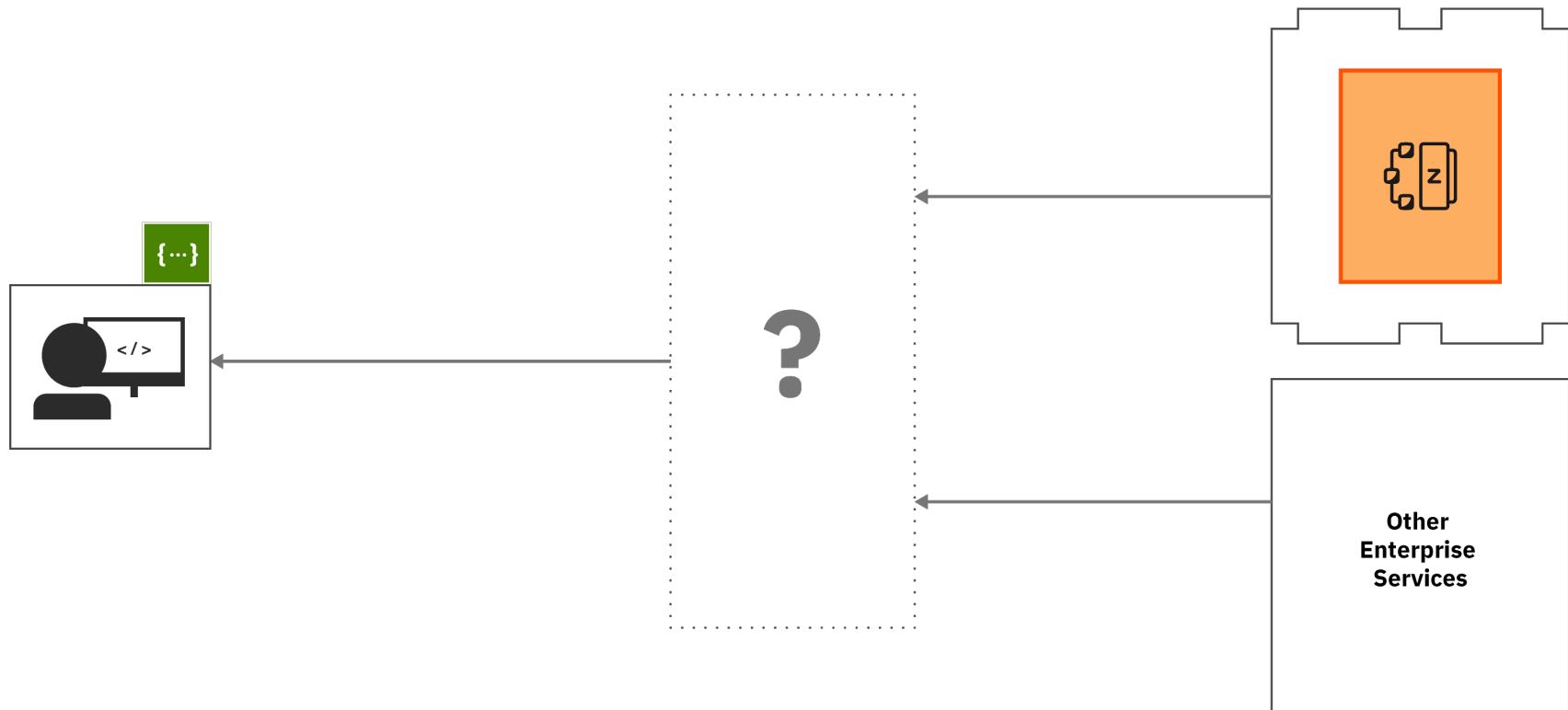


?
?
?

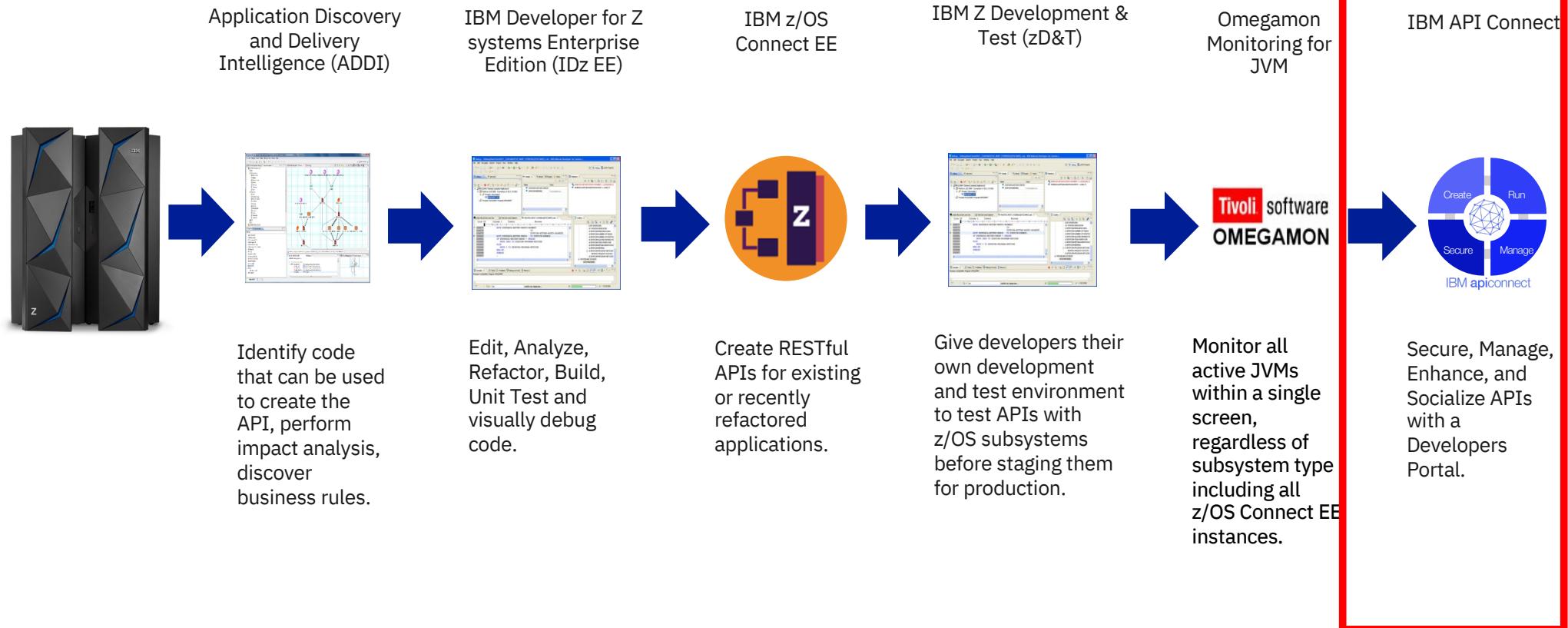


Publishing APIs

How do you publish your APIs to developers?

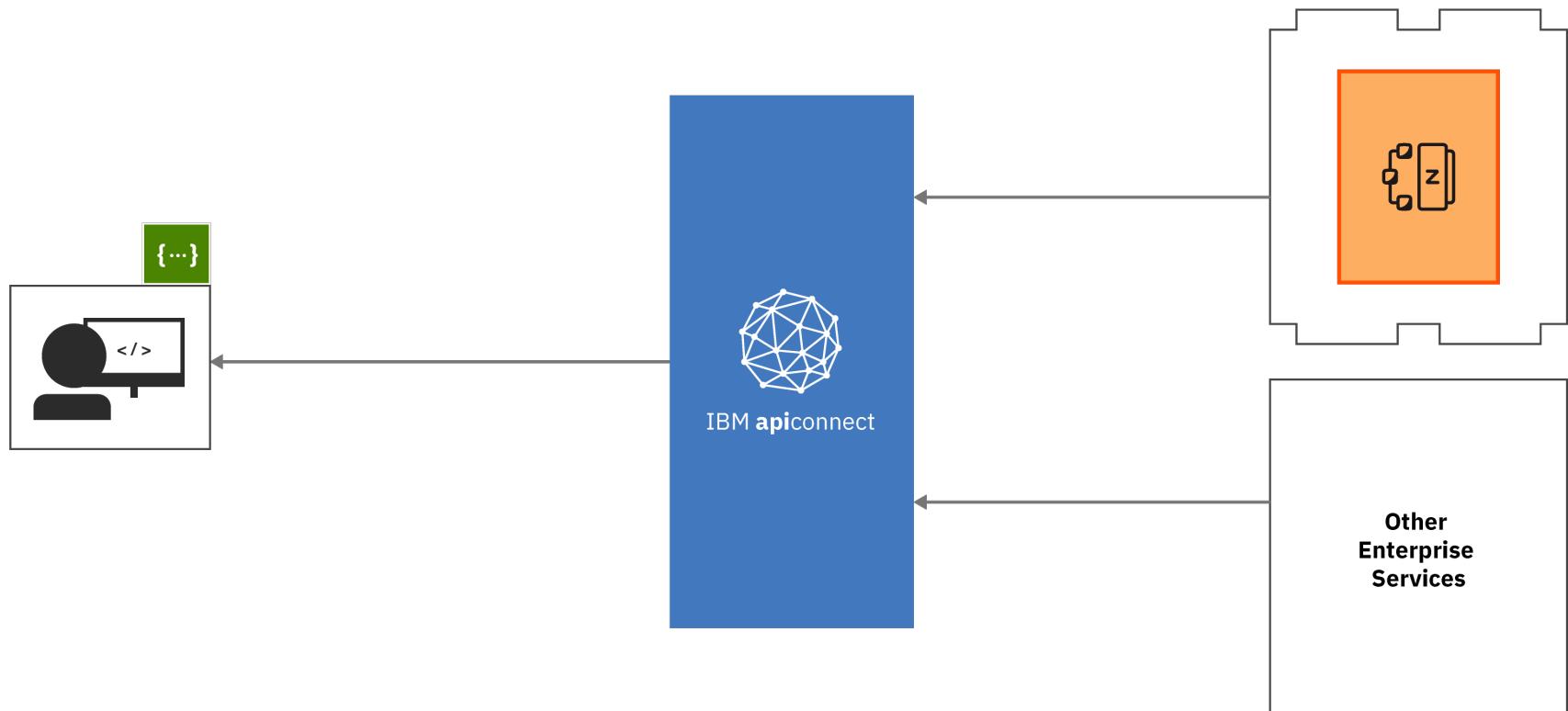


This is the Digital Transformation Roadmap for IBM Z

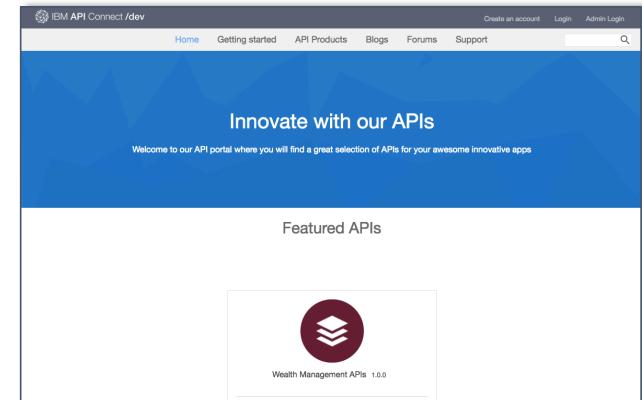


Publishing APIs

We need an API management gateway!

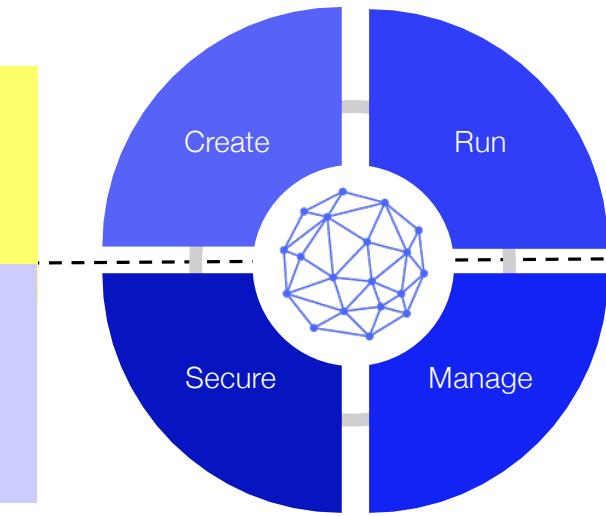


IBM API Connect Simplified & Comprehensive API Management



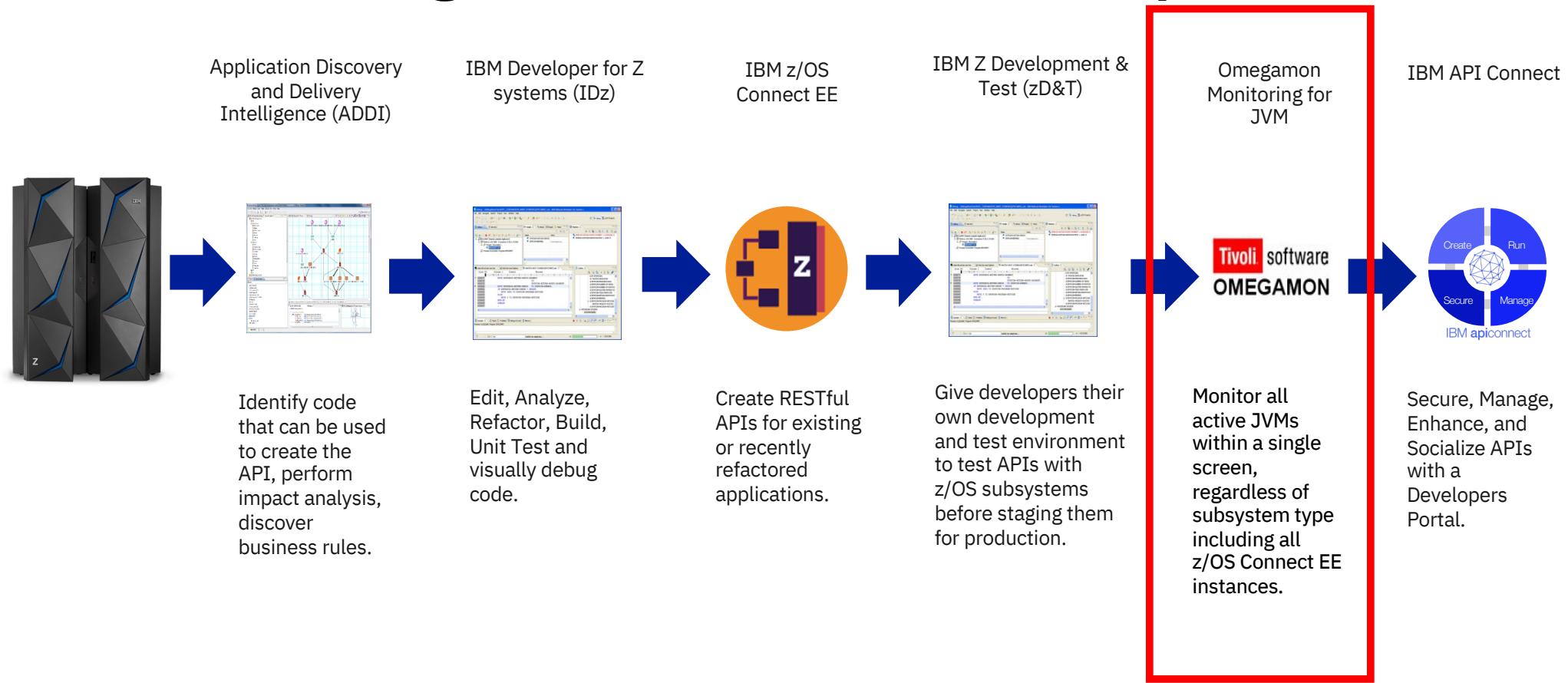
What does API Connect provide?

- Automated, visual and coding options for creating APIs
- Node.js and Java support for creating Microservices
- Integrated enterprise grade clustering, management and security for Node.js and Java
- Access control over API's, API Plans and API Products*
- Advanced API usage analytics
- Customizable, self service developer portal for publishing APIs
- Policy enforcement, API Monetization, security and control

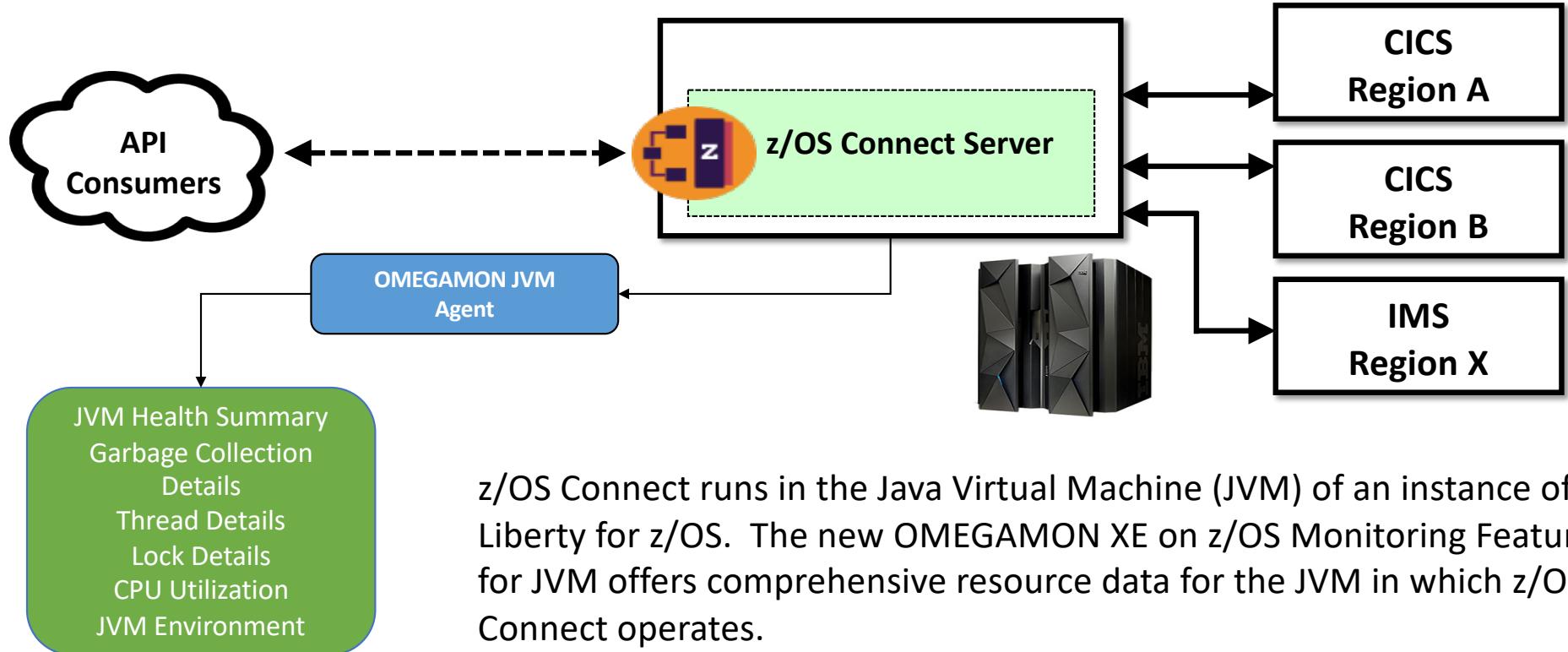


**Clients who may have already created SOAP or REST services on z can still benefit from API Connect*

This is the Digital Transformation Roadmap for IBM Z



OMEGAMON Monitoring for JVM



z/OS Connect runs in the Java Virtual Machine (JVM) of an instance of Liberty for z/OS. The new OMEGAMON XE on z/OS Monitoring Feature for JVM offers comprehensive resource data for the JVM in which z/OS Connect operates.

**Start with something small and useful
AND TALK TO YOUR DEVELOPERS!**



/resources

Useful links for next steps

New z/OS Connect EE LinkedIn user group



Al Grega
Joined group: Feb 2019

Recent

- z/OS Connect EE Users
- OCEAN User Group of Sou...
- WebSphere Business Part...
- IBM WebSphere Enthusiasts
- IBM i, iSeries, and AS/400...

Groups

- z/OS Connect EE Users
- OCEAN User Group of Sou...
- WebSphere Business Part...

Show more ▾

Events

Followed Hashtags

Discover more

z/OS Connect EE Users

Standard group

[Start a conversation in this group](#) [Photo](#) [Video](#) [File](#)

Anthony Papageorgiou • 1st
Offering Manager for API Enablement on IBM Z at IBM
1w

Hi everyone, FYI we just announced z/OS Connect EE Unlimited which provides unlimited server instances for a set price. For more information see the announce letter here: <https://lnkd.in/dsGP2bj>

IBM z/OS Connect Enterprise Edition Unlimited, V3 delivers a new, flexible pricing metric for enterprise clients

154 members [See all](#)

[Invite members](#)

About this group

This group is for users of z/OS Connect EE to discuss and share use cases, experiences and best practices around API enablement on IBM Z with z/OS Connect EE.

The User Group is community led, so [Show more ▾](#)

Group rules

This group is owned and run by the z/OS Connect EE user community.

Content here does not represent IBM's views.

This group is focused on open dialogue along [Show more ▾](#)



IBM z Trial Program

Try the latest z/OS Connect EE capabilities today at zero cost, and with no installation required.

- ⓘ Find out more, and sign up now at ibm.biz/ibmztrial

See what's new in z/OS Connect EE



IBM Developer Mainframe DEV Home Downloads Training Documentation Blogs Announcements Events Forum Videos

z/OS Connect Enterprise Edition

z/OS Connect Enterprise Edition gives you the power to call external APIs from your mainframe applications, and expose your mainframe assets as easily-consumable RESTful APIs.

Downloads Try it now What's new?

Learn more

Truly RESTful APIs to and from your mainframe

IBM Developer Mainframe DEV Home Downloads Training Documentation Blogs Announcements Events Forum Videos

What's new in z/OS Connect EE

Following the continuous delivery model, z/OS Connect Enterprise Edition offers regular release updates for you to make the most out of this leading capability. Read on to find the most up-to-date release information...

New to z/OS Connect EE? Ready to start? Go to the Developer Center Go to the Knowledge Center

Version 3.0.22 – 25th June 2019

See what's in this release:

- WebSphere Liberty Profile V19.0.0.3. The version of WebSphere Liberty Profile embedded in z/OS Connect EE V3.0.22 is V19.0.3.
- Angel V11. The version of the angel shipped with WebSphere Liberty Profile V19.0.3.
- API toolkit V3.0.6 / V3.2.6.6. The latest version of the z/OS Connect EE API plug-in to design and create your services and APIs is V3.0.6 for Aqua 3.0 users and V3.2.6.6 for Aqua 3.2 users.

Version 3.0.21 – 24th May 2019

See what's in this release:

- Upgrading from a previous release of z/OS Connect EE. After upgrading to z/OS Connect EE V3.0.21, the first time you start up your server, you must specify the --clean option on your started task PARMS statement.
- IBM MQ Service Provider. The IBM MQ service provider is now built into the z/OS Connect EE server, so do not need to copy it from an IBM MQ installation.
- More information about API requester security in the IBM Knowledge Center Security section of the IBM Knowledge Centre has been enhanced for API requests. This new content now includes more contextual information that will help you to choose the right security mechanism for your application.

Using Splunk with z/OS Connect EE

NigelWilliams Published on 05/22/2019 / Updated on 05/23/2019

All Documentation z/OS Connect EE Developer Center – Articles

The business case for APIs

- Get started with API enablement on Z
- Get started with z/OS Connect EE
- Test your APIs to z/OS assets
- Security

Managing API workloads

- Limiting the impact of API workloads on MLC
- Measuring API workloads with WLM
- Monitoring APIs with OMEGAMON for JVM
- Using OMEGAMON for JVM to diagnose z/OS Connect EE API failures
- Using Splunk with z/OS Connect EE

DevOps with z/OS Connect EE

Open Banking

Introduction

As you prepare to deploy APIs to z/OS Connect EE, you might be considering how to monitor API workloads, perform operational analytics and also track API requests across the enterprise. Figure 1 shows how z/OS Connect EE is now enabled for all these requirements.

Figure 1 z/OS Connect EE support for analytics, monitoring and transaction tracking

In the article Monitoring APIs with OMEGAMON for JVM we provided an overview of how OMEGAMON for JVM can be used to monitor the status of a z/OS Connect EE API workload. In this article we provide an overview of how z/OS Connect EE audit data can be viewed in a Splunk dashboard.

1.2 Common Data Provider for z Systems

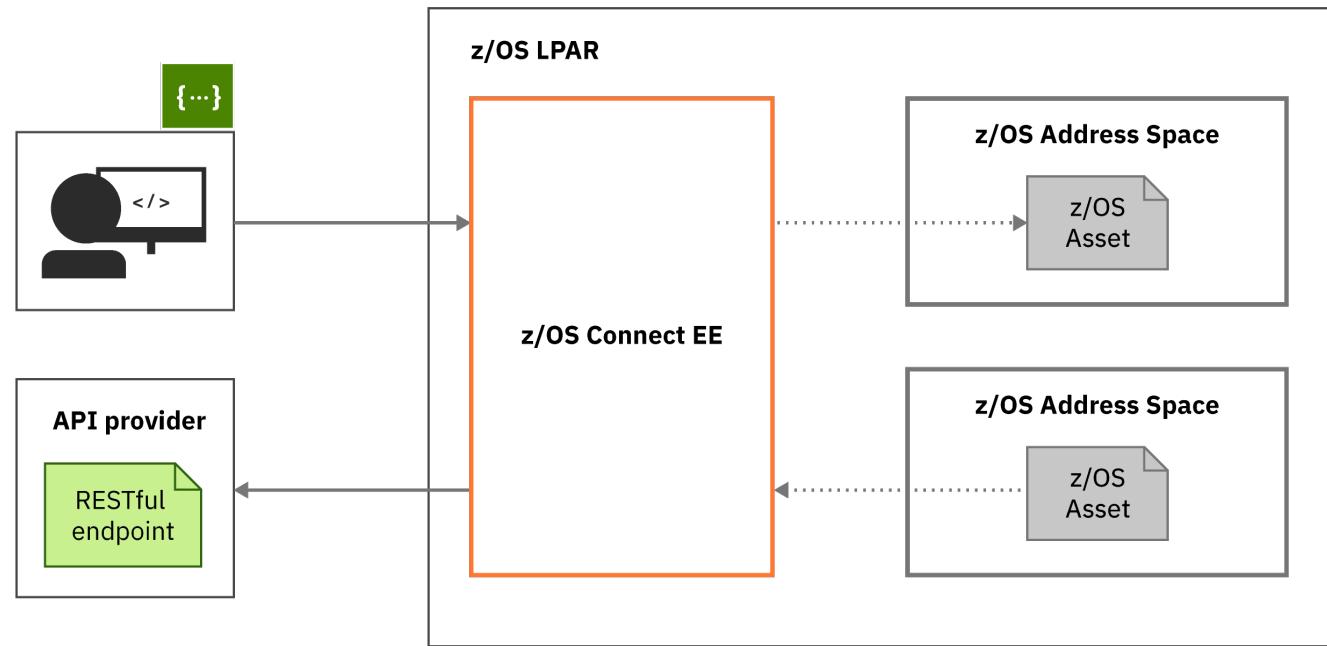
IBM Common Data Provider for z Systems provides the infrastructure for accessing IT operational data from z/OS systems and streaming it, in a consumable format, to an analytics platform such as Elasticsearch, Apache Hadoop or Splunk. It monitors z/OS log data and SMF data and forwards it to the configured destination. Over 140 different data sources are supported including all standard SMF records, Syslog and application logs from CICS, IMS, NetView and other subsystems. A web-based configuration tool is used to specify what data you want to collect from your z/OS system, where you want the data to be sent, and what form you want the data to arrive in at its destination. This configuration information is contained in a policy.

Visit the z/OS Connect EE Developer Center: ibm.biz/zosconnectdc

Make IBM Z the heart of your API strategy with truly RESTful APIs to and from your mainframe



z/OS Connect EE



- ⓘ Get started with the open beta today: ibm.biz/zosconnectdc

Resources



Downloads

⬇️ z/OS Connect EE open beta runtime



ibm.biz/zosconnect-open-beta

⬇️ z/OS Connect EE workstation tooling

ibm.biz/zosconnect-tooling-download

Explore the docs

ⓘ z/OS Connect EE Knowledge Center

ibm.biz/zosconnect-kc

ⓘ z/OS Connect EE Developer Center

ibm.biz/zosconnectdc

Where to get help

ⓘ dW Answers

ibm.biz/zosconnect-dw-answers

ⓘ z/OS Connect EE open beta forum

ibm.biz/zcee-beta-forum

One last thought...

***“We’re not trying to modernize the Mainframe,
We’re modernizing the way we use the Mainframe”***

*Director, DevOps
Large US based Property and Casualty Insurance Company.*



/questions?thanks=true

Thank you for listening.

About the Lab

- **The Lab environment is hosted on IBM Cloud, and accessed via the Remote Windows Desktop or a Web Browser**
- **The following Hands on Labs are available:**
 - Consume an API from a COBOL Application (API Requester)
 - Create an API from IMS
- **Lab Instructions are in the GitHub:**
 - <https://github.com/amgrega/zOSCEE-Workshop>
- **z/OS Connect EE Server** UID=Fred, pw=fredpwd
- **z/OS Explorer** UID= USER1, pw=USER1



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*	IBM*	IBM*
ibm.com*	ibm.com*	ibm.com*
IBM logo*	IBM logo*	IBM logo*

* Registered trademarks of IBM Corporation

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.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, 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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware, the VMware logo, VMware Cloud Foundation, VMware Cloud Foundation Service, VMware vCenter Server, and VMware vSphere are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice.

Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

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.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g. zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at

www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

September, 2019

2019 IBM Corporation