DB2 REST API and z/OS Connect – SQL/Stored Procedures Play a Role in Mobile and API Economics

Maryela Weihrauch,
IBM Distinguished Engineer, WW Analytics on System z

weihrau@us.ibm.com

Jun 5th, 2018

Please Note

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

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

- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

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

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Agenda

- The bigger picture - emerging business model “API Economy” is leveraging APIs to generate new business
- z/OS Connect provides a consistent REST(JSON) interface into z/OS- based business functions
- DB2 11 native RESTful service support provides a new interface into DB2 for z/OS

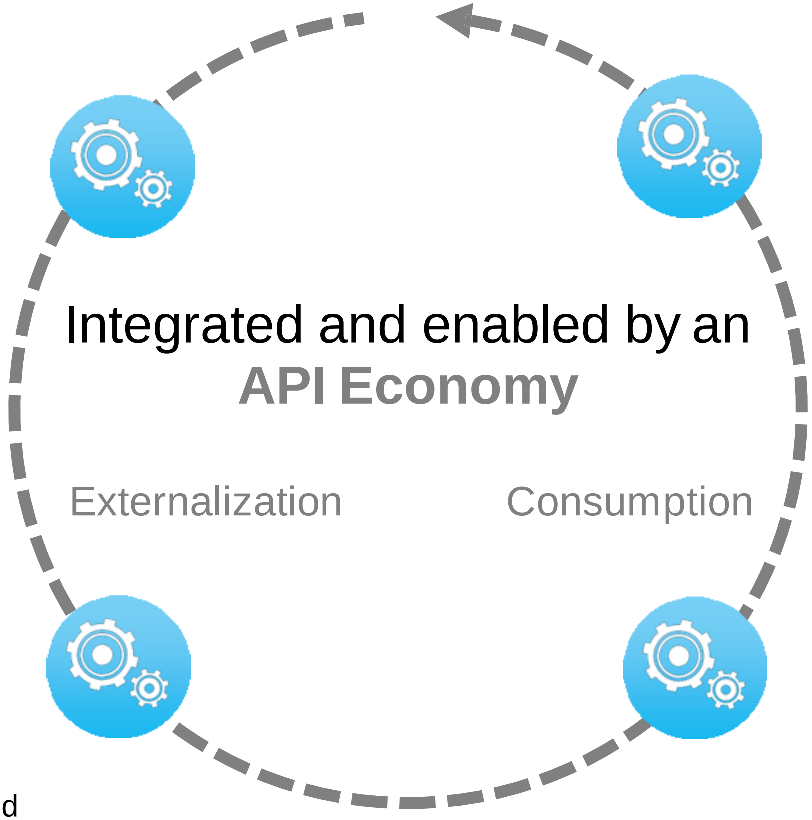
APIs are giving rise to a new value chain

Big Data | Assets

Big data sources provide insights that are shared and monetized through APIs

Social | Users

Social APIs fuel personalized experiences for users and new business models



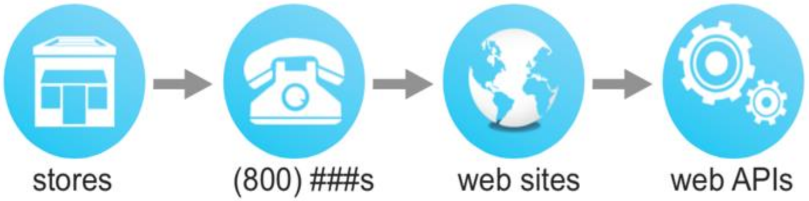
Cloud | Web Apps

Cloud services are exposed through Web APIs enabling rapid composition environments

Mobile | Apps

Mobile applications make calls to back end services through Web APIs

Business models are evolving

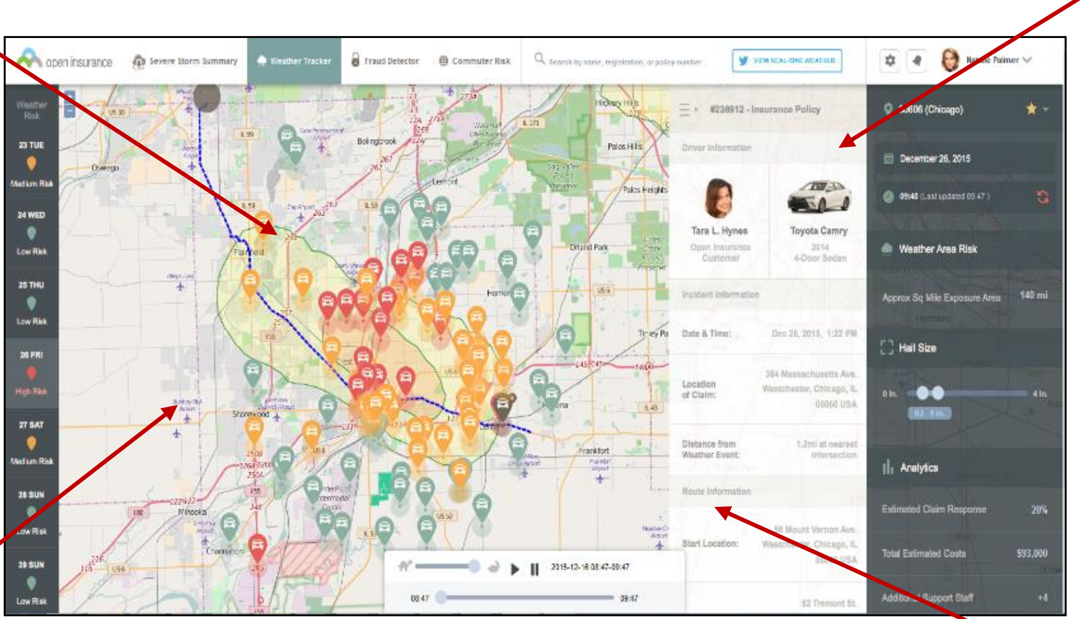


APIs represent a new, fast-growing channel opportunity

Weather Tracker Example

SoR
Driver/ Risk Data
API enabled

Public cloud
geospatial-temporal data



#238912 - Insurance Policy

Driver Information

Tara L. Hynes
Open Insurance Customer

Toyota Camry
2014
4-Door Sedan

Incident Information

Date & Time: Dec 26, 2015, 1:22 PM

Location of Claim: 384 Massachusetts Ave. Westchester, Chicago, IL 06060 USA

Distance from Weather Event: 1.2mi at nearest intersection

Route Information

Start Location: 58 Mount Vernon Ave. Westchester, Chicago, IL 06060 USA

82 Tremont St.

60606 (Chicago)

December 26, 2015

09:48 (Last updated 09:47)

Weather Area Risk

Approx Sq Mile Exposure Area: 140 mi

Hail Size

0 in. – 4 in.

0.2 - 1 in.

Analytics

Estimated Claim Response: 20%

Total Estimated Costs: $93,000

Additional Support Staff: +4

The Weather Company

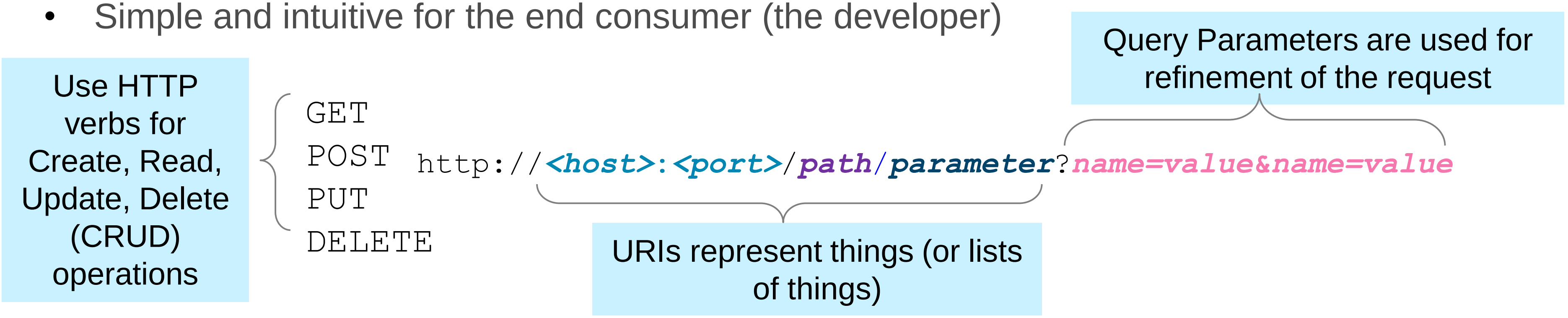
Public cloud
Weather data service

Public cloud /OnPrem
device data service

©2018 IBM Corporation

Introduction to REST

- **REST** - “Representational State Transfer” ... which uses HTTP/HTTPS and HTTP/HTTPS 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.
- An architectural style for **accessing** and **updating** data
- Simple and intuitive for the end consumer (the developer)



Use HTTP verbs for Create, Read, Update, Delete (CRUD) operations

```
GET
POST
PUT
DELETE
```

```
http://<host>:<port>/path/parameter?name=value&name=value
```

Query Parameters are used for refinement of the request

URIs represent things (or lists of things)

Request/Response Body is used to represent the data object

```
GET
http://www.dbs.com.sg:9080/API/Department?Location=tampines

RESPONSE: HTTP 200 OK
BODY { { “DEPTNAME” : “IT Operation SERVICE DIV”

       },

       { “DEPTNAME” : “DBMS SERVICE DIV”

       }
```

z/OS Connect Enterprise Edition

- **Provides single service channel for z/OS subsystem applications**
  - Industry Standard Interface (OpenAPI Initiative compliant)
  - Reduced interface layering, reduced complexity and costs
  - Scalable, highly available API
  - Secure, EAL5+

- **Delivers RESTful APIs as a discoverable, first-class resource with OpenAPI Spec (Swagger 2.0) descriptions**
  - Ready for consumption by today's enterprise application developers and integration with API management solutions
  - Swagger specification

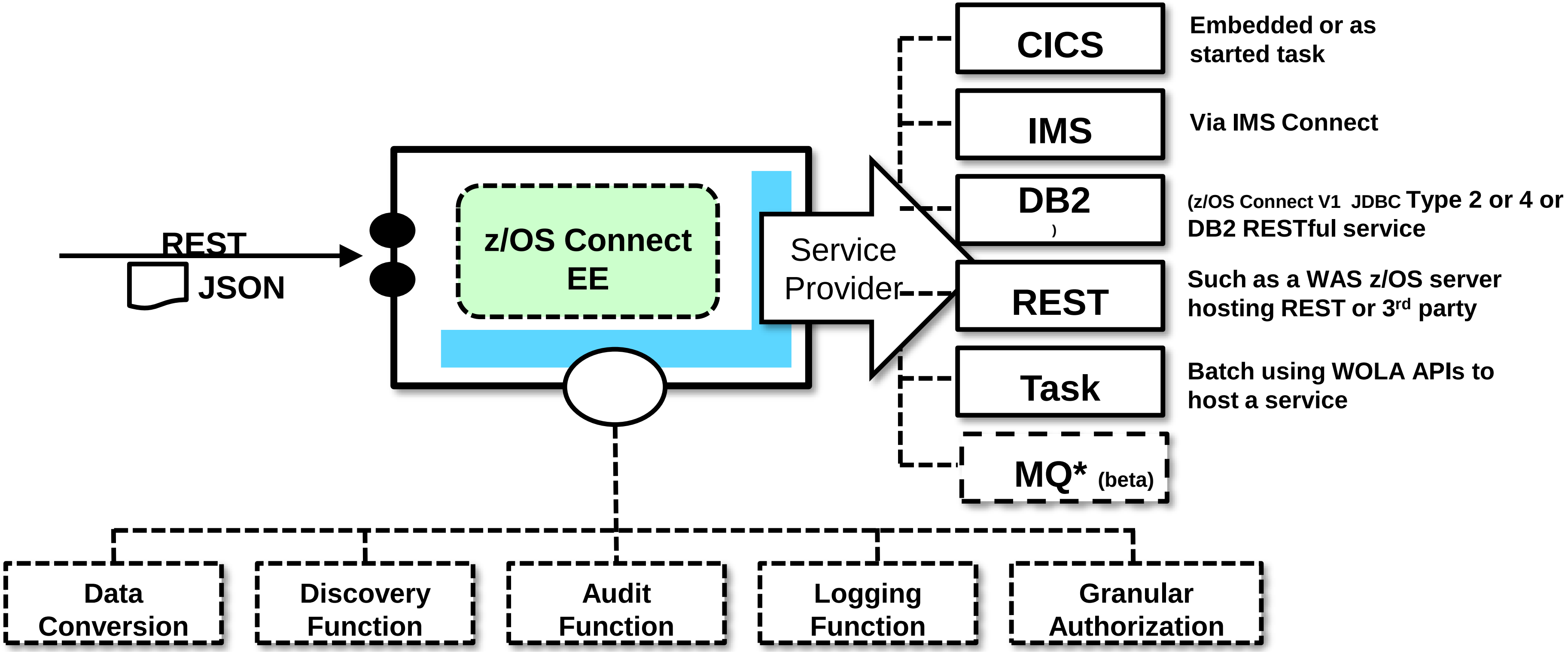
API
API
API
z/OS Connect EE
CICS
IMS
WebSphere
DB2

- **Comprehensive tooling that enables API developers to create RESTful APIs from z/OS-based assets**
  - API Editor download

- **Supports standard JSON message format** and conversion to z/OS subsystem backend format requirements

z/OS Connect EE Structure

- Provides common infrastructure for discovery, authorization, auditing, logging
- Data conversion to service provider subsystem


REST
JSON
z/OS Connect EE
Service Provider
CICS
Embedded or as started task
IMS
Via IMS Connect
DB2
)
(z/OS Connect V1 JDBC Type 2 or 4 or DB2 RESTful service
REST
Such as a WAS z/OS server hosting REST or 3rd party
Task
Batch using WOLA APIs to host a service
MQ* (beta)
Data Conversion
Discovery Function
Audit Function
Logging Function
Granular Authorization

* http://www.slideshare.net/MattLeming/hhm3481-ibm-mq-for-zos-enhancing-application-and-messaging-connectivity-in-a-hybrid-world

Why use Swagger?

More than just an API description framework…

There are number of Open Source tools available to aid consumption:

Write Swagger

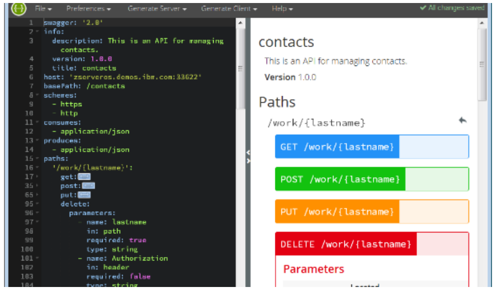
Swagger Editor allows API developers to design their Swagger documents.

Read Swagger

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

Consume Swagger

Swagger Codegen creates stub code to consume APIs from various languages.



```
swagger: '2.0'
info:
  description: This is an API for managing
    contacts.
  version: 1.0.0
  title: contacts
host: 'zserveros.demos.ibm.com:33622'
basePath: /contacts
schemes:
  - https
  - http
consumes:
  - application/json
produces:
  - application/json
paths:
  '/work/{lastname}':
    get:
    post:
    put:
    delete:
      parameters:
        - name: lastname
          in: path
          required: true
          type: string
        - name: Authorization
          in: header
          required: false
          type: string
```

contacts

This is an API for managing contacts.

Version 1.0.0

Paths

/work/{lastname}

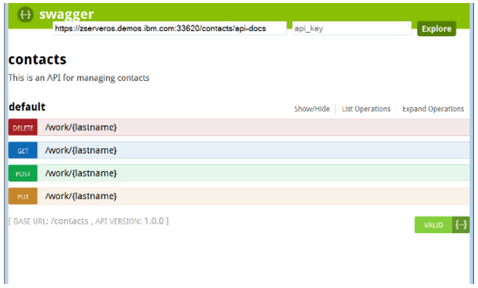
GET /work/{lastname}

POST /work/{lastname}

PUT /work/{lastname}

DELETE /work/{lastname}

Parameters



https://zserveros.demos.ibm.com:33620/contacts/api-docs

contacts

This is an API for managing contacts

default

DELETE /work/{lastname}

GET /work/{lastname}

POST /work/{lastname}

PUT /work/{lastname}

[BASE URL: /contacts , API VERSION: 1.0.0]

JS
iOS
Scala
php

Db2 RESTful Service Support

DB2 native RESTful service functionality **maintaining z/OS Connect externals** as applicable and

- Added by PI66828, PI70477 (Db2 11) or PI70652 (Db2 12)

- Is ready for use out of the box

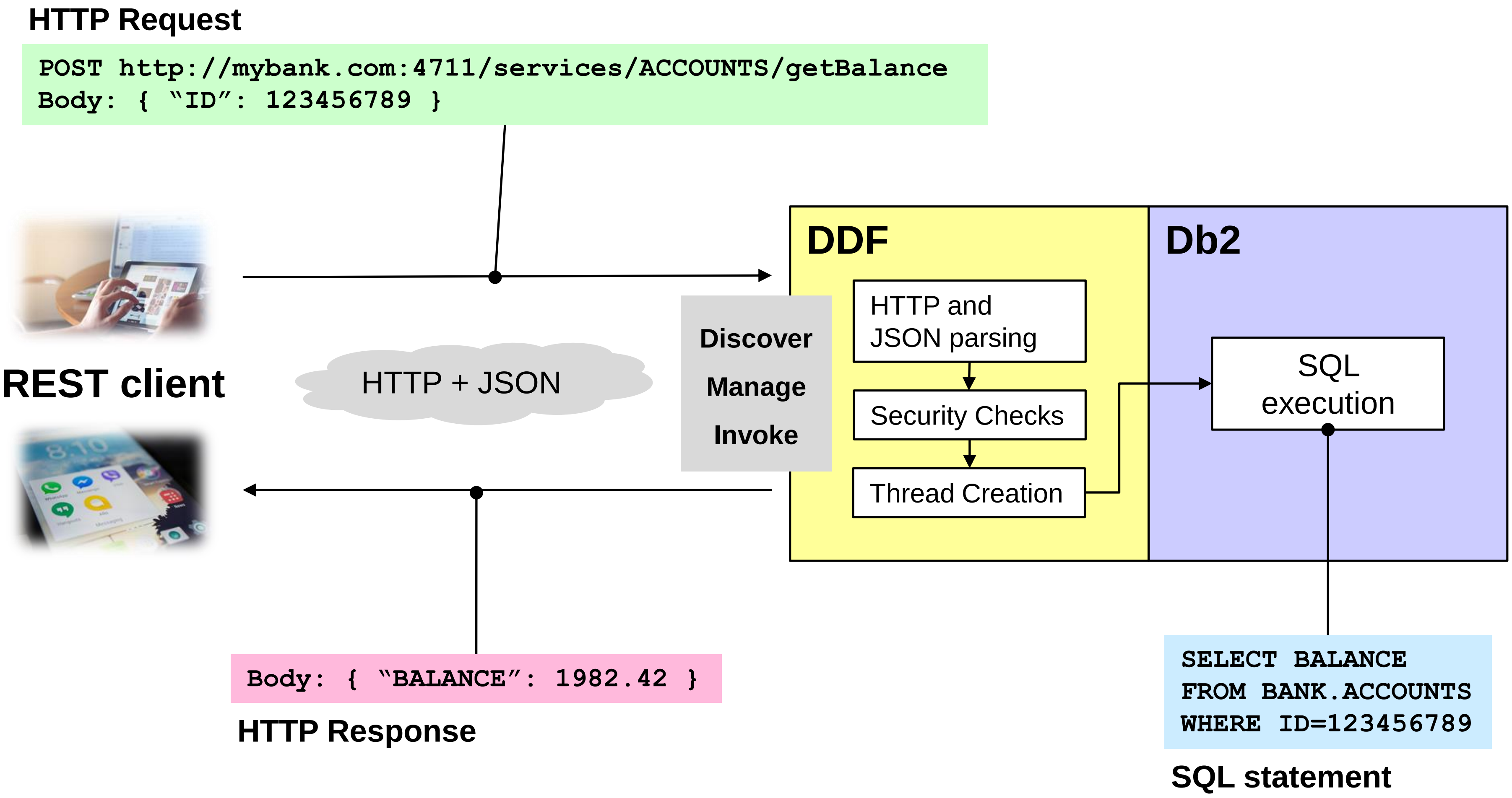
- Reuses DDF functionality that provides a highly available, highly scalable and highly serviceable network interface

- DDF provides security, auditing, and data transformation as being provided using z/OS Connect
  - PERMIT DSN.REST CLASS(DSNR) ID(DB2USER) ACCESS(READ)

- Support of clientInfo fields

- Allows a database administrator to manage services
  - New catalog table SYSIBM.DSNSERVICE
  - Static execution pattern – auth checking, object dependency tracking

Db2 RESTful Service Support



HTTP Request

```
POST http://mybank.com:4711/services/ACCOUNTS/getBalance
Body: { “ID”: 123456789 }
```

REST client

HTTP + JSON

Discover

Manage

Invoke

DDF

HTTP and JSON parsing

Security Checks

Thread Creation

Db2

SQL execution

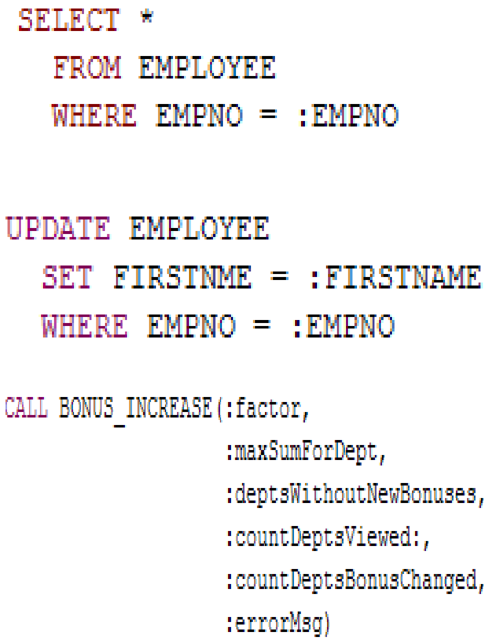
```
Body: { “BALANCE”: 1982.42 }
```

HTTP Response

```
SELECT BALANCE
FROM BANK.ACCOUNTS
WHERE ID=123456789
```

SQL statement

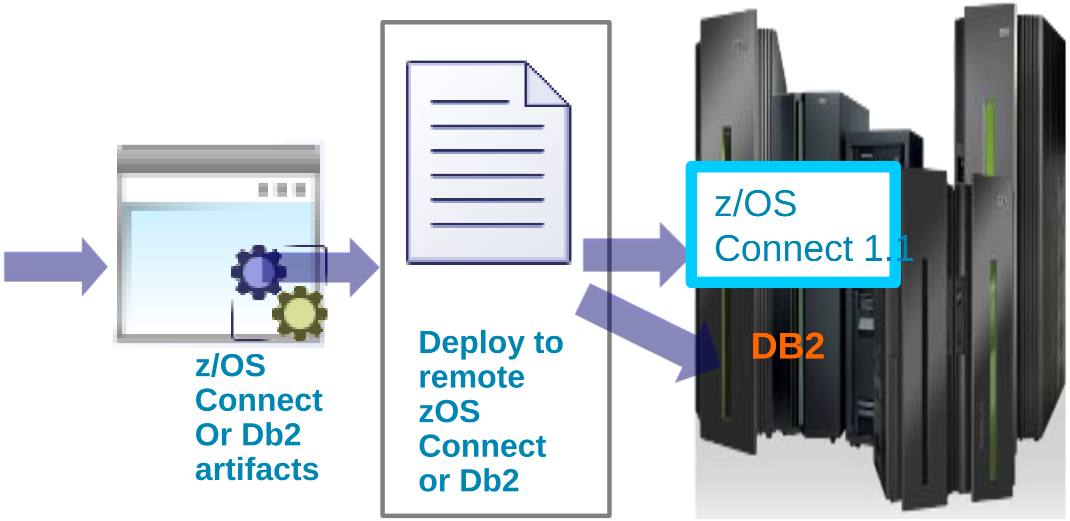
Data Studio Tooling for z/OS Connect – End to End Development



```
SELECT *
  FROM EMPLOYEE
  WHERE EMPNO = :EMPNO

UPDATE EMPLOYEE
  SET FIRSTNME = :FIRSTNAME
  WHERE EMPNO = :EMPNO

CALL BONUS_INCREASE(:factor,
                    :maxSumForDept,
                    :deptsWithoutNewBonuses,
                    :countDeptsViewed:,
                    :countDeptsBonusChanged,
                    :errorMsg)
```


z/OS Connect Or Db2 artifacts
Deploy to remote zOS Connect or Db2
z/OS Connect 1.1
DB2

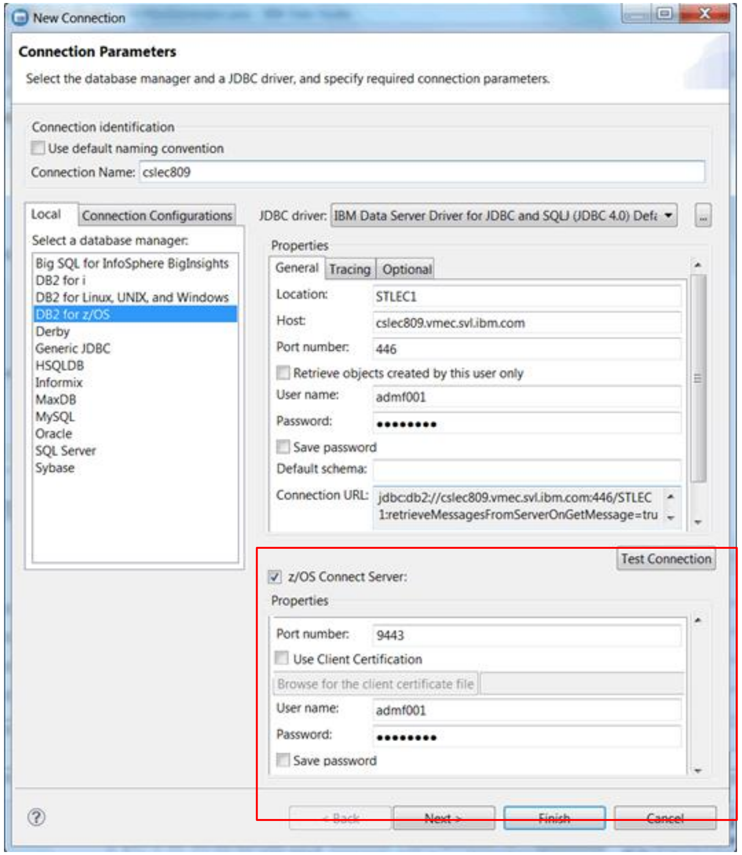
Database operations

Data Web Services of Data Studio

z/OS Connect on LPAR

User Interface

1. Define z/OS Connect together with Database connection



New Connection

Connection Parameters

Select the database manager and a JDBC driver, and specify required connection parameters.

Connection identification

□ Use default naming convention

Connection Name: cslec809

Select a database manager:

Big SQL for InfoSphere BigInsights
DB2 for i
DB2 for Linux, UNIX, and Windows
DB2 for z/OS
Derby
Generic JDBC
HSQLDB
Informix
MaxDB
MySQL
Oracle
SQL Server
Sybase

JDBC driver: IBM Data Server Driver for JDBC and SQLJ (JDBC 4.0) Defa

Properties

Location: STLEC1

Host: cslec809.vmec.svl.ibm.com

Port number: 446

□ Retrieve objects created by this user only

User name: admf001

Password: ●●●●●●●●●

□ Save password

Default schema:

Connection URL: jdbc:db2://cslec809.vmec.svl.ibm.com:446/STLEC1:retrieveMessagesFromServerOnGetMessage=tru

☑ z/OS Connect Server:

Properties

Port number: 9443

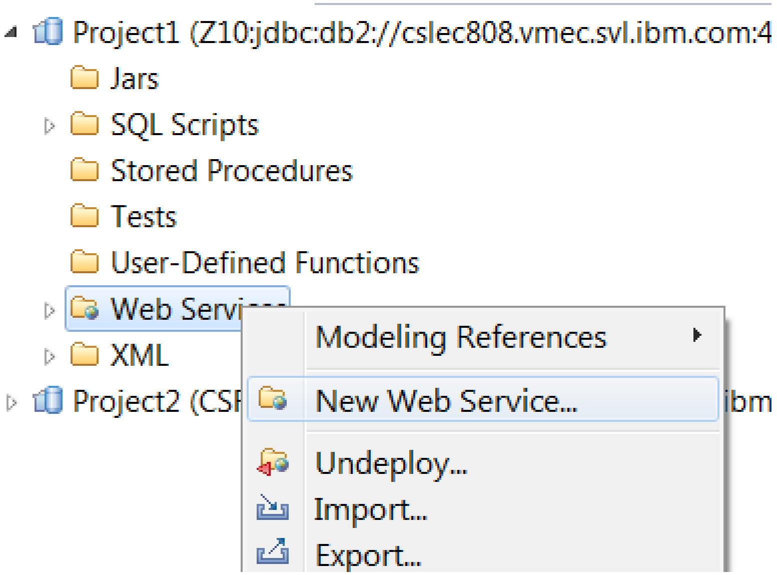
□ Use Client Certification

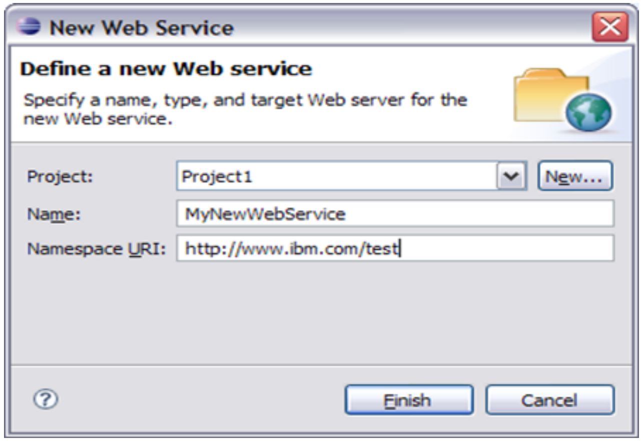
User name: admf001

Password: ●●●●●●●●●

□ Save password

2. Define a new service by right click on Web Services folder





New Web Service

Define a new Web service

Specify a name, type, and target Web server for the new Web service.

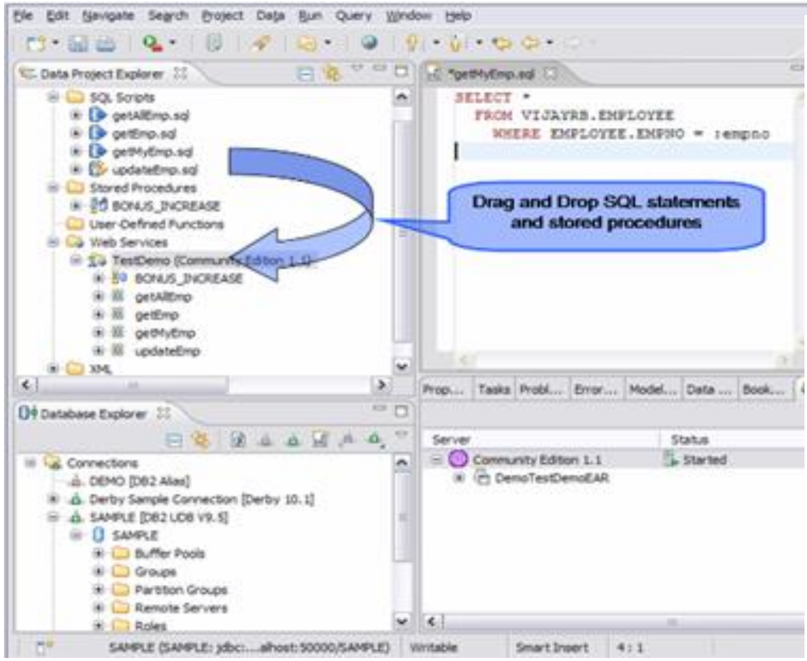
Project: Project1

Name: MyNewWebService

Namespace URI: http://www.ibm.com/test

User Interface…

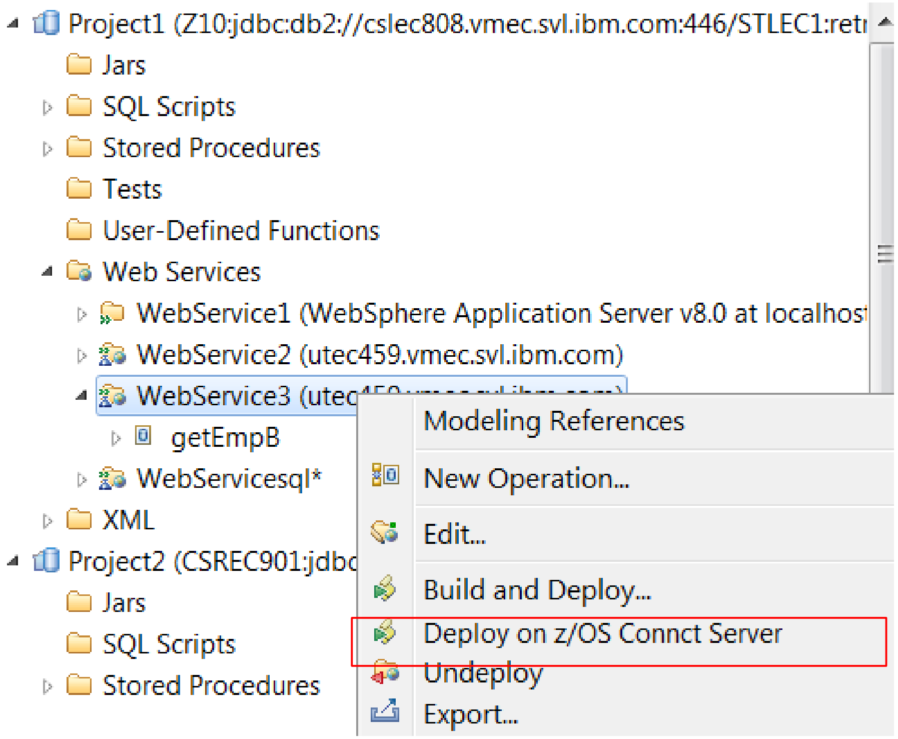
3. Drag ‘n drop



```
SELECT *
  FROM VIJAYRB.EMPLOYEE
    WHERE EMPLOYEE.EMPNO = :empno
```

Drag and Drop SQL statements and stored procedures

4. Right click to Deploy in z/OS Connect



Project1 (Z10:jdbc:db2://cslec808.vmec.svl.ibm.com:446/STLEC1:retr
Jars
SQL Scripts
Stored Procedures
Tests
User-Defined Functions
Web Services
WebService1 (WebSphere Application Server v8.0 at localhost
WebService2 (utec459.vmec.svl.ibm.com)
WebService3 (utec459.vmec.svl.ibm.com)
getEmpB
WebServicesql*
XML
Project2 (CSREC901:jdbc
Jars
SQL Scripts
Stored Procedures
Modeling References
New Operation...
Edit...
Build and Deploy...
Deploy on z/OS Connct Server
Undeploy
Export...

Service definitions are deployed in z/OS Connect – ready to use

DB2 Native RESTful Service Support …

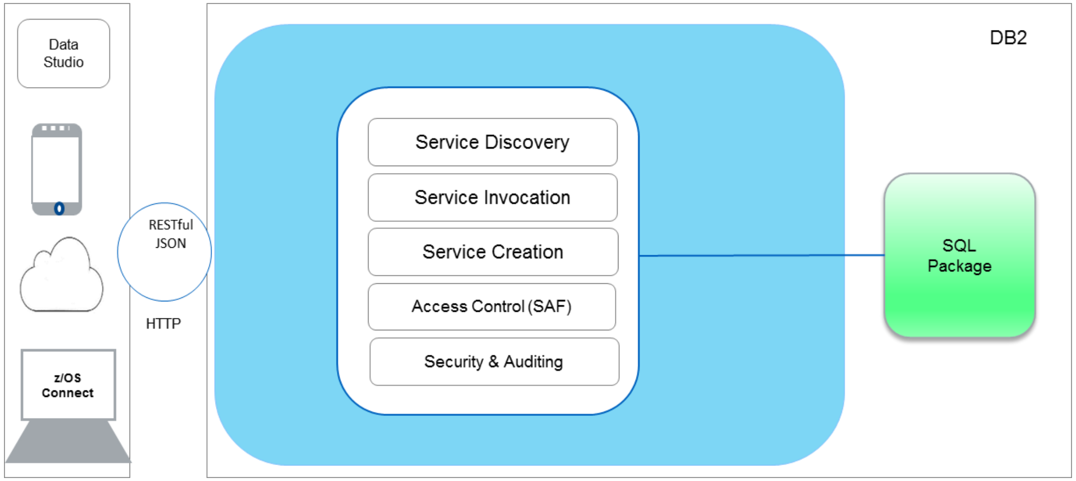
- Create|DROP|START|STOP Service using either Data Studio support, or direct DB2 RESTful service call

  POST https://<host>:<port>/services/DB2ServiceManager
  {"requestType": "createService«....}

- Discover all services and discover details for a specific service using direct DB2 RESTful service call

  POST https://<host>:<port>/services/DB2ServiceDiscover
  { ["collectionID": "<collectionID>"]

- DB2 RESTful service invocation.
  - DB2 Adapter for z/OS Connect V1 invocation API syntax
  - New DB2 native REST invoke API.



Creating a Db2 RESTful service using BIND

```
BIND SERVICE(collection-id) NAME(service-name)
[SQLDDNAME(ddname)]
```

- SQLDDNAME specifies the JCL DD containing the SQL
  - Can be inline, sequential data set, library member, zFS or UNIX file
- Supports long list of common BIND parameters
- Service can be deleted using FREE SERVICE command

```
FREE SERVICE(collection-id.service-name)
```

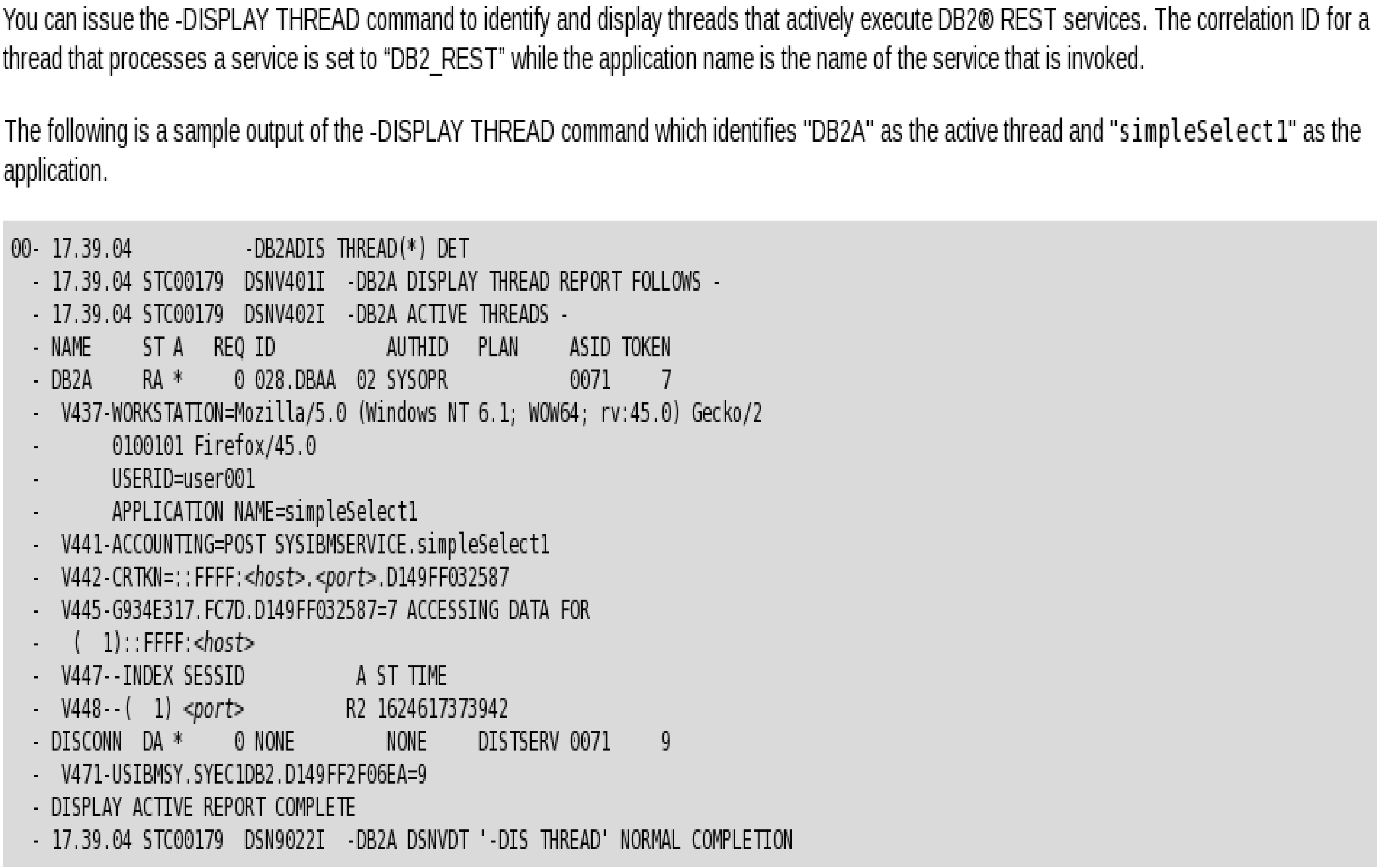
- Added by PI86867 (Db2 11) and PI86868 (Db2 12)

Example Create using DSN BIND SERVICE

```
//CR8SRVC EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD  DSN=DB2A.SDSNEXIT,DISP=SHR
//         DD  DSN=DB2A.SDSNLOAD,DISP=SHR
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//DSNSTMT  DD  DSN=SYSADM.SERVICE.SQL(SELECT1),
//             DISP=SHR
//SYSTSIN  DD  *
 DSN SYSTEM(DB2A)

 BIND SERVICE(SYSIBMSERVICE) -
  NAME("simpleSelect1") -
  SQLENCODING(1047) -
  DESCRIPTION('return a list of deptname-
 based on input location')
/*
```

Display of DB2 Native REST service locations and threads



You can issue the -DISPLAY THREAD command to identify and display threads that actively execute DB2® REST services. The correlation ID for a thread that processes a service is set to "DB2_REST" while the application name is the name of the service that is invoked.

The following is a sample output of the -DISPLAY THREAD command which identifies "DB2A" as the active thread and "`simpleSelect1`" as the application.

```
00- 17.39.04           -DB2ADIS THREAD(*) DET
  - 17.39.04 STC00179  DSNV401I  -DB2A DISPLAY THREAD REPORT FOLLOWS -
  - 17.39.04 STC00179  DSNV402I  -DB2A ACTIVE THREADS -
  - NAME     ST A   REQ ID           AUTHID   PLAN     ASID TOKEN
  - DB2A     RA *     0 028.DBAA  02 SYSOPR            0071    7
  -  V437-WORKSTATION=Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/2
  -       0100101 Firefox/45.0
  -       USERID=user001
  -       APPLICATION NAME=simpleSelect1
  -  V441-ACCOUNTING=POST SYSIBMSERVICE.simpleSelect1
  -  V442-CRTKN=::FFFF:<host>.<port>.D149FF032587
  -  V445-G934E317.FC7D.D149FF032587=7 ACCESSING DATA FOR
  -   (  1)::FFFF:<host>
  -  V447--INDEX SESSID           A ST TIME
  -  V448--(  1) <port>          R2 1624617373942
  - DISCONN  DA *     0 NONE         NONE     DISTSERV 0071    9
  -  V471-USIBMSY.SYEC1DB2.D149FF2F06EA=9
  - DISPLAY ACTIVE REPORT COMPLETE
  - 17.39.04 STC00179  DSN9022I  -DB2A DSNVDT '-DIS THREAD' NORMAL COMPLETION
```

DB2 Native RESTful Service Support

- Uses cloud and mobile application programming HTTP and HTTPS protocols
    - Needs to be HTTPS for connectivity from DataStudio
- zIIP eligibility of distributed workload
- All DB2 Native REST APIs uses static SQL packages
    - Dynamic SQL with SQL as parameter to SQL PL SP
- DB2 Native RESTful service integrates with z/OS Connect EE
    - Coming: Tooling to simplify service definition in z/OS Connect

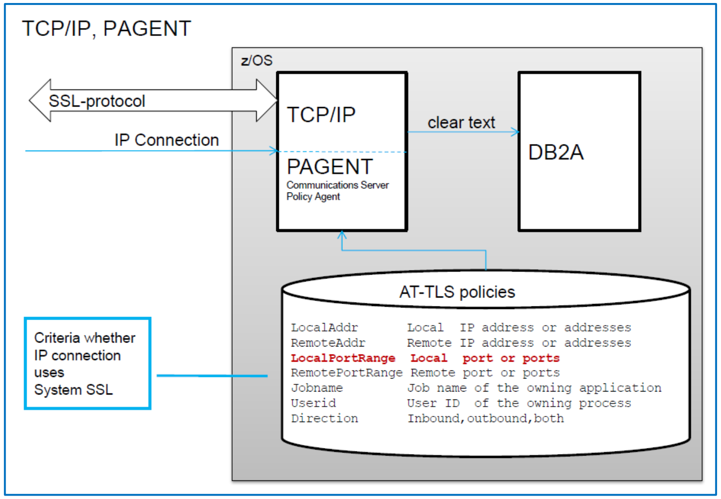
- **References**

The easiest way on the planet to create RESTful Web Services!

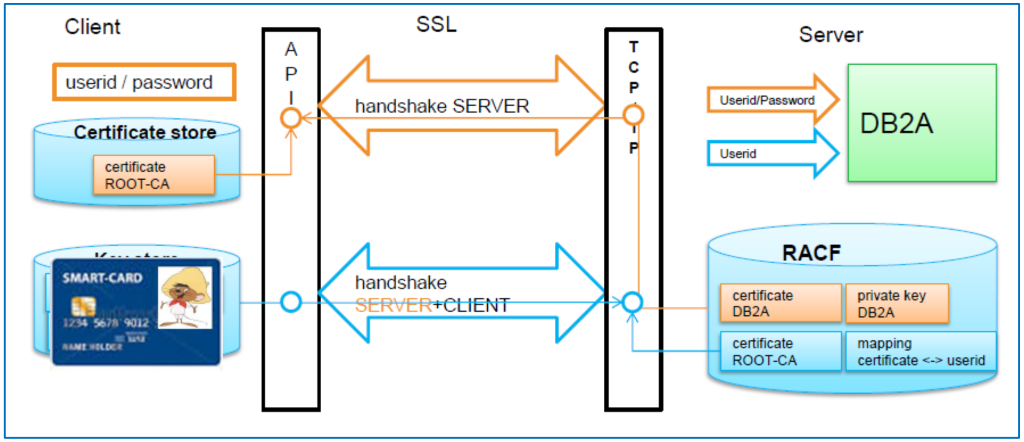
Security of DB2 Native REST service connections

DB2® supports HTTPS REST service requests by using the z/OS® Communications Server IP Application Transparent Transport Layer Security (AT-TLS) capability. The policy-driven AT-TLS support is configured in the TCP/IP stack and performs the TLS check on behalf of DB2 by invoking the z/OS system SSL feature in the TCP layer.

To support HTTPS requests, you must use a secure port for SSL connections. Make sure that the DDF TCP/IP SQL Listener service is capable of listening to a secondary secure port for inbound SSL connections. DDF verifies that all connections to DB2 through the secure port are protected by AT-TLS policies.



LocalAddr	Local IP address or addresses
RemoteAddr	Remote IP address or addresses
LocalPortRange	**Local port or ports**
RemotePortRange	Remote port or ports
Jobname	Job name of the owning application
Userid	User ID of the owning process
Direction	Inbound,outbound,both

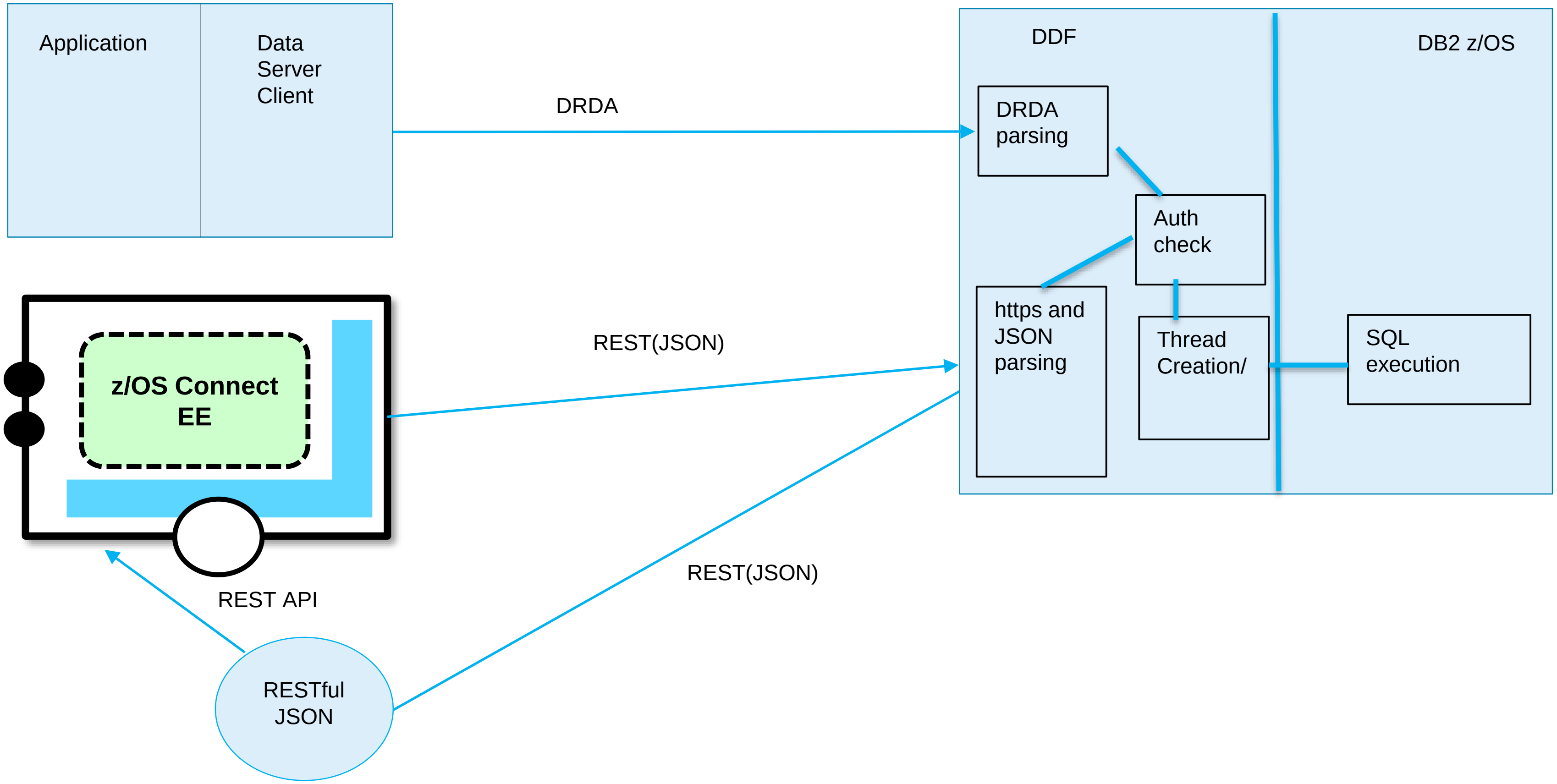


Setup SSL and client cert authentication
http://www.redbooks.ibm.com/redpapers/pdfs/redp4799.pdf
http://www.redbooks.ibm.com/redbooks/pdfs/sg248099.pdf (chapter 12)

DB2 Native RESTful Service Support …

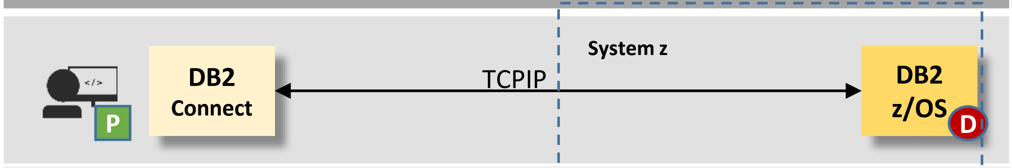
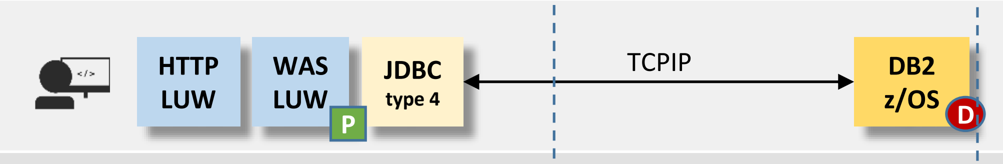
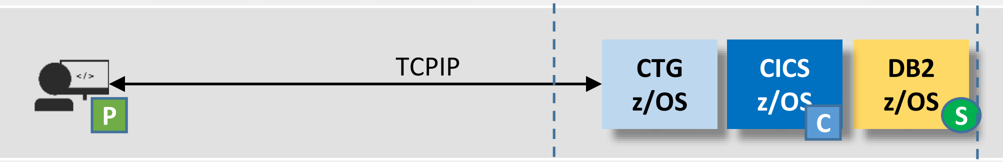
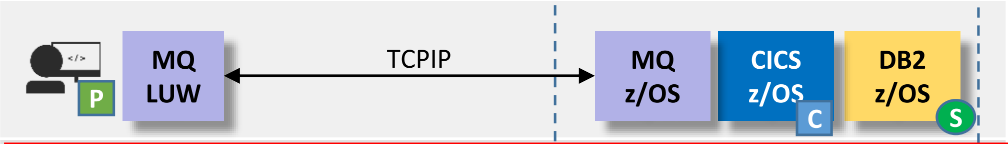
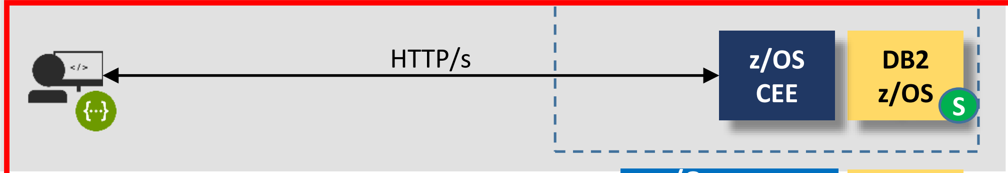
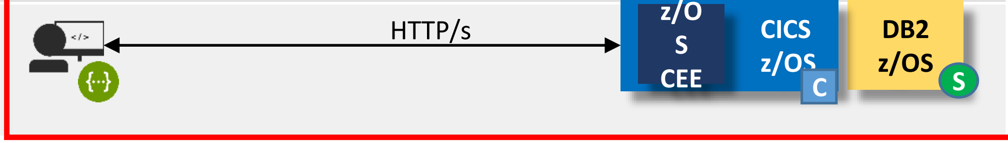
Alternative distributed connectivity to DB2,

- No need for DB2 drivers, DB2 Connect gateway or DB2 Connect license
- Service call direct or via z/OS Connect



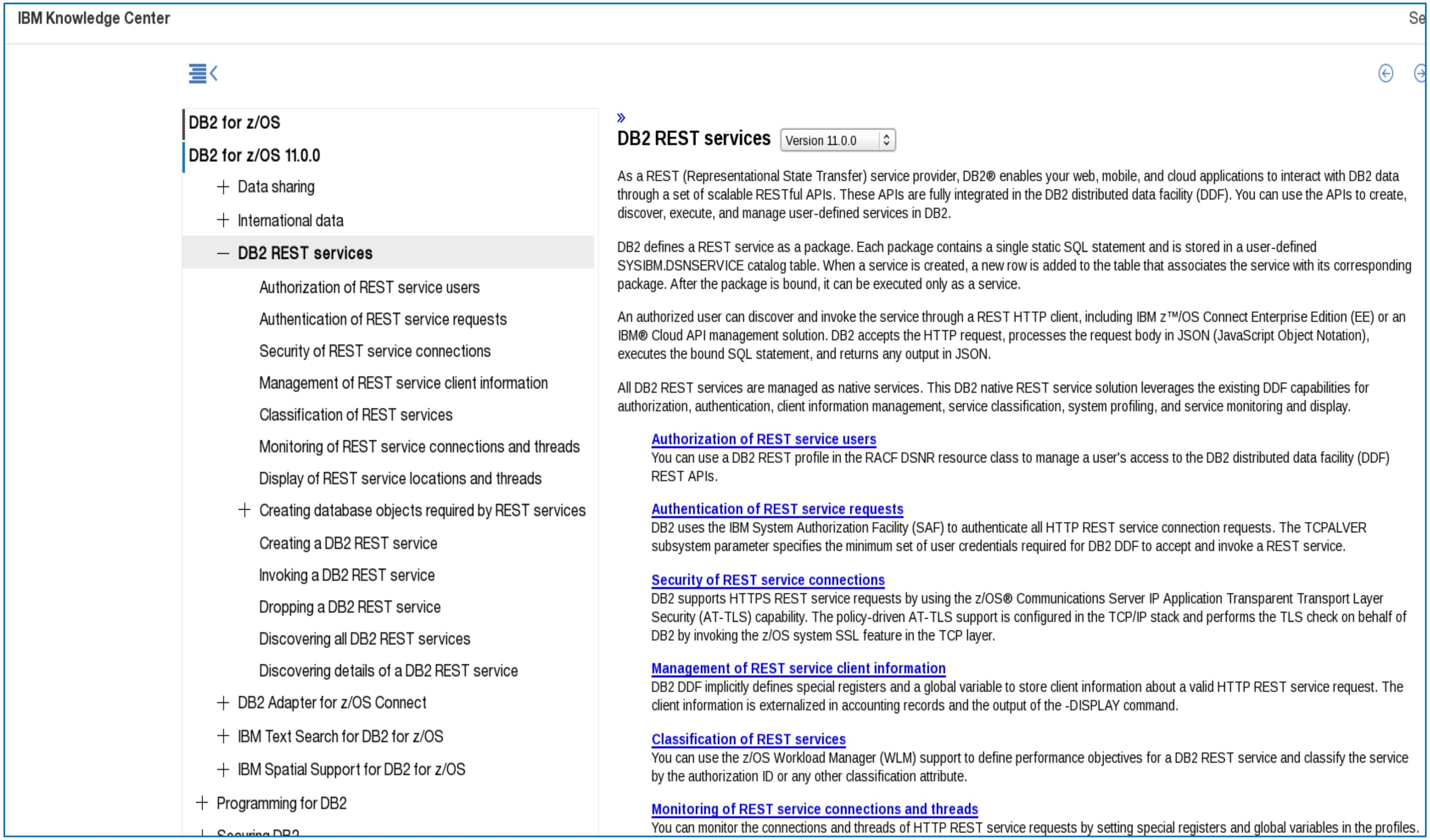
z/OS Connect EE Configuration Options with DB2 V11

{…} RESTAPI
P Program
C COBOL
S Static SQL
D Dynamic SQL

Diagram	Hop #	Pay load	Network Protocol	SQL Statement	zIIP Offload	Dev. Skill
 System z P DB2 Connect TCPIP DB2 z/OS D	3	DB2 nativ e data	TCPIP/ Sync Inbound	Coded in Client App. Dynamic SQL	DDF connection part only	Easy
 HTTP LUW WAS LUW P JDBC type 4 TCPIP DB2 z/OS D	4	DB2 nativ e data	TCPIP/ Sync Inbound	Coded in Client App. Dynamic SQL	DDF connection part only	Easy+
 P TCPIP CTG z/OS CICS z/OS C DB2 z/OS S	4	CTG nativ e data	TCPIP/ Sync Inbound	Coded in CICS App. Static SQL	Some CTG portion	Mode rate
 P MQ LUW TCPIP MQ z/OS CICS z/OS C DB2 z/OS S	5	MQ MSG	TCPIP/ Asyn In & Out bound	Coded in CICS App. Static SQL	All GCP	Mode rate+
 {…} HTTP/s z/OS CEE DB2 z/OS S	3	REST/ JSON	HTTP/ Sync Inbound*	Defined in DB2 z/OS. Static SQL	DDF connection partly. 90%+ on z/OSCEE	Super Easy
 {…} HTTP/s z/O S CEE CICS z/OS C DB2 z/OS S	4	REST/ JSON	HTTP/ Sync Inbound*	Coded in CICS App. Static SQL	90%+ on z/OSCEE	Easy

Note* : zOSCEE is planning to support outbound. (But, subject to be changed)

DB2 11 for z/OS Knowledge Center – RESTful Services


IBM Knowledge Center
Se
DB2 for z/OS
DB2 for z/OS 11.0.0
+ Data sharing
+ International data
– DB2 REST services
Authorization of REST service users
Authentication of REST service requests
Security of REST service connections
Management of REST service client information
Classification of REST services
Monitoring of REST service connections and threads
Display of REST service locations and threads
+ Creating database objects required by REST services
Creating a DB2 REST service
Invoking a DB2 REST service
Dropping a DB2 REST service
Discovering all DB2 REST services
Discovering details of a DB2 REST service
+ DB2 Adapter for z/OS Connect
+ IBM Text Search for DB2 for z/OS
+ IBM Spatial Support for DB2 for z/OS
+ Programming for DB2
+ Securing DB2
Version 11.0.0

DB2 REST services

As a REST (Representational State Transfer) service provider, DB2® enables your web, mobile, and cloud applications to interact with DB2 data through a set of scalable RESTful APIs. These APIs are fully integrated in the DB2 distributed data facility (DDF). You can use the APIs to create, discover, execute, and manage user-defined services in DB2.

DB2 defines a REST service as a package. Each package contains a single static SQL statement and is stored in a user-defined SYSIBM.DSNSERVICE catalog table. When a service is created, a new row is added to the table that associates the service with its corresponding package. After the package is bound, it can be executed only as a service.

An authorized user can discover and invoke the service through a REST HTTP client, including IBM z™/OS Connect Enterprise Edition (EE) or an IBM® Cloud API management solution. DB2 accepts the HTTP request, processes the request body in JSON (JavaScript Object Notation), executes the bound SQL statement, and returns any output in JSON.

All DB2 REST services are managed as native services. This DB2 native REST service solution leverages the existing DDF capabilities for authorization, authentication, client information management, service classification, system profiling, and service monitoring and display.

Authorization of REST service users
You can use a DB2 REST profile in the RACF DSNR resource class to manage a user's access to the DB2 distributed data facility (DDF) REST APIs.

Authentication of REST service requests
DB2 uses the IBM System Authorization Facility (SAF) to authenticate all HTTP REST service connection requests. The TCPALVER subsystem parameter specifies the minimum set of user credentials required for DB2 DDF to accept and invoke a REST service.

Security of REST service connections
DB2 supports HTTPS REST service requests by using the z/OS® Communications Server IP Application Transparent Transport Layer Security (AT-TLS) capability. The policy-driven AT-TLS support is configured in the TCP/IP stack and performs the TLS check on behalf of DB2 by invoking the z/OS system SSL feature in the TCP layer.

Management of REST service client information
DB2 DDF implicitly defines special registers and a global variable to store client information about a valid HTTP REST service request. The client information is externalized in accounting records and the output of the -DISPLAY command.

Classification of REST services
You can use the z/OS Workload Manager (WLM) support to define performance objectives for a DB2 REST service and classify the service by the authorization ID or any other classification attribute.

Monitoring of REST service connections and threads
You can monitor the connections and threads of HTTP REST service requests by setting special registers and global variables in the profiles.

II14827 – Db2 DDF native REST Info APAR, documents known restrictions, limitations, etc.

Demo video resources

- Db2 11 and 12 for z/OS Native REST Services – Part #1
  Demonstrates how to create and invoke Db2 native REST services using SQL and stored procedures
  https://youtu.be/6XkiEysnsCg

- Db2 11 and 12 for z/OS Native REST Services and z/OS Connect EE 2.0 – Part #2:
  The video shows how to enhance Db2 REST services using z/OS Connect EE 2.0 and transform them into REST APIs.
  https://youtu.be/91ji0w01E3A

Summary

- Emerging business model “API Economy” - leveraging APIs to generate new business
  - Companies are now exposing some of their key application assets as services to other users/application developers or external businesses

- DB2 Adapter for z/OS Connect is available for DB2 10 and DB2 11
  - Exploits z/OS Connect to REST(JSON)-enable SQL and Stored Procedures
  - It is a good start to get familiar with functionality when still on DB2 10

- DB2 11 RESTful service support provides a new native interface into DB2 for z/OS
  - Recommended functionality for DB2 11 NFM or later

- Integrates with z/OS Connect to provide a consistent REST(JSON) interface into z/OS- based business functions and development tooling support