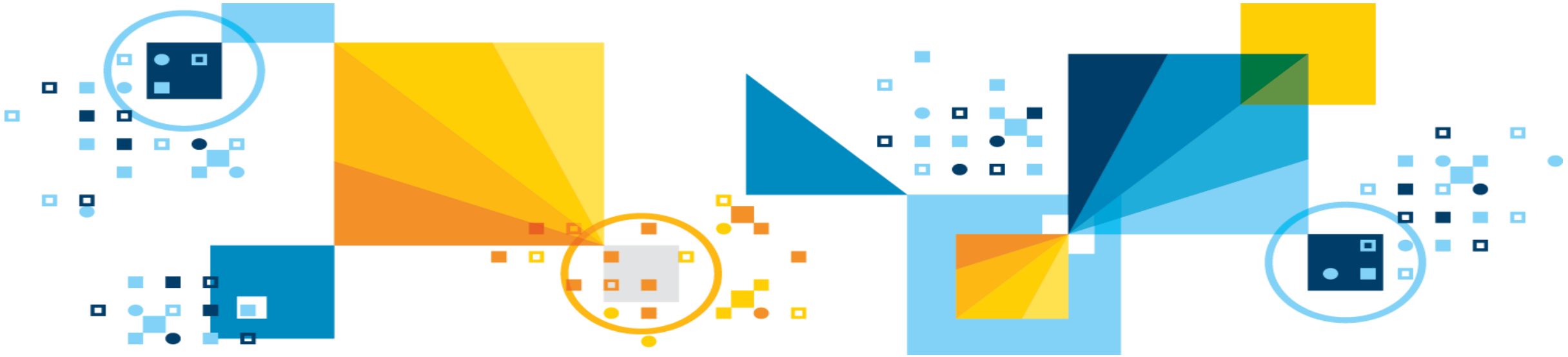
Dallas-area DB2 users group
April 19, 2017

DB2 for z/OS and ____-as-a-Service

Robert Catterall
IBM Senior Consulting DB2 for z/OS Specialist



Agenda

- The distinction between data-as-a-service and database-as-a-service
- DB2 for z/OS and data-as-a-service
- DB2 for z/OS and database-as-a-service

The distinction between data-as-a-service and database-as-a-service

Data-as-a-service versus database-as-a-service

- The key distinction is in the name
  - With DBaaS, you want the functionality of a database management system, provided as a service
- What does “as a service” mean, in a DBaaS context?

  - It can mean an off-premise cloud deployment of the DBMS
    - An example of an IBM offering of this nature is dashDB
    - dashDB for Transactions is available on IBM’s Bluemix cloud
    - dashDB for Analytics is available on Bluemix and also on AWS
  - “As a service” can also mean an on-premise deployment
    - Again, IBM’s dashDB plays here, in the form called dashDB Local – a data warehouse solution that can be deployed, via Docker container technology, where you want it (private cloud, virtual private cloud, software-defined infrastructure)
    - A key objective of database-as-a-service in an on-premise context is ease and speed of provisioning – a dashDB Local environment can be deployed in minutes

Data-as-a-service

- Compared to DBaaS, DaaS is more about *the programmatic interface to data server*
- "Database" is not part of the term, because there is no need (or desire on the part of a programmer) to know that a database is on the other end of a data request
  - Might be a database (could be relational, like DB2, or hierarchical, like IMS)
  - Might be a file system (such as VSAM in a z/OS system)
  - Might be a Hadoop-managed data store
  - Might be none of the above

=

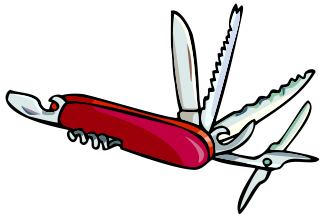
It doesn't matter. An application developer simply wants to invoke a data service of some kind (create, read, update, delete data) via a straightforward and consistent interface, *regardless of the mechanism by which the request is executed*.

REST – Representational State Transfer – provides that straightforward and consistent service invocation interface

REST and one of its antecedents: SOAP

- Not too many years ago, a dominant mode of service invocation from an application was Simple Object Access Protocol, or SOAP
  - The thing is, SOAP is not all that “simple” from a programmer’s perspective
  - Among other things, it involves the use of XML documents
    - XML is robust, but not always easy to use
  - Additionally, SOAP is designed to be neutral with regard to communications protocols
    - Being able to use it with communications protocols such as SMTP or JMS might be helpful in some cases, but what if you just want to use HTTP?

- The fact of the matter is, SOAP came to be seen as a “heavyweight” protocol for service invocation, with quite a lot in the way of attendant baggage
  - REST is more specialized and focused, and very much slimmed down versus SOAP

Sometimes you want this  instead of this 

RESTful services – front-end perspective

URI = Uniform resource identifier

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

- With REST, a service is invoked by way of a URI, which is appended to the URL of an HTTP request
- If the URI is understood by the receiving server, the requested action is taken

What about data “payloads” (input/output) for REST calls?

URI = Uniform resource identifier

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

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "1542 Main Street",
        "city": "Anytown",
        "state": "NY",
        "postalCode": "10021-1004"
    },
}
```

- Data associated with REST calls is sent in JSON format (JavaScript Object Notation) – a series of name/value pairs
- Input data is appended to the URL associated with the REST call
- Output data is returned to the requester in JSON format

DB2 for z/OS and data-as-a-service

DB2's native REST interface

- Introduced with DB2 12 for z/OS, retrofitted to DB2 11 via the fix for APAR PI66828
- An extension of DB2 distributed data facility (DDF) functionality
  - Leverages existing DDF capabilities including thread pooling, profiling, classification, accounting and statistics tracing
  - Leverages existing DB2 package management capabilities (package = compiled form of DB2 static SQL statements)
    - DB2 RESTful services are recorded in DB2 catalog tables and stored in the DB2 directory
  - SQL statements executed by way of DB2 REST API calls run under preemptible SRBs in the DDF address space
    - SQL executing under DDF preemptible SRBs is up to 60% zIIP-eligible
- Designed for high performance
  - IBM tests: 540 million transactions per hour through the DB2 for z/OS REST API

A closer look at DB2 for z/OS RESTful services

- A single static SQL statement can be exposed for execution via a REST call
  - Could be a single SQL DML statement (SELECT, INSERT, UPDATE, DELETE)
  - Could be a call to a DB2 stored procedure
    - In that case, I'd recommend a native SQL procedure, to get zIIP offload
- Not just RESTful service creation – also support for service discovery
  - Allows client-side developers to get information about function provided by a service, input data required, and content and form of output data
- Also access control
  - Authorize users of services

Where z/OS Connect fits in

- For one thing, z/OS Connect EE adds value to DB2's built-in REST API
  - DB2 can be a REST provider to z/OS Connect
  - z/OS Connect EE provides capabilities beyond DB2's for managing, discovering, securing, and auditing DB2-provided RESTful services
  - z/OS Connect also makes life easier for client-side and server-side developers
    - Client-side: service discovery via the Open API Initiative's Swagger specification
    - Client-side: RESTful services can be invoked via the full range of HTTP verbs (for example, GET and PUT – DB2's native REST interface only supports POST), so REST calls can be more intuitive
    - Server-side: intuitive, workstation-based tooling that facilitates creation of REST APIs from DB2 SQL statements

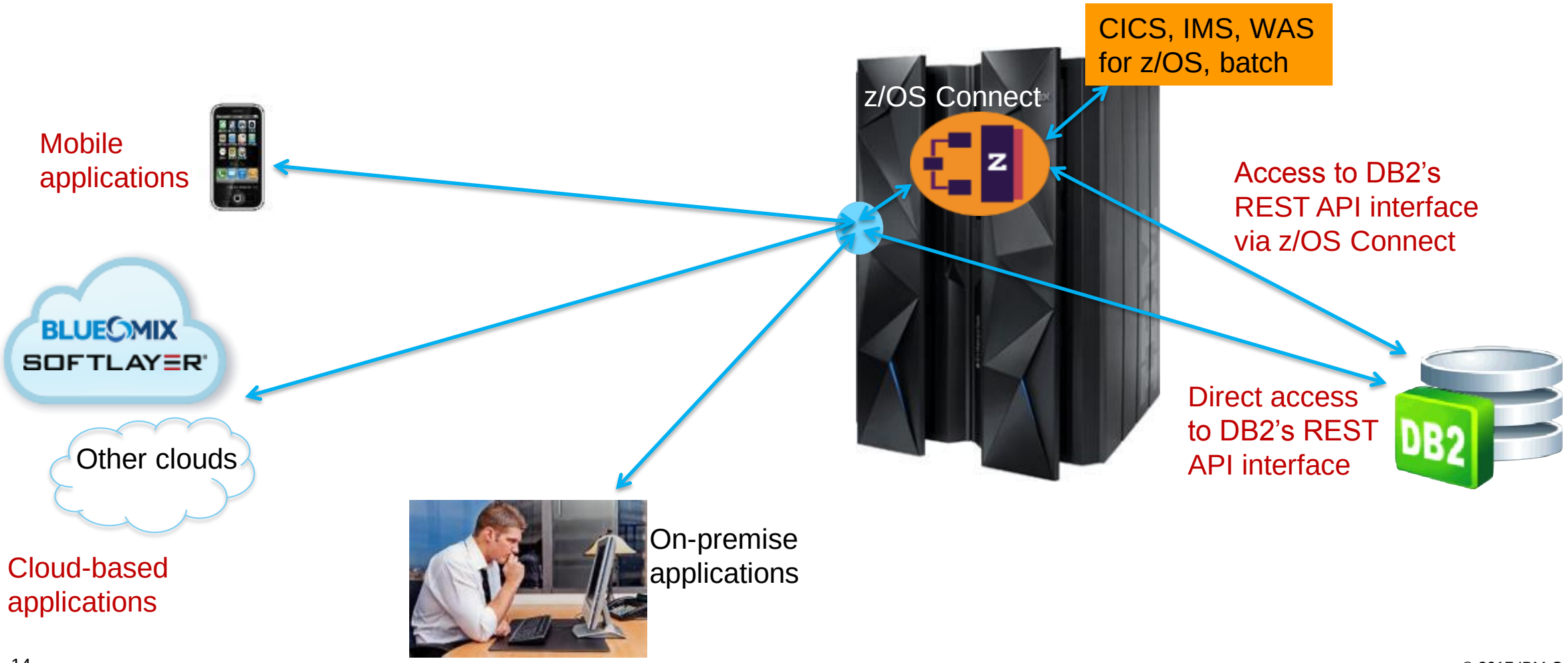
More on z/OS Connect

- z/OS Connect Enterprise Edition (EE) expands the range of z/OS-based programmatic assets that can be exposed as RESTful services
  - CICS transactions (might access DB2 data, might access VSAM data)
  - IMS transactions
  - WebSphere Application Server for z/OS transactions
  - Batch jobs

The big picture

Accessing DB2 data with RESTful APIs and JSON

z/OS Connect also makes other z/OS-based data processing services available through REST APIs



What about DB2 Connect (or the IBM Data Server Driver)?

- DB2 Connect (and the IBM Data Server Driver) continue to allow remote, network-connected applications to interact with DB2 for z/OS using non-DBMS-specific interfaces such as JDBC and ODBC
- Some situations will favor use of z/OS Connect, while in others DB2 Connect/Data Server Driver will be a better fit

z/OS Connect

✓REST APIs are simple, consistent

✓No SQL skills needed

✓Growing demand for data-as-a-service development model

✓Very well suited to cloud-based applications and applications with a mobile front-end

DB2 Connect/Data Server Driver

✓SQL skills required

✓Better workload isolation

✓High-volume transaction processing

✓Resource pooling

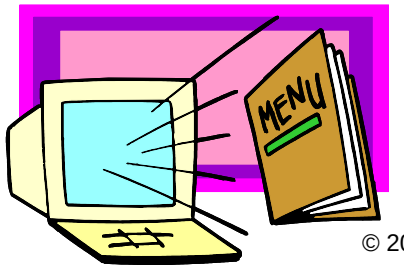
✓Sysplex workload balancing

✓Transaction fault-tolerance

DB2 for z/OS and database-as-a-service

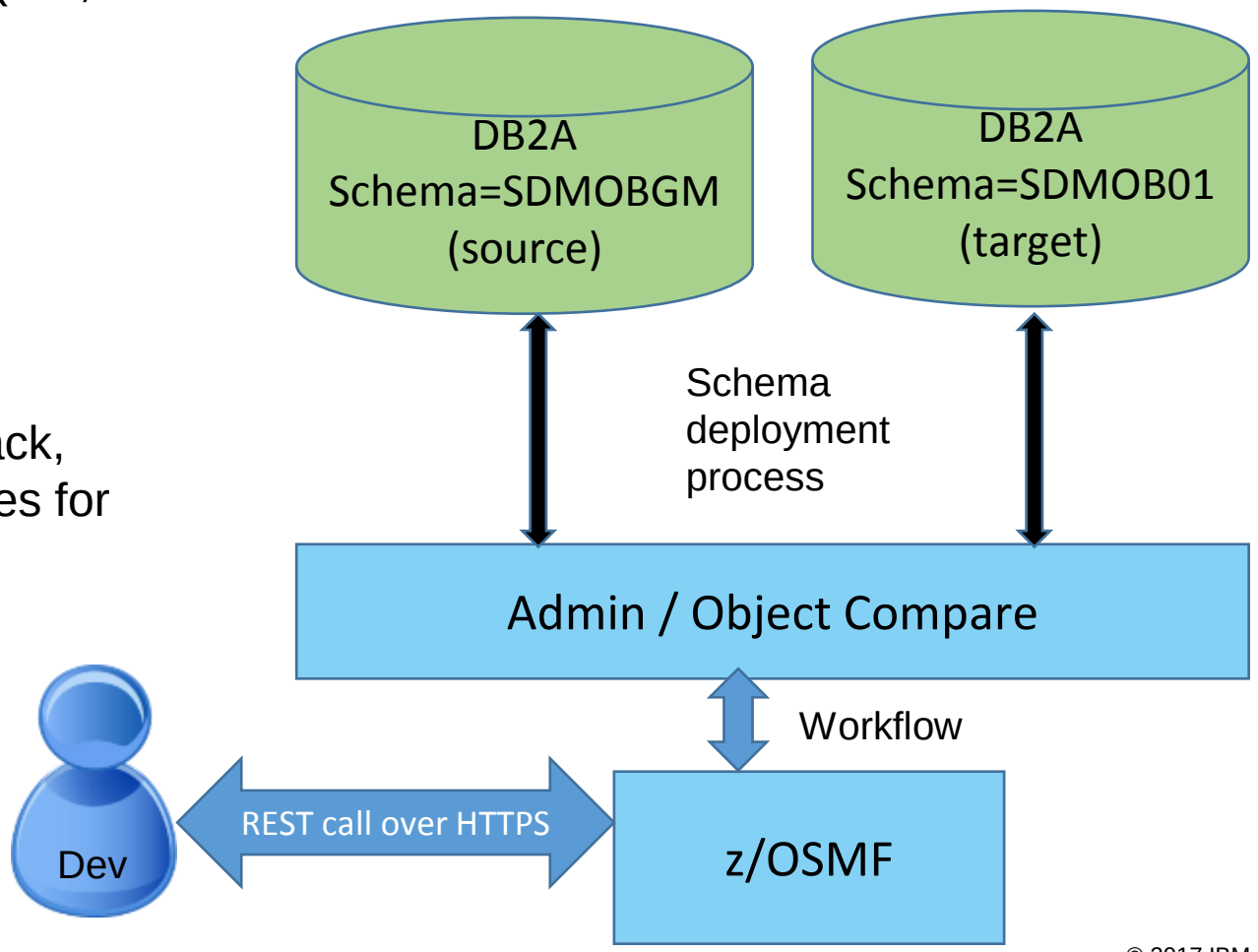
DB2 for z/OS and DBaaS

- IBM dashDB is a distributed systems technology
- Question: Is database-as-a-service meaningful in a DB2 for z/OS context?
- Answer: Yes
  - DBaaS in a DB2 for z/OS context is primarily about ease and speed of provisioning
  - That could mean creating a new DB2 environment (i.e., a new DB2 subsystem), or it could mean creating a new schema (a set of tables and related database objects) to support development of a new application
    - With regard to standing up a new subsystem, the DB2 12 for z/OS installation CLIST was enhanced to generate artifacts that can be used to automate subsystem installation by way of a z/OSMF workflow (z/OSMF is short for z/OS Management Facility, a feature of z/OS)
    - Creating a new schema is facilitated by DB2's ability to implicitly create databases, table spaces, indexes, and other items required for tables (and for data types such as LOBs and XML)



DB2 for z/OS DBaaS is still a work in progress

- An early objective: provide a service to enable easy creation of a new database schema (i.e., a set of tables and associated objects)
- The idea: utilize REST-enabled IBM DB2 Administration Tool and DB2 Object Comparison Tool
- Product requirements
  - IBM DB2 Change Management Solution Pack, including Administration Tool V11.2 with fixes for APARs PI67731, PI72396, and PI76054
  - DB2 for z/OS with fixes for these APARs:
    - DB2 11 – PI73168
    - DB2 12 – PI73492



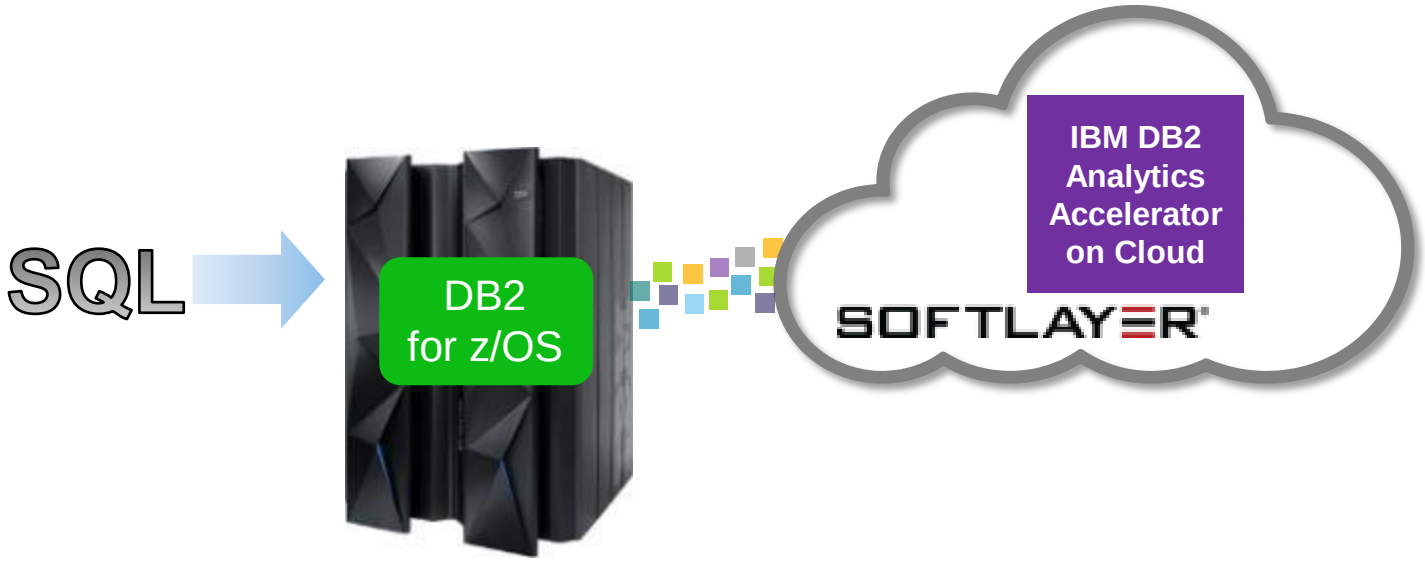
Future DB2 for z/OS DBaaS capabilities under consideration

- Deliver REST services allowing a developer to…
  - Provision/de-provision a DB2 application environment
  - Automate (via self-service) application deployments including schema changes
  - Configure automated backups
  - Snapshot backup/restore services
  - Monitor/add/remove storage
  - …

A DB2 for z/OS DBaaS capability that is available now

The IBM DB2 Analytics Accelerator on Cloud

- Like an on-premise Accelerator, the DB2 Analytics Accelerator on Cloud…
  - …is an extension of a DB2 for z/OS system
  - …is logically transparent (queries are directed to the front-end DB2 system)
  - …protects data through DB2 for z/OS security controls
  - …dramatically speeds execution of complex queries



Cloud deployment benefits:

- *Flexibility* (upsize/downsize on-demand)
- *Agility* (deploy within hours)
- *Secure cloud environment* (based on dedicated, bare-metal deployment)

Thanks for your time

Robert Catterall
rfcatter@us.ibm.com