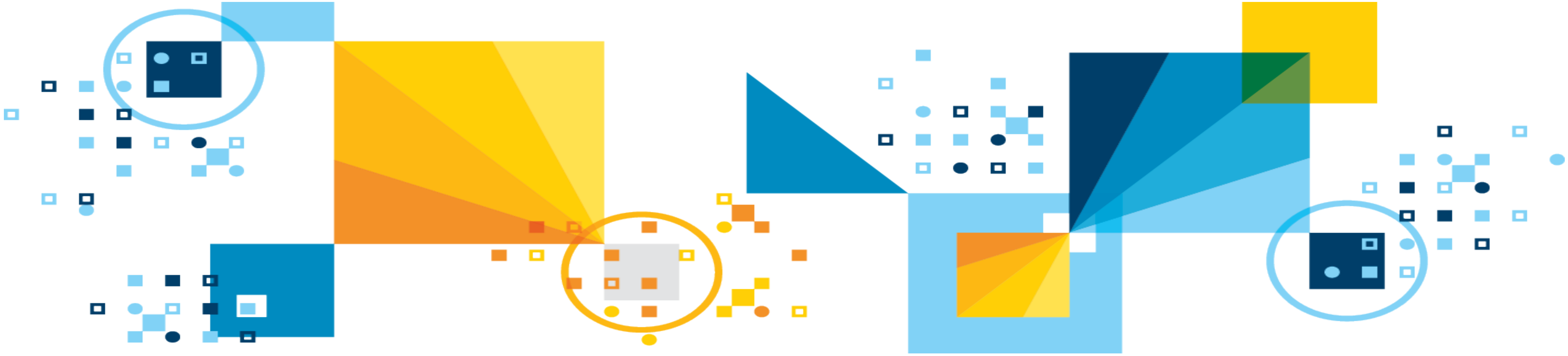
IBM Analytics

TRIDEXG
October 5, 2017

Db2 for z/OS Gets Agile: Data-as-a-Service and Database-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

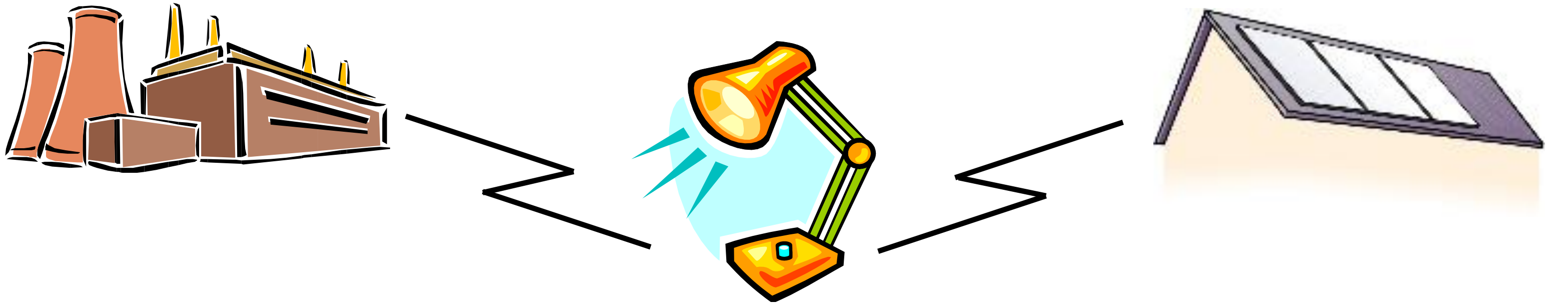
Data-as-a-service | database-as-a-service – not the same thing

- Key distinction is in the name
  - With DBaaS, you want the functionality of a database management system, provided as a service
- What does “provided 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 Db2 on Cloud (formerly dashDB)
  - “As a service” can also mean an on-premise deployment
    - Can you *really* do “as-a-service” in an on-premise way? Sure you can – the “where” of the service-providing mechanism is often not a primary consideration

See next slide

Database-as-a-service, on-prem and off

- Consider the point of view of consumers of the provided services
- Key objectives of database-as-a-service: easy requisitioning, fast provisioning
  - If consumers can easily request and quickly receive the database services they want, do they care if the requisitioning and provisioning mechanism is on-site or off site?
  - Analogy: if you want light to read a book, and you can get it easily (flip a switch) and quickly (light comes on right away), do you care if the electricity comes from a distant power plant or from solar panels on the roof?



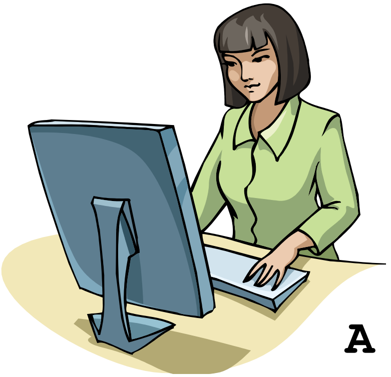
- How is the requisitioning and provisioning of database services made quick and easy?

(answer to come)

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 (often no 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

=



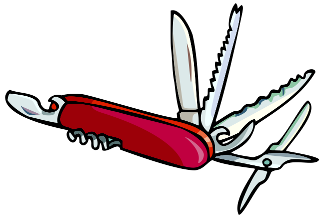
It doesn't matter. Many application developers just want to invoke a data service of some kind (create, read, update, delete data) via a straightforward and consistent interface, *regardless of the mechanism by which the request is executed.*

A straightforward and consistent service invocation interface that a lot of developers like is called REST (REpresentational State Transfer)

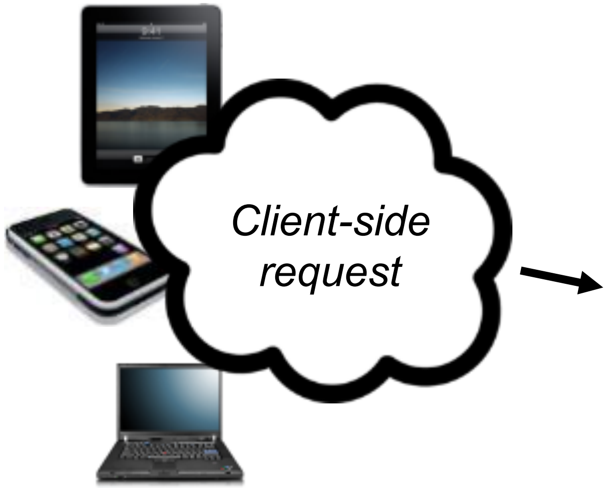
REST and one of its antecedents: SOAP

- 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?

- SOAP came to be seen as an overly “heavy” protocol for service invocation, with a good bit 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 – client-side 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?

Client-side request

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 can be 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, classification, accounting and statistics tracing
  - Leverages existing Db2 package management capabilities (package = compiled form of Db2 static SQL statements)
  - 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 data manipulation SQL statement (SELECT, INSERT, UPDATE, DELETE)
  - Could be a call to a Db2 stored procedure
    - In that case, I'd recommend a native SQL procedure (written in SQL PL), to get zIIP offload (runs under preemptible SRB in Db2 DDF address space when)
- 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

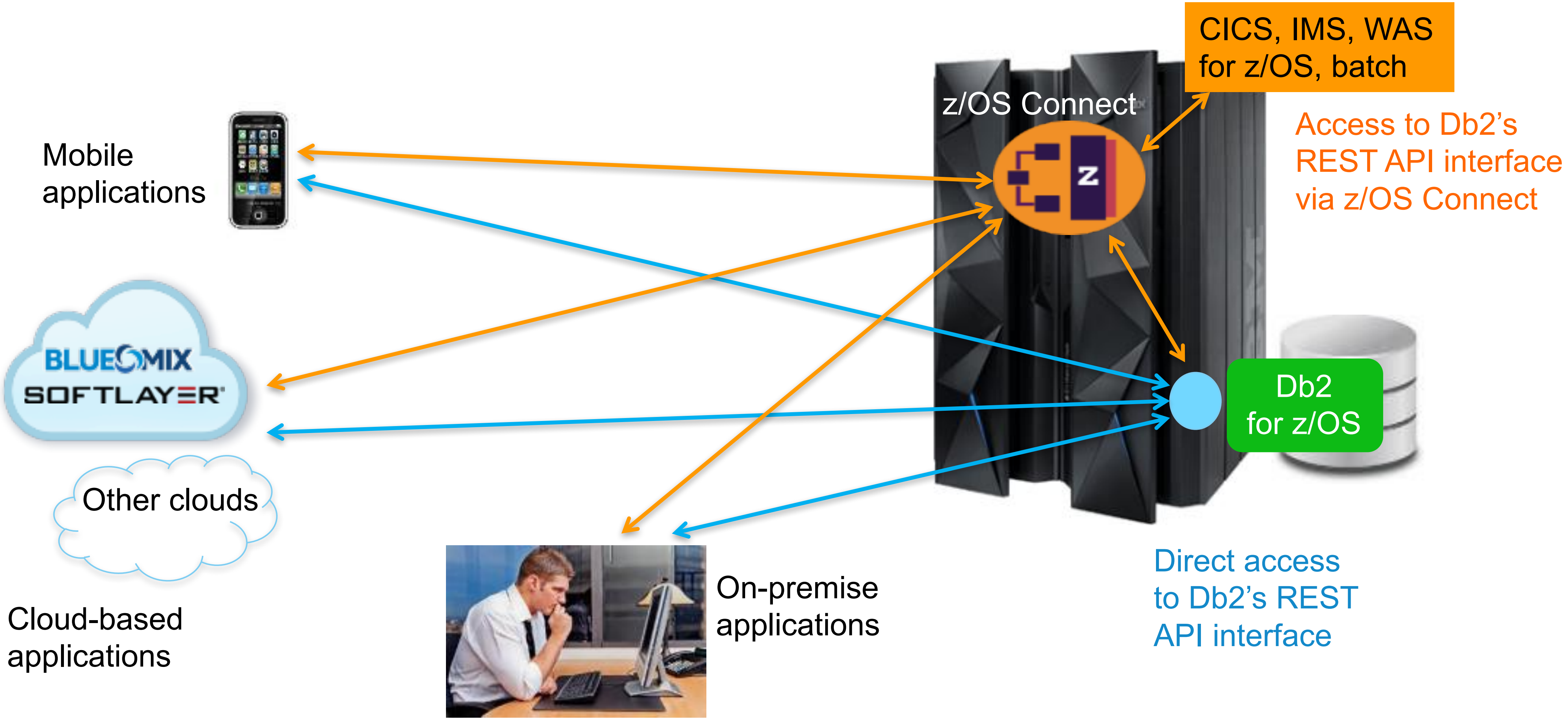
- z/OS Connect Enterprise Edition (EE) provides another path to Db2’s native REST interface, versus direct access from client
  - In other words, Db2 can be a REST provider to z/OS Connect
- Not just another path – z/OS Connect EE enhances Db2’s built-in REST support
  - z/OS Connect EE provides capabilities 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 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
- Does going through z/OS Connect affect the cost of executing a REST-invoked Db2 SQL service?
  - Not much
    - Some additional CPU consumption, but z/OS Connect is written in Java, so additional mainframe MIPS consumed are zIIP MIPs

The big picture

Accessing Db2 data with RESTful APIs and JSON



Access to Db2's REST API interface via z/OS Connect

Direct access to Db2's REST API interface

z/OS Connect, or Db2 Connect (IBM Data Server Driver)?

- Db2 Connect and the IBM Data Server Driver allow applications running on distributed systems servers (or workstations) to interact with Db2 for z/OS using non-DBMS-specific SQL 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

✓No need for Db2 client code on requester

✓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

✓Leverages developers' SQL skills

✓Better workload isolation

✓Very CPU-efficient transaction processing

✓Resource pooling (e.g., connection pooling)

✓Sysplex workload balancing

✓Transaction fault-tolerance

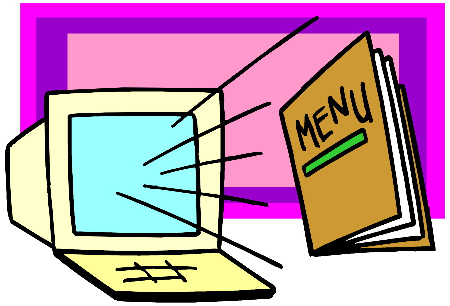
Db2's native REST interface: a few current restrictions

- These are documented in "info" APAR II14827
  - http://www-01.ibm.com/support/docview.wss?uid=isg1II14827
- Db2's native REST interface does not currently support:
  - Access to data in a Db2 Analytics Accelerator (will be addressed by a future APAR)
  - Use of Db2 system profile monitoring (will be addressed by a future APAR)
  - Access to data at a remote DRDA server site by way of the Db2 system targeted by the REST request (in other words, cannot, by way of Db2's REST interface, invoke a SQL statement at Db2 location A that references data at Db2 location B)
    - May be addressed by a future APAR

Db2 for z/OS and database-as-a-service

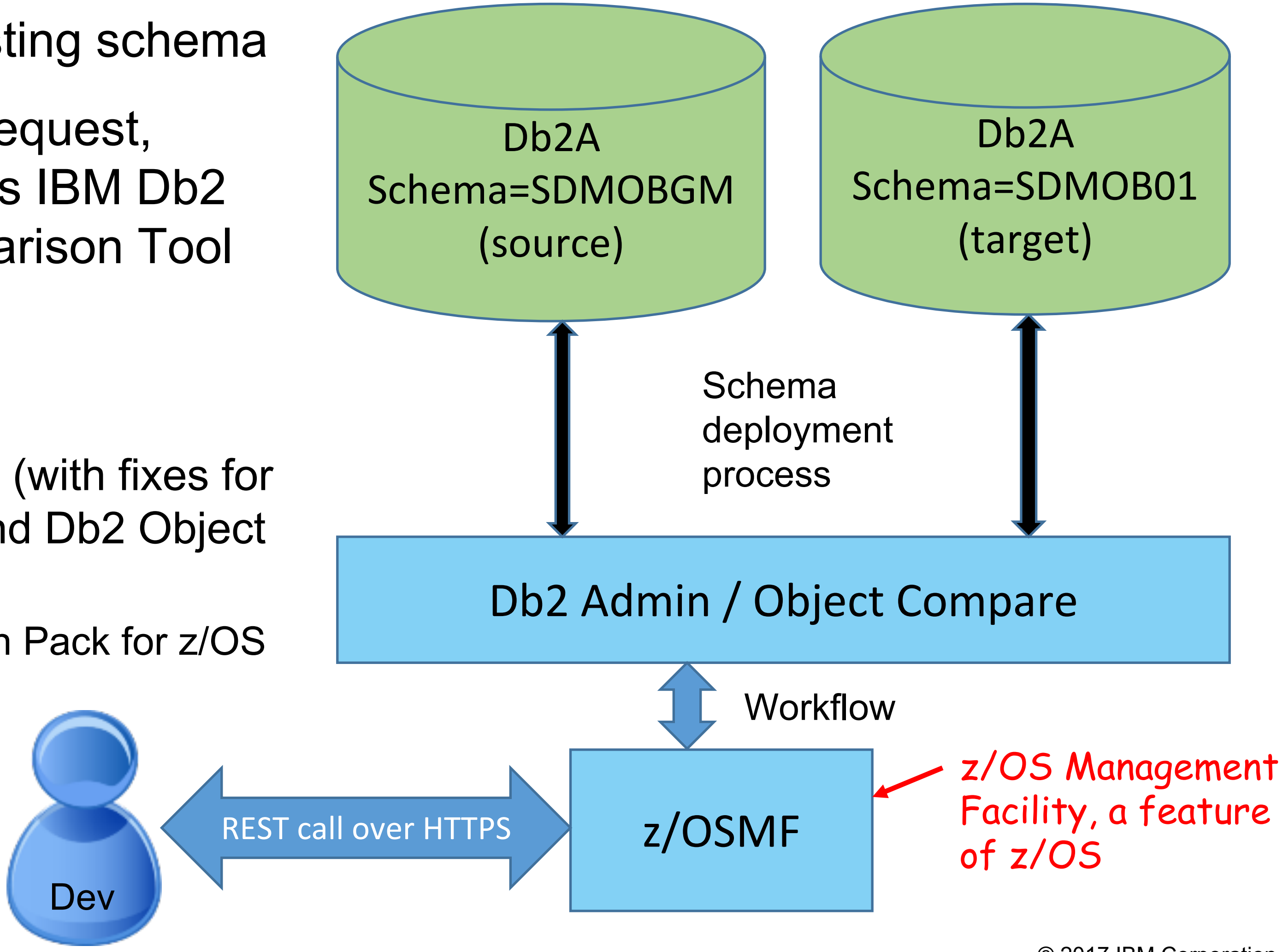
Db2 for z/OS and DBaaS

- In a Db2 for z/OS context, the key consumer group for database-as-a-service is application developers
- As I mentioned previously, what you want from “as-a-service” is ease and speed of resource requisitioning and provisioning
- What database services do application developers want?
  - Examples: stand up a new instance of Db2, create a new schema, deploy an application (and any required database objects) to a new environment
- How can these services be easily requisitioned and quickly provisioned?
  - Answer: automation
    - Users request database services via an easy-to-use interface, and the requested resources are rapidly delivered through pre-designed processes – dependency on DBA action is removed (but DBAs have a very important role, as I’ll explain)



A Db2 for z/OS DBaaS capability available now

- “Create schema like” – provides a service that creates a new database schema that is like an existing schema
- Mechanism: developer uses GUI to enter request, REST call goes to z/OSMF, z/OSMF directs IBM Db2 Administration Tool and Db2 Object Comparison Tool to perform the requested service
- Software prerequisites:
  - IBM Db2 Administration Tool for z/OS V11.2 (with fixes for APARs PI67731, PI72396, and PI76054), and Db2 Object Comparison Tool for z/OS
    - Both included in Change Management Solution Pack for z/OS
  - Db2 for z/OS with fixes for these APARs:
    - Db2 11 – PI73168 (PTF in error, fixed via APAR PI79222)
    - Db2 12 – PI73492



z/OS Management Facility, a feature of z/OS

More on CREATE SCHEMA LIKE (and other Db2 services)

- What about that GUI a developer would use to request the service? What is that?
  - Could be in any number of forms – multiple types of front-end will be supported (just has to be able to issue the required REST call)
    - Could be IBM UrbanCode Deploy (widely used, already geared towards application deployment)
    - Could be IBM Bluemix (would provide cloud-based interface for Db2 DBaaS capabilities)
    - Could be something else
- How about Db2 Administration Tool and Db2 Object Comparison Tool?
  - Very important – these tools, often referred to collectively as AOC (Admin/Object Compare), are and will be the foundation of Db2's DBaaS capabilities
    - They are designed to automate what a DBA would otherwise have to do in a manual fashion
- The other key component – z/OSMF
  - It can run workflows (also called worklists) that drive execution of various tasks – similar in concept to what distributed systems people call a script

Db2 DBaaS: what might follow CREATE SCHEMA LIKE?

- The Db2 for z/OS development team wants to provide capabilities that will enable application developers to easily request and quickly receive services such as:
  - Provision/de-provision a Db2 application environment (note that the Db2 12 for z/OS: installation/migration process generates artifacts that can become part of a z/OSMF workflow)
  - Automate (via self-service) application deployments including schema changes
  - Configure automated backups
  - Snapshot backup/restore services
  - Monitor/add/remove storage
  - Refresh statistics
  - …

What does all this mean for application developers?

- In a word: speed
- The speed advantage doesn’t only come from quick provisioning of database resources such as replicating a schema or populating a set of tables with test data – it also comes through rapid validation of schema changes
  - Suppose a developer creates some new code that has associated with it a database schema change (e.g., a new column in a table, or maybe a new table) – what if that change could be automatically and quickly validated when the new code is checked in?
    - Schema validation routines could verify things like adherence to naming conventions and data type standards, and could run some unit tests utilizing the new schema?
    - Why that’s important: errors can be more productively resolved when they are detected quickly, *while the developer’s mind is still on the new code that he/she created*

That’s what “shift-left” development is all about – find and fix problems as early as possible

What does all this mean for Db2 for z/OS DBAs?

- Some mainframe Db2 DBAs think that DBaaS is going to take away their job
  - ***Wrong*** – distributed systems DBAs have been involved with DBaaS for a while now, and their jobs aren't disappearing (in part because DBaaS makes a system more attractive *for new applications*)
- What DBaaS will do is change the nature of the Db2 for z/OS DBA's job
  - The DBA becomes more of a developer and designer – the automatically executed database service-providing processes first have to be designed
    - What will be automated, and in what way?
    - Where will "gates" be placed in these processes – what will require review by a DBA, and when? How will DBAs be notified of "must-review" situations, and how will they know what needs to be reviewed?
  - And, once a particular automated process has been designed, that story's not over – it should be reviewed and improved on a regular basis, just as developers do with their applications
  - In short: DBAs will become an integral part of organizations' overall CI/CD capabilities

continuous integration / continuous delivery

Things are about to get very interesting.

Robert Catterall
rfcatter@us.ibm.com