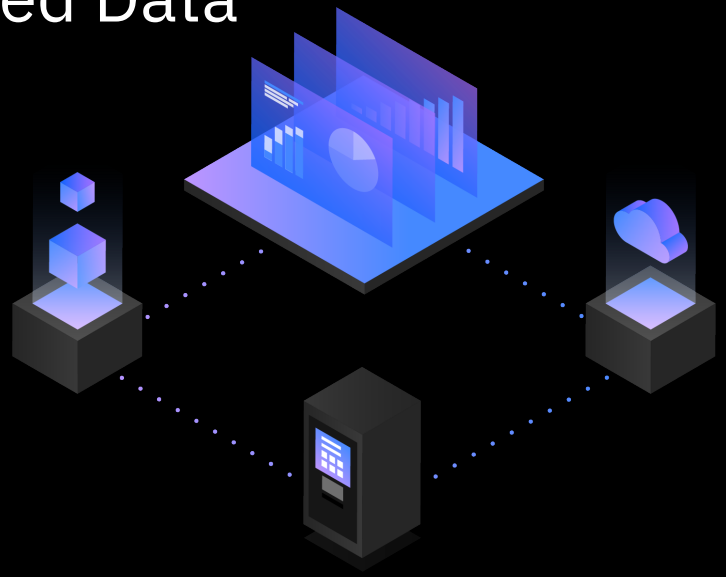
Modernizing Access to z/OS-Based Data



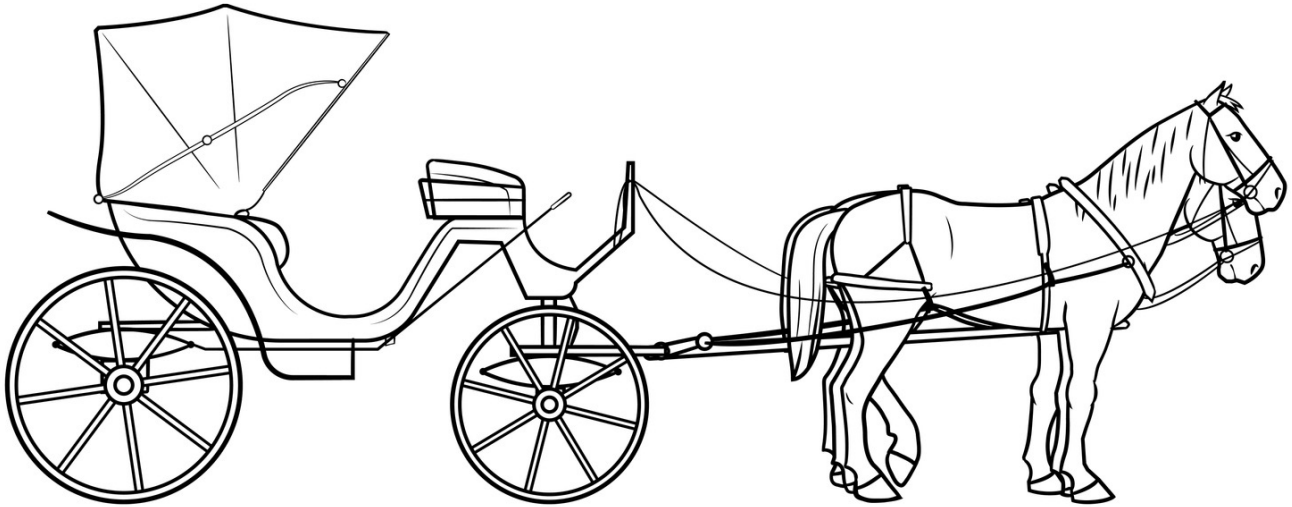
IBM Z Customer Council
Sao Paulo

May 23, 2023

Robert Catterall, IBM
Principal Db2 for z/OS Technical Specialist

To begin, a question…

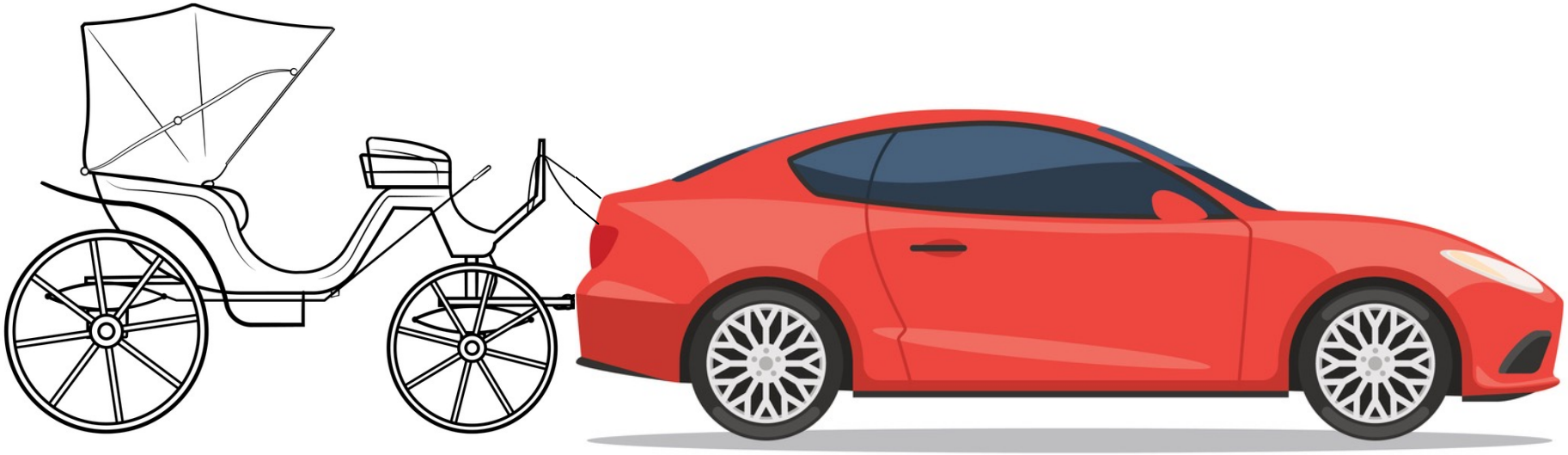
Is IBM Z old-fashioned technology?



The answer…

*IBM Z is **very modern technology** that is sometimes used **in old-fashioned ways***

- *And "old fashioned use" can apply to ways in which Z-based (especially z/OS-based) data is accessed*



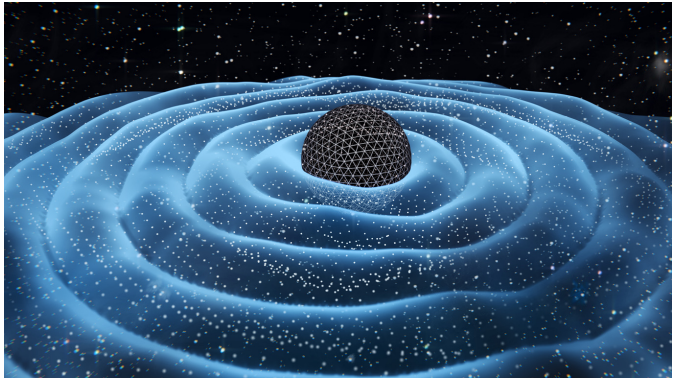
Agenda

- The primary objectives of z/OS data access modernization
- Modernizing access to Db2 for z/OS data
- Modernizing access to IMS data
- IBM Data Virtualization Manager – more options for z/OS data access modernization
- The Db2 Analytics Accelerator for z/OS – high-performance data analysis on the IBM Z platform
- IBM z/OS Connect – expanding REST access to z/OS-based data
- Infusing analytics into z/OS-based applications

The primary objectives of z/OS data access modernization

Access z/OS-based data “where it lives”…
(in other words, on the system of origin)

- This approach, which acknowledges the reality and the importance of data gravity, delivers multiple benefits:
  - Data latency is minimized (data “freshness” is maximized)
  - Data security is optimized (fewer data copies = reduced “threat area”)
  - Data consistency is preserved (one version of the truth)
  - Infrastructure and IT staff are utilized cost-effectively



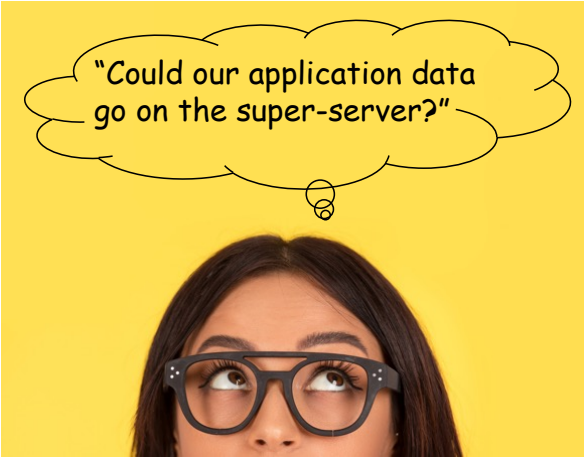
...but do that in developer-friendly way

- The key to “developer-friendly”: make the mainframe invisible
  - What that means: enable application developers to access z/OS-based data *the same way they access data on other platforms*
  - In other words, eliminate the need for developers to write mainframe-specific code

“What mainframe? I only see data.”

An important aspect of mainframe invisibility

- When a mainframe system looks the same (to a developer) as other data-serving platforms, what becomes more noticeable is that it does not act the same way as other platforms
  - It is the system that is always up, is never hacked, and performs well even at peak load
  - A mainframe database administrator once suggested a different name for IBM Z platform: “They should just call it the super-server”



Modernizing access to Db2 for z/OS data

One option: “generic” (non-DBMS-specific) SQL interfaces

- When using these interfaces, a developer does not have to know the particulars of the specific relational database management system being accessed
- The two most popular of these interfaces are JDBC (Java database connectivity) and ODBC (open database connectivity)
- A driver turns the non-DBMS-specific JDBC or ODBC statements into a form that is compatible with the particular DBMS targeted by the statements
- z/OS-based applications can use the JDBC and ODBC drivers that are packaged with Db2 for z/OS

What about applications that run on Linux- or UNIX- or Windows-based servers (which might be on-premise or in-cloud servers)?

JDBC/ODBC drivers for Linux/UNIX/Windows-based apps

- For these applications, the JDBC and ODBC drivers that support access to Db2 for z/OS are provided by the IBM Data Server Driver
- An organization is entitled to use the IBM Data Server Driver by way of a license for IBM Db2 Connect
- The IBM Data Server Driver runs on the same server as the JDBC- or ODBC-using application that is accessing Db2 for z/OS data
  - In the case of JDBC, this kind of "straight to the database server" driver is sometimes referred to as a type 4 driver

"I'm a relational database management system – do you know which one?"

"I don't need to know – I'm using JDBC"

Another option: the built-in REST interface to Db2 for z/OS

- REST – short for Representational State Transfer – is an architectural style that enables invocation of services through a straightforward and consistent interface
  - "Consistent": with a REST service request, the particulars of the service-providing system are completely abstracted from the perspective of the requesting program

What type or server platform is it? What is the operating system? What type of database is it? Is there even a database on the service-providing system?

Who cares? Thanks to the REST architectural style, all of this is just "plumbing" to a client-side developer

The Db2 for z/OS REST interface

- An extension of the distributed data facility (DDF) – the part of Db2 that has handled communications with client-server applications for 30+ years
- Scalable: one Db2 subsystem can drive thousands of transactions per second via the REST interface
- Secure: SQL is static, every request requires auth ID + password/certificate
- Cost-effective: SQL statement invoked through REST request is up to 60% zIIP offload-able when executed
- Service creation: a single static SQL statement (SELECT, INSERT, UPDATE, DELETE, TRUNCATE or stored procedure CALL) turned into REST service…
  - Through execution of Db2 command BIND SERVICE -or-
  - Through execution of Db2-provided DB2ServiceManager REST service

The big picture…

REST call (ACCOUNTS is collection name, getBalance is service name)

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

REST client

HTTP + JSON

DDF

HTTP and JSON parsing

Security checks

Thread management

Db2 for z/OS
(database services address space)

SQL execution

```
Body:
{
  "FIRSTNAME" : "John",
  "LASTNAME"  : "Smith",
  "BALANCE"   : 1982.42,
  "LIMIT"     : 3000.00
}
```

HTTP response in JSON format (JavaScript Object Notation)

```
SELECT C.FIRSTNAME,
C.LASTNAME, A.BALANCE,
A.LIMIT
FROM ACCOUNTS A,
CUSTOMERS C
WHERE A.ID = ?
AND A.CUSTNO = C.CUSTNO
```

SQL statement (in form of a package)

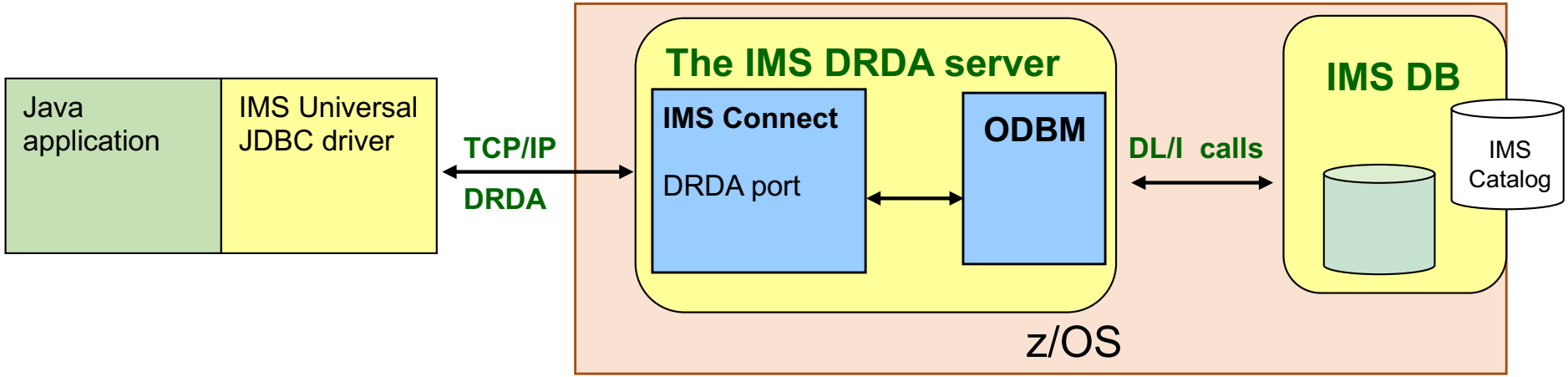
Modernizing access to IMS data

JDBC access to IMS data

Your IMS DB license entitles you to use these components

- JDBC access to IMS data is provided by way of three IMS components:
  - IMS Connect
    - Separate address space in z/OS system
    - Functions as IMS TCP/IP socket server, enabling client access via TCP/IP
  - Open Database Manager (ODBM)
    - Separate address space in z/OS system
    - Enables IMS DB to function as a DRDA server for JDBC-issuing client programs
  - IMS Universal JDBC driver
    - Runs on same server as application that connects to IMS DB via TCP/IP
    - Enables application access to IMS data using JDBC

The big picture...



IBM Data Virtualization Manager – more options for z/OS data access modernization

The DVM value proposition: “any data, any interface”

- Important: DVM is a z/OS-based virtualization technology – a major factor in delivering optimal performance
- The virtualization work that DVM does (i.e., making non-relational data appear to be relational) is almost 100% zIIP offload-able

DVM and IMS and VSAM (and Db2)

- What DVM does for IMS DB:
  - Adds ODBC as an access option (IMS has a JDBC interface)
  - Enables REST access to IMS data (when used with z/OS Connect)
- What DVM does for VSAM:
  - Enables application access to VSAM data using JDBC and ODBC
  - Enables access to VSAM data using REST requests (when used with z/OS Connect)
- What DVM does for Db2 for z/OS (and for IMS DB and VSAM):
  - Provides logical co-location of data in physically different sources
    - Example: with DVM, program can join a Db2 table with an IMS or VSAM “table”
    - DVM can also make off-mainframe data (e.g., Db2 for LUW, Oracle, SQL Server) appear to be co-located with mainframe data

The Db2 Analytics Accelerator for z/OS – high-performance data analysis on the IBM Z platform

Overview

- An Analytics-optimized system that is tightly coupled with a front-end Db2 for z/OS system
  - How tightly coupled? Db2 for z/OS code was modified to make the Accelerator, essentially, *another access path for Db2 query execution*
  - How that works:
    - Copy of “accelerated” Db2 tables maintained in the Analytics Accelerator
    - Accelerator is logically invisible: query is directed to front-end Db2 for z/OS system, and Db2 query optimizer determines where query would be executed fastest – on Accelerator or on Db2 front-end
    - If “on Accelerator,” query routed accordingly, and query result is routed back to the submitting user or application

A high-impact solution

- Complex, data-intensive queries can execute 1000X faster (or more) on Accelerator vs. front-end Db2 system (e.g., from hours to < 1 minute)
- Users can ask questions (in the form of queries) that they would not ask before (because the queries would never complete before)
  - At one site, users refer to the Accelerator as “the magic box”
- Besides speed, users appreciate access to data that is…
  - Fresh – Integrated Synchronization technology can keep data on Accelerator within a couple of seconds of currency relative to source data on front-end Db2
  - Detailed – Atomic-level (i.e., “original-state”)

Benefits for IT team, as well as for business users

- *Protects mainframe performance* – queries routed to Accelerator have virtually no impact on front-end operational applications (front-end Db2 just routes query to Accelerator, routes result back to user or application)
- *Ease of use for Db2 for z/OS DBAs*
  - Easy to administer via graphical interface (“point and click”)
  - No need to create indexes to speed query execution – no indexes on Accelerator
- *Data security* – protection of data on Accelerator controlled through front-end Db2 (no access to Accelerator outside of front-end Db2)

Deployment flexibility

- Accelerator is *software*, in containerized form, that runs on IFLs (IBM Z processors that can run Linux)

So, no need for a Linux administrator

- Can be deployed on traditional IBM Z server – *could be same one in which front-end Db2 for z/OS system is running*

Leverage HyperSocket connectivity

- Can be deployed on IBM LinuxONE server that is connected to IBM Z server on which front-end Db2 for z/OS system is running

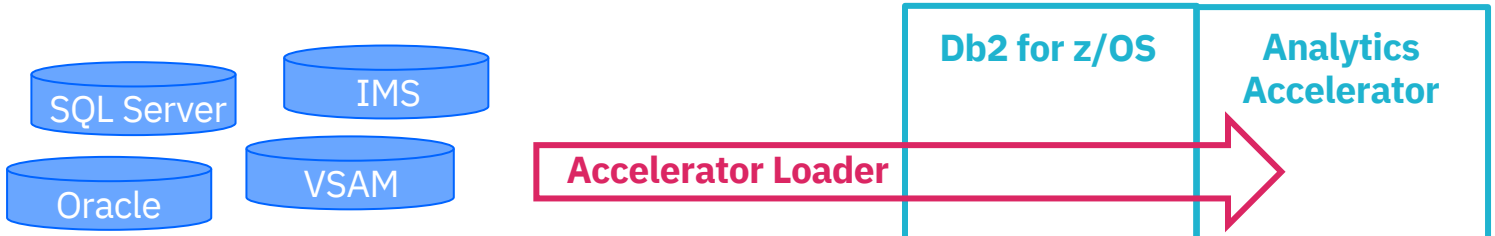
Either way, one Accelerator can be attached to multiple Db2 for z/OS systems, and multiple Accelerators can be attached to one Db2 for z/OS system

At the core of Accelerator: Db2 Warehouse

- An edition of Db2 for Linux featuring BLU Acceleration technology (in-memory, column-oriented arrangement of data that is optimized for analytical processing)
- Db2 for Linux SQL is very rich from an analytics perspective, and includes numerous functions that are not available with Db2 for z/OS SQL
  - Examples: LEAD, LAG, regular expression functions (e.g., REGEXP_COUNT), regression functions (e.g., REGR_INTERCEPT)
- When Analytics Accelerator is part of a Db2 for z/OS system, front-end Db2 can pass queries with these analytical functions to the Accelerator
  - In other words, Accelerator effectively *makes these analytical functions available in a Db2 for z/OS environment*

Extending Accelerator benefits to IMS and VSAM data

- With the Db2 Analytics Accelerator Loader, an organization can load non-relational z/OS-based data (e.g., IMS, VSAM) into Accelerator
  - And these could be “Accelerator-only” tables – with only a logical representation on the front-end Db2 system
    - Using Accelerator Loader with IMS data, one organization brought in-house data analytics work that had previously been outsourced, saving millions of $$
  - Loader could also be used to load Accelerator tables with data sourced from non-z/OS systems (e.g., Oracle, SQL Server) to co-locate that with z/OS-sourced data for analysis



IBM z/OS Connect – expanding REST access to z/OS-based data

Overview

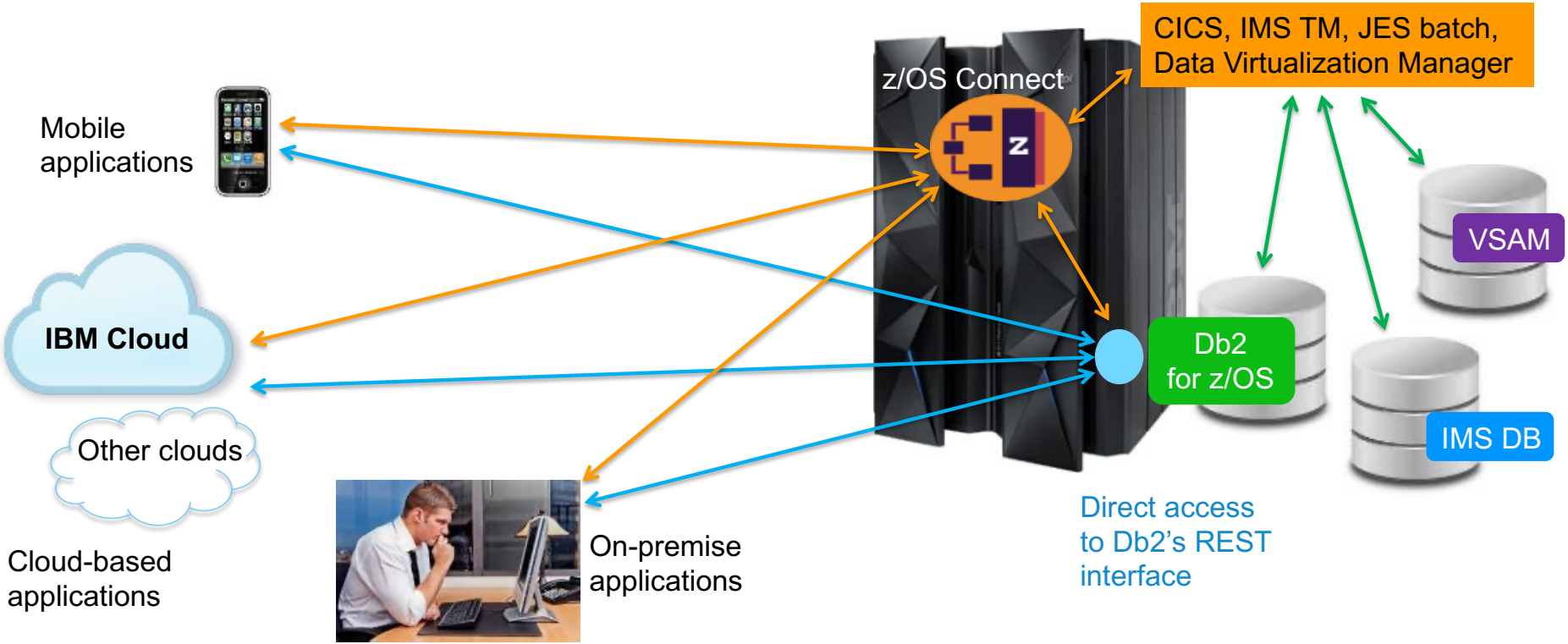
- z/OS Connect is a z/OS-based software product that enables REST access to a wide variety of z/OS data and programmatic assets – for example:
  - REST-enable a CICS-VSAM transaction
  - REST-enable an IMS TM-IMS DB transaction
  - REST-enable a batch job
  - REST-enable a Db2 for z/OS stored procedure
- z/OS Connect provides GUI tooling that makes creation of REST services from z/OS data and programmatic assets quick and easy
  - And, for REST services created with this tooling, z/OS Connect generates associated descriptions using the Swagger specification
    - Standard service description specification – helpful for service discovery

Db2 has a REST interface – why use z/OS Connect with Db2?

- Several reasons:
  - z/OS Connect's GUI tooling makes creation of REST services from Db2 SQL statements and stored procedures easier versus Db2-provided mechanisms
  - With z/OS Connect, REST services based on Db2 SQL statements and stored procedures can be invoked using any HTTP verbs (e.g., GET, PUT)
    - When accessing Db2's REST interface directly, request has to use POST
  - z/OS Connect provides options for formatting a REST service's JSON output document that Db2 does not provide
  - As previously noted, Swagger description of REST service generated by z/OS Connect can be helpful for service discovery

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

The big picture…



Infusing analytics into z/OS-based applications

Analytical applications – Db2 SQL Data Insights (SQL DI)

- New feature of Db2 13 for z/OS – advanced machine learning technology incorporated with the Db2 database "engine"
- No data scientist required to activate and utilize the feature
- Three new associated built-in Db2 functions:
  - AI_SIMILARITY
  - AI_SEMANTIC_CLUSTER
  - AI_ANALOGY
- SQL DI provides ability to execute "fuzzy" queries
  - Example: "Here is the account ID of someone who engaged in fraudulent activity – show me the 10 account IDs most like this one"

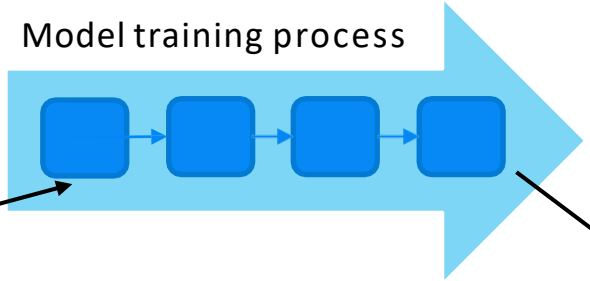
The key: you don't have to tell Db2 what you mean by "like"

The big picture…

A model is a Db2 table containing encoded vectors for each distinct entity in source table

SQL semantic functions retrieve the vectors to calculate their results

Model training process



USER.DATA_TABLE

CUSTOMER_ID	GENDER	BILLING	…
3668-QPYBK	F	auto	…
…	…	…	…

DBA

DSNAIDB.<generated vector table name>

Column	Value	vector
CUSTOMER_ID	3668-QPYBK	<1280 byte vector>
…	…	…
CUSTOMER_ID	8923-VFGHT	<1280 byte vector>
GENDER	F	<1280 byte vector>
…	…	…

App Developer

Business Analyst

```
SELECT CustomerID,
AI_SIMILARITY(CUSTOMER_ID, '3668-QPYBK')
FROM USER.DATA_TABLE
WHERE ...
```

And, through IBM Data Virtualization Manager the power of SQL Data Insights can be extended to IMS and VSAM data

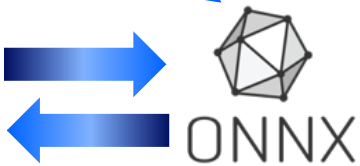
Operational applications – predictive models deployed in z/OS

- "Train anywhere, deploy in z/OS"
  - Allow data scientist to develop and train predictive model on platform of choice, using the model framework of choice, and enable deployment of that model in a z/OS system, with no need for the data scientist to know anything about z/OS
  - And, that predictive model will not just tolerate the mainframe platform – it will leverage IBM Z features that enhance the performance of *model inferencing*

This is the term often used to refer to the invocation of a predictive model by (for example) an application to get a predictive indicator called a "score"

The big picture…

ONNX – **O**pen **N**eural **N**etwork e**X**change – is a standard format for representing machine learning (ML) and deep learning (DL) models



IBM Deep Learning Compiler

Generated inference program

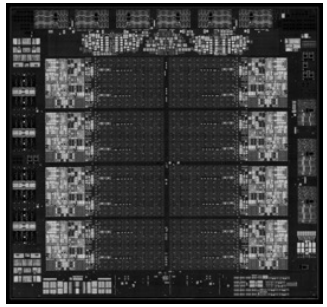
The IBM Deep Learning Compiler (DLC) can take a model in ONNX format as input, and the output will be an inference program that is optimized for performance on IBM Z

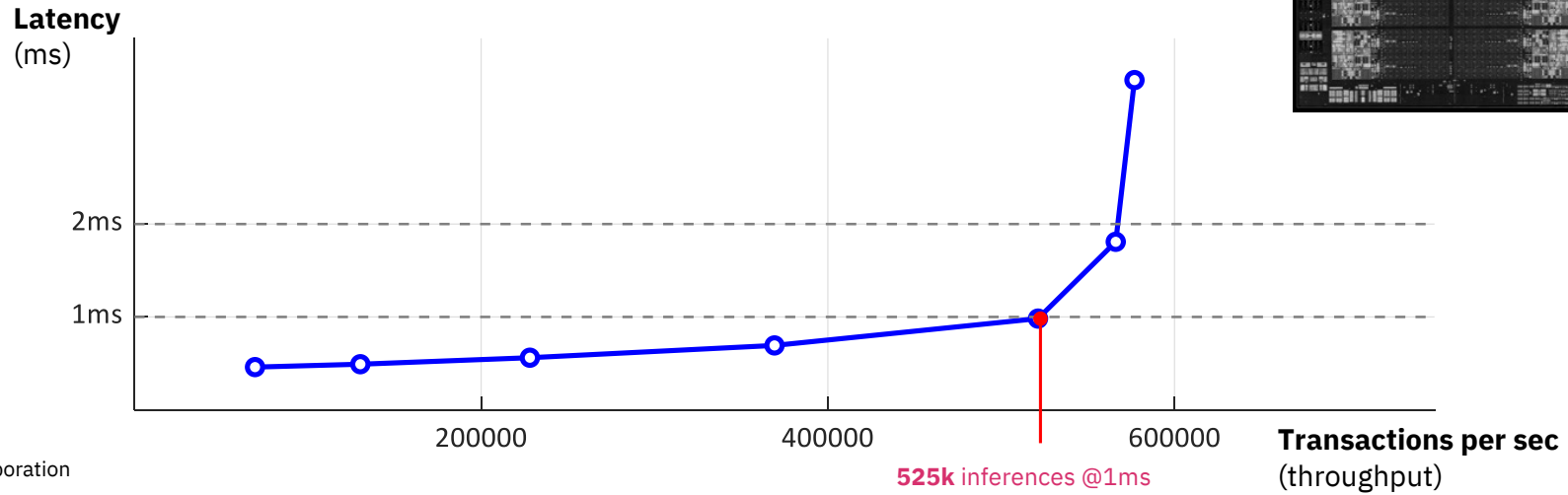
Model can be deployed natively in z/OS using Watson Machine Learning for z/OS (WMLz), or in containerized form via Z Container Extensions (z/CX)

Scalable? YES

- Especially with the z16 and its on-chip Integrated Accelerator for AI (the “AIU”), huge transaction volumes can be executed with 100% inferencing and 1 ms model response time
  - Credit card transactions, ATM withdrawals, insurance claim adjudication, filings for government aid, online shopping up-sell and cross-sell, etc.

Use case:	Customer transaction prediction (open source bank dataset)
Model:	XGBoost
Hardware:	IBM z16, 4 cores, 1 Integrated Accelerator for AI





A closing thought…

IBM Z can be the most modern data server in your IT environment – if you let it

Robert Catterall

rfcatter@us.ibm.com