# A Restful Journey

John Pearson

*Wakefern Food Corp*

Session code: E03

Mon, Jun 03, 2019 (03:10 PM - 04:10 PM)

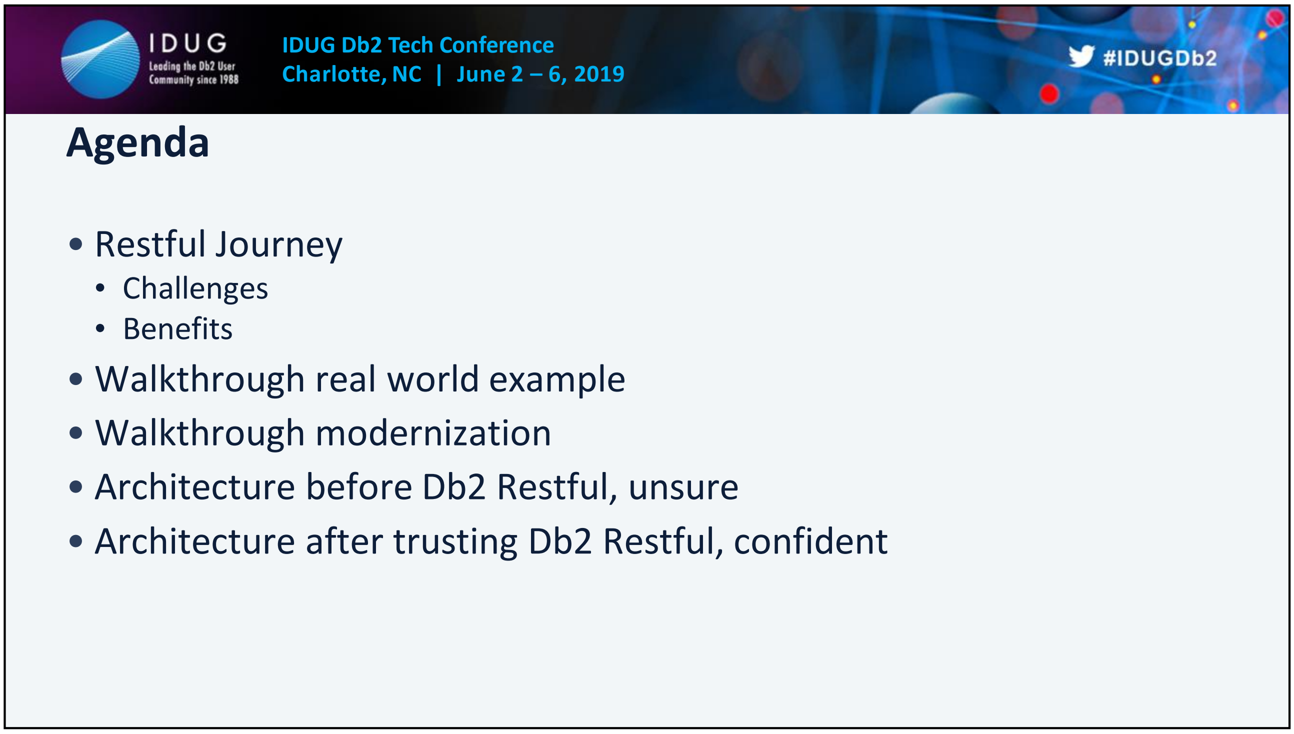
Db2 for z/OS

IDUG

Leading the Db2 User Community since 1988

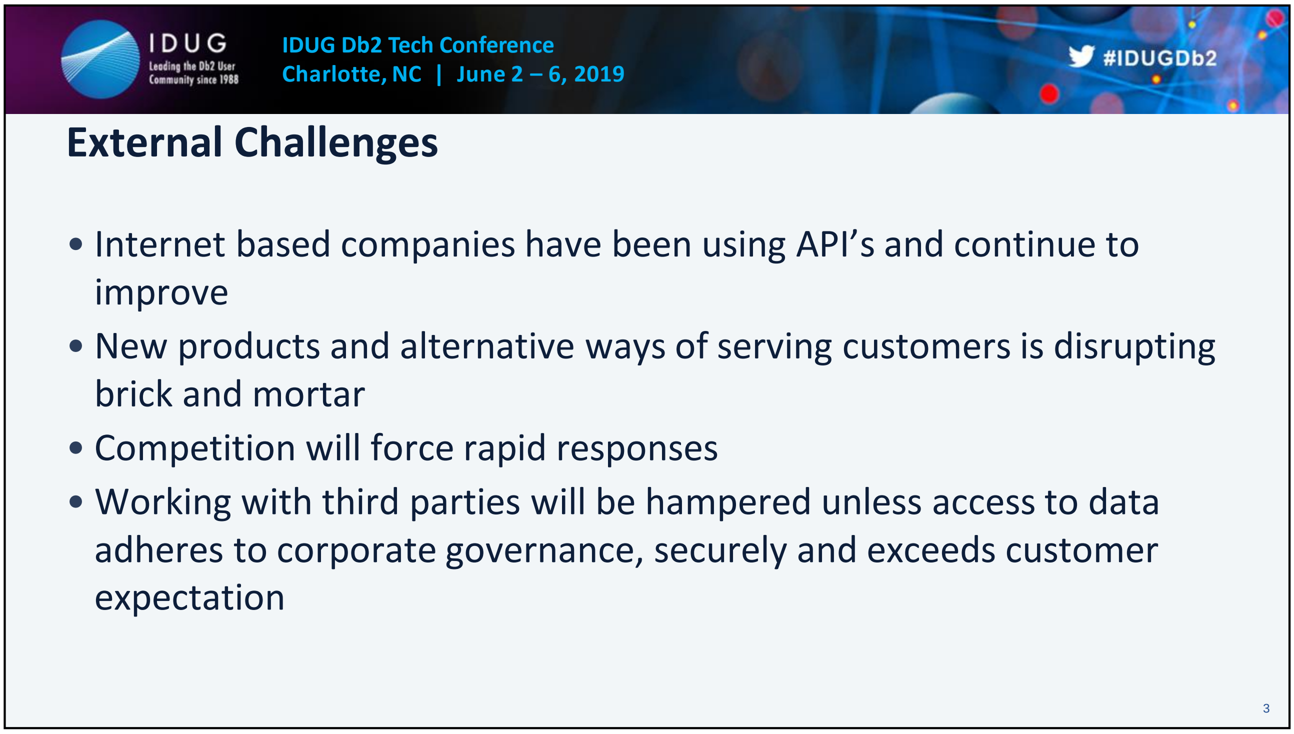
Modern is the new normal for mobile, web or back office development. Take new or existing resources without any zOS skills and make them productive in less than an hour. Send requests directly to DB2 to create, update or delete data securely. DB2 Restful services are easily consumable, fast to build and deploy, cost effective and safe.

This journey enabled the most junior of developers with the most valuable enterprise data influencing customers. Web and mobile aware customers have high expectations. You will meet those expectations, exceed those expectations or regret your shortsightedness. Let's walk through the architecture from web and mobile clients, security, monitoring, then onto zOS DB2 Restful services.



## Agenda

- Restful Journey
  - Challenges
  - Benefits
- Walkthrough real world example
- Walkthrough modernization
- Architecture before Db2 Restful, unsure
- Architecture after trusting Db2 Restful, confident



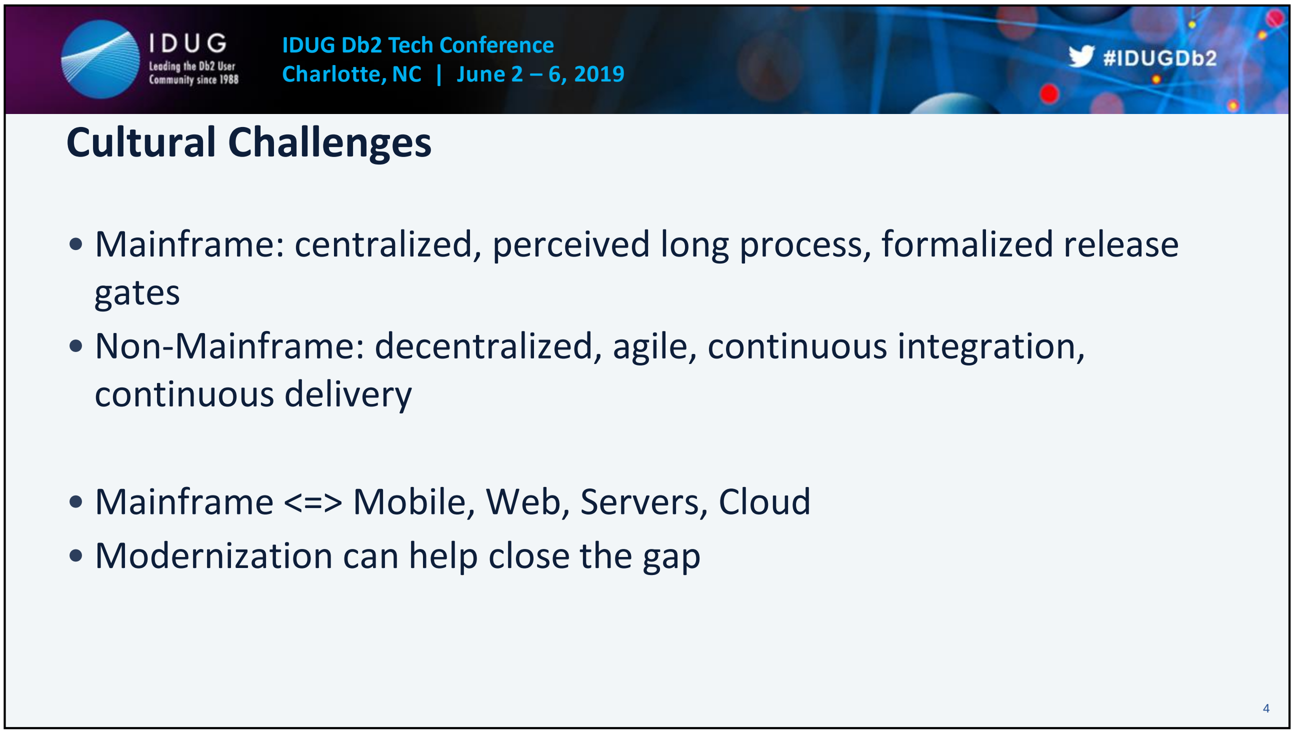
## External Challenges

- Internet based companies have been using API's and continue to improve
- New products and alternative ways of serving customers is disrupting brick and mortar
- Competition will force rapid responses
- Working with third parties will be hampered unless access to data adheres to corporate governance, securely and exceeds customer expectation

Internal challenges

Difficult to reuse common capabilities

Departments tend not to support holistic view across departments so sharing data is difficult



## Cultural Challenges

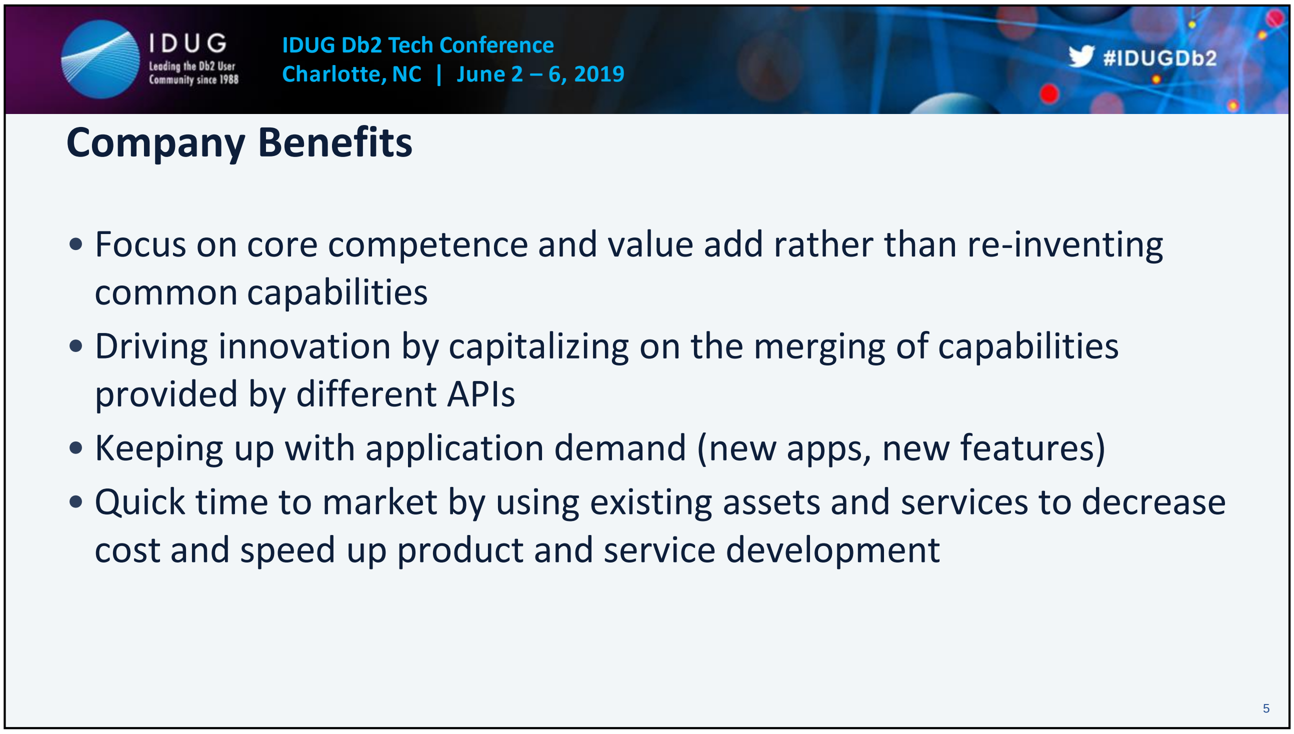
- Mainframe: centralized, perceived long process, formalized release gates
- Non-Mainframe: decentralized, agile, continuous integration, continuous delivery

- Mainframe <=> Mobile, Web, Servers, Cloud
- Modernization can help close the gap

4

Many examples of challenges:

Architecture: Monolithic verses Microservice

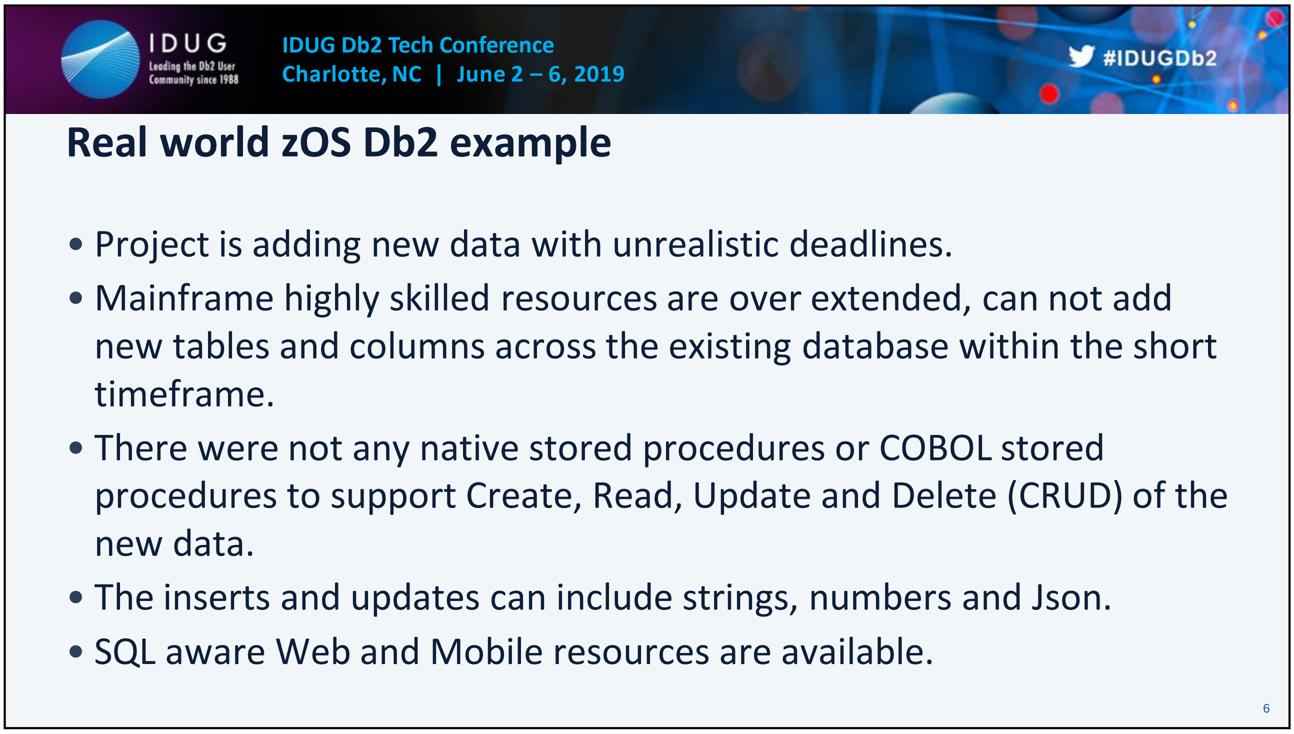


## Company Benefits

- Focus on core competence and value add rather than re-inventing common capabilities
- Driving innovation by capitalizing on the merging of capabilities provided by different APIs
- Keeping up with application demand (new apps, new features)
- Quick time to market by using existing assets and services to decrease cost and speed up product and service development

These benefits are easy to reference

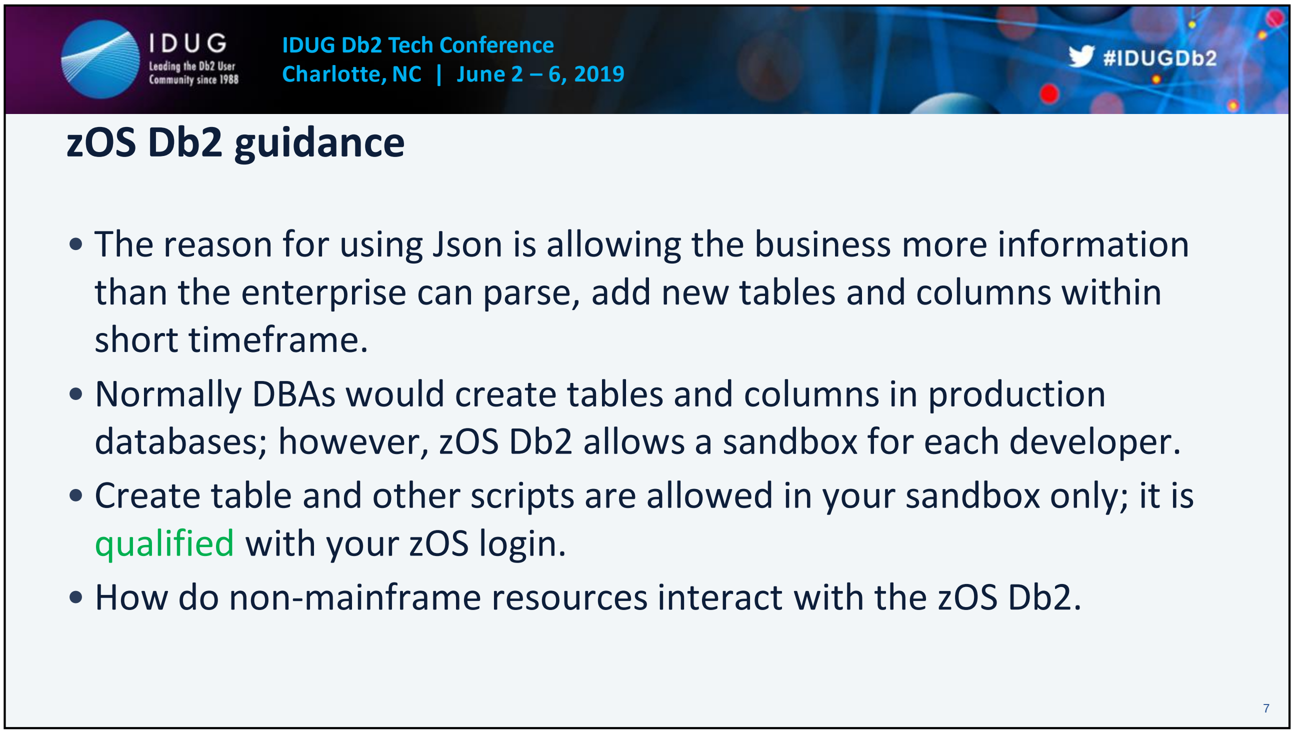
Eliminating batch, replicated interface processing



## Real world zOS Db2 example

- Project is adding new data with unrealistic deadlines.
- Mainframe highly skilled resources are over extended, can not add new tables and columns across the existing database within the short timeframe.
- There were not any native stored procedures or COBOL stored procedures to support Create, Read, Update and Delete (CRUD) of the new data.
- The inserts and updates can include strings, numbers and Json.
- SQL aware Web and Mobile resources are available.

At the time there was and procedures able to do so we used static sql parameterize queries.


IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019
#IDUGDb2

## zOS Db2 guidance

- The reason for using Json is allowing the business more information than the enterprise can parse, add new tables and columns within short timeframe.
- Normally DBAs would create tables and columns in production databases; however, zOS Db2 allows a sandbox for each developer.
- Create table and other scripts are allowed in your sandbox only; it is qualified with your zOS login.
- How do non-mainframe resources interact with the zOS Db2.

7

It was tested before submitting to DBAs

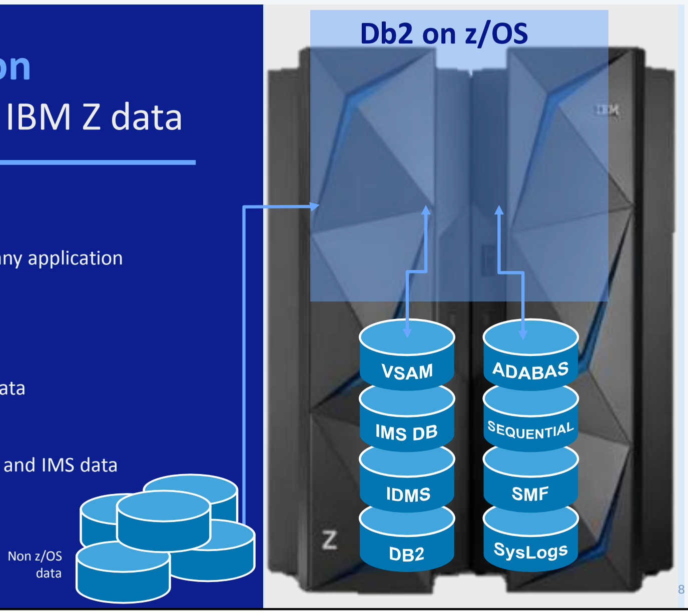
## QMF FOR Workstation
Analytics optimized for IBM Z data

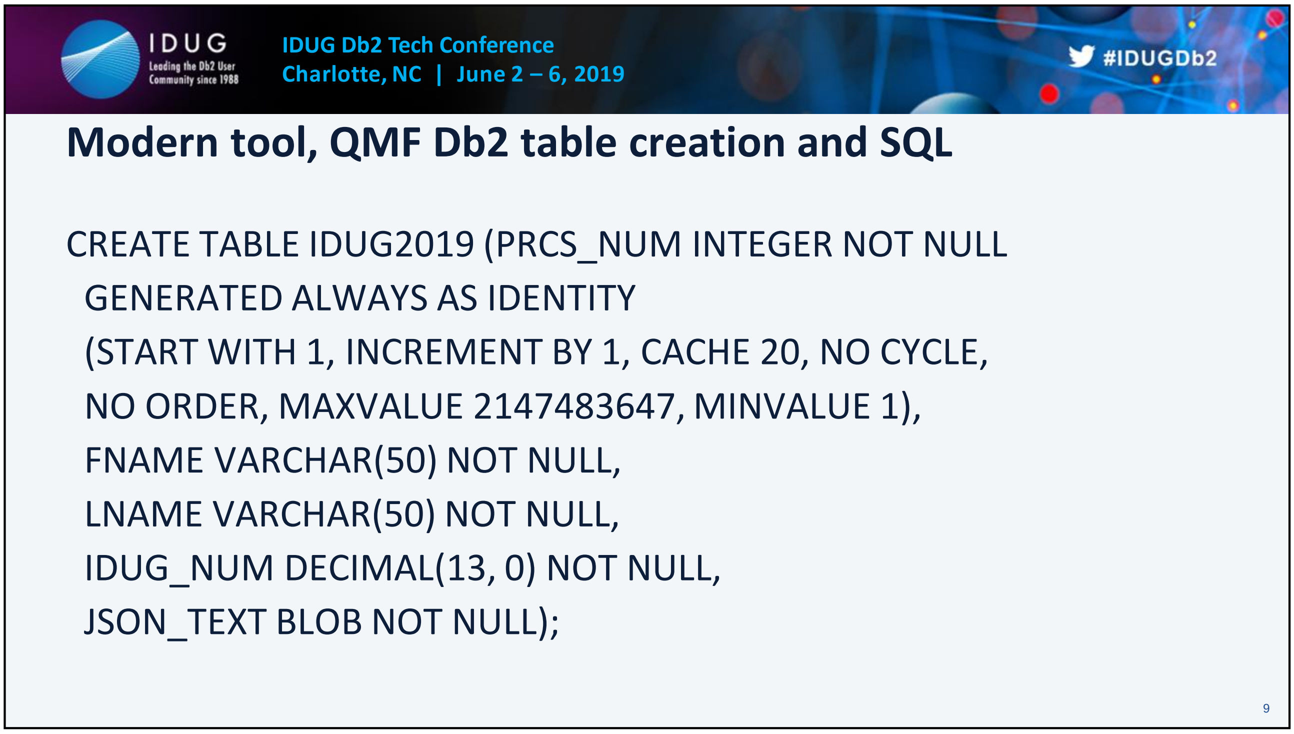
**Data Preparation**

- Data from multiple data sources can be easily readied for any application

**Data Access and Virtualization**

- Includes data virtualization technology
- Access to Z and non-Z data: Structured and unstructured data
- Discover schema and build SQL
- Save costs with zIIP engine utilization: up to 99% for VSAM and IMS data





## Modern tool, QMF Db2 table creation and SQL

```
CREATE TABLE IDUG2019 (PRCS_NUM INTEGER NOT NULL
  GENERATED ALWAYS AS IDENTITY
  (START WITH 1, INCREMENT BY 1, CACHE 20, NO CYCLE,
  NO ORDER, MAXVALUE 2147483647, MINVALUE 1),
  FNAME VARCHAR(50) NOT NULL,
  LNAME VARCHAR(50) NOT NULL,
  IDUG_NUM DECIMAL(13, 0) NOT NULL,
  JSON_TEXT BLOB NOT NULL);
```

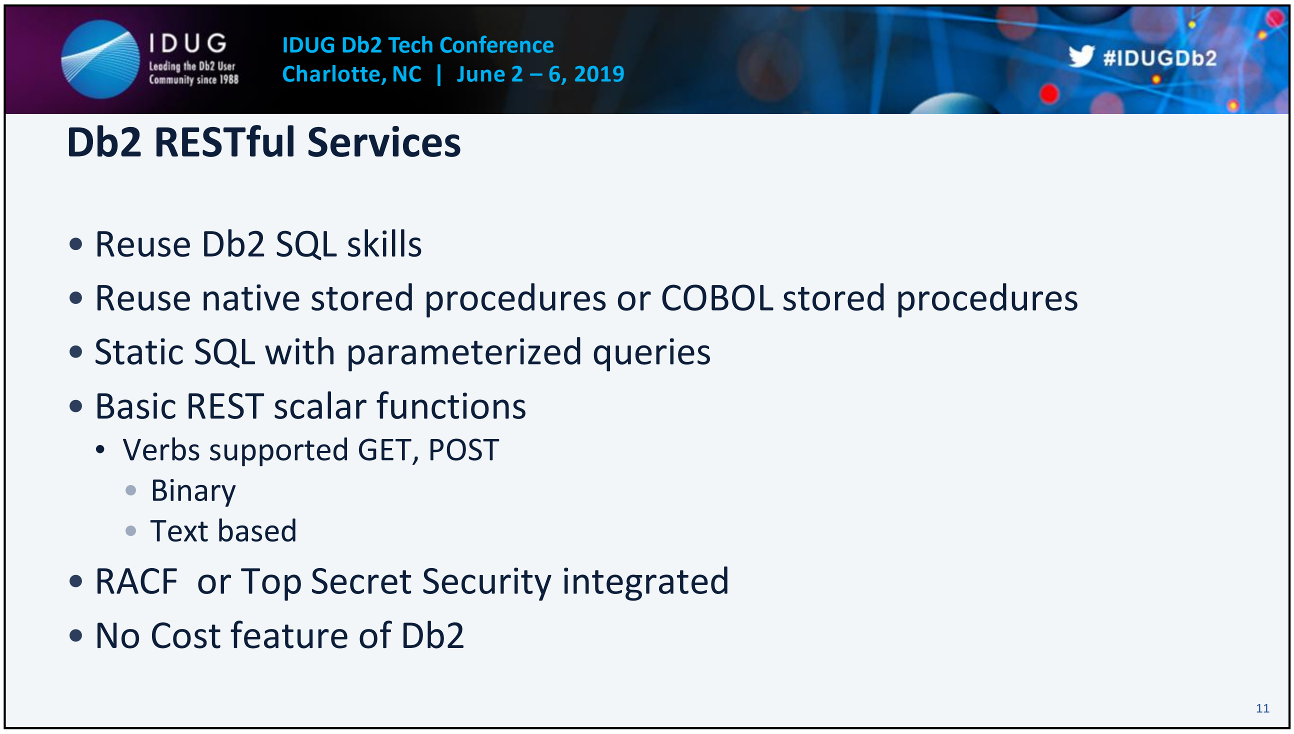
## Modern tools and standards for creating APIs

- CURL, very useful for a quick share
- Postman, collaboration

- Open API "Swagger", be careful of versions

- NodeJS for testing

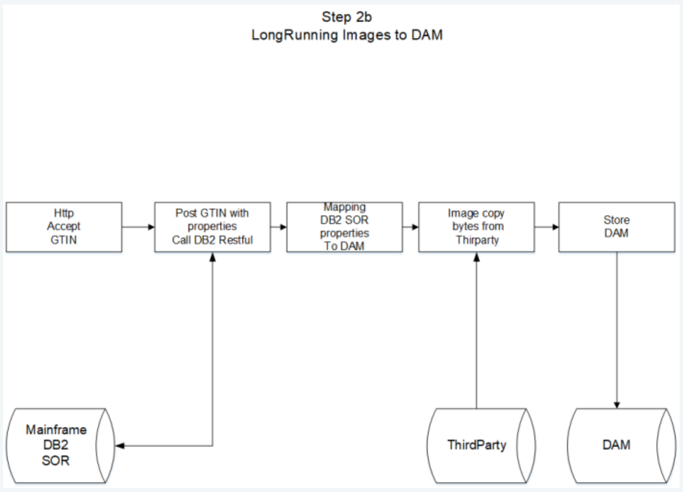
- Db2 Restful Services



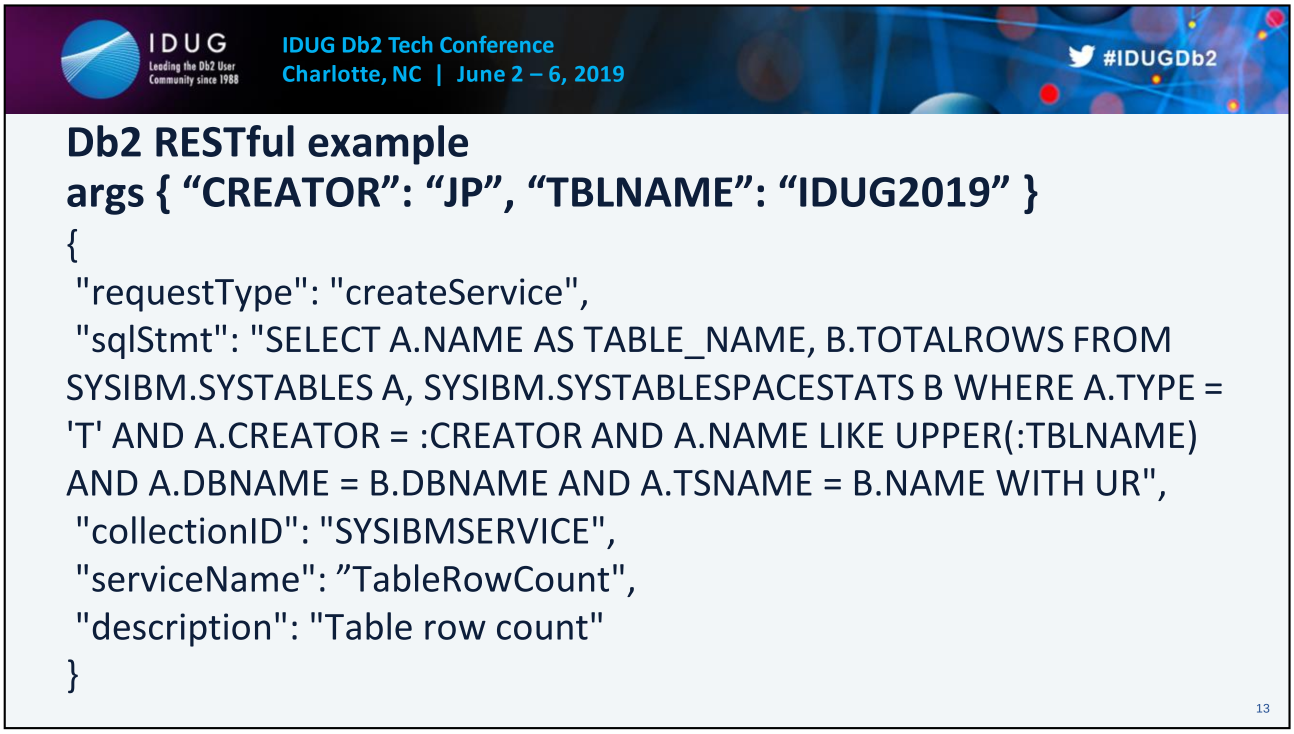
## Db2 RESTful Services

- Reuse Db2 SQL skills
- Reuse native stored procedures or COBOL stored procedures
- Static SQL with parameterized queries
- Basic REST scalar functions
  - Verbs supported GET, POST
    - Binary
    - Text based
- RACF  or Top Secret Security integrated
- No Cost feature of Db2

## July 2018 Architect a plan to use Db2 Restful

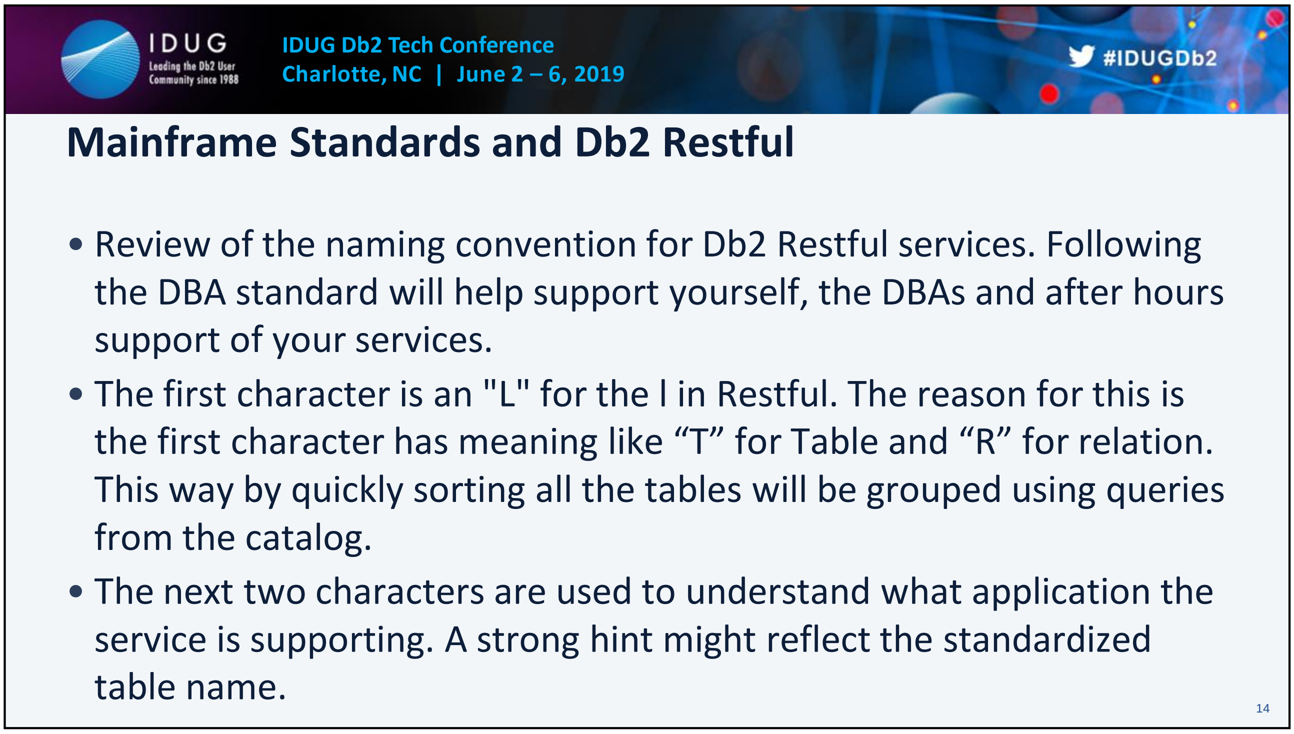

Step 2b
LongRunning Images to DAM
Http Accept GTIN
Post GTIN with properties Call DB2 Restful
Mapping DB2 SOR properties To DAM
Image copy bytes from Thirparty
Store DAM
Mainframe DB2 SOR
ThirdParty
DAM

IBM Cloud functions



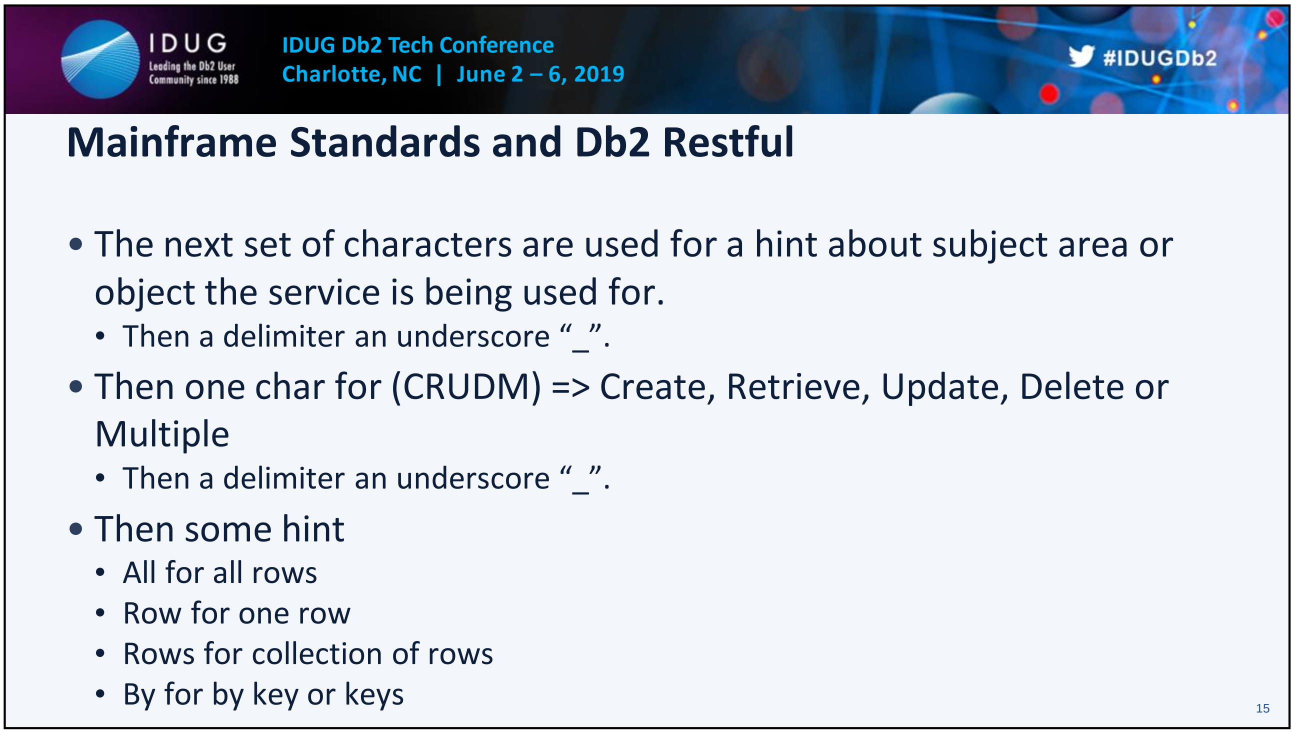
## Db2 RESTful example
## args { “CREATOR”: “JP”, “TBLNAME”: “IDUG2019” }

```
{
 "requestType": "createService",
 "sqlStmt": "SELECT A.NAME AS TABLE_NAME, B.TOTALROWS FROM
SYSIBM.SYSTABLES A, SYSIBM.SYSTABLESPACESTATS B WHERE A.TYPE =
'T' AND A.CREATOR = :CREATOR AND A.NAME LIKE UPPER(:TBLNAME)
AND A.DBNAME = B.DBNAME AND A.TSNAME = B.NAME WITH UR",
 "collectionID": "SYSIBMSERVICE",
 "serviceName": ”TableRowCount",
 "description": "Table row count"
}
```



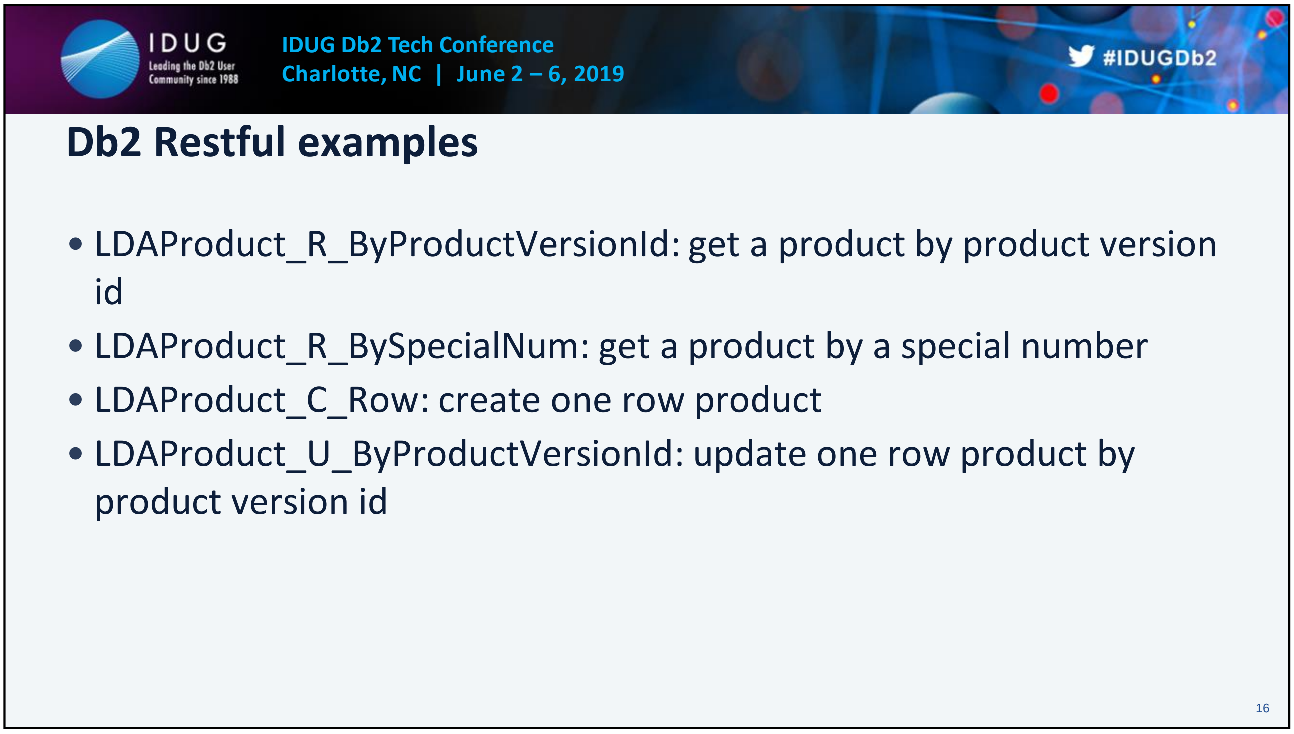
## Mainframe Standards and Db2 Restful

- Review of the naming convention for Db2 Restful services. Following the DBA standard will help support yourself, the DBAs and after hours support of your services.
- The first character is an "L" for the l in Restful. The reason for this is the first character has meaning like “T” for Table and “R” for relation. This way by quickly sorting all the tables will be grouped using queries from the catalog.
- The next two characters are used to understand what application the service is supporting. A strong hint might reflect the standardized table name.



## Mainframe Standards and Db2 Restful

- The next set of characters are used for a hint about subject area or object the service is being used for.
  - Then a delimiter an underscore “_”.
- Then one char for (CRUDM) => Create, Retrieve, Update, Delete or Multiple
  - Then a delimiter an underscore “_”.
- Then some hint
  - All for all rows
  - Row for one row
  - Rows for collection of rows
  - By for by key or keys



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

#IDUGDb2

## Db2 Restful examples

- LDAProduct_R_ByProductVersionId: get a product by product version id
- LDAProduct_R_BySpecialNum: get a product by a special number
- LDAProduct_C_Row: create one row product
- LDAProduct_U_ByProductVersionId: update one row product by product version id

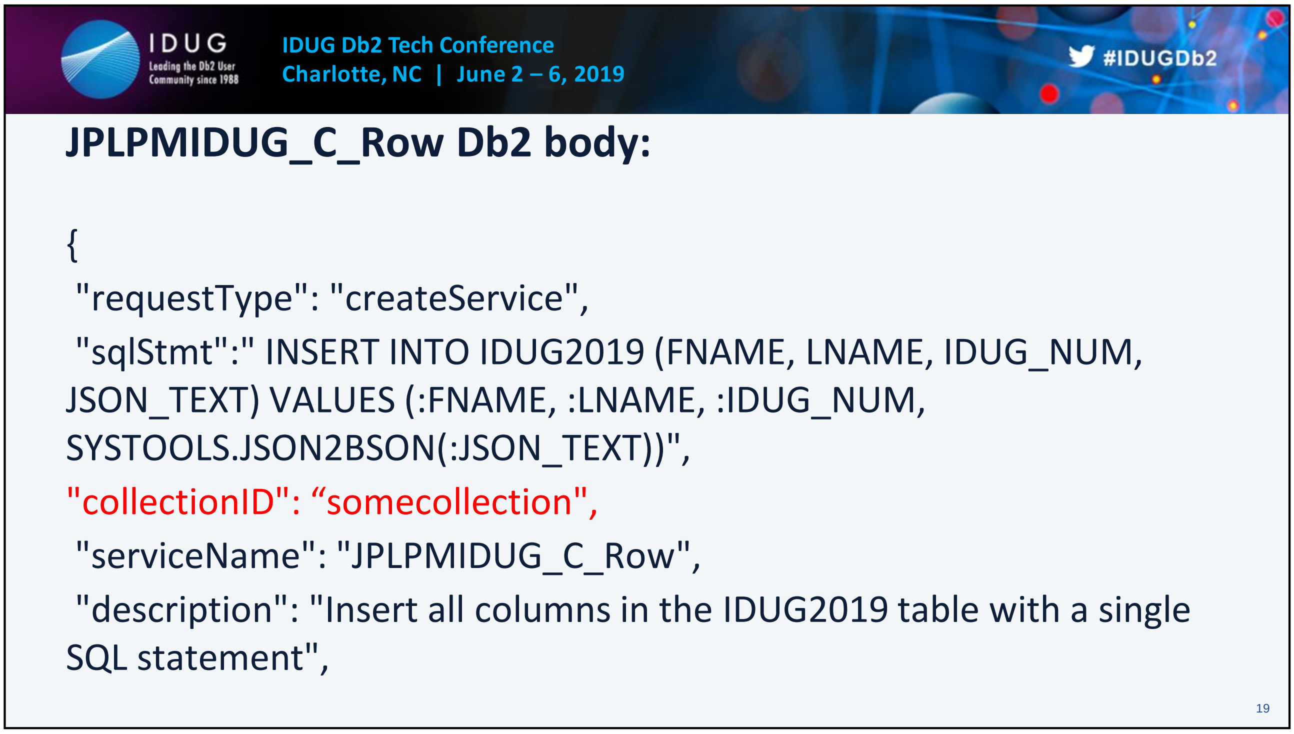
16

## Db2 Restful lesson learned

- In keeping with an agile way of developing with a group of people name your service with your initials, if your name is John Pearson, create your sand box services JPLDAProduct_C_Row. The namespace for Db2 Restful is flat within a collection.

## Db2 Restful creation service json

- Post http://hostname:port/services/Db2ServiceManager
- API Headers
  - Content-Type: application/json
  - Accept: application/json
  - Authorization: Basic A0ZKUAA6d2ludAVyNw==
- Your mainframe user name and password.
  - The Authorization: <type> <credentials> pattern was introduced by the W3C in HTTP 1.0, and has been reused in many places since. Many web servers support multiple methods of authorization. In those cases sending just the token isn't sufficient.



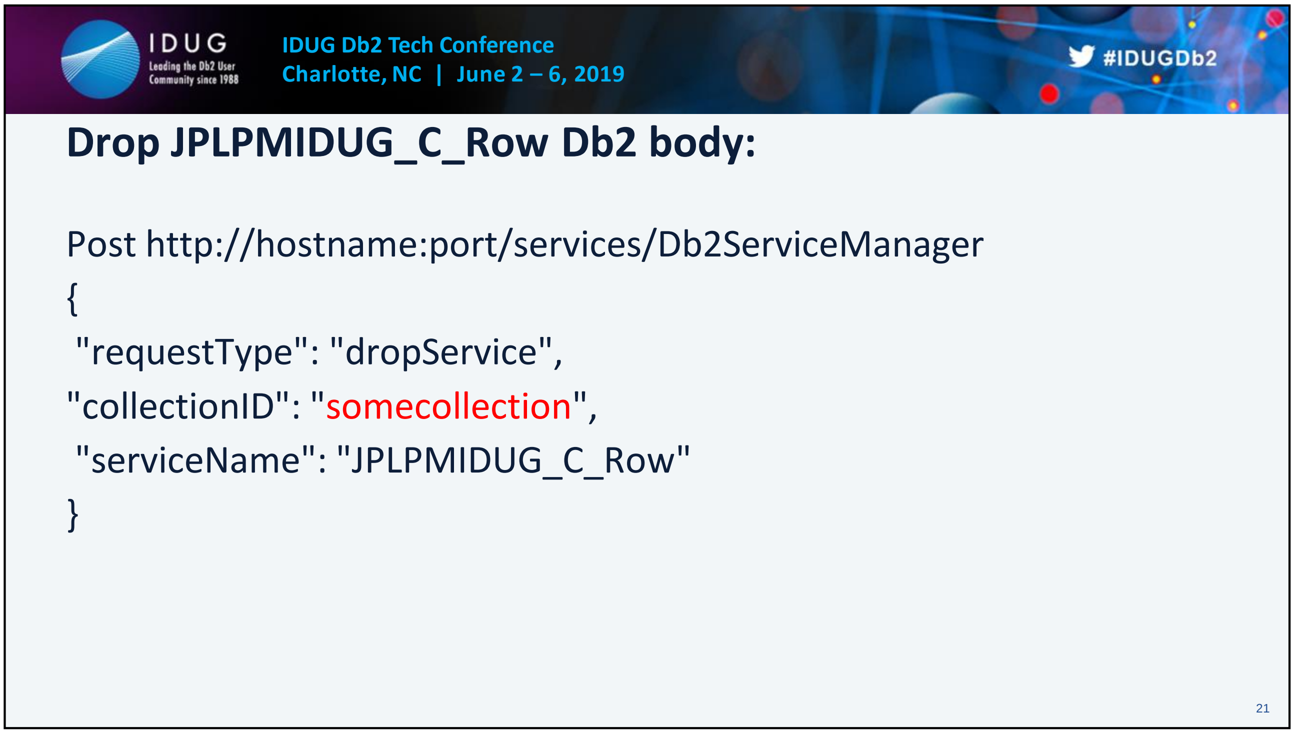
## JPLPMIDUG_C_Row Db2 body:

```
{
 "requestType": "createService",
 "sqlStmt":" INSERT INTO IDUG2019 (FNAME, LNAME, IDUG_NUM,
JSON_TEXT) VALUES (:FNAME, :LNAME, :IDUG_NUM,
SYSTOOLS.JSON2BSON(:JSON_TEXT))",
"collectionID": “somecollection",
 "serviceName": "JPLPMIDUG_C_Row",
 "description": "Insert all columns in the IDUG2019 table with a single
SQL statement",
```

This could also be done in a batch, DSN subcommand.

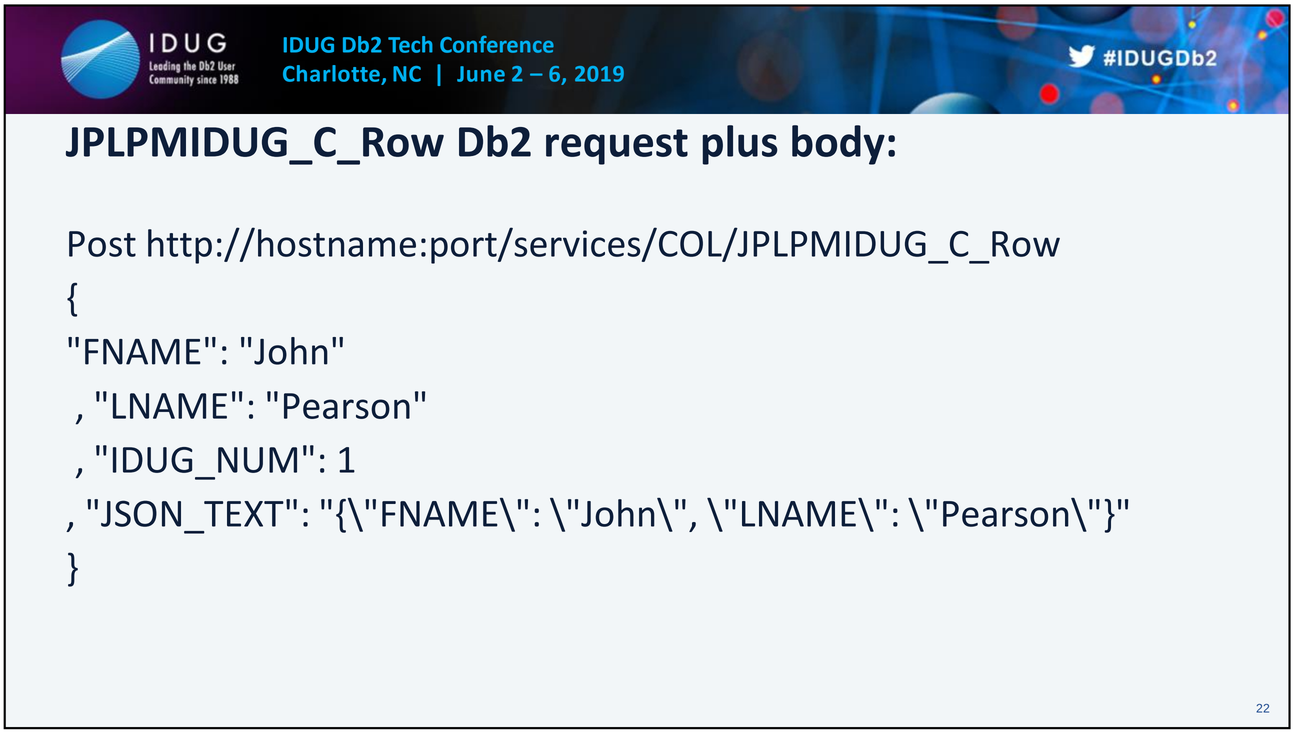
## JPLPMIDUG_C_Row Db2 and API Management body:

```
"OWNER":“DEV",
"EXPLAIN":"YES",
"ISOLATION":"CS",
"VALIDATE":"BIND",
"QUALIFIER": "JP"
}
```



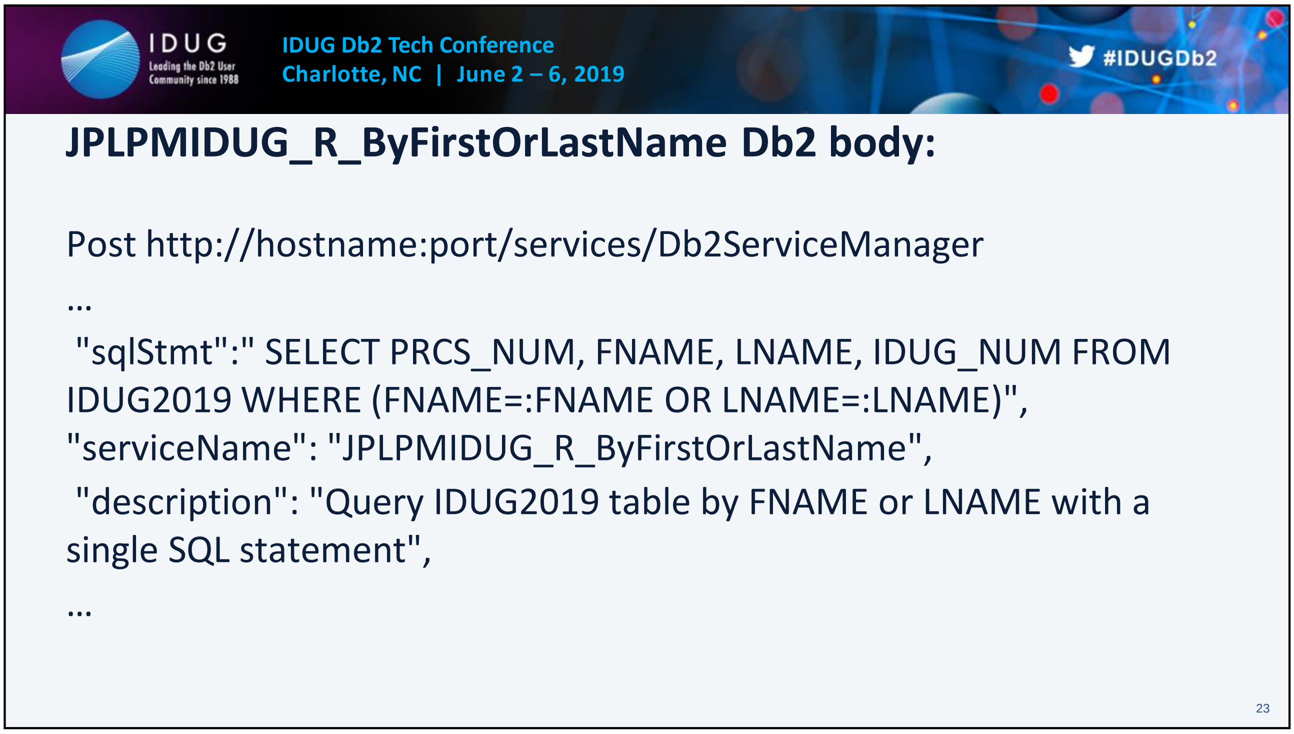
## Drop JPLPMIDUG_C_Row Db2 body:

```
Post http://hostname:port/services/Db2ServiceManager
{
 "requestType": "dropService",
"collectionID": "somecollection",
 "serviceName": "JPLPMIDUG_C_Row"
}
```



## JPLPMIDUG_C_Row Db2 request plus body:

```
Post http://hostname:port/services/COL/JPLPMIDUG_C_Row
{
"FNAME": "John"
 , "LNAME": "Pearson"
 , "IDUG_NUM": 1
, "JSON_TEXT": "{\"FNAME\": \"John\", \"LNAME\": \"Pearson\"}"
}
```



## JPLPMIDUG_R_ByFirstOrLastName Db2 body:

Post http://hostname:port/services/Db2ServiceManager

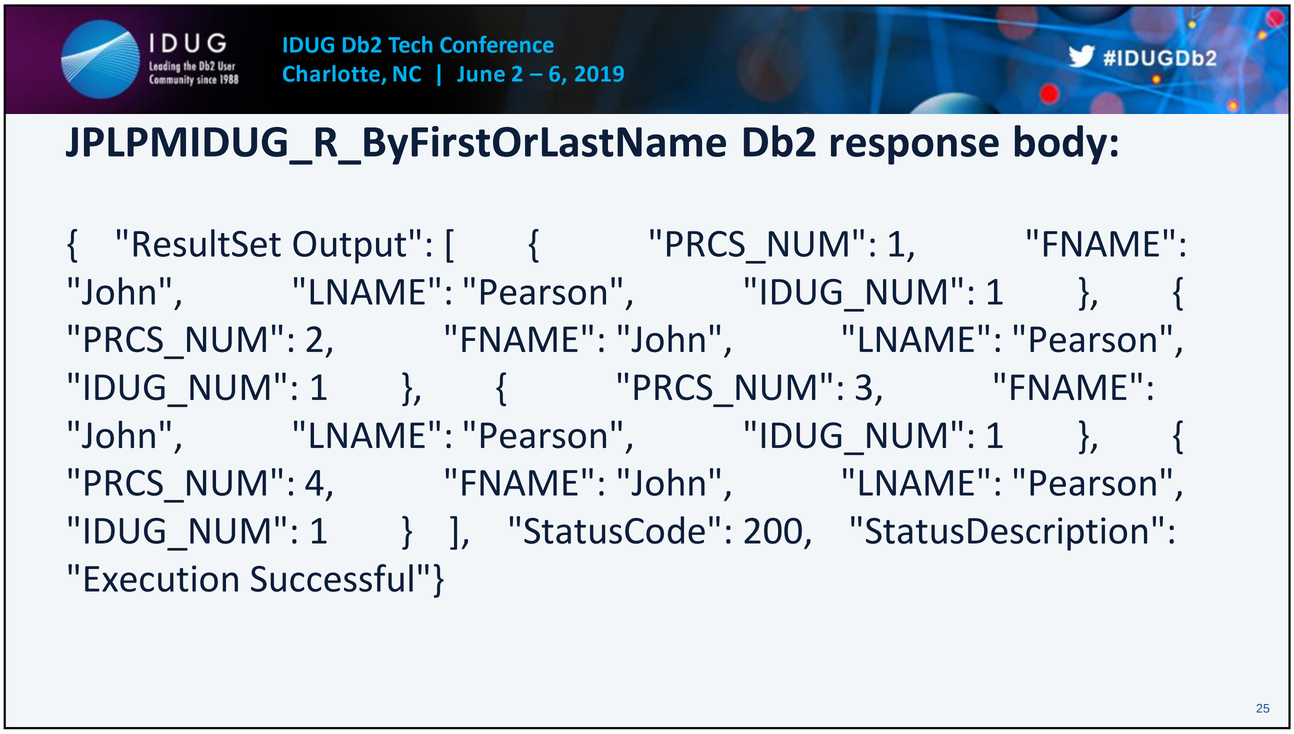
…

 "sqlStmt":" SELECT PRCS_NUM, FNAME, LNAME, IDUG_NUM FROM IDUG2019 WHERE (FNAME=:FNAME OR LNAME=:LNAME)",
"serviceName": "JPLPMIDUG_R_ByFirstOrLastName",
 "description": "Query IDUG2019 table by FNAME or LNAME with a single SQL statement",

…

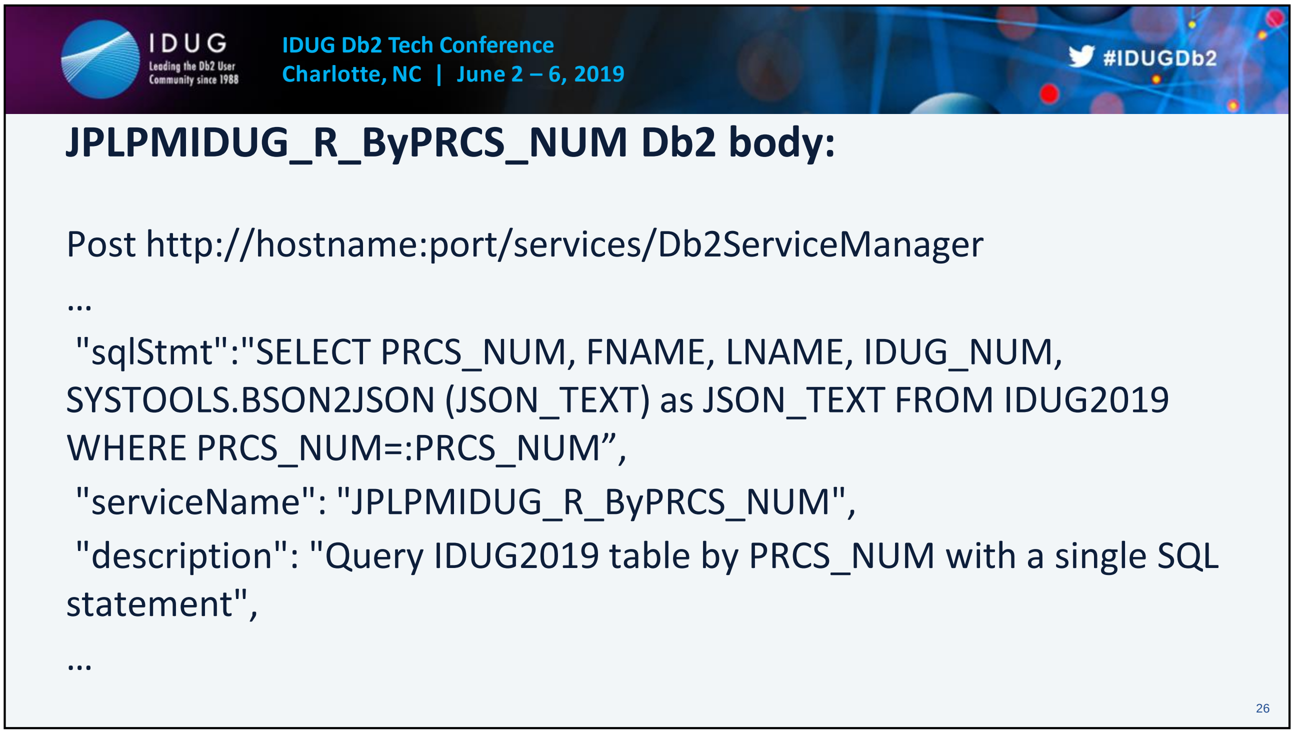
## JPLPMIDUG_R_ByFirstOrLastName Db2 request plus body:

```
Post
http://hostname:port/services/COL/JPLPMIDUG_R_ByFirstOrLastName
{
"FNAME": "John"
 , "LNAME": "Pearson"
}
```



## JPLPMIDUG_R_ByFirstOrLastName Db2 response body:

```
{   "ResultSet Output": [      {         "PRCS_NUM": 1,         "FNAME":
"John",         "LNAME": "Pearson",         "IDUG_NUM": 1      },      {
"PRCS_NUM": 2,         "FNAME": "John",         "LNAME": "Pearson",
"IDUG_NUM": 1      },      {         "PRCS_NUM": 3,         "FNAME":
"John",         "LNAME": "Pearson",         "IDUG_NUM": 1      },      {
"PRCS_NUM": 4,         "FNAME": "John",         "LNAME": "Pearson",
"IDUG_NUM": 1      }   ],   "StatusCode": 200,   "StatusDescription":
"Execution Successful"}
```



## JPLPMIDUG_R_ByPRCS_NUM Db2 body:

Post http://hostname:port/services/Db2ServiceManager

…

"sqlStmt":"SELECT PRCS_NUM, FNAME, LNAME, IDUG_NUM, SYSTOOLS.BSON2JSON (JSON_TEXT) as JSON_TEXT FROM IDUG2019 WHERE PRCS_NUM=:PRCS_NUM”,

"serviceName": "JPLPMIDUG_R_ByPRCS_NUM",

"description": "Query IDUG2019 table by PRCS_NUM with a single SQL statement",

…

## JPLPMIDUG_R_ByPRCS_NUM Db2 request plus body:

```
Post http://hostname:port/services/COL/JPLPMIDUG_R_ByPRCS_NUM
{
"PRCS_NUM": 2
}
```

## JPLPMIDUG_R_ByPRCS_NUM Db2 response body:

{ "ResultSet Output": [
{ "PRCS_NUM": 2, "FNAME": "John", "LNAME":
"Pearson", "IDUG_NUM": 1, "JSON_TEXT":
"{\"FNAME\":\"John\",\"LNAME\":\"Pearson\"}" }
]
, "StatusCode": 200
, "StatusDescription": "Execution Successful"}

## JPLPMIDUG_U_RowByPRCS_NUM Db2 body:

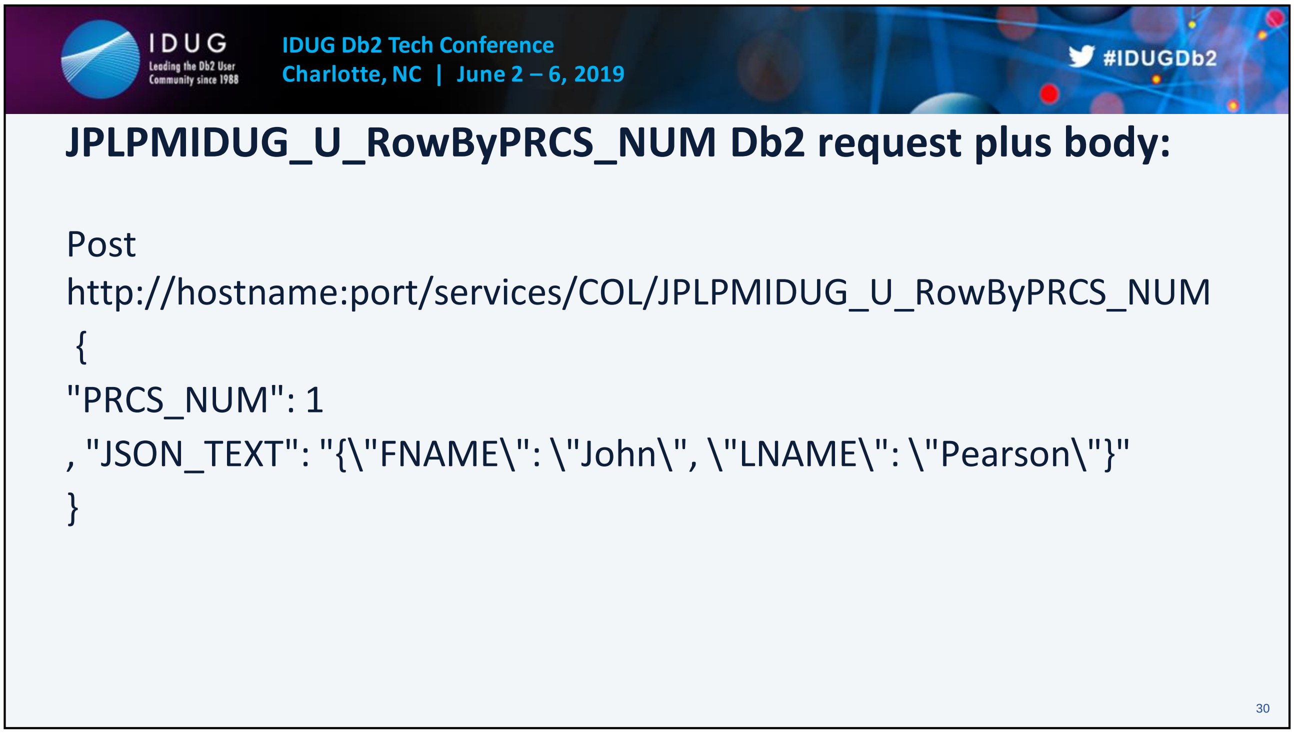
Post http://hostname:port/services/Db2ServiceManager

…

"sqlStmt":" UPDATE IDUG2019 SET JSON_TEXT=SYSTOOLS.JSON2BSON(:JSON_TEXT) WHERE PRCS_NUM=:PRCS_NUM", "serviceName": "JPLPMIDUG_U_RowByPRCS_NUM",

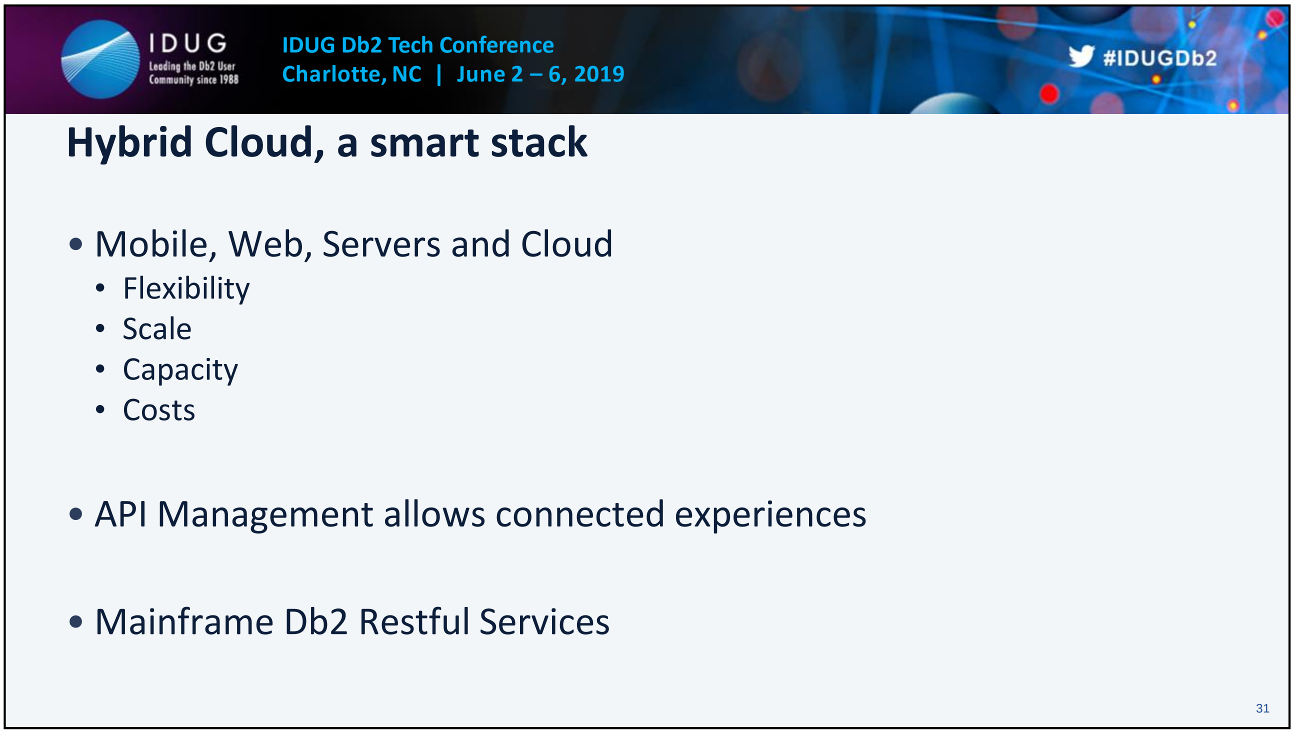
"description": "Update JSON_TEXT column by PRCS_NUM in the IDUG2019 table with a single SQL statement",

…



## JPLPMIDUG_U_RowByPRCS_NUM Db2 request plus body:

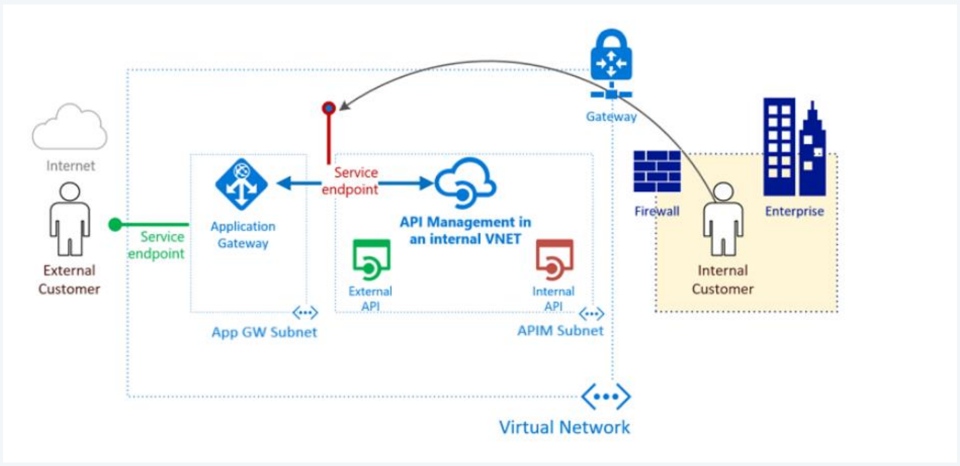
```
Post
http://hostname:port/services/COL/JPLPMIDUG_U_RowByPRCS_NUM
 {
"PRCS_NUM": 1
, "JSON_TEXT": "{\"FNAME\": \"John\", \"LNAME\": \"Pearson\"}"
}
```



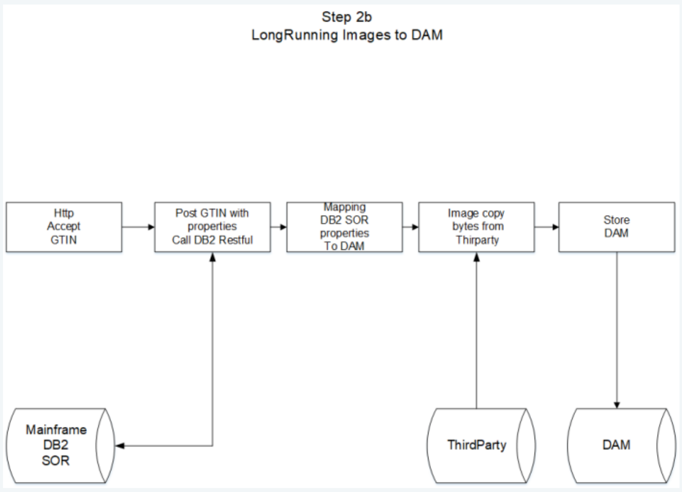
## Hybrid Cloud, a smart stack

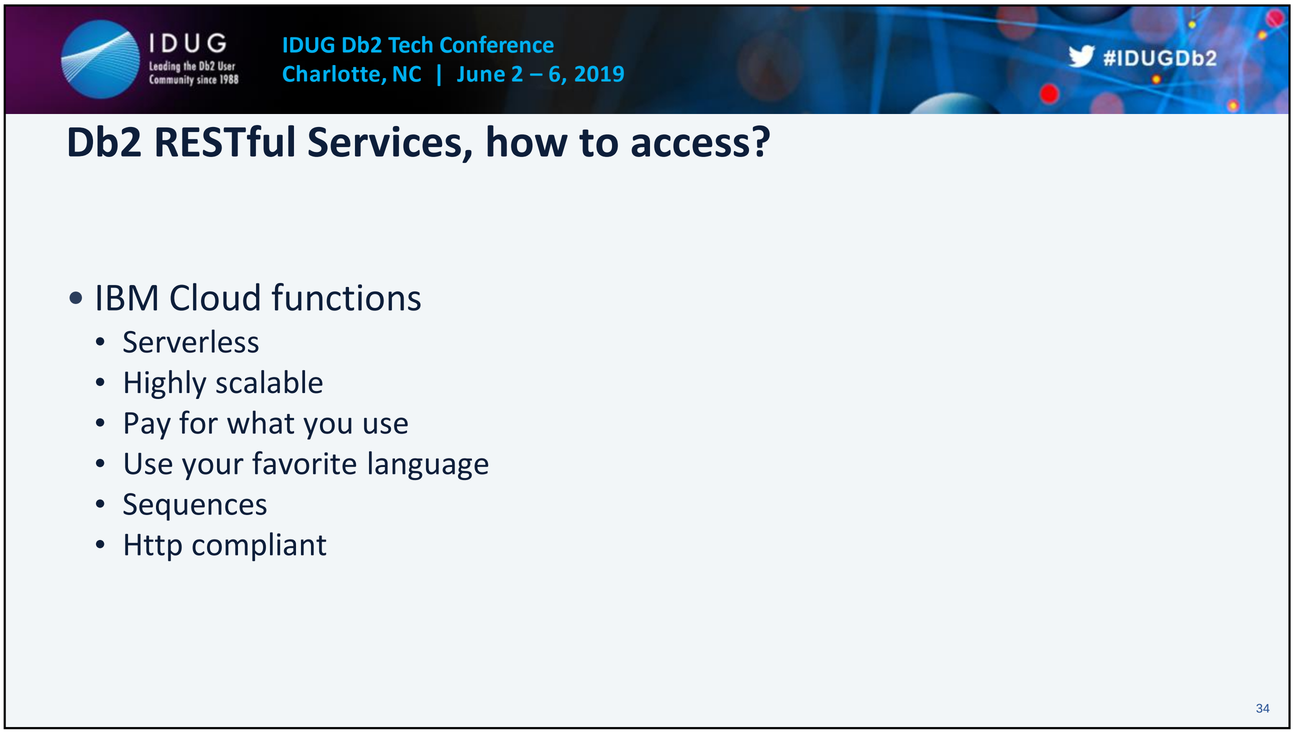
- Mobile, Web, Servers and Cloud
  - Flexibility
  - Scale
  - Capacity
  - Costs
- API Management allows connected experiences
- Mainframe Db2 Restful Services

## Microsoft API Management for on-premise access



## July 2018 rear view mirror

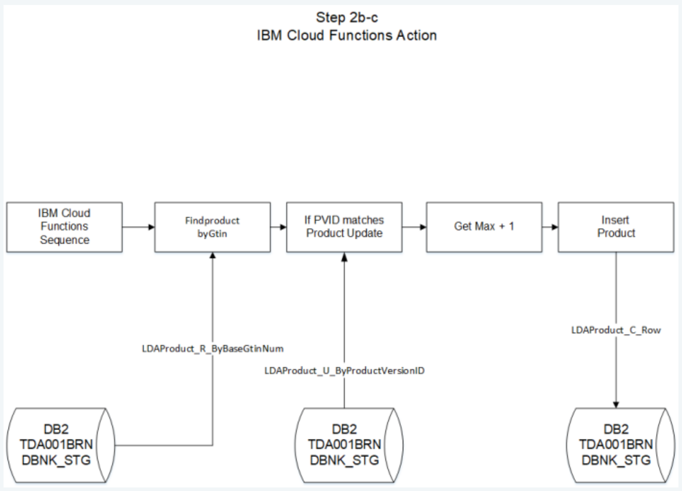

Step 2b
LongRunning Images to DAM
Http Accept GTIN
Post GTIN with properties Call DB2 Restful
Mapping DB2 SOR properties To DAM
Image copy bytes from Thirparty
Store DAM
Mainframe DB2 SOR
ThirdParty
DAM



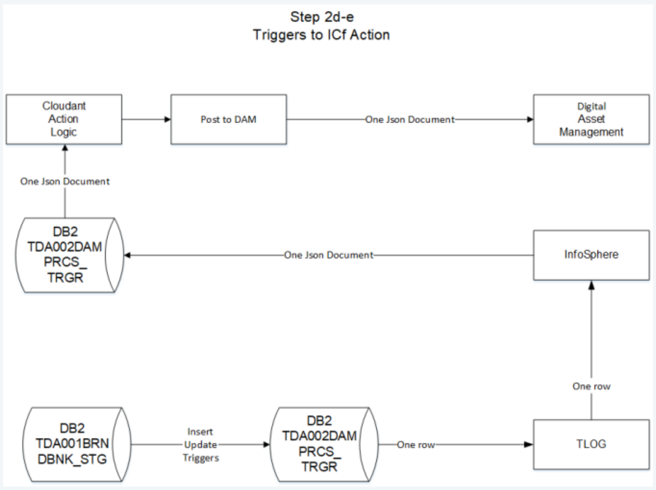
## Db2 RESTful Services, how to access?

- IBM Cloud functions
  - Serverless
  - Highly scalable
  - Pay for what you use
  - Use your favorite language
  - Sequences
  - Http compliant

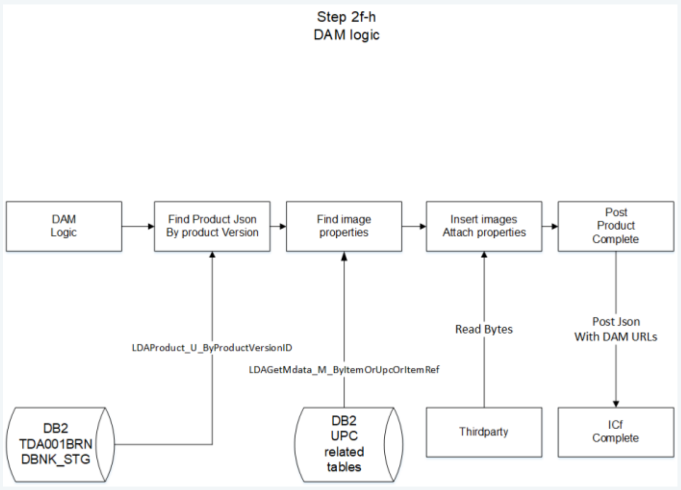
## January 2019 Architecture using Db2 Restful


Step 2b-c
IBM Cloud Functions Action
IBM Cloud Functions Sequence
Findproduct byGtin
If PVID matches Product Update
Get Max + 1
Insert Product
LDAProduct_R_ByBaseGtinNum
LDAProduct_U_ByProductVersionID
LDAProduct_C_Row
DB2 TDA001BRN DBNK_STG
DB2 TDA001BRN DBNK_STG
DB2 TDA001BRN DBNK_STG

## January 2019 Architecture using Db2 Restful



## January 2019 Architecture using Db2 Restful



**John Pearson**
**Wakefern**
**John.Pearson@wakefern.com**

Session code: E03

IDUG
Leading the Db2 User Community since 1988

*Please fill out your session evaluation before leaving!*