



OpenShift & IBM i Demonstration

<https://github.com/bmarolleau/acmeair-customerservice-java-jdbc>

Benoit MAROLLEAU – Cloud/AI Architect

IBM Systems Center Europe, Montpellier, France

benoit.marolleau@fr.ibm.com



Replays & Presentations

<https://ibm.biz/bma-wiki>





Survey 2: Cloud & AI

The IBM hybrid cloud approach

Hybrid cloud Applications



Infuse AI
for business throughout
our software

Integrated with
Cloud Paks,
on any cloud

Cloud Paks

Unify and simplify software experiences
with seamless integration of related capabilities



Hybrid cloud platform

Enable software to run anywhere, on any
cloud, as agile cloud-native services



IBM i

- The Integrated Promise of IBM i :
- Deliver high value platform for business apps
 - Provide exceptional security and resiliency for critical business data
 - Leverage IBM systems, storage and software technologies



Application Modernization

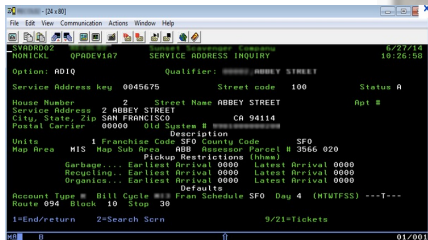
Extending Traditional Apps with Cloud Native

Capacity to Innovate vs. Risk

Spaghetti Architecture

Lasagna Architecture

Ravioli Architecture



Cut & Paste
(1990's)



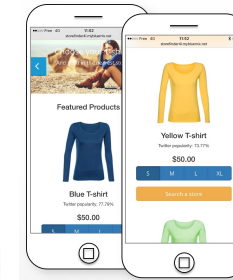
Layered Monolith
(2000's)



Microservices
(2010's)



OPENSIFT



Why Microservices?

→ Continuous Innovation
vs. Business Needs

How?

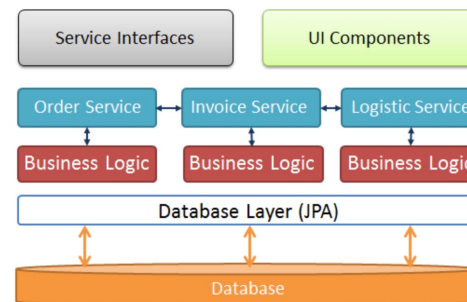
→ Flexible User facing components
(Micro-Services, 12-Factor, K8s)
→ Integrated with rock solid core
Business apps.

DevOps Ready

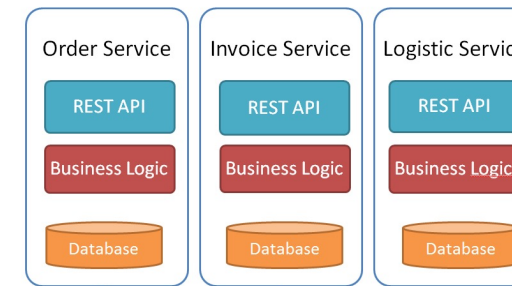
Verticals

```
0001.00 H DECEIT('')
0002.00 PAYEMP IF E K DISK
0003.00 PAYEMP CF E WORKSTN
0004.00 C 2-ADD 10.75 TRUX 4 2
0005.00 C *LIKE DEFINE TRUX PAY 2
0006.00 C *LIKE DEFINE TRUX OTPAY 2
0007.00 C *LIKE DEFINE NBRHRS OHEUR 10
0008.00 C MOVE 'JULIE' PREOPR 10
0009.00 C MOVE 'LARDUSSE' SEP 1
0010.00 C MOVE 'LARDUSSE' NBRHRS 10
0011.00 C MOVE PREOPR INITIA 1
0012.00 *
0013.00 C INITIA CAT SEP:0 WK1 5
0014.00 C WK1 CAT NBRHRS:1 NOMP 20
0015.00 C REOP PAYEMP
0016.00 C *INBO SOURCE NON
0017.00 C NBRHRS TPLE 35
0018.00 C NBRHRS MULT TRUX PAY
0019.00 C ELSE
0020.00 C TRUX MULT 35 PAY
0021.00 C NBRHRS SUB 35 OHEUR 9 4
0022.00 C TRUX MULT 1.75 OTTRUX
0023.00 C OTTRUX MULT OHEUR OTPAY
0024.00 C ADD OTPAY PAY
0025.00 C ENDIF
0026.00 C EXPMT FMT1
0027.00 C REOP PAYEMP
0028.00 C ENDD
0029.00 C SETON
```

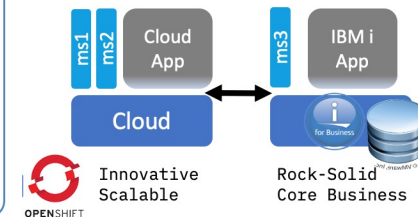
App Centric Monolith,
Single Program
Hard to Maintain & Change



Data Centric, Modular & Layered,
Modern Tech, Design Patterns (MVC...)
Horizontal (technical) Layers



Microservice, 12 Factor Design
Loosely coupled services
Vertical (business) layers



Demonstration



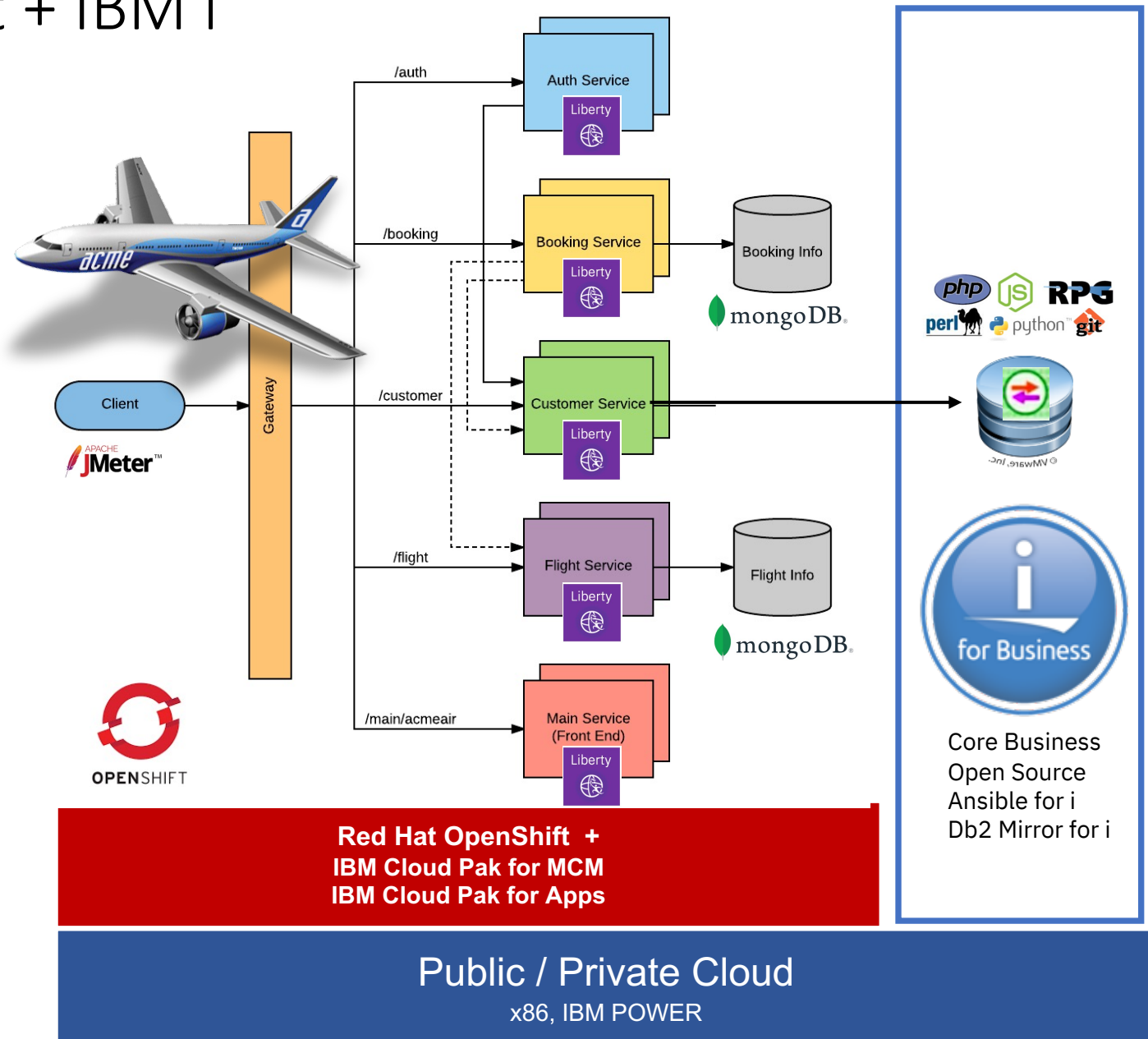
Unleash: Build new cloud-native solutions

- ✓ OCP on Power: 2.5 X more container density vs. x86 and Unequaled Reliability
- ✓ IBM Cloud Pak for Apps for easy Re-platforming & DevOps



Unlock: Modernize and leverage existing investments

- ✓ Real time replication w/ **PowerHA**
- ✓ Zero downtime with **Db2 Mirror for i**
- ✓ Maximum Resilience, Security (CVS Reports)
- ✓ Modernize and Expose existing Business Logic (RPG, COBOL, Node.js, Python etc.)



Micro-Service App : OpenShift + IBM i

Demonstration

The Acmeair microservices app is developed to analyze Cloud Environment performance.

Acmeair-CustomerService rewritten for IBM i



acmeair-mainservice-java

This service contains the front end of AcmeAir Microservices.

● HTML Apache-2.0 49 17 0 0 Updated 5 days ago



acmeair-flightservice-java

This service queries flights and reward miles.

● Java Apache-2.0 23 0 0 Updated 5 days ago



acmeair-customerservice-java

This service handle getting and updating customer data.

● Java Apache-2.0 23 0 0 Updated 5 days ago



acmeair-bookingservice-java

This service handles getting, making, and cancelling flight bookings.

● Java Apache-2.0 24 1 0 0 Updated 5 days ago



acmeair-authservice-java

During the login it generates a JWT which will be used by booking and customer services to validate the user.

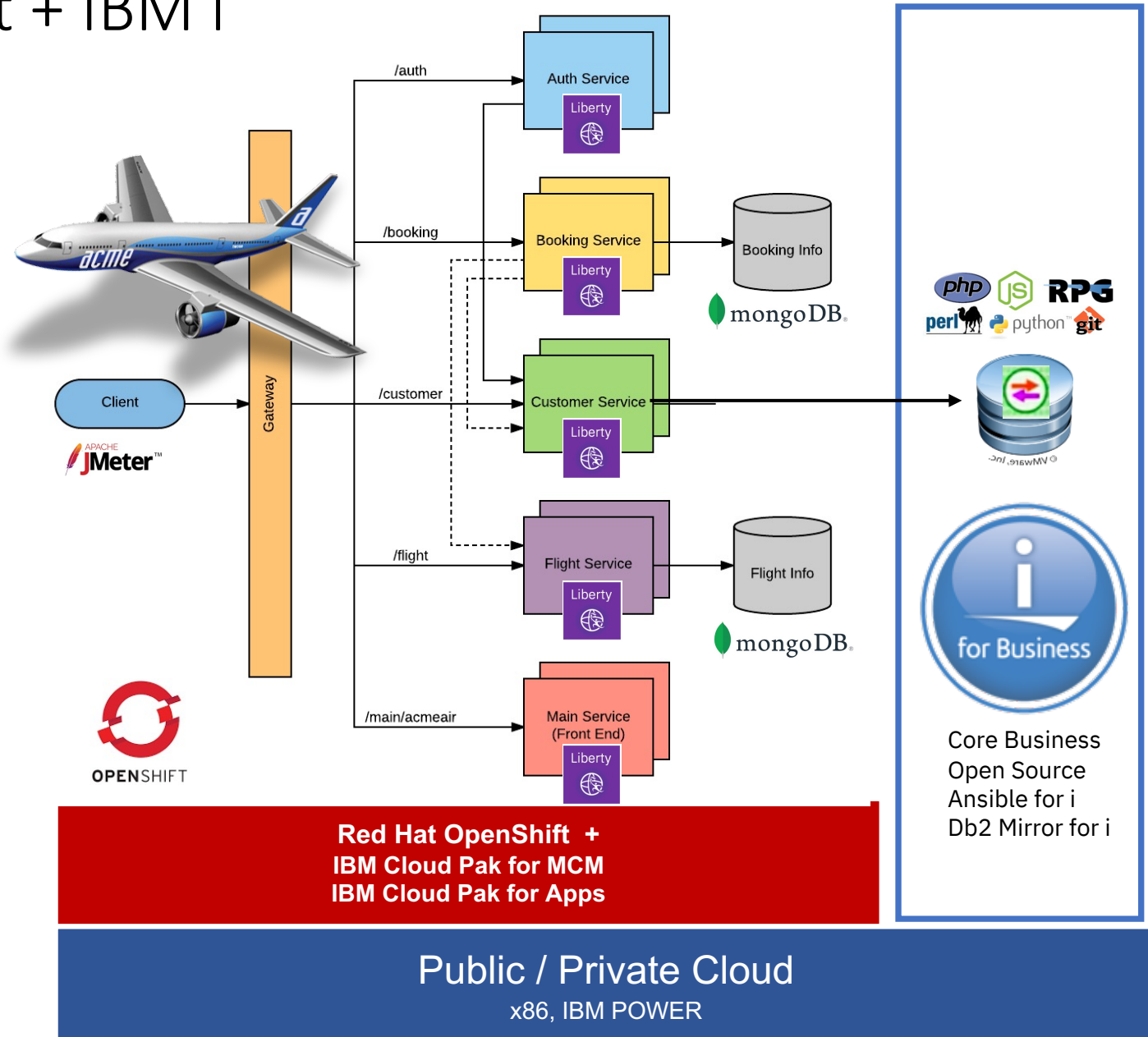
● Java Apache-2.0 23 1 0 0 Updated 5 days ago

acmeair-driver

Forked from acmeair/acmeair-driver

A workload driver for the Acme Air Sample Application.

● Java Apache-2.0 17 3 0 0 Updated on May 14



Source Code Version/ Branch used : microprofile-3.3

<https://github.com/blueperf/>

Micro-Service App : OpenShift + IBM i

Demonstration



App Modernization & Cloud Journey Showcase

A day in the life of an IBM i shop



Dev:

- Cloud Pak for Applications
- OCP for IBM i applications
- DevOps : 3rd Party + CP4MCM

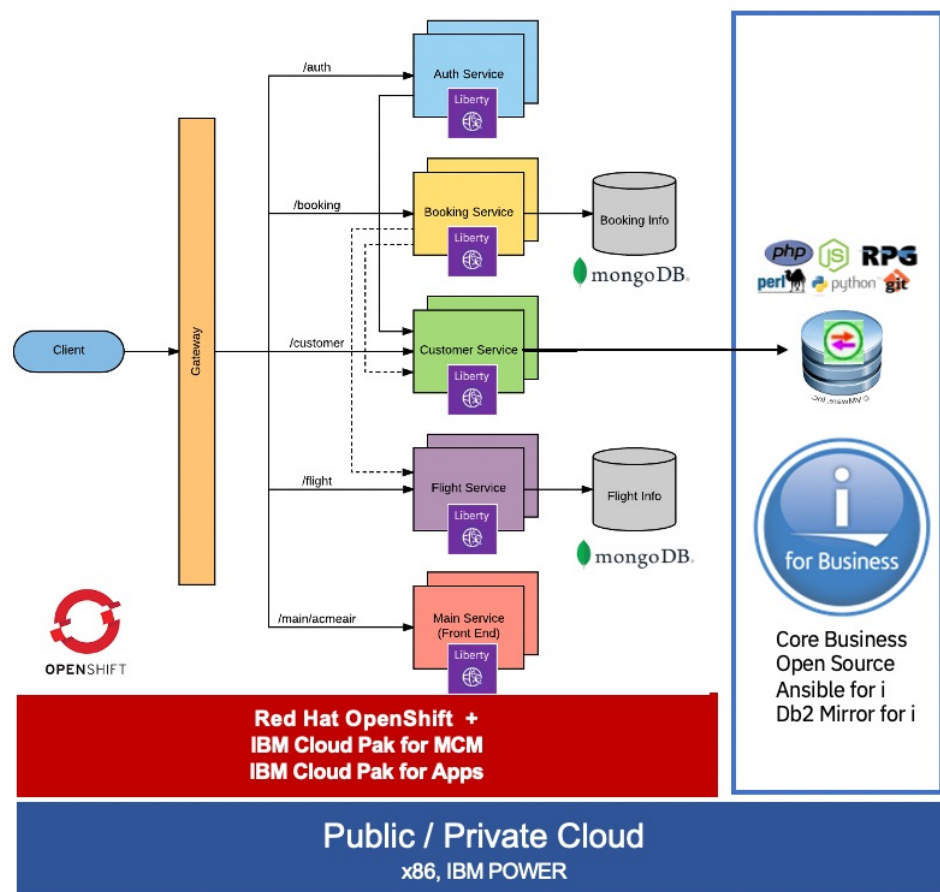


Ops:

- OpenShift vs. IBM i administration
- Cloud Pak for MCM (PVC/PVS)
- IBM i on PowerVS (IBM Cloud)



OPENSIFT



Permanent Demo / Workshop environment

POWER9 & IBM i 7.4
Move to Cloud
Db2 Mirror



Dev:

- RDi
- Open Source (Vscode, Node.js...)
- APIm / Web Services (IWS, Loopback)
- Ansible for i



Ops:

- Db2 Mirror
- Ansible AWX/Tower (GUI with RBAC)
- PowerHA/NVMe/Storage



AI/ Analytics:

- Db2 Web Query (hybrid data sources)
- AI on IBM i (PASE) , H2O.ai



Security :

- IBM i + Qradar , IBM i Security

Micro-Service App : OpenShift + IBM i

Demonstration Scenario

1. **Business Continuity & Multi-Cloud** : Db2 Mirror, PowerHA (IBM Cloud PVS Ready)

- Workload injection on OCP+IBM i (Acmair Booking engine) while failing over / failing back
- Showing the best of both worlds , IBM i Transactional & Stateless/OCP

2. **Modernization & Integration** :

Ansible for i: integration IBM i playbooks / AWX or Tower in the demo

Open Source & Integration : IBM i with exposed business logic in the demo

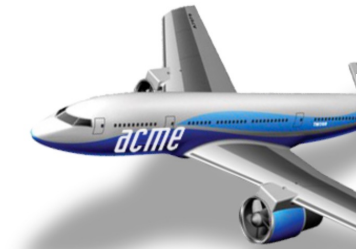
Development & DevOps: Rdi , Partner tools in action

PVS / Cloud – Multicloud management, ICC/Cloud Object Storage backup

Initial Acmeair App <https://github.com/blueperf/>

- Time to rewrite & test a jdbc based micro-service for IBM i : 3 days

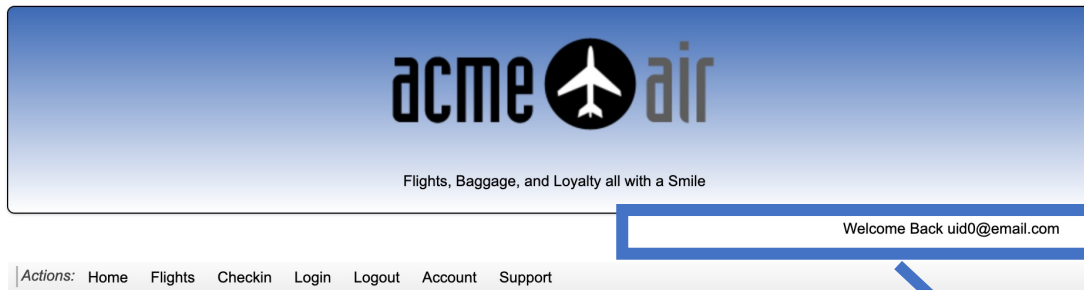
Principle: Distributed data stores are not always the best choice



This is a sample application for performance test the cloud.



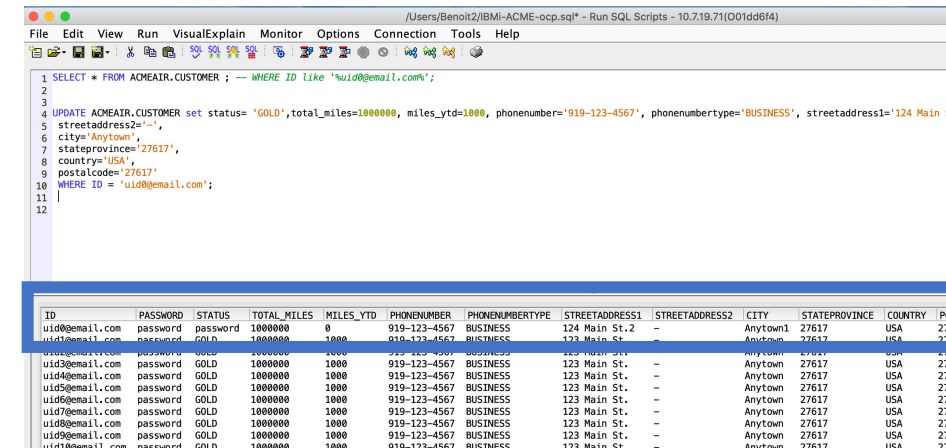
OPENSIFT



account id:	uid0@email.com
password:	

Phone Number:	919-123-4567
Phone Type:	Business
Street Address:	124 Main St.2
Street Address 2:	-
City:	Anytown1
State (Province):	27617
Country:	USA
Postal Code:	27617

- Login (Db2 for i customer data)
- Display & Update Account (Db2 for i)
- Book Flights
- Check-in
- Update
- Logout



OpenShift & IBM i integration : K8s Service

- Several alternatives to consume IBM i services from OpenShift
 - Use of K8s Services, a reverse proxy network component / abstraction layer between K8s micro-services and IBM i
- IBM i integration:

Database server (jdbc, odbc)

Ex: HA with jdbc + Db2 Mirror aware

Application server (Logic & Web Services on IBM i)

Ex: HA/DR with REST API + PowerHA

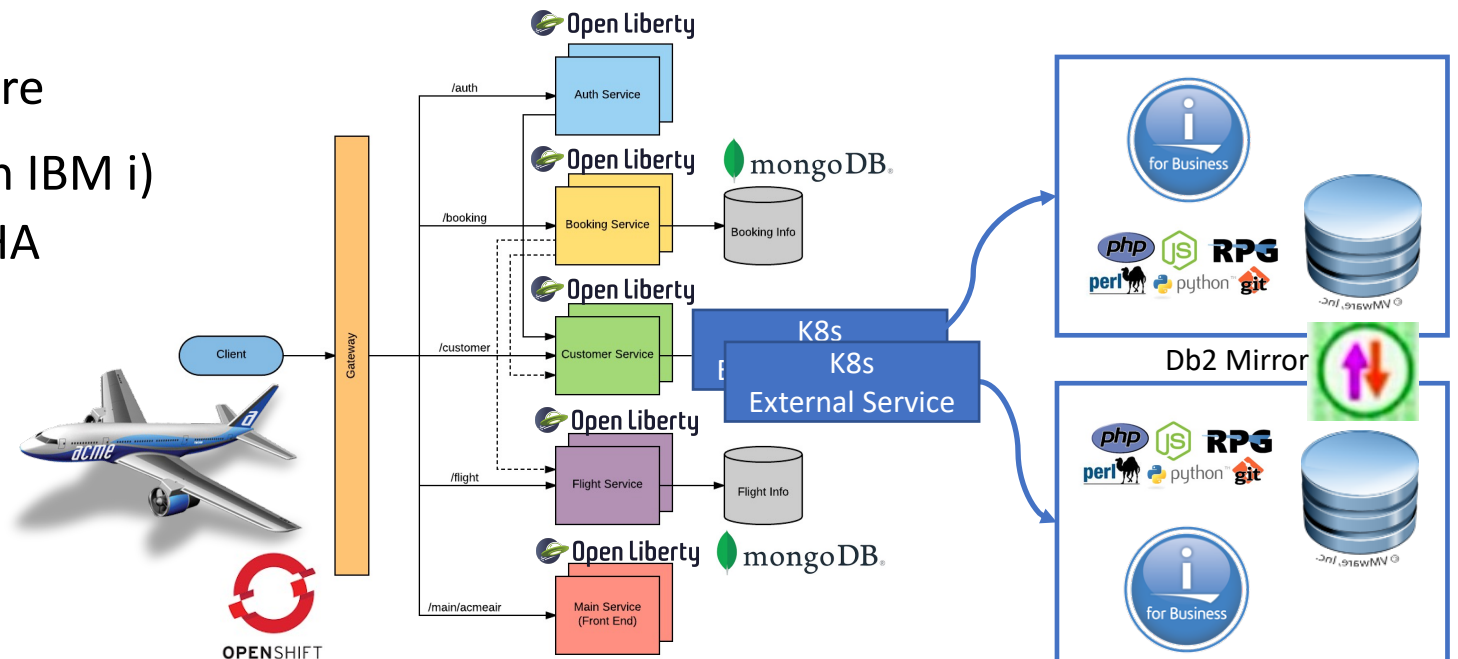
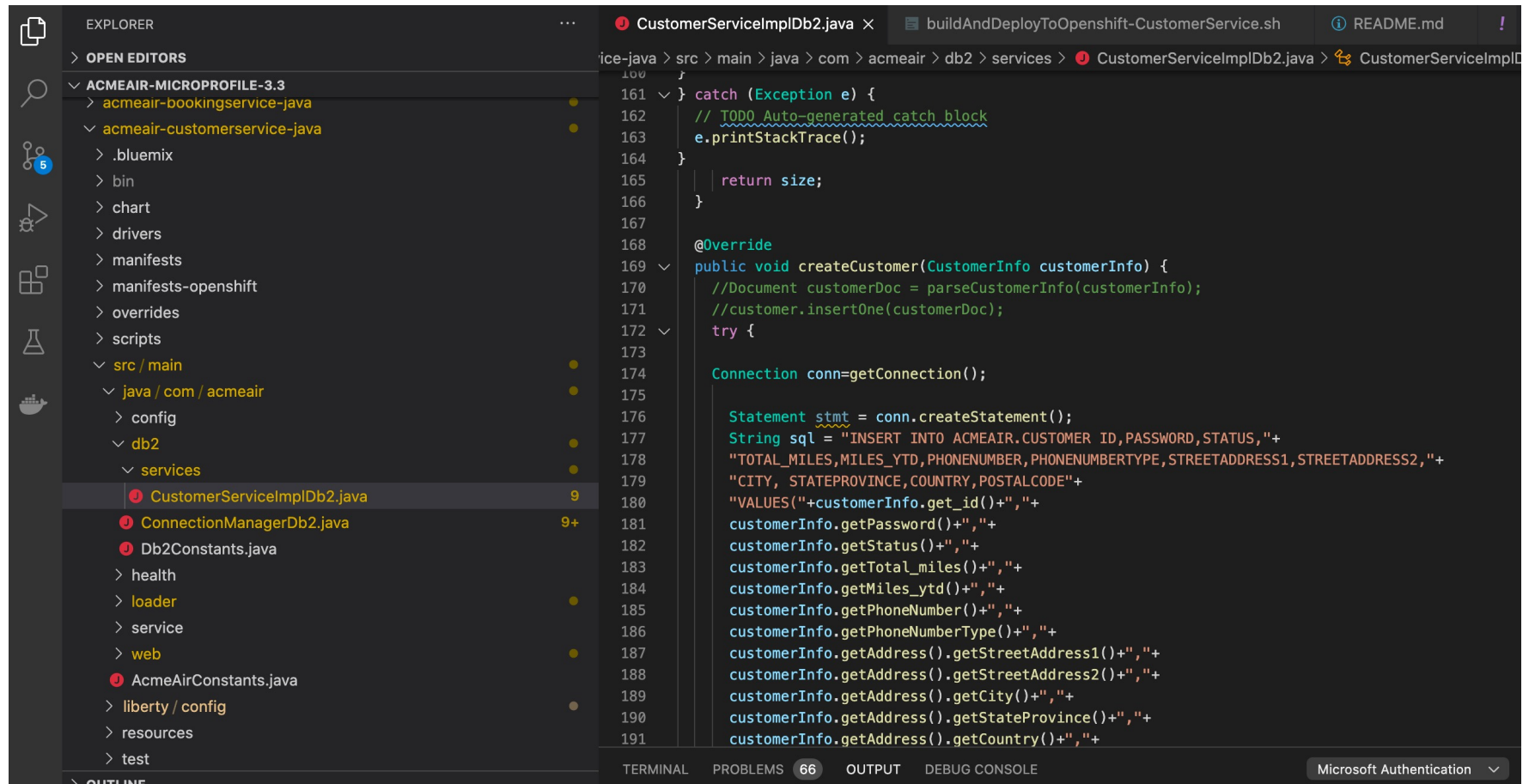


fig: OCP+ IBM i + Db2 Mirror

Dev: Micro-Service for IBM i

- IDE : Use of vscode or RDi for both micro-service (OCP) based and IBM i development (RPG, Python, Node.js...)

Simple micro-service implementation with Db2 for i integration (SQL, jdbc driver)



The screenshot shows the Visual Studio Code (VS Code) IDE interface. On the left, the Explorer sidebar displays a project structure for 'ACMEAIR-MICROPROFILE-3.3'. The project includes several sub-projects like 'acmeair-bookingservice-java' and 'acmeair-customerservice-java', along with various configuration and script files. The 'src/main/java/com/acmeair/db2/services' directory is expanded, showing 'CustomerServiceImplDb2.java' as the active file. The main editor area displays the code for 'CustomerServiceImplDb2.java', which implements a 'createCustomer' method. This method uses a JDBC connection to insert customer data into a Db2 database. The code includes a try-catch block for database operations and a return statement for the size of the data. The bottom status bar indicates '66' lines of code and 'Microsoft Authentication' as the active extension.

```
100 }
101
161 } catch (Exception e) {
162     // TODO Auto-generated catch block
163     e.printStackTrace();
164 }
165
166 return size;
167
168 @Override
169 public void createCustomer(CustomerInfo customerInfo) {
170     //Document customerDoc = parseCustomerInfo(customerInfo);
171     //customer.insertOne(customerDoc);
172     try {
173
174         Connection conn=getConnection();
175
176         Statement stmt = conn.createStatement();
177         String sql = "INSERT INTO ACMEAIR.CUSTOMER ID,PASSWORD,STATUS,"+
178             "TOTAL_MILES,MILES_YTD,PHONENUMBER,PHONENUMBERTYPE,STREETADDRESS1,STREETADDRESS2,"+
179             "CITY, STATEPROVINCE,COUNTRY,POSTALCODE"+
180             "VALUES('"+customerInfo.get_id()+"','"+
181             customerInfo.getPassword()+"','"+
182             customerInfo.getStatus()+"','"+
183             customerInfo.getTotal_miles()+"','"+
184             customerInfo.getMiles_ytd()+"','"+
185             customerInfo.getPhoneNumber()+"','"+
186             customerInfo.getPhoneNumberType()+"','"+
187             customerInfo.getAddress().getStreetAddress1()+"','"+
188             customerInfo.getAddress().getStreetAddress2()+"','"+
189             customerInfo.getAddress().getCity()+"','"+
190             customerInfo.getAddress().getStateProvince()+"','"+
191             customerInfo.getAddress().getCountry()+"','"+
```

Build: Micro-Service for IBM i

JDBC based micro-service example: Settings for accessing the highly available acmeair customer DB hosted on IBM i

The image shows an IDE interface with a project named 'ACMEAIR-MICROPROFILE-3.3'. The left sidebar shows the project structure, with the 'drivers' folder highlighted. The main editor displays the 'server.xml' file, which contains configuration for a JDBC data source. The configuration includes a library reference to 'DB2iToolboxLib', a connection manager, and a data source named 'acmeairdb'. The data source configuration includes properties for the database name, password, server name, user, and libraries. The configuration is annotated with green text and brackets indicating the purpose of different sections.

drivers

- customer-table.csv
- ddl-db2fori.sql
- jt400.jar

server.xml

```
<library id="DB2iToolboxLib">
  <fileset dir="/db2" includes="*.jar"/>
</library>

<connectionManager id="conMgr4" reapTime="-1" purgePolicy="FailingConnectionOnly" minPoolSize="50" maxPo
<dataSource id="acmeairdb" jndiName="jdbc/acmeairdb" statementCacheSize="60" connectionManagerRef="con
  <jdbcDriver libraryRef="DB2iToolboxLib"/>
  <properties.db2.i.toolbox
    databaseName="acmeair"
    password="${env.PASSWORD}"
    serverName="${env.DATABASE_HOST}"
    user="${env.USERNAME}"
    libraries="${env.LIBRARY_LIST}"
    clientRerouteAlternateServerName="${env.DATABASE_ALT_HOST}"
    enableClientAffinitiesList="1"
    enableSeamlessFailover="1"
  />
</dataSource>

<jndiEntry value="db2" jndiName="com/acmeair/repository/type"/>

<cors domain="/customer"
  allowedOrigins="*"
  allowedMethods="GET, DELETE, POST, OPTIONS"
  allowedHeaders="*"
  allowCredentials="true"
  maxAge="3600" />

</server>
```

Db2 for i credentials

Optional: Alternate server (ex: Db2 Mirror)

Build: Micro-Service for IBM i

- Docker & microservice build

Injection of Db2 for i drivers in the “acmeair-customer-service” image

```
INFO] Server defaultServer package complete in /Users/Benoit2/acmeair-microprofile-3.3/acmeair-customerservice-java-jdbc/target/acmeair-customer
INFO] -----
INFO] BUILD SUCCESS
INFO] -----
INFO] Total time: 12.314 s
INFO] Finished at: 2020-12-08T20:57:07+01:00
INFO] -----
Sending build context to Docker daemon 227.4MB
Step 1/9 : FROM open-liberty:full
Pull: Pulling from library/open-liberty
Digest: sha256:ff9b703e6b6d99f15e4056508a4bcf4228547fbcf6f89c98fd6f03bc3584b449
Status: Image is up to date for open-liberty:full
--> f01dbd22d389
Step 2/9 : COPY --chown=1001:0 src/main/liberty/config/server.xml /config/server.xml
--> 554d1715e619
Step 3/9 : COPY --chown=1001:0 src/main/liberty/config/server.env /config/server.env
--> 3ed4d667c26e
Step 4/9 : COPY --chown=1001:0 src/main/liberty/config/ivm.options /config/ivm.options
--> 5ceca45dedf5
Step 5/9 : COPY --chown=1001:0 target/acmeair-customerservice-java-3.3.war /config/apps/
--> 83777b9eb71a
Step 6/9 : COPY --chown=1001:0 drivers/jt400.jar /db2/jt400.jar
--> 101cd735dd8e
Step 7/9 : ARG CREATE_OPENJ9_SCC=true
--> Running in f74ead8cc333
Removing intermediate container f74ead8cc333
--> d4f0fc7e53c1
Step 8/9 : ENV OPENJ9_SCC=${CREATE_OPENJ9_SCC}
--> Running in b79d4cb321d1
Removing intermediate container b79d4cb321d1
--> 9cf51fa94c6b
Step 9/9 : RUN configure.sh
--> Running in 6210e26dff2
```

- Result

AcmeAir Web Page w/ microservices details

Acme Air Configuration information

Database Information

Entity	Count
Bookings	0
Customers	10000
Flights	2364
Flight Segments	394
Airports	31



LIB ACMEAIR
Table CUSTOMER

Auth Service

Java,1.8.0_275,Eclipse OpenJ9

Booking Service

mongo
Java,1.8.0_275,Eclipse OpenJ9

Customer Service

db2 for i
Java,1.8.0_275,Eclipse OpenJ9

Flight Service

mongo
Java,1.8.0_275,Eclipse OpenJ9

How to connect OpenShift to IBM i / Db2 for i ?

Environment variables are used by each micro-service
Credentials can be encoded (even encrypted) using K8s Secrets



Db2 for i connection information
and credentials (K8s Secret protected)

Deployments > Deployment Details

D acmeair-customerservice

- Details
- YAML
- Replica Sets
- Pods
- Environment
- Events

Container: acmeair-customerservice-java

Single values (env)

NAME	VALUE	
DATABASE_HOST	ibmi-database	
LIBRARY_LIST	acmeair	
SECURE_SERVICE_CALLS	true	
ACMEAIR_STACKAA_AUTH_URL	http://acmeair-auth-service:9080/auth	
USERNAME	ibmi-acmeair user-profile	
PASSWORD	ibmi-acmeair password	

How to connect OpenShift to IBM i / Db2 for i ?

Like Internal resources, external resources can be reached through K8s services

service name used by the app →

real IBM i hostname →

```
kind: Service
apiVersion: v1
metadata:
  name: ibmi-database
spec:
  type: ExternalName
  sessionAffinity: None
  externalName: bendemo.10.7.19.71.nip.io
```

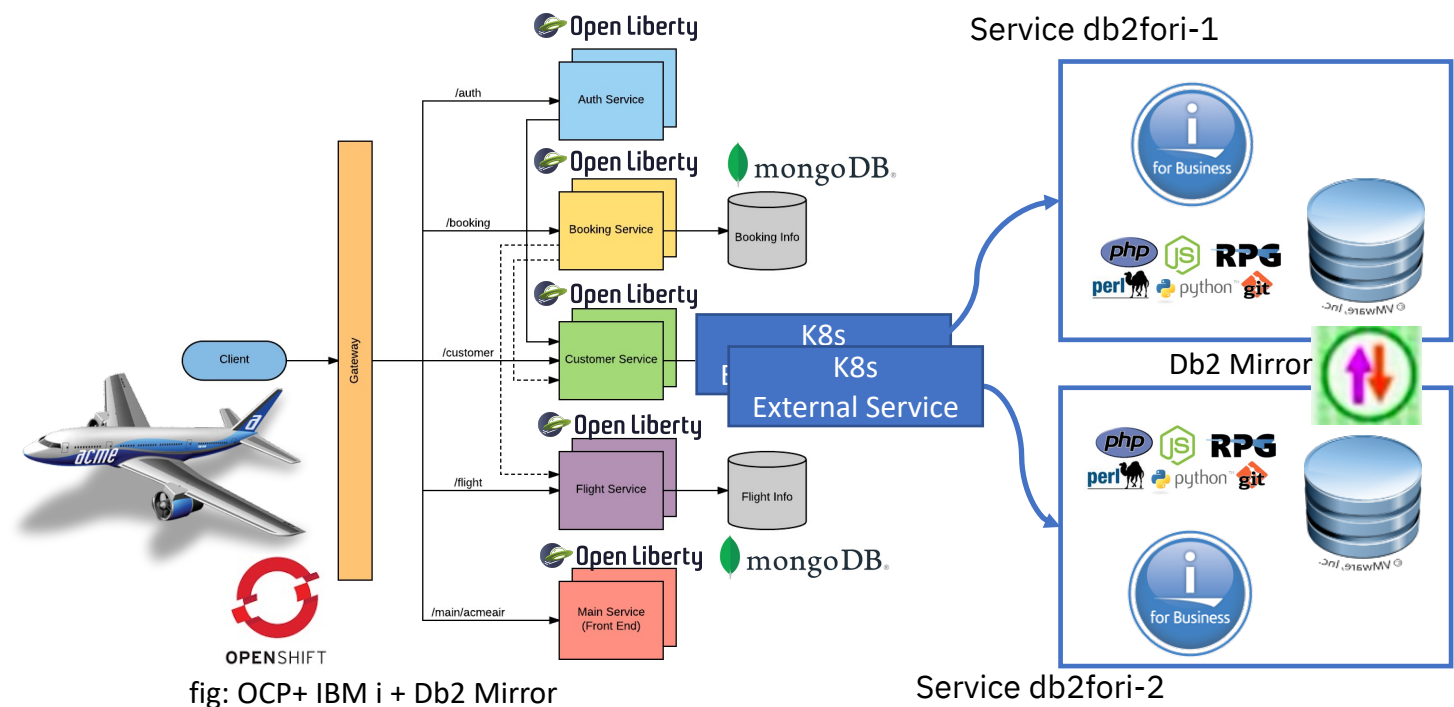


fig: OCP+ IBM i + Db2 Mirror

Need 24x7 availability?

IBM Db2 Mirror for i

Application Evaluation - Replication Implementation

Primary - SYSTEMASecondary - SYSTEMBBoth Nodes

Calculate Object Count

Default Inclusion State: + Include

Library Name

acmeair

Replication State

All

Object Count

-

ACMEAIR

+ Include

-

<<<1>>>

300

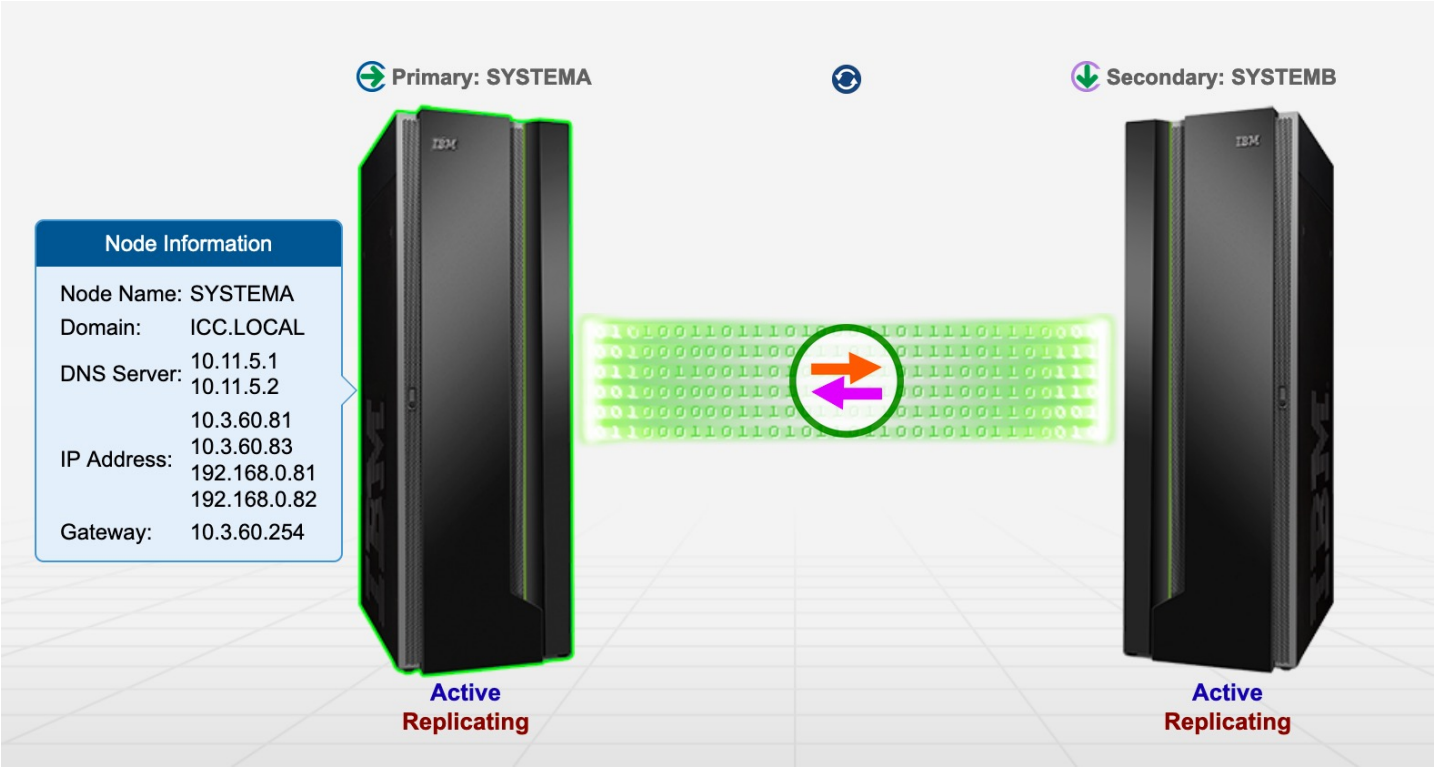
Showing 1 of 1

ACMEAIR

Object Type	Replication State	Object Cou
*FILE	+ Include	25
*FILE	⊘ Ineligible	1
*JRN	+ Include	1
*JRNRCV	⊘ Ineligible	20
*PGM	+ Include	1

<<<1>>>

100



Open
standards
and open
source
create the
best cloud
architecture



Enterprise-grade
software on the most
secure public cloud

Deep industry
expertise

Built on Linux OS,
delivered by
Containers, managed
by Kubernetes

Robust open source
ecosystem



Transform, run, and
manage your critical
workloads —
anywhere

Thanks !

Benoit MAROLLEAU – Cloud/AI Architect

IBM Systems - Montpellier, France

benoit.marolleau@fr.ibm.com

#IBMi #IBMiOSS Fan



Modernization Journey on Power Systems

IBM Systems Technical Sales
Lab Services | IBM Garage for Systems

Workshops

Infrastructure Modernization - Ops Efficiency

Rehosting – Replatforming - Private/Public Cloud Automation – Business Continuity



Application & Data Integration

Standardized back-end integration, **Repackaging** - Master Data Mgt



App Modernization , DevOps & CI/CD

Maximize ROI (TCO), Digital transformation (UI/UX, **AI**) , **Refactoring** with DevOps toolchain & Micro-services



How ? IBM Systems Technology, and Open Source Software.
At scale with Red Hat & IBM Cloud Paks



IBM Cloud Pak



Introduction: Application Modernization

DEMO PART 1 : Containerization / Replatforming using IBM Transformation Advisor

DEMO PART 2 : IBM i (VM) based apps Integration with OpenShift

DEMO PART 3 : Performance Management with RedHat OpenShift on Power Systems
HA/DR considerations



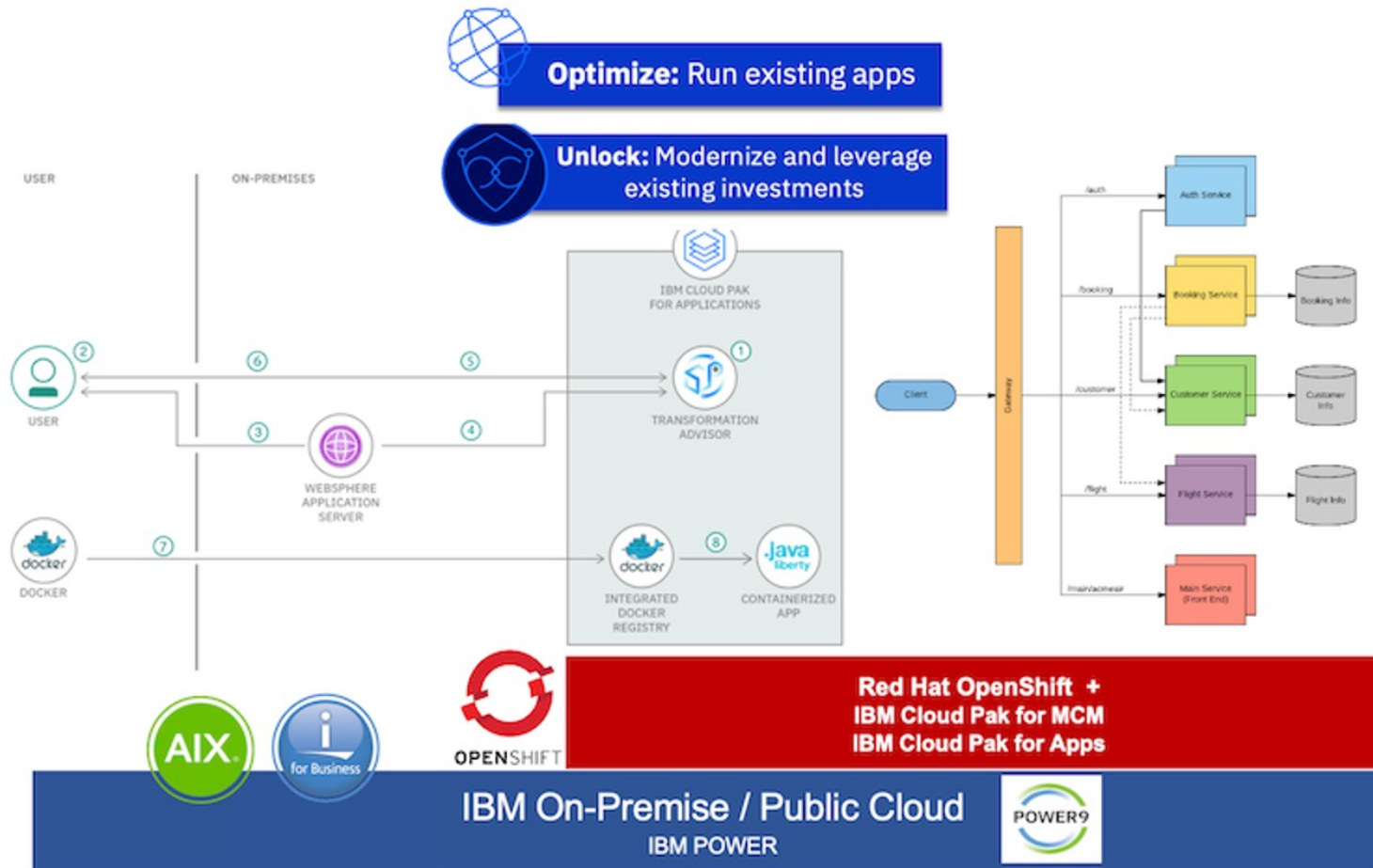
DEMO PART1 : Modernization with Containerization / Replatforming

Modernize your traditional on-premises app and deploy it as a containerized app on OpenShift

Find the original app (.war in the target dir) and the generate bundle here:

https://github.com/bmarolleau/acmeair_migrationBundle

Demo Video here: <https://www.youtube.com/watch?v=tdStP9Ck> 



CP4A / Transformation Advisor

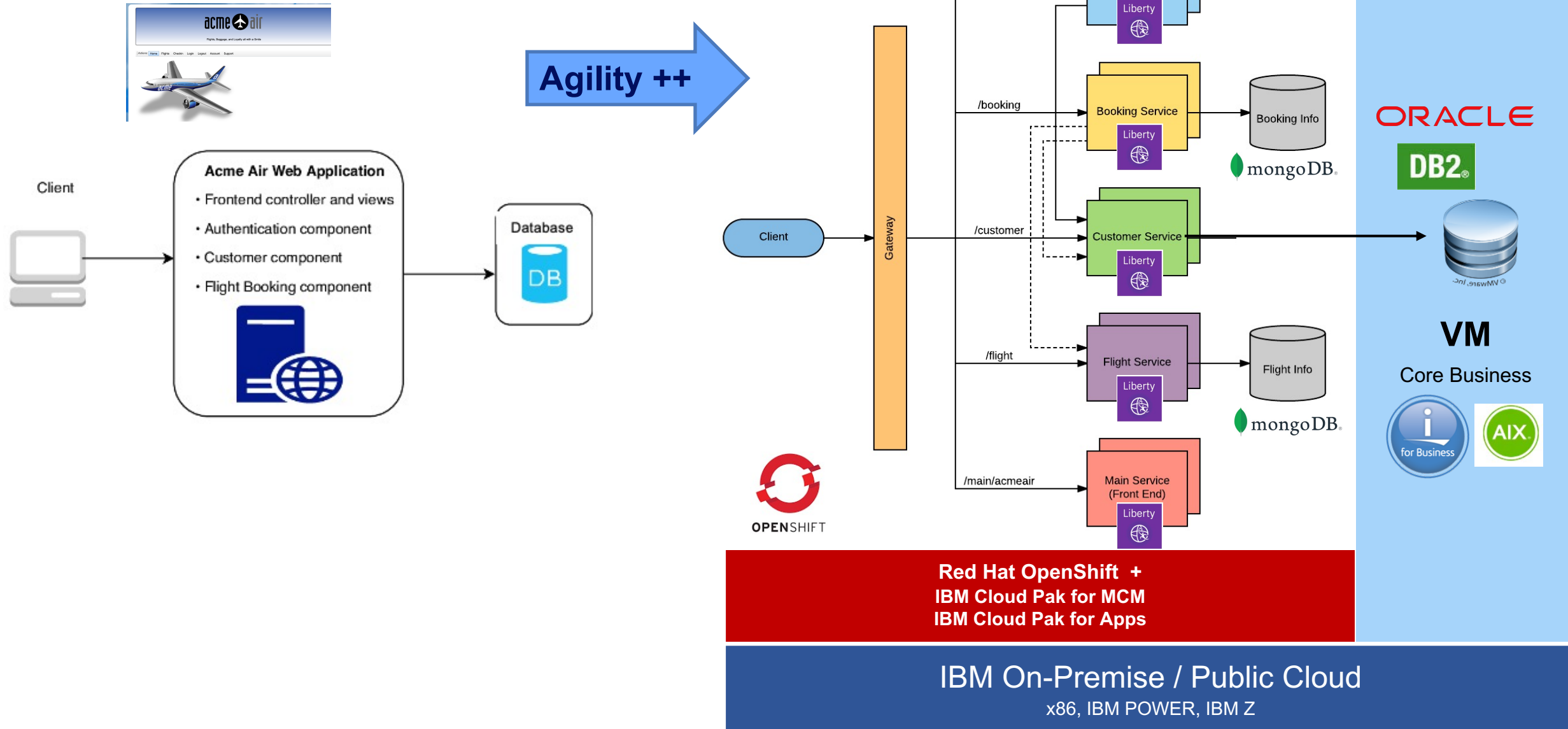
1. Developer uses IBM Transformation Advisor Local or on IBM Cloud Pak for Applications on the IBM managed OpenShift cluster.
2. Developer downloads a custom Data Collector from IBM Transformation Advisor
3. Developer runs the Data Collector on the traditional WebSphere Application Server host where application (to be migrated) is running. In this example, tWAS (WAS Base) version 9.0 with the Acmeair application running.
4. Data Collector analysis is uploaded (automatically or manually)
5. Developer reviews recommendations in Transformation Advisor and creates a migration bundle
6. Developer downloads migration bundle

OCP / S2i/ Multi-Arch :

7. Developer uses Docker to build an image and upload it to OpenShift Docker Registry
8. Developer creates an app using the pushed image and runs the containerized app

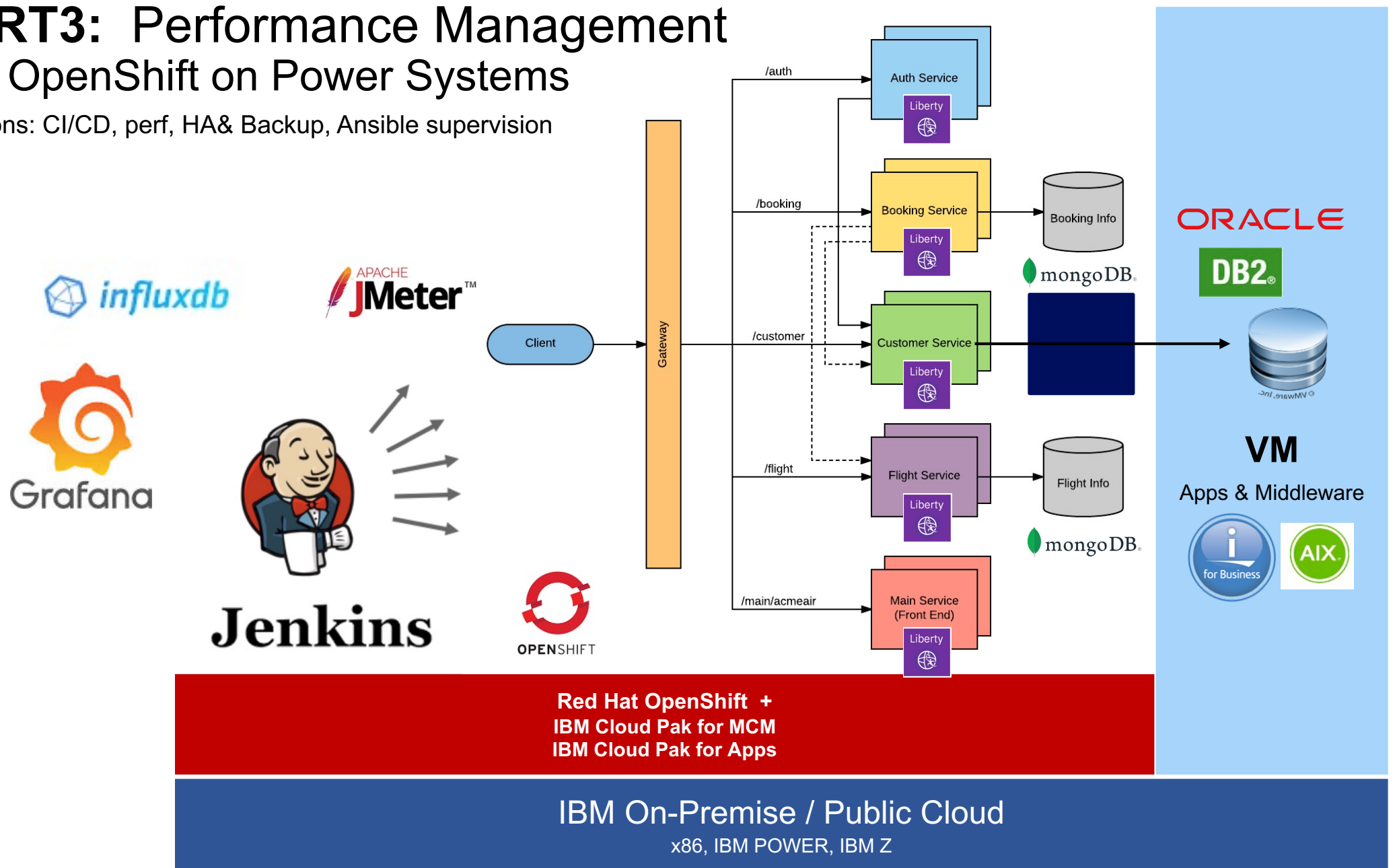
DEMO PART2 : Monolith to Micro-service

Integration of your traditional apps with OpenShift



DEMO PART3: Performance Management with RedHat OpenShift on Power Systems

Day to day Operations: CI/CD, perf, HA& Backup, Ansible supervision



Why OCP on Power?

Openshift on IBM Power Key differentiators

Performance & Scalability : Better performance vs x86

- 2 to 5 X more containers per core on POWER (Container density)
→ 8 thread per core (SMT8), Memory Bandwidth...
- **PowerVM Scalability** 500 containers/VM
- Unrivalled Power Server Performance with OCP
- 3.2X per core on OCP - MongoDB vs. Intel Xeon SP Gold 6150 Skylake
- **Java+MongoDB verified by a benchmark in Montpellier** → → →
2.6X throughput per core , 2.8x lower response times



IBM S924:

2 sockets system
24 Power 9 cores / 384 GB Memory
10Gb/s Ethernet card
PowerVM



versus



Lenovo SR650:

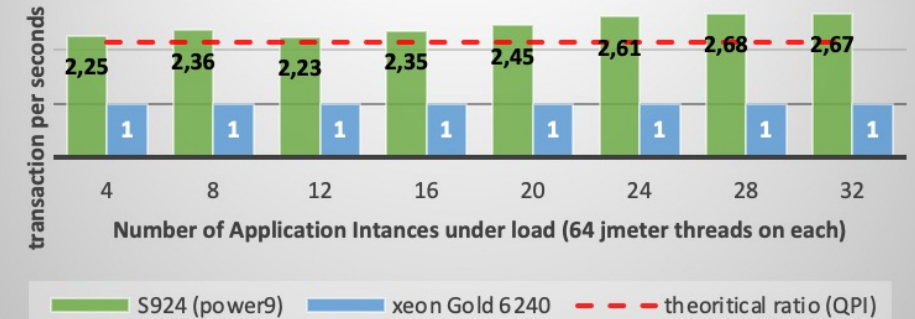
2 sockets system
36 Xeon Gold 6240 @2.6Ghz / 384 GB Memory
10Gb/s Ethernet card
VMWare ESXi 6.7

x2.3
x1.5

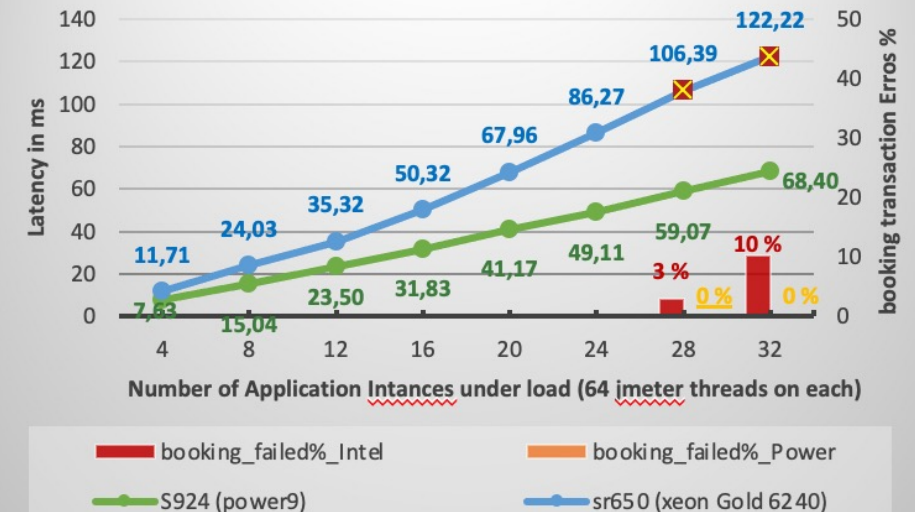
more Transactions /Core on Power

more Transactions per \$

Application TPS per core ratio
while increasing application instances



Application average latency & SLA



Real Life Testing : Microservice benchmark OCP + AcmeAir
x86 Cascade Lake vs. POWER9

Why OCP + IBM i ?

Application Modernization

Extending Traditional Apps with Cloud Native

Capacity to Innovate vs. Risk

Spaghetti Architecture

Lasagna Architecture

Ravioli Architecture



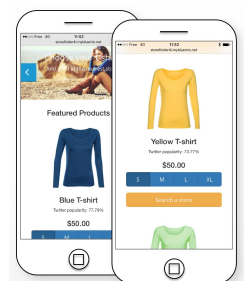
Cut & Paste
(1990's)



Layered Monolith
(2000's)



Microservices
(2010's)



Why Microservices?
→ Continuous Innovation
vs. Business Needs

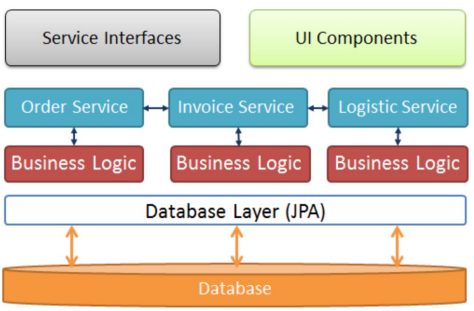
How?
→ Flexible User facing components
(Micro-Services, 12-Factor, K8s)
→ Integrated with rock solid core
Business apps.

DevOps Ready

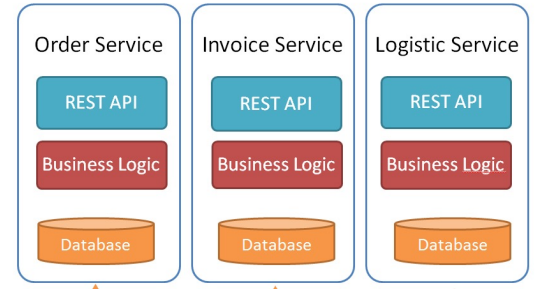
Verticals

```
0001.00 H DECEIT('')
0002.00 PAYEMP IF E K DISK
0003.00 PAYEMP CF E WORKSTN
0004.00 C 2-ADD 10.75 TRUX 4 2
0005.00 C *LIKE DEFINE TRUX PAY + 2
0006.00 C *LIKE DEFINE TRUX OTPAY + 2
0007.00 C *LIKE DEFINE NBRHRS OHEUR 10
0008.00 C MOVE 'JULIE' PREOPR 10
0009.00 C MOVE 'LARDUSSE' SEP 1
0010.00 C MOVE 'NOMPR' NOMP 10
0011.00 C MOVE PREOPR INITIA 1
0012.00 C *
0013.00 C INITIA CAT SEP:0 WK1 5
0014.00 C WK1 CAT NOMPR:1 NOMP 20
0015.00 C REOP PAYEMP
0016.00 C *INBO SUMME *ON
0017.00 C NBRHRS TPLE 35
0018.00 C NBRHRS MULT TRUX PAY
0019.00 C ELSE
0020.00 C TRUX MULT 35 PAY
0021.00 C NBRHRS SUB 35 OHEUR 9 4
0022.00 C TRUX MULT 1.75 OTTRUX
0023.00 C OTTRUX MULT OHEUR OTPAY
0024.00 C ADD OTPAY PAY
0025.00 C ENDIF
0026.00 C EXPMT FMT1
0027.00 C REOP PAYEMP
0028.00 C ENDDO
0029.00 C SETON
```

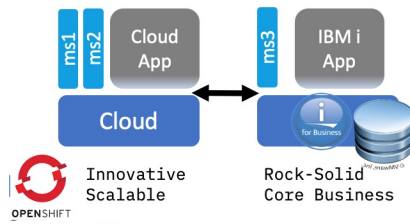
App Centric Monolith,
Single Program
Hard to Maintain & Change



Data Centric, Modular & Layered,
Modern Tech, Design Patterns (MVC...)
Horizontal (technical) Layers



Microservice, 12 Factor Design
Loosely coupled services
Vertical (business) layers

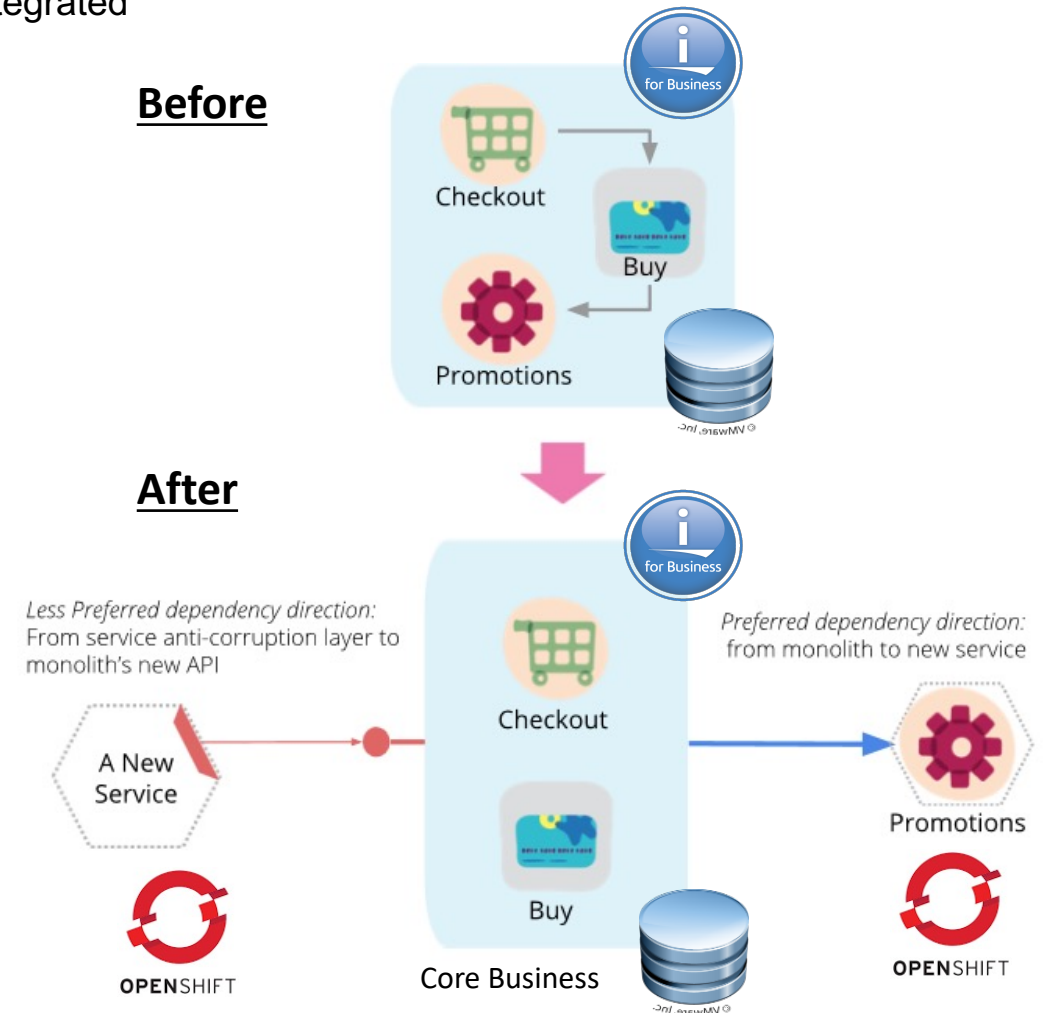


Traditional Apps & Cloud Native Apps

Everything is at your finger tips for building Cloud Native & Microservices solutions integrated with any traditional applications.

Modernize your apps on IBM i !

1. Upgrade your Traditional environment (HW/SW)
2. Re-discover , map your apps & data
(using IBM or Third party tooling)
 1. Modernize with free open source & latest IBM i / Db2 features on IBM i
2. Need to create inovative/continuously changing services/apps ? cloud native technologies: OpenShift Container Platform
3. Need full support & Enterprise middleware? Evaluate IBM Cloud Paks on OCP



Why CP4Apps ?

Cloud Pak for Applications



Optimize: Run existing apps

WebSphere Application Server

WebSphere ND | WebSphere Base
Liberty Core | Mobile Foundation

JBoss Enterprise Application Platform



Unlock: Modernize and leverage existing investments

IBM Modernization & Developer Tools

Included with all components

- Transformation Advisor
- Mono2Micro*
- Application Navigator
- WebSphere Migration Toolkit

Enterprise Dev tools & extensions for local IDE's
Supported when used with Cloud Pak for Applications, no charge

**open beta*

Perpetual and Term licensing options available
w/ no functional restrictions to OpenShift



Unleash: Build new cloud-native solutions

Accelerators for Teams & Enterprise Governance

Frictionless cloud-native development for multi-disciplinary teams.
Enable developers to rapidly innovate knowing they comply with your unique operational, security, and technology standards.

Red Hat CodeReady Workspaces

Collaborative OpenShift-native IDE

Enterprise Runtimes

- | | |
|-------------------------|--------------------------------|
| • Traditional WebSphere | • Node.js |
| • Liberty | • Spring Boot |
| • Mobile Foundation | • JBoss WS |
| • Open Liberty | • Vert.x |
| • JBoss EAP | • Cloud Functions (Serverless) |
| • Quarkus | • OpenJDK |

Distributed Data

SSO

Messaging

Red Hat OpenShift Container Platform

Modernize AIX/IBM i Apps with the Cloud Pak for Applications on OpenShift on Power

6x lower TCO than AWS,
2x lower TCO than x86
(on-premises)

Cloud Pak for Applications

Build, deploy and run applications

Red Hat OpenShift 4.3

Marry AIX / IBM i applications with cloud-native services to modernize one microservice at a time. Develop apps once, run anywhere.

- Improve agility, innovate faster with Kubernetes container based PaaS
 - WebSphere Liberty, node.js, NGINX
 - Red Hat Runtimes, JBOSS, Service Mesh
 - Developer productivity, Ops efficiency
- Proven Enterprise Server Attributes
 - Secure, Scalable, Resilient, Performant
 - Lower Cloud TCO – 6x less than AWS and 2x less than x86 private cloud (on-



✓ [Learn more](#) ✓ [Try it now](#)

Solution Stack / Config Type	Proof-of-Concept	Entry Production
Cloud Native DevOps stack	Cloud Pak for Apps on OCP 4.3	Cloud Pak for Apps on OCP 4.3
Cores: Master-Infra / Worker	1 x Master: 2c, 32GB VM 3 x Workers: 6c, 96gb VM	3 x Masters: 2c, 16 gb / VM each 3 x Workers: 2c, 16 gb VM each 1 x Shared Svc: 2c, 32GB VM
IBM Servers	1 x Power System L922s 20c @ 2.9GHz, 256GB	3 x Power System L922s 16c @ 3.4GHz, 256GB
Storage (internal per server)	2x1TB SSD, 2x2TB HDD	8x480GB SSD, 2x128GB HDD
Storage (external)	1 TB NFS storage	1 TB NFS storage
IBM System SW	PowerVM	PowerVM
Operating System	Red Hat CoreOS	Red Hat CoreOS
List price for system stack	Insert local price for L922+PowerVM (OpenShift 4.3 included in Cloud Pak)	Insert local price for L922s+PowerVM (OpenShift 4.3 included in Cloud Pak)