

Power Week 2025

#pw2025

18 - 19 - 20 novembre 2025

IBM Innovation Studio Paris

S67- Intégration et Open Source

20 novembre 11:15 - 12:15

Benoit Marolleau

IBM Client Engineering

benoit.marolleau@fr.ibm.com

IBM

common
FRANCE

IBM Client Engineering | EMEA

Custom Demos, Architecture Workshops, MVP Prototyping...

Let's create ↻
value together

Data & AI
Squad

01 API-zation / REST / Core Business Integration

02 DB2 Data Modernization: from Native to SQL

IBM Z &

03 OpenShift and DevOps

Modernization Squad



04 Open-Source Adoption with Legacy on IBM i

05 Power/IBM i move to cloud: IBM Cloud PowerVS

06 IBM i Resilience: PowerHA, DB2 Mirror

Sustainability Squad

07 Ansible Automation and AIOps

Hybrid Cloud
& AIOps Squad

08 Data Governance & MLOps with CP4Data

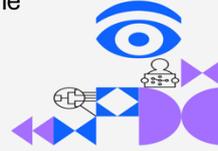
09 Infuse AI in core business apps

NEW: Watsonx Code Assistant for Z / Bob Code Assistant



IBM TechXchange : Power11 - Shaping the Future of Hybrid Cloud and AI

IBM Montpellier, France
Wednesday 3rd December
Thursday 4th December



JOIN US, for 2 days of INNOVATION, December 3rd and 4th in **Montpellier** (South of France), to discover how **New IBM Power11** can help you swiftly adopt future of Hybrid Cloud and AI.

Our technical experts will present the New Power11 features and provide best practices to take benefits from IBM solutions based on Power systems and AI Accelerator Cards (Spyre).

Over demonstrations and **HANDS-ON**, you will learn how to build a Hybrid Cloud Environment based on Power Virtual Server (PVS)

You will create a GenAI test case based on typical AI use cases (Q&A, Summarization, Entity Extraction).

Meet our experts to share your business cases, IT projects, security constraints to shape the Future with IBM Power11.

Audience: CIO, CTO, Chief Data Officer, Data Scientist, AI & Data engineer, Architect, DBAs, SMEs and Partners

Agenda:

Day1 : Presentations and Demos

- Introduction and Power11 Overview
- AI on Power: Presentation, Best Practices and demonstration (including Spyre cards)
- Empowering Power11 with Open Databases and watsonx.data.
- IBM P11 security, Quantum Safe, Power with CyberVault bundle

Day2 : Hands-On workshop

- Configuring a Power Virtual Server Environment
- Build an AI environment on Power
- Heterogeneous Databases Integration to watsonx.data

Date:
Wednesday 3rd December 12pm – 6pm
Thursday 4th December 9am – 2pm

Location:
IBM Montpellier
P.I.T La Pompi gnane,
rue de la Vieille Poste
34006 Montpellier, France

Language: English

Audience:
Clients & Business Partners

Event is free of charge
Travel and lodging to be covered by participants.
Limited Seating

Registration Link and QR Code:
<https://www.ibm.com/events/regflow/ibm/HK514CMB/landing/page/landing>



Agenda

- Open Source & IBM i
- Mission : intégration
- Design Patterns et exemples
- Liens utiles et contacts



Power Week

18 -19 - 20 novembre
2025

IBM
common
FRANCE

IBM

1. Open Source & IBM i

Open Source et IBM i

Ouverture de l'IBM i , des compétences

Pour les **administrateurs** système, Pour les **développeurs** (outils, portage),
standardisation.

Intégration Entreprise

Passerelle vers IA, IoT, Data, Web, Quantique...

Open Source et IBM i

Beaucoup de technologies Open Source pour IBM i et sur IBM i

Favorise l'intégration de l'IBM i dans l'architecture d'entreprise à l'ère du DevOps, du Cloud, de l'IA.

Technologies co-développées par la communauté, par **IBM** ou par **Red Hat**.

Support **IBM** (TSS Support) ou **Red Hat** (Streams, Kafka, Kafka Connect) , selon déploiement cloud (containers) ou sur IBM i.

Open Source sur IBM i

Bases de données

PostgreSQL, MariaDB, SQLite

Langages

PHP, Node.js, Python, R, Perl...

Développement

vscode for i , Node-RED, GCC, Driver ODBC, Mapepire

DevOps

Git, Ant, Maven

Intégration/Echange de données

ActiveMQ, Kafka, Manzan, Debezium

Utilitaires

Ansible, cURL, jq, rsync, bash, chroot, cloud-init, updatedb, locate, man, 7zip, sed, nano, vim

Serveur HTTP

Nginx

IA/Machine Learning/Real time scoring

Librairies Python: scikit-learn, matplotlib, OpenBLAS (MMA) + NEW : Serveur MCP

Gestion du système/logs

Prometheus/Grafana

NEW IBM i Extensions for VS Code Enterprise Support by IBM

<https://www.ibm.com/docs/en/announcements/i-extensions-vs-code-enterprise-support?region=US>

Open Source et IBM i – Documentation

[View on GitHub](#) 

ibmi-oss-resources

Important resources for anyone interested in open source on IBM i

IBM i Open Source Resources

Important resources for anyone interested in open source on IBM i

Documentation - General

- [IBM i Open Source documentation from IBM](#)
- [Getting Started with IBM i RPMs](#)

Documentation - Specific

- [Node.js toolkit](#)
- [Node.js odbc module](#)
- [Node.js idb-connector module](#)
- [Node.js idb-pconnector module](#)
- [Loopback \(Node.js API framework\) connector](#)
- [Python toolkit](#)

- [Python SQLAlchemy Adapter](#)
- [XMLService](#)
- [ODBC driver](#)
- [PostgreSQL](#)
- [PHP options for IBM i](#)
- [Ruby on IBM i](#)
- [IBM i chroot containers](#)
- [Self-signed TLS certificate validation in IBM i OSS](#)

Support

- [Open Source Support for IBM i](#)

Examples

- [IBM i Open Source Examples](#)

Community

- [IBM i Open Source Chat on Ryver](#) (must first join at [this link](#))
- [IBM i Community Slack](#)
- [IBM Community pages for IBM i](#)
- [IBMiOSS LinkedIn group](#)
- [#IBMiOSS hashtag on Twitter](#)

Blogs

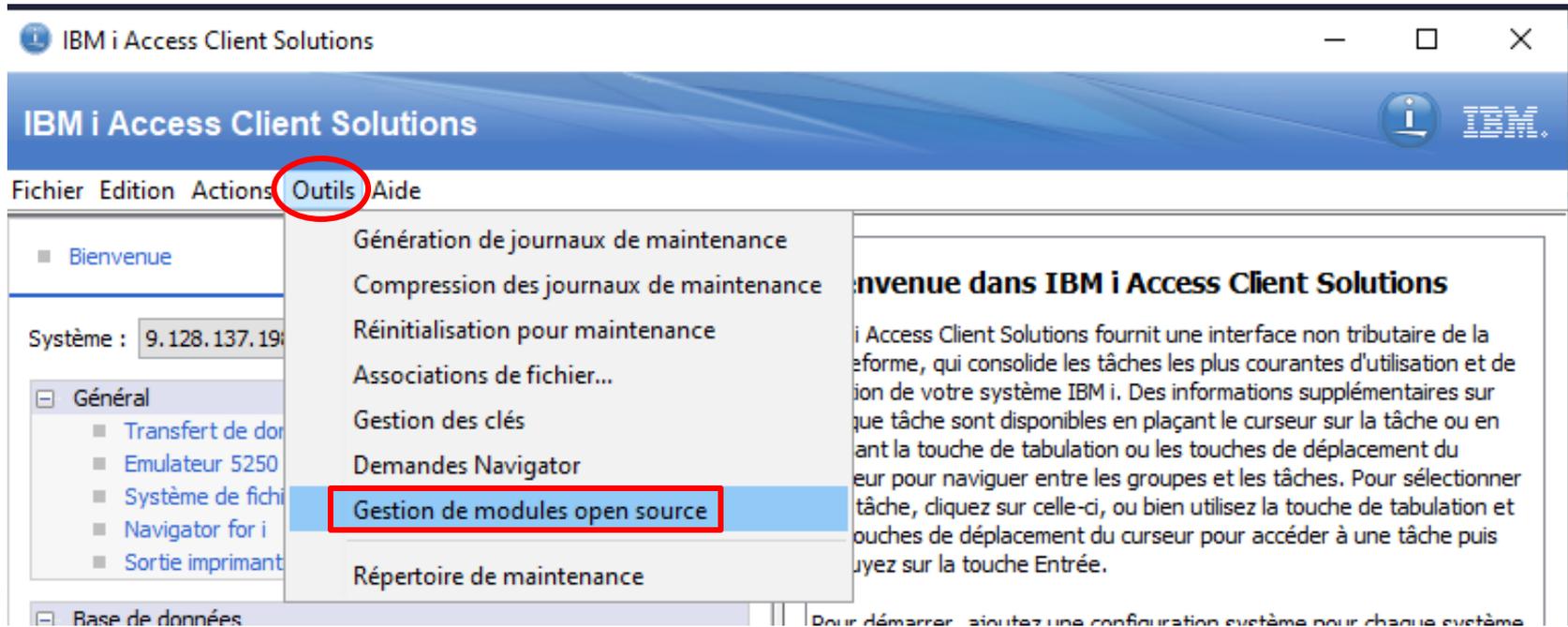
- [Kevin Adler's Blog](#)
- [Seiden Group Blog](#)
- [FAQ400 Blog](#)
- [Anand Khakale's Technical Musings](#)

Videos

- [JORI + IBM i: Designing digital transformation](#)
- [FormaServ video library](#)
- [The Bearded Geek on IBM i - Youtube Channel](#)

<https://ibmi-oss-docs.readthedocs.io/en/latest/README.html>

Open Source et IBM i – *Installation*



Open Source et IBM i – *Support*

- Support compris dans la **SWMA IBM i** :
 - L'installation des packages par YUM ou par ACS
 - Les produits sous licence (5733-DG1 et 5733-SC1)
- Pour les packages **RPM**, deux possibilités :
 - 1. Support **communautaire**, gratuit
 - Suivi d'incidents : <https://github.com/IBM/ibmi-oss-issues/>
 - Chat : <http://ibm.biz/ibmiOSS-chat> (inscription : <http://ibm.biz/ibmiOSS-chat-join>)
 - 2. Support **IBM TSS** (Technology Support Services), facturable
 - <https://www.ibm.com/support/pages/open-source-support-ibm-i>

Power Week

18 -19 - 20 novembre
2025

IBM
common
FRANCE

IBM

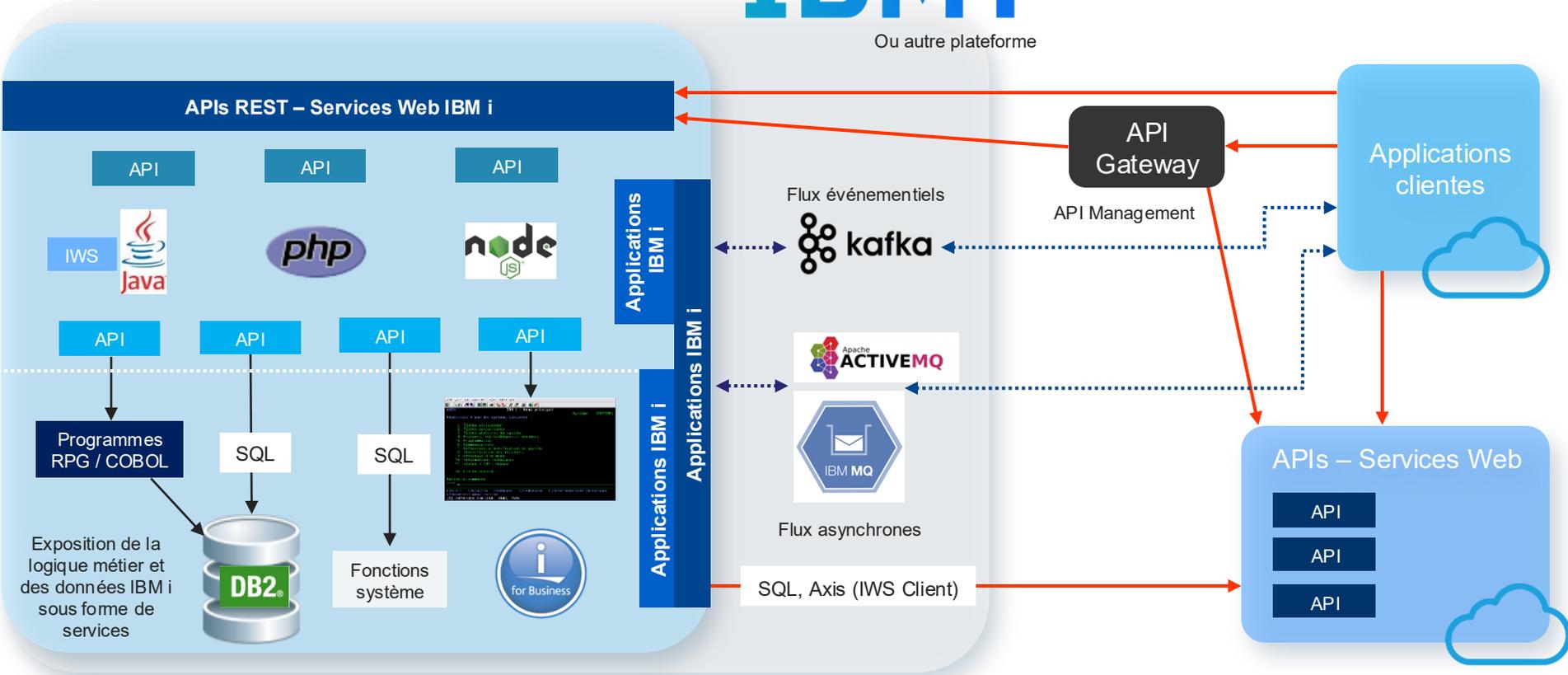
2. IBM i : mission intégration



Intégration IBM i

IBM i

Ou autre plateforme



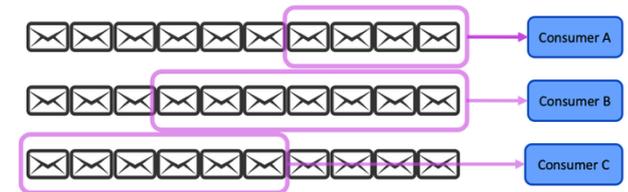
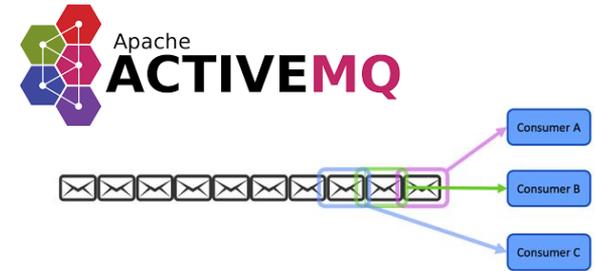
Open source et Intégration IBM i

- Apache **Kafka**: la super Data Queue cross-plateforme
- **ActiveMQ**: le collecteur MQTT
- Apache **Camel**: le couteau suisse multi-protocole
- **Manzan**: le super **Camel** , à la sauce IBM i
- **Node-RED** : l'outil low-code pour l'intégration et les tableaux de bord
- **Debezium** : Change Data Capture avec Db2 for i

MQTT & Kafka sur IBM i

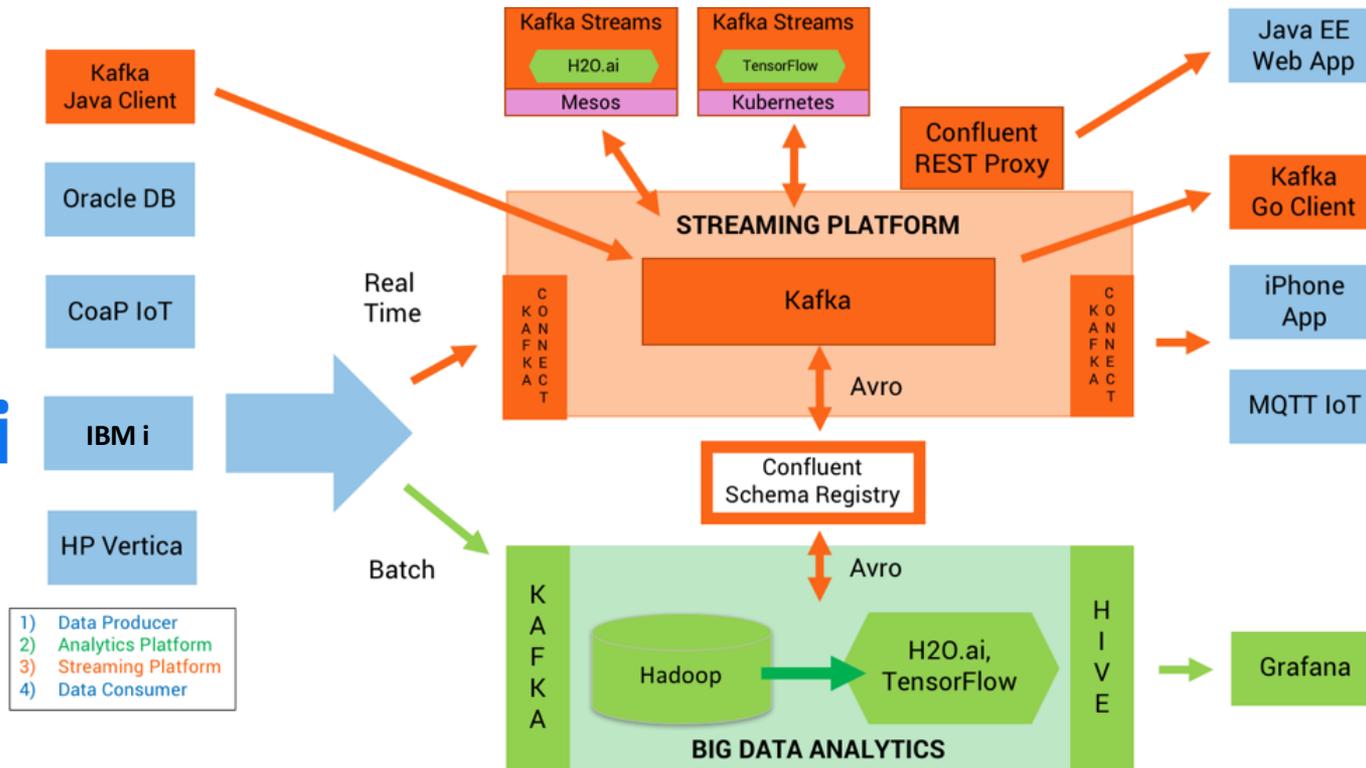
Disponibles sur IBM i, gratuit, Open Source sous licence Apache 2.0

- Apache **ActiveMQ** : implémentation JMS
 - Fédération de systèmes & Intégration asynchrone
 - Message Broker supportant divers protocoles : JMS, AMQP, MQTT, etc.
- Apache **Kafka** : high-throughput streaming event engine
 - Conçu par LinkedIn, Scalable++, pour traiter des grands volumes de données
 - S'intègre facilement avec Camel...
 - Streaming = on traite **plusieurs** lots/ séries de données à la fois



Kafka dans l'entreprise

IBM i



Source: <https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/>

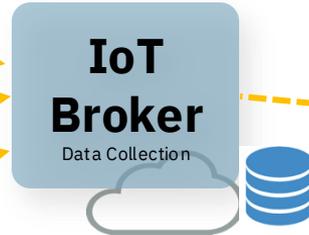
MQTT dans l'entreprise

Industrial Control Systems, SCADA*
 *SCADA : Supervisory Control And Data Acquisition



Legacy Industrial Devices

Edge/ Fog Computing (Machine Learning, etc)



← **SUBSCRIBE**



IoT Data (raw data: "data lake")

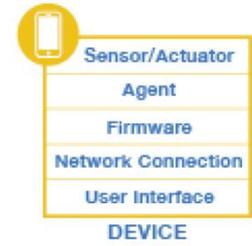


Ex: NoSQL Databases,
 Time-series data...

Secure Protocols for data (MQTT)
 Infrastructure & Physical layer (Wireless, wired...)

PUBLISH →

Private or Public/Dedicated Cloud

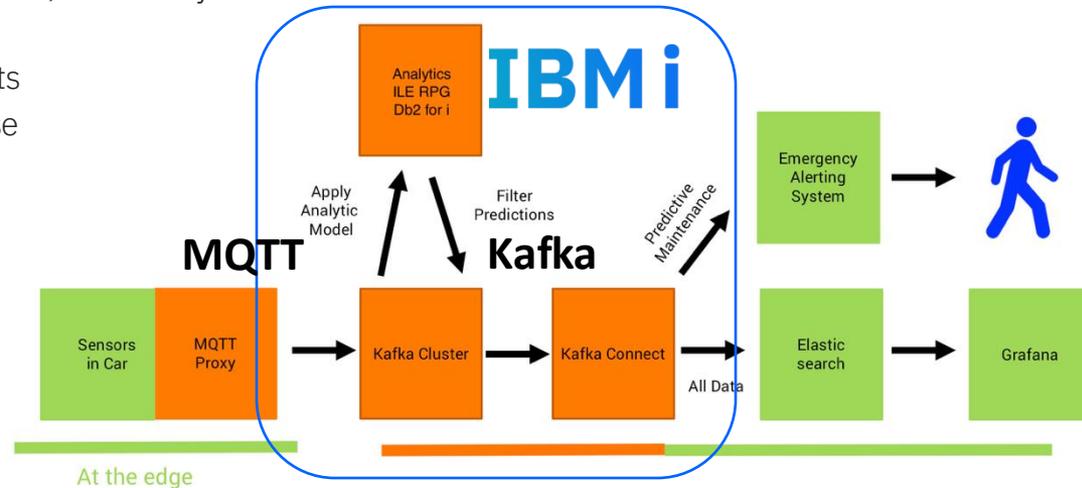


IoT Devices
 Sensors, Cameras, Web data, Social Media, Weather, GPS... embedded intelligence "Edge Computing"



MQTT ou Kafka ?

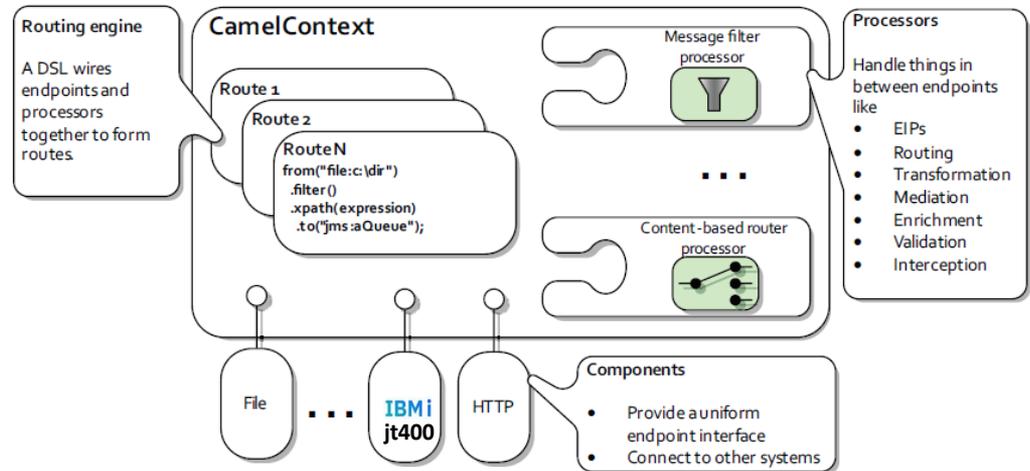
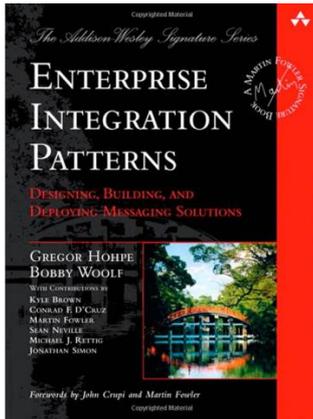
- Architecture de type Publish/Subscribe (pubsub)
- MQTT : principalement responsable de la transmission des données provenant des objets et applications connectés
 - Léger, conçu pour connexions faibles et latences élevées (ex: mobiles)
- Kafka : dédié au stockage et à la lecture d'évènements. Une fois les données publiées, Kafka peut les traiter, les analyser et les stocker pour une utilisation ultérieure.
 - long terme, retraitement des événements
 - Bonne intégration au reste de l'entreprise
 - Nécessite un réseau stable
 - Use case: Event-driven architecture, Change Data Capture Db2 for i etc.



Apache Camel

Couteau Suisse de l'intégration, disponible sur IBM i

- Projet Open Source Apache fondé sur les "Enterprise Integration Patterns" (EIP)
- Centré autour de 60+ patterns que l'on rencontre dans des projets "Enterprise integration"
- Fournit un langage afin d'implémenter ces patterns (style UNIX pipeline)



IBM i et Apache Camel

- Documentation : Apache Camel Component
<https://camel.apache.org/components/latest/jt400-component.html>

To send or receive data from a data queue

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.DTAQ[?options]
```

To send or receive messages from a message queue



```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.MSGQ[?options]
```

To call remote program

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/program.PGM[?options]
```

You can append query options to the URI in the following format, `?option=value&option=value&...`

PATH PARAMETERS (5 PARAMETERS):

Name	Description
userID	Required Returns the ID of the IBM i user.
password	Required Returns the password of the IBM i user.
systemName	Required Returns the name of the IBM i system.

QUERY PARAMETERS (33 PARAMETERS):

Name	Description
ccsid (common)	Sets the CCSID to use for the connection with the IBM i system.
format (common)	Sets the data format for sending messages. There are 2 enums a binary
guiAvailable (common)	Sets whether IBM i prompting is enabled in the environment run
keyed (common)	Whether to use keyed or non-keyed data queues.
searchKey (common)	Search key for keyed data queues.

DB2 et Apache Camel

- Les fonctions SQL de publication JSON renvoient les données d'une façon compréhensible pour les consommateurs Kafka/ActiveMQ

```
SELECT JSON_OBJECT(  
    KEY 'Department' VALUE  
    JSON_ARRAYAGG(JSON_OBJECT(  
        KEY 'Id' VALUE X.DEPTNO,  
        KEY 'Name' VALUE X.DEPTNAME)))  
    AS DEPT_JSON  
FROM TOYSTORE.DEPT X;
```

```
call qsys2.send_data_queue_utf8(  
    message_data      => scottf.dq_json,  
    data_queue        => 'HANDOFF_DQ',  
    data_queue_library => 'BANKONOSS');
```

- Les fonctions SQL de gestion des Data Queue permettent une intégration avec les files d'attente (ainsi qu'Apache Camel) directement depuis la base de données

Manzan

<https://theprez.github.io/Manzan>



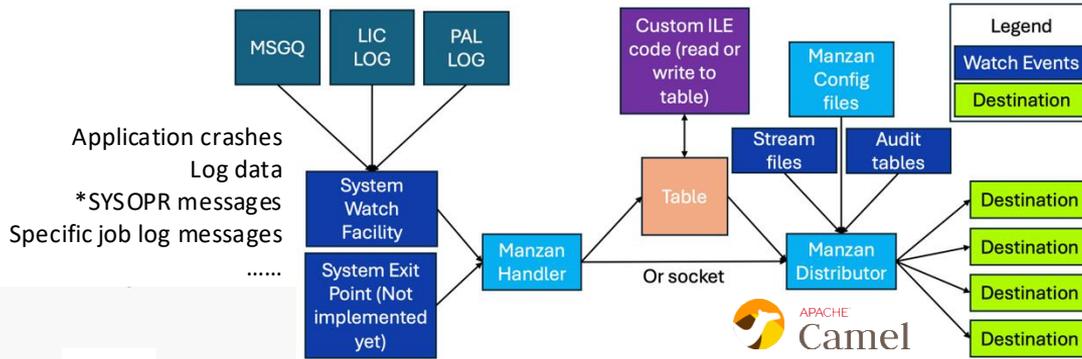
Jesse Gorzinski
ThePrez

AI/Open Source Architect
<https://github.com/theprez>

100% Open Source, basé sur le système de surveillance (STRWCH) , les points d'exit et Apache Camel

Filtre et formate les données d'événements

“Conçu pour simplifier les opérations et la surveillance du système IBM i”



data.ini

```
[watchout]
type=watch
id=jesse
destinations=slackme
format=$MESSAGE_ID$ (severity $SEVERITY$): $MESSAGES$
strwch=WCHMSGQ(*ALL) WCHMSGQ(*HSTLOG)
```

dests.ini

```
[slackme]
type=slack
channel=open-source-system-status
webhook=https://hooks.slack.com/services/TA3EF58G4...
```

- [ActiveMQ](#) 🕒
- [AWS Simple Email Service \(SES\)](#) 🕒
- [AWS Simple Notification System \(SNS\)](#) 🕒
- [ElasticSearch](#) 🕒
- Email (SMTP) ✓
- [FluentD](#) ✓
- [Google Drive](#) 🕒
- [Google Mail \(gmail\)](#) 🕒
- [Google Pub/Sub](#) 🕒
- [Grafana Loki](#) 🕒
- HTTP endpoints (REST, etc) ✓
- HTTPS endpoints (REST, etc) 🕒
- [Internet of Things \(mqtt\)](#) 🕒
- [Kafka](#) ✓
- [Mezmo](#) 🕒
- [Sentry](#) 🌐
- [Slack](#) ✓
- SMS (via [Twilio](#)) ✓
- [Splunk](#) 🕒

open-source-system-status - Dec 20th, 2022



ibm i system monitor APP 9:01 PM

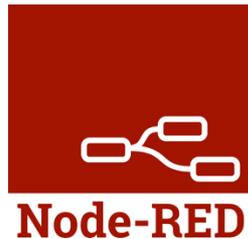
CPIAD09 (severity 0): User LINUX from client 9.163.40.192 connected to job 492988/QUSER/QZDA subsystem QUSRWRK in QSYS on 12/20/22 20:17:06.



Node-RED

<https://nodered.org>

- Node-RED – outil gratuit et open-source développé par IBM initialement fait pour l'internet des objets (IoT)
- Permet de coder graphiquement et rapidement, et de connecter matériels, API, applications ensemble
- Editeur Web - flow editor – contient un grand nombre de nodes dans sa palette évolutive : on trouve tous types de nodes dans un repository en ligne
- Basé sur JavaScript (runtime Nodejs)
- Node-RED sur i:
 - Tableaux de bord techniques ou métiers
 - APIisation
 - Intégration & Protocole MQTT



Node-RED & IBM i

<https://flows.nodered.org/node/node-red-contrib-db2-for-i>

- Noeud [Node-RED](#) pour lire et écrire dans une base de données DB2 for i
- Projet Open Source sur [GitHub](#) créé en juillet 2017
- Téléchargé [+ de 10 000 fois](#)
 - `npm install node-red-contrib-db2-for-i`
- DB2 for i native driver : [idb-connector](#)
 - Plan* : odbc mode (portability) , iToolkit node ...
- Utilisé par tous types de solutions (ERP, CRM,...) et développements en Dev/Test et Production



Power Week

18 -19 - 20 novembre
2025

IBM
common
FRANCE

IBM

3. Design Patterns, Exemples

Cloud design patterns

Solution générique et réutilisable à un problème courant dans un contexte donné d'architecture logicielle.

Il ne s'agit pas d'une conception finalisée pouvant être directement transformée en code.

Les modèles de conception cloud sont des bonnes pratiques formalisées qui ont fait leurs preuves dans le domaine du cloud computing & du développement logiciel

Différentes catégories de patterns:

- Security: Identity & access management,...
- **Event and Data Management**
- Cloud Native application development
- Performance and scalability
- Resilience and fault tolerance

Cloud design patterns & IBM i

Focus sur les « Event Driven & Data Management » Patterns

L'IBM i est un moteur transactionnel, une source d'évènements dans l'entreprise.

Comparée à une architecture applicative de type requête/réponse, l'architecture orientée événements (EDA) offre plusieurs avantages et opportunités pour les développeurs et les organisations:

Réactivité et analyse en temps réel

Applications capables de réagir immédiatement aux changements et de prédire grâce aux données actuelles et historiques (ère de l'IA)

Tolérance aux pannes et scalabilité

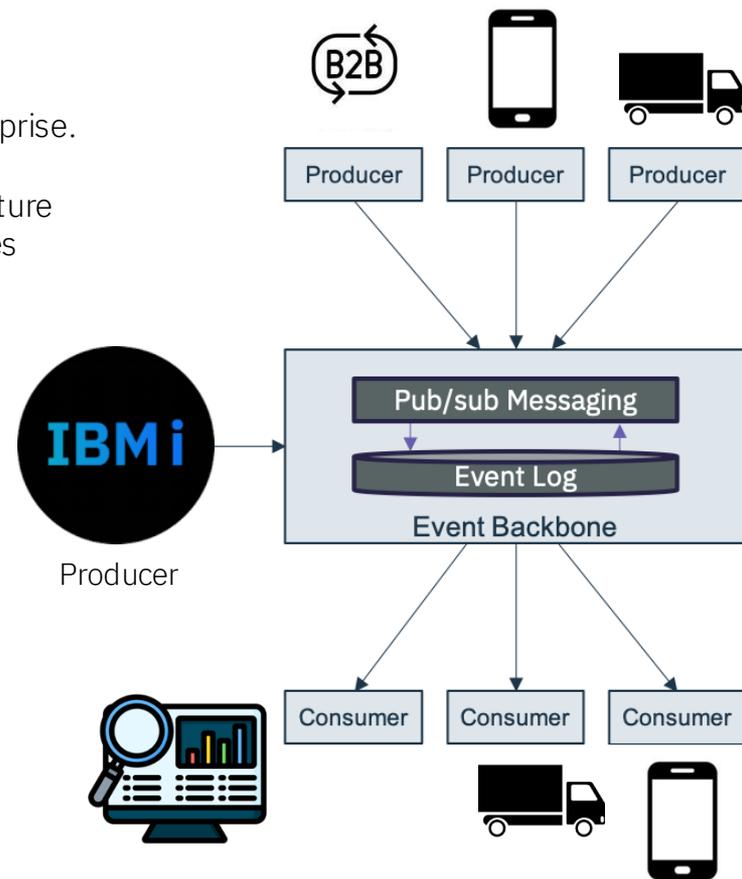
Composants indépendants, mise à jour et déploiement sans interruption, reprise possible via la persistance des événements.

Maintenance simplifiée et faible couplage

Moins de dépendances entre composants, flexibilité accrue.

Messages asynchrones

Producteurs publient des événements sans attendre la réception par les consommateurs



Cloud design patterns & IBM i

Focus sur les « Event Driven & Data Management » Patterns

[Decompose by subdomain](#): L'approche domain-driven est utile pour identifier et classer les fonctions métier et les microservices correspondants qui leur seraient associés.

[Database per service](#): Chaque service stocke des données de manière privée et n'est accessible que via son API. Les services sont faiblement couplés, ce qui limite l'impact des modifications de schéma dans la base de données sur les autres services.

[Strangler pattern](#): Utilisé pour migrer progressivement une application monolithique existante en remplaçant un ensemble de fonctionnalités par un microservice, tout en maintenant les deux fonctionnant en parallèle.

[Event sourcing](#): Conserver, dans un journal d'ajout, les états d'un business entity, telle qu'une commande, sous forme d'une séquence d'événements immuables modifiant l'état

[Change Data Capture](#): permet de surveiller les modifications apportées à une base de données et de les propager aux systèmes en aval. C'est un moyen efficace de permettre une intégration fiable des microservices et de résoudre des problèmes courants, tels que l'extraction progressive de microservices à partir d'applications monolithiques existantes.

[Command Query Responsibility Segregation](#): permet de séparer les requêtes (read) des commandes (write)

Reference: <https://martinfowler.com/articles/201701-event-driven.html>

Exemple 1 - Event Sourcing pour de l'analytique (BI) et du prédictif (IA)

Chaque event (transaction DB2) déclenche l'envoi de données dans un composant externe pour un scoring

Db2 Trigger → Data Queue → Job Camel+Kafka → AI Scoring

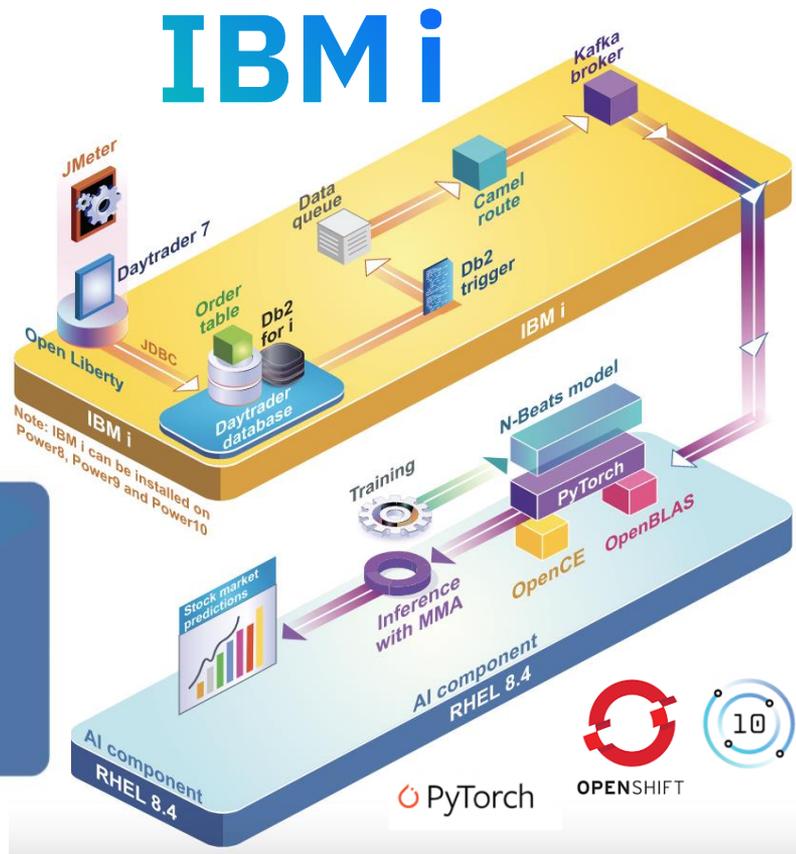
```
-- description: Create ORDER trigger
CREATE OR REPLACE TRIGGER OCPDT.ORDEREJB_TRIGGER
AFTER INSERT OR DELETE OR UPDATE ON OCPDT.ORDEREJB
...
CALL QSYS2.SEND_DATA_QUEUE_UTF8 (
MESSAGE_DATA => OCPDT.DQ_ORDER ,
DATA_QUEUE => 'AI' ,
DATA_QUEUE_LIBRARY => 'OCPDT'
) ;
END ;
```

- The DayTrader 7 as a Liberty Application runs on IBM i and creates trade data
- Trade data is stored in Db2 for IBM i
- JMeter runs on IBM i to simulate trades in DayTrader 7 and generates large trades data
- Kafka produces the real-time data stream on IBM i
- PyTorch consumes the Kafka stream as the input of the AI component
- The AI component leverages Power10 MMA to make sequence-based predictions (For example, stock price prediction)

IBM Developer

<https://developer.ibm.com/tutorials/power10-business-inferencing-at-scale-with-mma/>

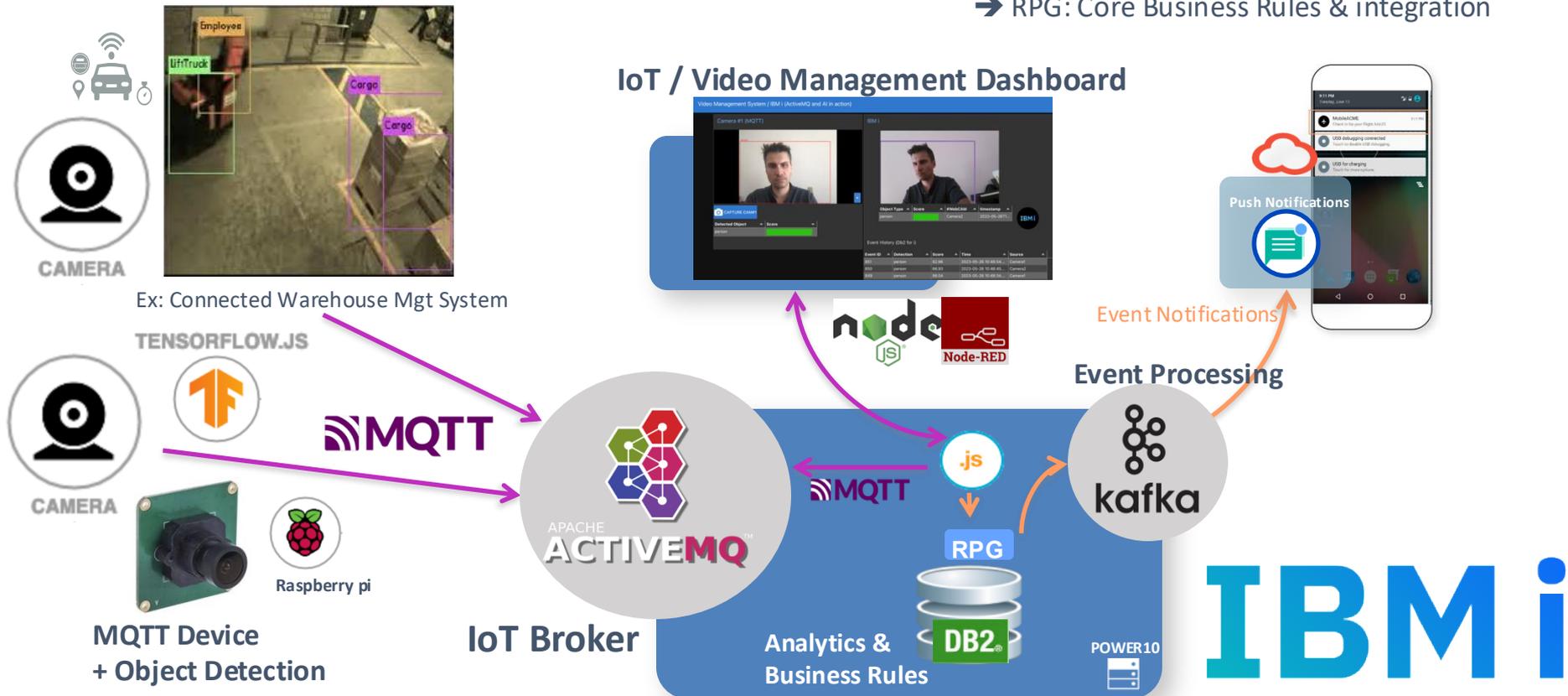
<https://github.com/ThePrez/Kafka-DayTrader-AI-example/>



Exemple 2 : Event Streaming et IBM i

IBM i in action

- IoT & Event Processing & dashboard
- Db2 for i: NoSQL/JSON & Geospatial Analytics
- RPG: Core Business Rules & integration



IBM i

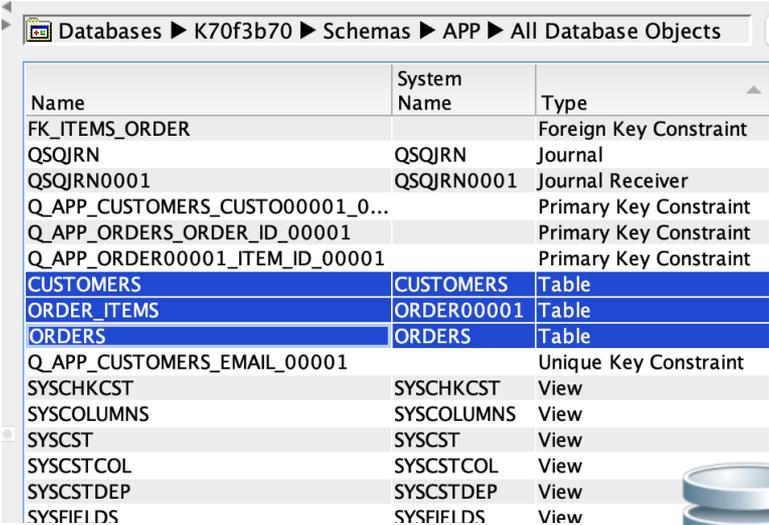
Exemple 3 : Réplication de données via Debezium CDC

Scénario

Propagation d'évènements relative aux commandes et factures

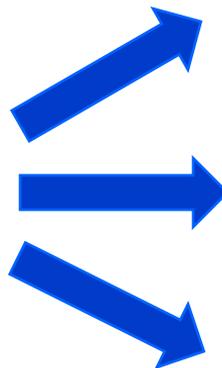
- Pour un besoin de reporting / BI déporté sur un autre système
- Afin d'utiliser ces données sur d'autres applications métier (micro-service, mobile etc.)

Ex: communiquer les informations de facturation à un une Plateforme de Dématérialisation Partenaire (PDP) pour gérer les factures électroniques.



Databases ► K70f3b70 ► Schemas ► APP ► All Database Objects

Name	System Name	Type
FK_ITEMS_ORDER		Foreign Key Constraint
QSQRN	QSQRN	Journal
QSQRN0001	QSQRN0001	Journal Receiver
Q_APP_CUSTOMERS_CUSTO00001_0...		Primary Key Constraint
Q_APP_ORDERS_ORDER_ID_00001		Primary Key Constraint
Q_APP_ORDER00001_ITEM_ID_00001		Primary Key Constraint
CUSTOMERS	CUSTOMERS	Table
ORDER_ITEMS	ORDER00001	Table
ORDERS	ORDERS	Table
Q_APP_CUSTOMERS_EMAIL_00001		Unique Key Constraint
SYCHKCST	SYCHKCST	View
SYSCOLUMNNS	SYSCOLUMNNS	View
SYSCST	SYSCST	View
SYSCSTCOL	SYSCSTCOL	View
SYSCSTDEP	SYSCSTDEP	View
SYSFIFIDS	SYSFIFIDS	View

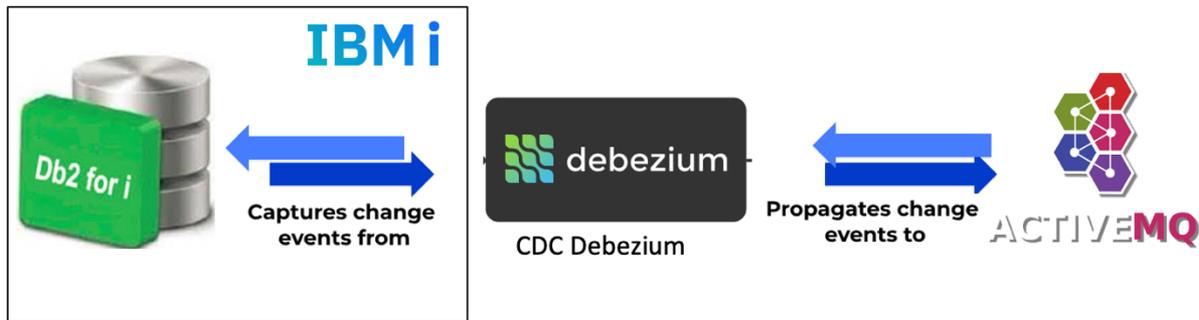


Exemple 3 : Réplication de données via Debezium CDC

<https://github.com/debezium/debezium-connector-ibmi>

La solution CDC permet de capturer et de diffuser en temps réel les événements de changement sur DB2 for i, et de publier ces événements — c'est-à-dire les enregistrements nouveaux, supprimés ou modifiés — vers un autre consommateur, une base de données, ou un autre système.

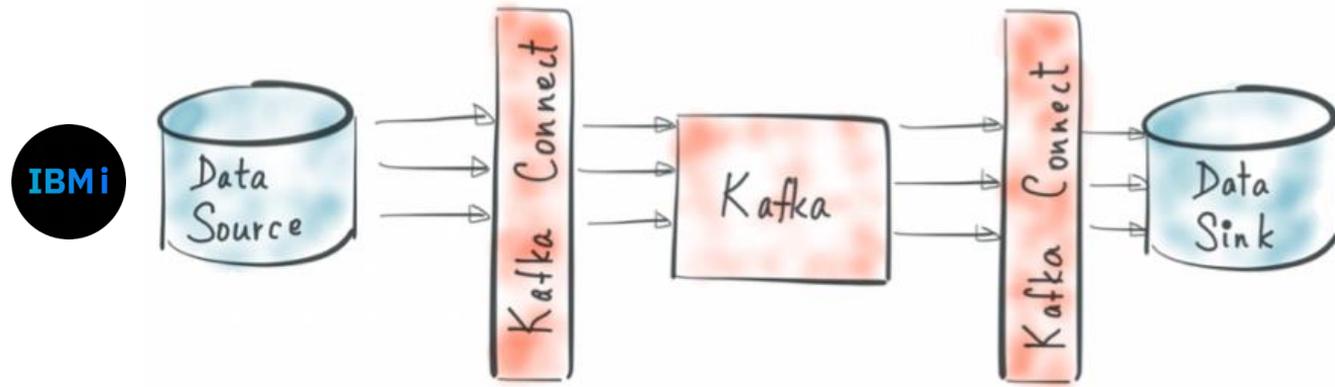
Les événements sont publiés sur un topic MQTT, afin que toutes les applications autorisées puissent consommer ces données.



Code et explications pour démarrer rapidement :

<https://github.com/bmarolleau/debezium-ibmi-demo>

Exemple 3 : Réplication de données via Debezium CDC

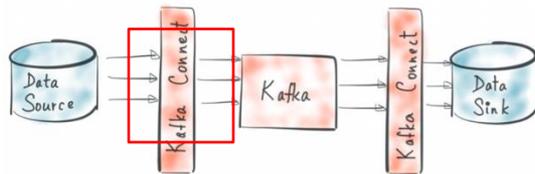


Dans la démonstration, tout s'exécute sur sur podman/docker, excepté IBM i / Db2 for i ,mais tout peut aussi tourner sur IBM i.

Composant	Rôle	Langage	Exemple
Apache Kafka	Le broker de messages : stocke et distribue les événements.	Java	kafka-server-start.sh  
Kafka Connect	Le framework d'intégration : exécute des "connecteurs" source/sink.	Java	Standalone ou Distributed (cluster)  
Debezium	Une collection de connecteurs Kafka Connect spécialisés dans la CDC (Change Data Capture) . Dépendance: jt400.jar	Java (plug-in de Connect)	io.debezium.connector.db2i.Db2iConnector
Base source	Système de données émettant les changements (ex: DB2 for i).	—	Journaux DB2 
Sink	Cible recevant les données (Kafka topics, JDBC sink, Elasticsearch, MQTT.)	—	JDBC, Postgres, ActiveMQ (MQTT) etc. 

Exemple 3 : Réplication de données via Debezium CDC

➔ Configuration du connecteur Debezium pour Db2 for i



Création d'un connecteur par schema (lib) ou table (ici sur 3 tables):

```
curl -X POST -H "Content-Type: application/json" \
  --data @db2i-connector.json \
  http://myKafkaConnect:8083/connectors
```

Kafka Connect Debezium UI:

Name	Status	Tasks
db2i-orders-cdc	RUNNING	1 RUNNING: 1

Overview

Message/sec: 0
Max lag in last min: 0
Percentiles: 0

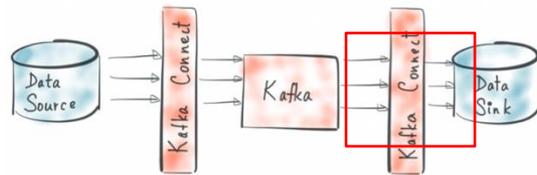
Task Id: 0
Status: RUNNING

db2i-connector.json

```
{"name": "db2i-orders-cdc",
"config": {
"connector.class": "io.debezium.connector.db2.As400As400RpcConnector",
"transforms.unwrap.delete.handling.mode": "rewrite",
"transforms.unwrap.drop.tombstones": "false",
"transforms.unwrap.type": "io.debezium.transforms.ExtractNewRecordState",
"snapshot.mode": "initial",
"snapshot.max.threads": 4,
"database.hostname": "10.3.61.2",
"database.dbname": "K70F3B70",
"database.user": "BENOIT",
"database.password": "xxxxx",
"database.driver.secure": "false",
"database.schema": "APP",
"table.include.list": "APP.CUSTOMERS,APP.ORDERS,APP.ORDER_ITEMS",
"topic.prefix": "db2i",
"decimal.handling.mode": "double"}}
```

Exemple 3 : Réplication de données via Debezium CDC

➔ Configuration des connecteurs MQTT (Sink)



Création d'un connecteur par topic Kafka (=par table):

```
curl -X POST -H "Content-Type: application/json" \
  --data @mqtt-customers-sink.json \
  http://mykafkaConnect:8083/connectors
```

```
curl -X POST -H "Content-Type: application/json" \
  --data @mqtt-orders-sink.json \
  http://mykafkaConnect:8083/connectors
```

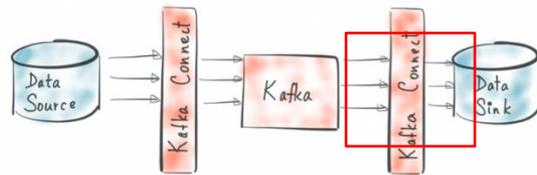
```
curl -X POST -H "Content-Type: application/json" \
  --data @mqtt-order-items-sink.json \
  http://mykafkaConnect:8083/connectors
```

mqtt-customers-sink.json

```
{
  "name": "mqtt-orders-sink",
  "config": {
    "connector.class": "io.lenses.streamreactor.connect.mqtt.sink.MqttSinkConnector",
    "tasks.max": "1",
    "topics": "db2i.APP.ORDERS",
    "connect.mqtt.hosts": "tcp://activemq:1883",
    "connect.mqtt.kcql": "INSERT INTO mqtt/orders SELECT * FROM db2i.APP.ORDERS WITHTYPE JSON",
    "connect.mqtt.username": "admin",
    "connect.mqtt.password": "xxxx",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
    "value.converter.schemas.enable": "false",
    "key.converter": "org.apache.kafka.connect.storage.StringConverter",
    "connect.mqtt.service.quality": 1 }}}
```

Exemple 3 : Réplication de données via Debezium CDC

➔ Configuration du Sink jdbc pour DB2 for i



Création d'un connecteur par topic Kafka (=par table):

```
curl -X POST -H "Content-Type: application/json" \
  --data @jdbc-sink-db2i-orders-confluent.json \
  http://mykafkaConnect:8083/connectors
```

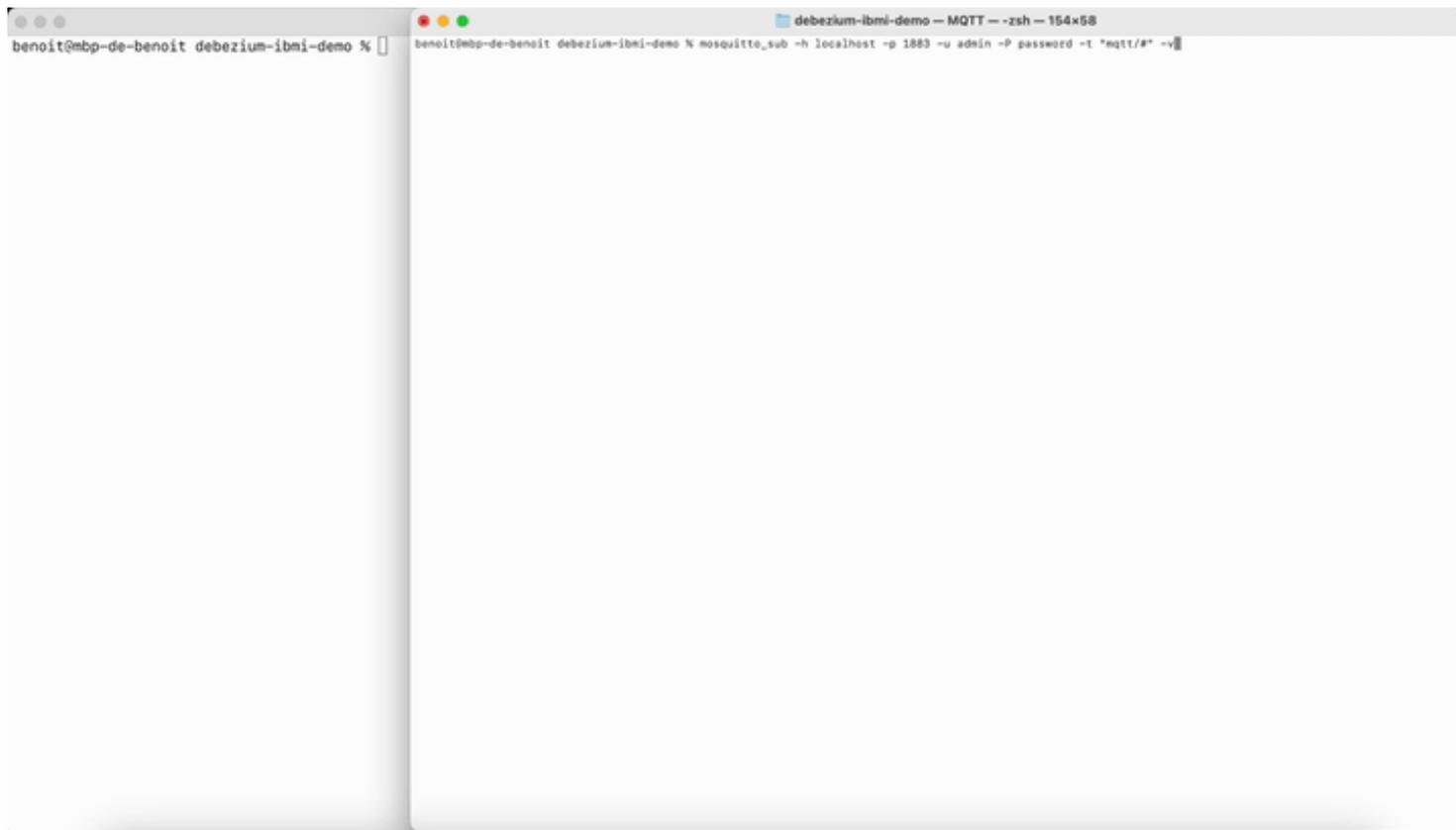
jdbc-sink-db2i-orders-confluent.json

```
CREATE SCHEMA APP1;
CREATE TABLE APP1.ORDSINK (
  ORDER_ID INTEGER,
  CUSTOMER_ID FOR COLUMN CUST000001 INTEGER NOT NULL ,
  STATUS VARCHAR(20) CCSID 37 NOT NULL ,
  TOTAL_AMOUNT FOR COLUMN TOTAL00001 DECIMAL(12, 2) NOT NULL)
RCDFMT ORDERS ;
GRANT ALTER , DELETE , INDEX , INSERT , REFERENCES , SELECT , UPDATE
ON APP1.ORDSINK TO DBZUSER WITH GRANT OPTION ;
```

```
{
  "name": "jdbc-sink-db2i-orders",
  "config": {
    "connector.class": "io.confluent.connect.jdbc.JdbcSinkConnector",
    "topics": "db2iAPP.ORDERS",
    "connection.url": "jdbc:as400://10.3.61.2/K70F3B70;user=benoit;password=xxx;libraries=APP1;prompt=false;",
    "driver.class": "com.ibm.as400.access.AS400JDBCDriver",
    "insert.mode": "insert",
    "delete.enabled": "true",
    "primary.key.mode": "record_key",
    "schema.evolution": "basic",
    "auto.create": "true", "batch.size": "300",
    "retries": "5", "max.retries": "10", "retry.backoff.ms": "3000",
    "tasks.max": "1",
    "table.name.format": "APP1.ORDSINK", "schema.pattern": "APP1",
    "pk.mode": "record_key", "pk.fields": "ORDER_ID",
    "dialect.name": "Db2DatabaseDialect" }}}
```

Exemple 3 : Réplication de données via Debezium CDC

➔ Démonstration live



```
benoit@mbp-de-benoit debezium-ibmi-demo %  
benoit@mbp-de-benoit debezium-ibmi-demo % mosquitto_sub -h localhost -p 1883 -u admin -P password -t "mqtt/#" -v
```

Exemple 3 : Réplication de données via Debezium CDC

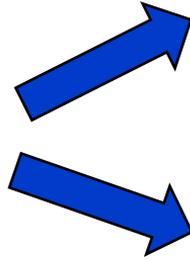
→ Démonstration live

Résultat:

Propagation d'évènements relative aux commandes et factures

```
✓ Connected to DB2 for i at 10.3.61.2
Updated item 1 → QTY=4

Updated order 29 → status=PENDING
Updated item 24 → QTY=2
Updated order 36 → status=SHIPPED
Updated order 34 → status=SHIPPED
Updated order 11 → status=PENDING
Updated order 25 → status=DELIVERED
Updated order 10 → status=CANCELLED
Updated order 34 → status=SHIPPED
Updated order 20 → status=PENDING
Updated item 8 → QTY=8
Updated order 31 → status=PENDING
Updated order 30 → status=PENDING
Updated order 31 → status=DELIVERED
Updated item 31 → QTY=3
Updated order 4 → status=DELIVERED
Updated item 28 → QTY=4
```



Event Bus (Kafka → MQTT) :

mqtt/orders

```
payload":{"STATUS":"CANCELLED","ORDER_ID":10,"CUSTOMER_ID":10,"TOTAL_AMOUNT":430.0}}
```

```
mqtt/orders payload":{"STATUS":"PENDING","ORDER_ID":20,"CUSTOMER_ID":20,"TOTAL_AMOUNT":75.75}}
```

```
mqtt/order_items payload":{"ORDER_ID":6,"QTY":8,"ITEM_ID":8,"UNIT_PRICE":90.1,"SKU":"SKU-2002"}}
```

```
mqtt/orders {"payload":{"STATUS":"PENDING","ORDER_ID":30,"CUSTOMER_ID":30,"TOTAL_AMOUNT":330.6}}
```

Remote Db2 for i (Kafka → Any db) :

ORDER_ID	CUSTOMER_ID	STATUS	TOTAL_AMOUNT
36	1	CANCELLED	120.50
33	1	PENDING	120.50
19	19	PENDING	590.00
2	2	PENDING	89.99
3	3	DELIVERED	250.00
5	5	CANCELLED	340.75
7	7	CANCELLED	560.00
9	9	DELIVERED	75.50
11	11	CANCELLED	99.90
12	12	DELIVERED	130.45
14	14	DELIVERED	200.20
15	15	CANCELLED	67.45
31	1	CANCELLED	120.50
2	2	SHIPPED	89.99
32	1	CANCELLED	120.50
32	1	SHIPPED	120.50

Debezium – Mode de snapshot

Mode	Signification	Comportement	Cas d'usage typique
initial	Capture complète initiale	Au premier démarrage du connecteur, Debezium lit toutes les lignes des tables surveillées, puis continue avec le flux CDC (changements du journal).	Déploiement initial quand il faut synchroniser toute la base existante avant de commencer à capter les changements.
initial_only	Capture initiale sans CDC	Effectue un snapshot complet des tables, puis s'arrête sans lire le journal .	Export ponctuel de données (type "dump"), sans besoin de suivi continu.
never	Aucune capture initiale	Ne fait pas de snapshot : Debezium commence uniquement avec les événements du journal à partir de la position courante.	Quand les données initiales ont déjà été chargées par ailleurs et que seules les modifications futures comptent.
when_needed	Snapshot si nécessaire	Debezium fait un snapshot uniquement si aucun offset précédent n'existe (ex : premier lancement), sinon il reprend le journal là où il s'était arrêté.	Redéploiement d'un connecteur existant : permet d'éviter les doublons.
incremental	Snapshot incrémental (par lots)	Capture les données d'une ou plusieurs tables par petits lots , en parallèle de la lecture CDC. Permet d'éviter les verrous et gros volumes instantanés. Nécessite une clé primaire dans la table!	Très utile pour bases volumineuses (DB2, Oracle, MySQL...) — mode progressif et sans impact.

NEW

Novembre 2025
avec le connecteur
DB2 for I 😊

Debezium – Snapshot incremental avec Db2 for i

<https://issues.redhat.com/browse/DBZ-7958>

Aspect	Snapshot classique	Snapshot incrémental
Blocage sur base	Oui (verrou de cohérence)	Non
Lecture en parallèle du CDC	Non	Oui
Relance partielle (par table)	Non	Oui
Impact sur performance	Fort	Faible et contrôlé
Contrôle via signaux	Non	Oui (debezium_signals)
Relance automatique sur incident	Non	Oui, grâce à l'offset de chunk

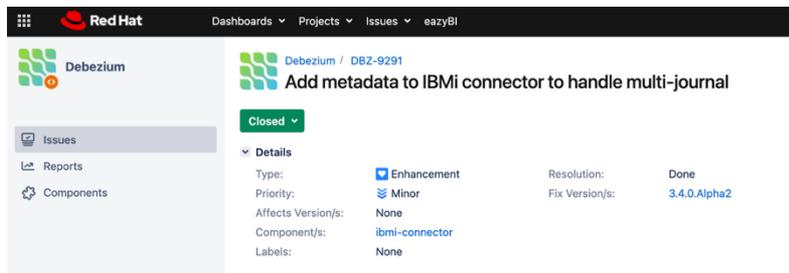
Champs additionnels dans la configuration du connecteur DB2 for i:

```
"incremental.snapshot.allow": "true",  
"incremental.snapshot.chunk.size": "2000"  
"signal.data.collection": "MYLIB.DEBEZIUM_SIGNAL",  
"signal.enabled.channels": "source"
```

Debezium 3.4 IBM i Connector: ameliorations/limitations actuelles

<https://debezium.io/releases/3.4/release-notes>

<https://repo1.maven.org/maven2/io/debezium/debezium-connector-ibmi/3.4.0.Alpha2/>



Red Hat Jira issue DBZ-9291: Add metadata to IBMi connector to handle multi-journal. Status: Closed. Type: Enhancement, Priority: Minor, Resolution: Done, Fix Version/s: 3.4.0.Alpha2. Component/s: ibmi-connector.



Red Hat Jira issue DBZ-7958: Support for incremental snapshots in IBMi connector. Status: Closed. Type: Enhancement, Priority: Major, Resolution: Done, Fix Version/s: 3.4.0.Alpha2. Component/s: ibmi-connector.

Limitations

- TODO integrate with exit program to prevent journal loss <https://github.com/jhc-systems/debezium-ibmi-exitpgm>
- Limited support for table changes - the journal entries for table changes are not documented so rely on fetching table structure at runtime and refetching when table change detected
- No support for remote journals and fail over
- No support for clobs/xml and similar large text/blobs

Power Week

18 -19 - 20 novembre
2025

IBM
common
FRANCE

IBM

4. Liens utiles, Contacts

Ressources

Debezium: Code et explications pour démarrer rapidement : <https://github.com/bmarolleau/debezium-ibmi-demo>

IBM i RPMs <http://ibm.biz/ibmi-rpms>

IBM i Open Source Support <http://ibm.biz/ibmi-oss-support>

AI & IBM i : <https://ibm.github.io/AI-IBMi-guide/>

ZULIP CHAT <https://chat.ibmioss.org/>

Jesse Gorzinski's blog. <http://ibm.biz/open-your-i>
<https://ibmsystemsmag.com/Power-Systems/06/2020/common-open-source-questions-answered>

Exemples Open Source <http://github.com/IBM/ibmi-oss-examples>

IBM i customer stories <http://ibm.biz/ibmistories>

Open Source , ILE , git

- <https://github.com/OSSILE>
- <https://github.com/richardschoen/iforgit>
- [GIT source control on IBM i](#) (presentation par Nathanael Bonnet, Common France)

Quel IDE choisir ? RDi mais également....[vscode - Code for IBM I](#)

Un besoin? Contactez-nous!!

- Votre partenaire IBM
- Votre Sales IBM
- IBM Client Engineering (moi)
- Les guru IBM sur l'Intégration



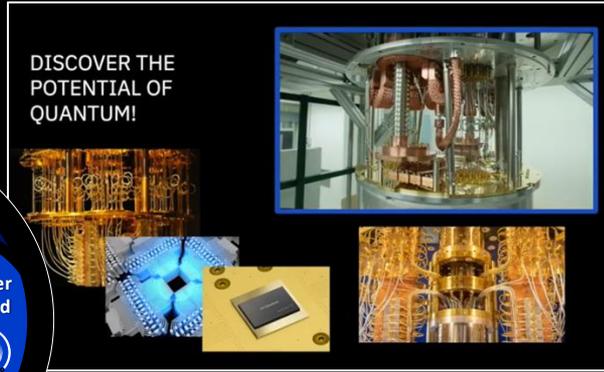
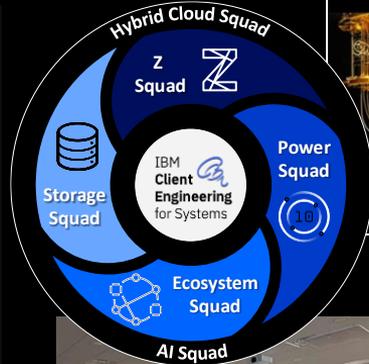
ARNAULD DESPRETS

Hybrid Cloud Integration Architect

arnauld_desprets@fr.ibm.com



IBM Client Engineering



<https://www.ibm.com/ibm/clientcenter/montpellier>
Client Support PoC, Workshops, Benchmarks...
Partners Workshops & Training
ISV , Integrators....



IBM Z

**AIX
IBM i**

IBM Blockchain

IBM Q

IBM Cloud



OPENSIFT



Power Week

18 -19 - 20 novembre
2025

IBM
common
FRANCE

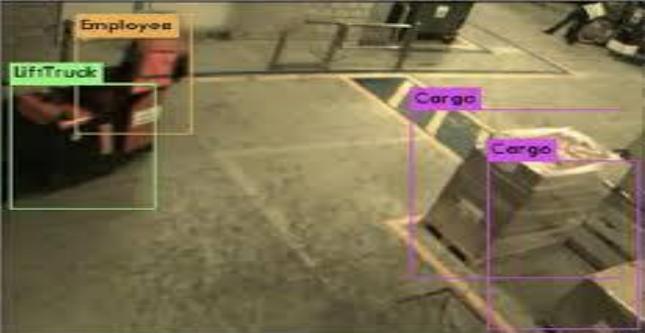
IBM

4. Références, Backup slides

IBM i Next Gen Apps: VMS demo

Video Management System / IBM i (ActiveMQ and AI in action)

Camera #1 (MQTT)



CAPTURE CAM#1

Detected Object	Score
person	[Green bar]

IBM i

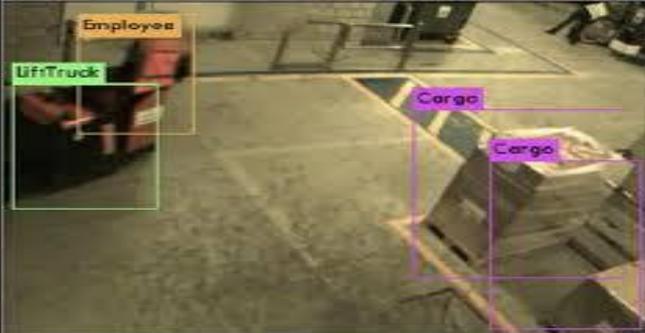


Image received on IBM i (MQTT)

Object Type	Score	#WebCAM	timestamp
person	[Green bar]	Camera1	2023-06-01T1...

IBM i

Aggregated data on IBM i (raw JSON & Relational)

Event ID	Detection	Score	Time	Source	Location	Distance (m)
236	person person	56.38	2023-06-01 18:13:...	Camera1	POINT (-92.503 44.0...	78
235	person	83.79	2023-06-01 18:13:...	Camera1	POINT (-92.503 44.0...	78
234	person	84.48	2023-06-01 18:13:...	Camera1	POINT (-92.503 44.0...	78



Source Video Camera#1
with embedded AI Model inference
(Tensorflow cocossd model)

Object Classes

Camera#

Geospatial
real time processing



Demo

Video Management System / IBM i (ActiveMQ and AI in action)

Camera #1 (MQTT)



CAPTURE CAM#1

Detected Object	Score
person	[Green bar]

IBM i - Event Detection Hub (ActiveMQ)



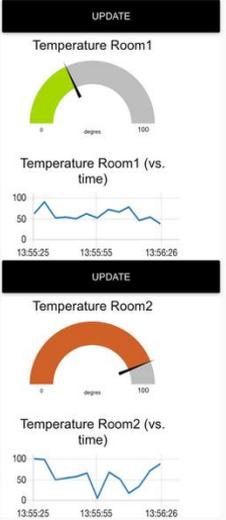
Object Type	Score	#WebCAM	timestamp
person	[Green bar]	Camera2	2023-11-06T12...
laptop	[Green bar]	Camera2	2023-11-06T12...

Event Records (Db2 for i)

Event ID	Detection	Score	Time	Source	Location	Distance (m)
on/laptop		53.17	2023-11-06 12:56:1...	Camera2	POINT (-92.503 44.01)	523
on/laptop#1...		51.58	2023-11-06 12:56:0...	Camera2	POINT (-92.503 44.01)	523
on		86.72	2023-11-06 13:55:5...	Camera1	POINT (-92.503 44.006)	78
on		84.35	2023-11-06 12:55:4...	Camera2	POINT (-92.503 44.01)	523
on		88.81	2023-11-06 12:55:3...	Camera1	POINT (-92.503 44.006)	78
on/laptop#1...		61.27	2023-11-06 12:55:3...	Camera2	POINT (-92.503 44.01)	523
on/person		79.66	2023-11-06 13:55:2...	Camera1	POINT (-92.503 44.006)	78

- CAPTURE CAM#1
- CAPTURE CAM#2
- RESTART CAM#2

IBM i - Sensor Data (ActiveMQ)



Dashboard source code (Node-Red)

Node-RED interface showing flows for event detection and image dashboard.

Dashboard rendering (HTML5 Javascript)

INTEGRATION & DASHBOARD WITH NODE-RED, ACTIVEMQ AND KAFKA

Demo

```
benoit@10.3.61.2's password:
***** IBM i 7.5 Montpellier Client Engineering EMEA *****
```

```
### ##### # # ##### #####
# # # ## ## # # # #
# # # # # # # # #
# ##### # # # # # #
# # # # # # # # #
# # # # # # # # #
### ##### # # # # # #
```

```
***** Contact: benoit.marolleau@fr.ibm.com *****
```

```
[14:10:32][DEMOP.IBM.COM][~/home/benoit]# sc status all
RUNNING | cameltrade (Camel Route for VMS events Kafka streaming)
NOT RUNNING | daytrader (DayTrader)
RUNNING | kafka (Apache Kafka bootstrap server) Port 9092
RUNNING | kafkaviz (Apache Kafka Visualizer) Port 8080
NOT RUNNING | node-red (node-red)
RUNNING | zookeeper (Apache Zookeeper Server using OpenJDK)
```

← PASE service management with Service Commander

```
[14:10:37][DEMOP.IBM.COM][~/home/benoit]#
```

Messages Consumers

Showing messages on partition 0 of vmsevents

Publish Message Search

(116) -1699275949967 -
{\"table\":\"vmsevents\",\"operation\":\"INSERT\",\"row\":{\"id\":\"1703\",\"objecttype\":\"personlaptop\",\"detection_score\":58.52,\"event_ts\":\"2023-11-06-12.58.18.871000\",\"event_src\":\"Camera2\"}}

(115) -1699275934797 -
{\"table\":\"vmsevents\",\"operation\":\"INSERT\",\"row\":{\"id\":\"1702\",\"objecttype\":\"personlaptop\",\"detection_score\":56.96,\"event_ts\":\"2023-11-06-12.58.03.824000\",\"event_src\":\"Camera2\"}}

(114) -1699275919762 -
{\"table\":\"vmsevents\",\"operation\":\"INSERT\",\"row\":{\"id\":\"1701\",\"objecttype\":\"personlaptop\",\"detection_score\":78.61,\"event_ts\":\"2023-11-06-12.57.48.745000\",\"event_src\":\"Camera2\"}}

Kafka server on IBM i
Published message to
(external) Kafka applications →

Demo - Kafka Consumer & Mobile (ntfy.sh) Push notification

Subscription to a topic (“vmsevent”), and send an alert based on a threshold (2 “persons” detected)

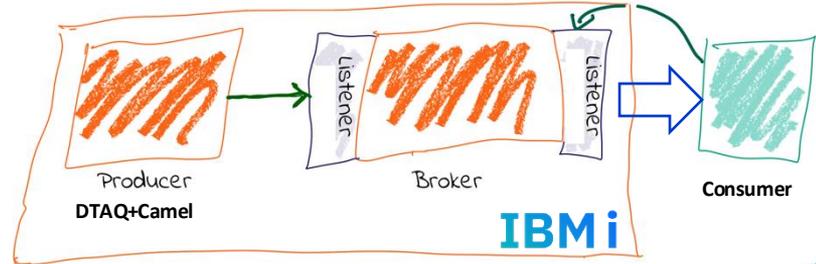
kafka_consume.py

```
import kafka, time, json, numpy as np, requests

topic = "vmsevents"
bootstrap_servers = "10.3.61.2:9092"
consumer = kafka.KafkaConsumer(topic, bootstrap_servers=bootstrap_servers, auto_offset_reset="latest")

assert(consumer.bootstrap_connected())
print(f"Bootstrap '{bootstrap_servers}' connected. Listening...")
print(consumer.topics())

for message in consumer:
    print(message)
    #message = f""
    #Message received: {message.value}
    #Message key: {message.key}
    #Message partition: {message.partition}
    #Message offset: {message.offset}
    #""
    message = f""{message.value}""
    count = message.count("person")
    print(count)
    if (count>=2):
        print("alert")
        requests.post("https://ntfy.sh/iotvms-events", data="Alert 🚨 2 personnes detectees".encode(encoding='utf-8'))
```



Watson Geospatial Analytics + *HTTP functions*

```
--  
-- Get Geodetic data for one of Scott's fave restaurants  
--  
select st_point(longitude, latitude) as thai_pop_geo  
from json_table(QSYS2.HTTP_GET(  
  'https://geocoding.geo.census.gov/geocoder/locations/onlineaddress?address='  
concat ADDRESS concat '&benchmark=2020&format=json',  
  '{"header":  
"User-Agent,Scott","sslCertificateStoreFile":"/home/javaTrustStore/fromJava.KDB"}'  
),  
  'lax $.result.addressMatches[0].coordinates'  
columns(  
  longitude varchar(100) path 'lax $.x',  
  latitude varchar(100) path 'lax $.y'));
```

THAI_POP_GEO
01000000E61000000400000001000000786B8AFF9D024640786B8AF

<https://www.ibm.com/docs/en/i/7.5?topic=concepts-spatial-reference-system>

<https://api-adresse.data.gouv.fr/>

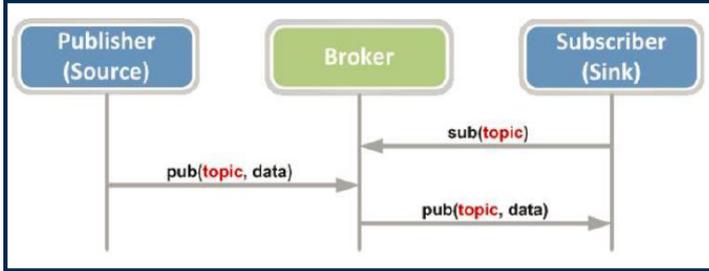
https://www.ibm.com/docs/api/v1/content/ssw_ibm_i_75/sqlp/rbafyhttpoverview.htm

Kafka, Camel, Comment démarrer?

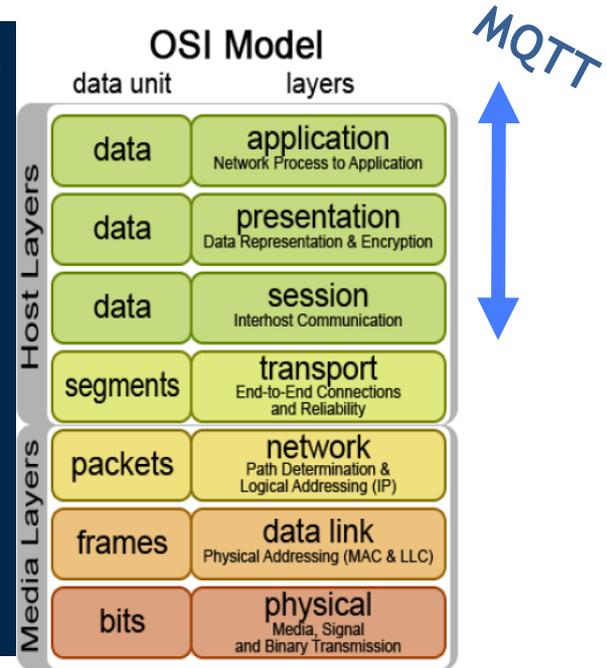
- Exemples "IBM i OSS" <https://github.com/IBM/ibmi-oss-examples/>
 - Dossier « [Camel](#) »
 - Bridge entre data queue & Kafka (Db2->Kafka bridge)
 - Nécessite Kafka
 - Bridge Message queue to email
 - Nécessite un SMTP server
 - Monitoring Disque avec email
 - Nécessite un SMTP server
 - Monitoring Disque avec message queue
 - envoi de messages via message queue *SYSOPR
- Workshop Camel / Kafka en ligne:
 - <https://github.com/ThePrez/FOCUS2020-Workshop/>

Le protocole MQTT

- MQTT(Message Queuing Telemetry Transport) is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol.
- It was designed as an extremely lightweight **publish/subscribe** messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium
- MQTT was found by IBM and in 2013 it was donated to OASIS (Organization for the Advancement of Structured Information Standards)



- The publish-subscribe model requires a message broker. The broker is for distributing messages to interested clients based on the topic of a message.



Topic **Publish** Example:

temperature sent from a 'pipeline' device : `iot-2/type/NI/id/pipeline/cmd/msg/fmt/json{"name":"temperature","value":15}`

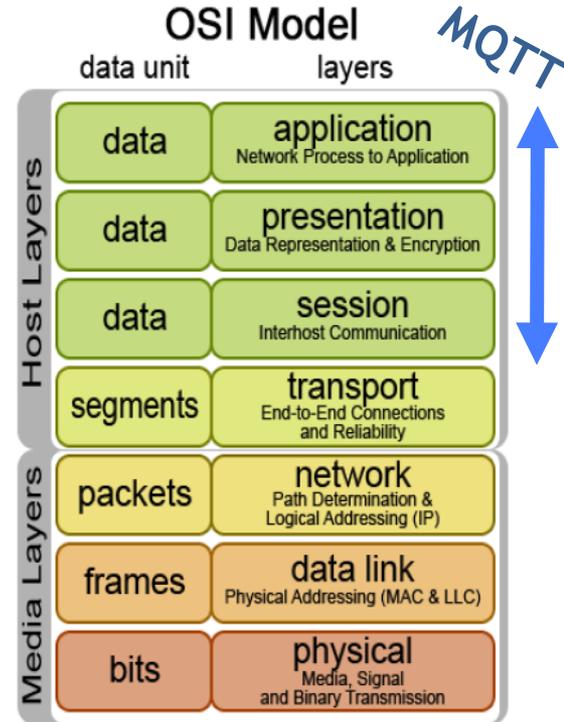
Topic **Subscribe** Example:

`myhome/groundfloor/livingroom/temperature` or `Germany/Bavaria/car/2382340923453/latitude`
`myhome/groundfloor+/temperature` (+ is a wildcard equivalent to "any")

Le protocole MQTT

MQTT Characteristics

- Lightweight
 - smallest packet size 2 bytes (header)
 - reduced clients footprint
- Security
 - MQTT is over TCP...use SSL/TLS for security
 - username/password on connection
 - Encrypt payload
- Reliable
 - Three QoS levels (0: fire & forget / at most once, 1: At least once, 2: exactly once)
 - Avoid packet loss on disconnections - In case of unreliable networks, MQTT guarantees retransmissions and the delivery of the messages
 - Persistent Session / Clean Session
- Simple
 - TCP Based: socket connection oriented
 - Asynchronous : no wait for response
 - Publish / Subscribe : decoupling producer and consumers
 - Payload agnostic: any data format (text, binary, JSON, XML...)



<https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>

Kafka sur IBM i

Basé sur Java, tout comme Apache Camel
Kafka Client existe aussi en Node.js....

// **Prérequis** : JVM (Openjdk11...)

```
# yum install wget ca-certificates-mozilla  
gzip tar-gnu openjdk-11 coreutils-gnu sed-  
gnu grep-gnu
```

// **Installation du produit (téléchargement)**

```
# wget
```

```
https://apache.osuosl.org/kafka/2.6.0/kafka\_2.13-2.6.0.tgz
```

// **Démarrage Zookeeper** (coordination vs. Brokers & partitions) dans PASE

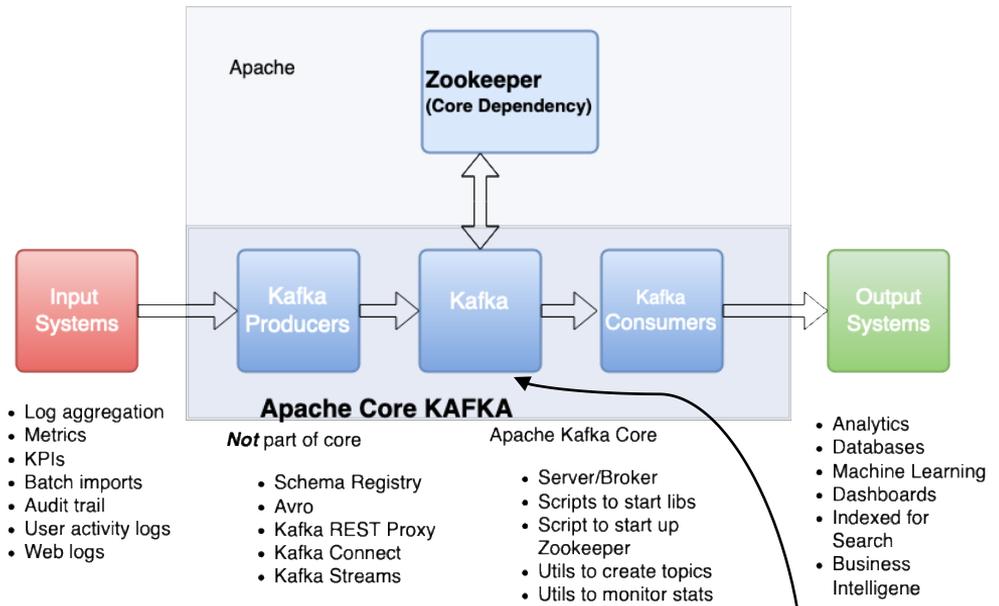
```
# kafka/bin/zookeeper-server-start.sh kafka/config/zookeeper.properties
```

// **Démarrage du serveur** (Broker) dans PASE

```
# kafka/bin/kafka-server-start.sh kafka/config/server.properties
```

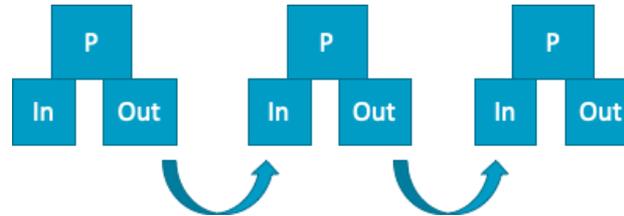
// **Test avec un producteur** (ici Application Java) publiant sur un topic "my-topic"

```
# kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my-topic
```



[Kafka Visualizer](#)
(Graphique)

- Bibliothèques Java → Une application Camel est une application Java
D'autres langages peuvent être utilisés pour décrire des « Expressions »
- Prérequis : JVM sur IBM i (JV1, Openjdk etc.)
- Camel utilise une concaténation répétable et normalisée d'objets "Processor" et de "Messages" groupés dans un ensemble appelé "Exchange"
 - Message "In"
 - "Processor"
 - Message "Out"



- Note : Camel peut être considéré comme un Node-RED en version programmatique Java : simple, beaucoup de connecteurs, passe-plat entre applications

Kafka, Camel : Service Commander

- Gérer les services IBM I dans PASE facilement :

```
bash-4.4$ sc check all
RUNNING      | as-database (System *DATABASE Host Server)
NOT RUNNING  | ftp (System FTP server)
NOT RUNNING  | zookeeper (Apache Zookeeper Server using OpenJDK)
NOT RUNNING  | kafka (Apache Kafka bootstrap server)
RUNNING      | as-central (System *CENTRAL Host Server)
RUNNING      | navigator (IBM Navigator for i)
NOT RUNNING  | kafkaviz (Apache Kafka Visualizer)
RUNNING      | gitbucket (GitBucket git web platform)

bash-4.4$ sc start kafka
Performing operation 'START' on service 'kafka'
Attempting to start service dependency 'zookeeper' (Apache Zookeeper Server using OpenJDK)...
Service 'Apache Zookeeper Server using OpenJDK' successfully started
For details, see log file at: /home/JGORZINS/.sc/logs/2021-03-12T13:23:31.093-0500.zookeeper.log
Service 'Apache Kafka bootstrap server' successfully started
For details, see log file at: /home/JGORZINS/.sc/logs/2021-03-12T13:23:31.086-0500.kafka.log
```

- Exemples, code, vidéo : <https://github.com/ThePrez/ServiceCommander-IBMi>