Kafka and IBM i

Presented by:
Jesse Gorzinski
jgorzins@us.ibm.com
Twitter: @IBMJesseG

About your speaker

- Business Architect of Open Source Technologies

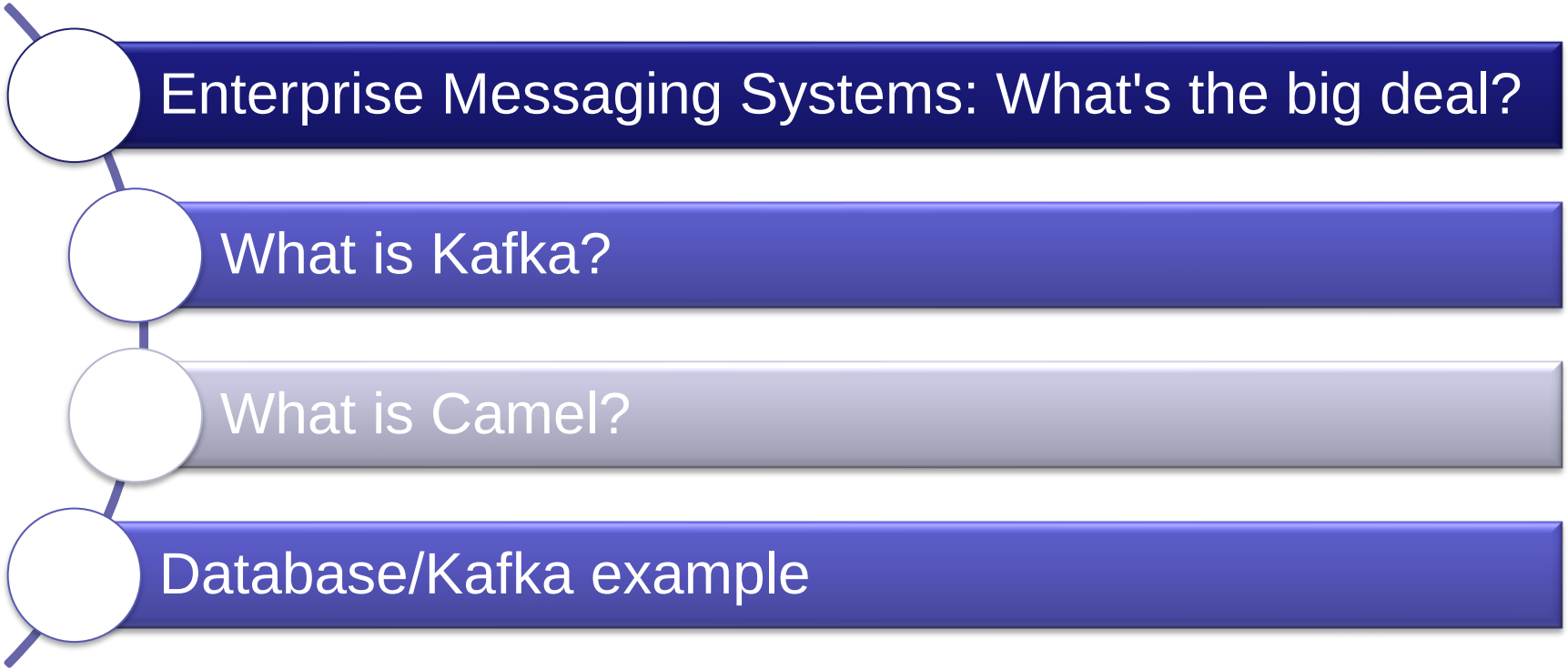
- Reach me at:
  - jgorzins@us.ibm.com
  - Twitter: @IBMJesseG

Justin Reock

Justin has over 20 years' experience working in various software roles and is an outspoken free software evangelist, delivering enterprise solutions and community education on databases, integration work, architecture, and technical leadership.

He is currently the Field CTO and Chief Evangelist at Gradle Enterprises

Agenda



Enterprise Messaging Systems: What's the big deal?

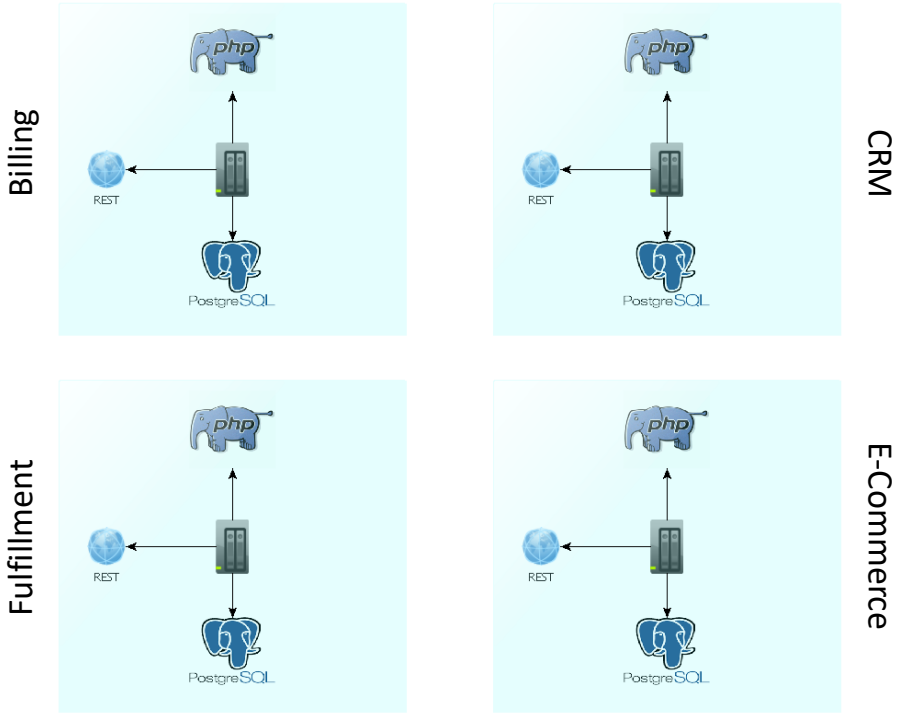
Quick Exercise…

- I’m going to show you three slides
- They all have something in common…
- See if you can figure out what it is!

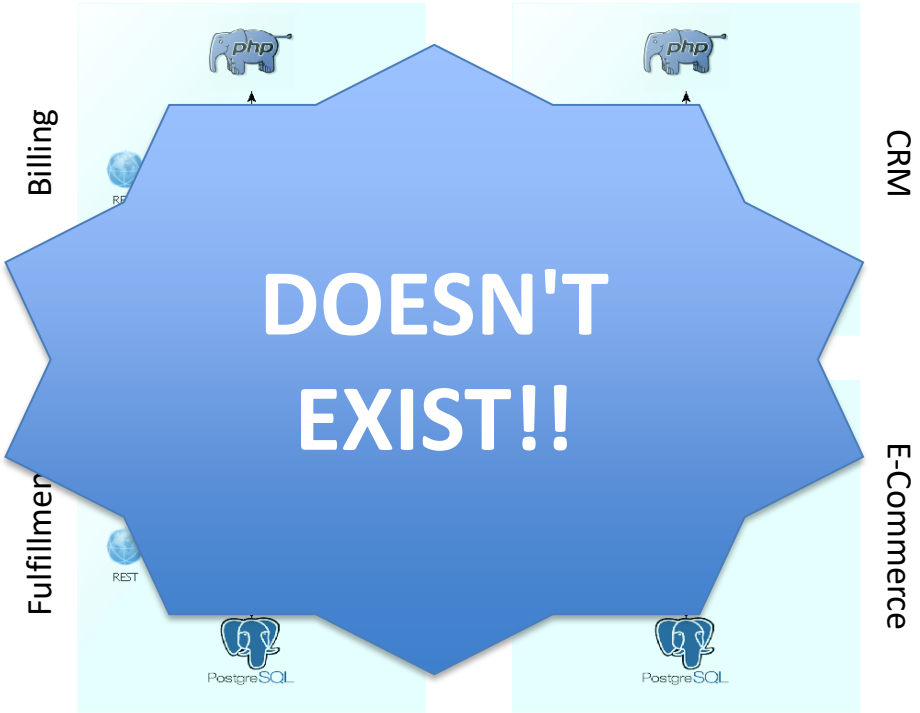
The Chupacabra

A Magical Unicorn…

A Fully Homogeneous Enterprise Landscape…

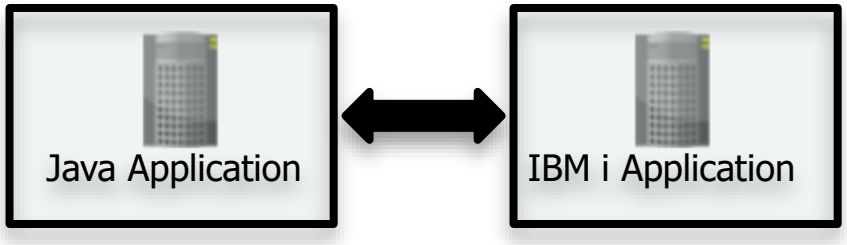


A Fully Homogeneous Enterprise Landscape…



Traditional Messaging Middleware

- Applications often have a need to send information back and forth to one another in a normalized fashion
- Before the advent of messaging systems, it was often difficult if not impossible to “federate” applications written in disparate languages or residing on heterogeneous platforms
- For instance, JMS, the Java Messaging Service, arose out of a growing need to federate very different systems with each other

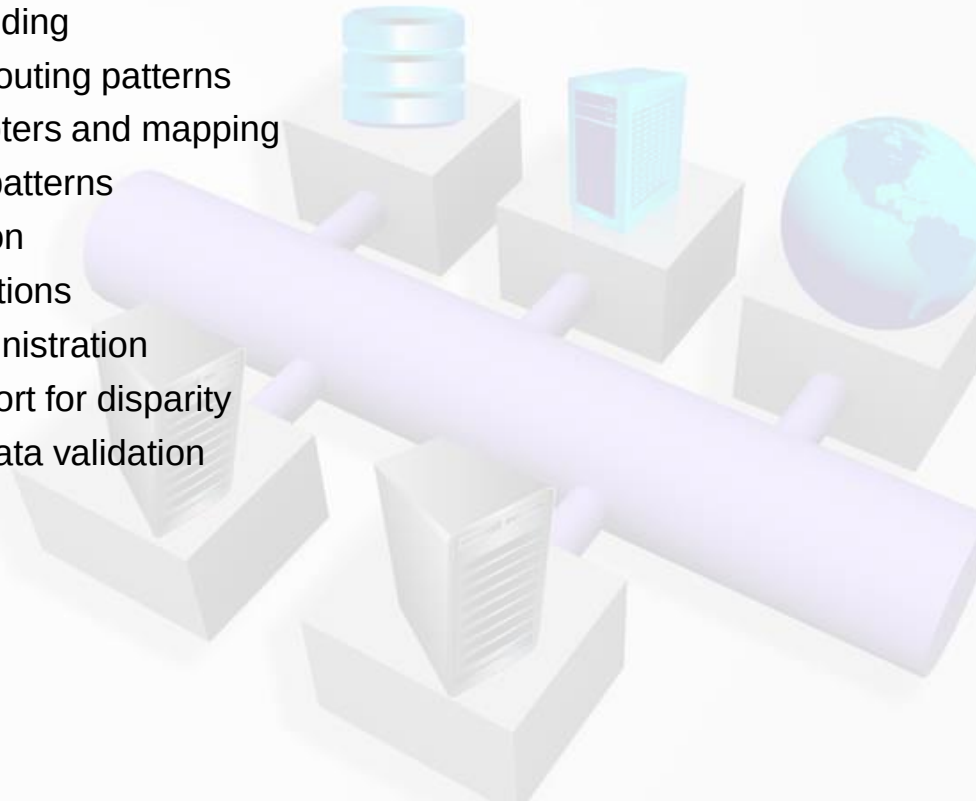


A Word on ESBs

- Perhaps the most recognizable solution for this problem of heterogeneous systems integration is the pattern of an Enterprise Service Bus or ESB
- An ESB is an architectural design pattern that focuses on providing federation between systems using a common runtime
- Ideally, a Service Bus will provide loosely coupled endpoints to which various, heterogeneous systems can communicate with one another via their native frameworks
- So, for instance, a .NET SOAP-driven application should be able to send data along the bus, which could be received by, say, a Java-based JMS destination
- The ESB will provide payload and protocol normalization to facilitate that communication
- Loose coupling will ensure that services can remain unaware of federated services' languages and frameworks

ESB Capabilities

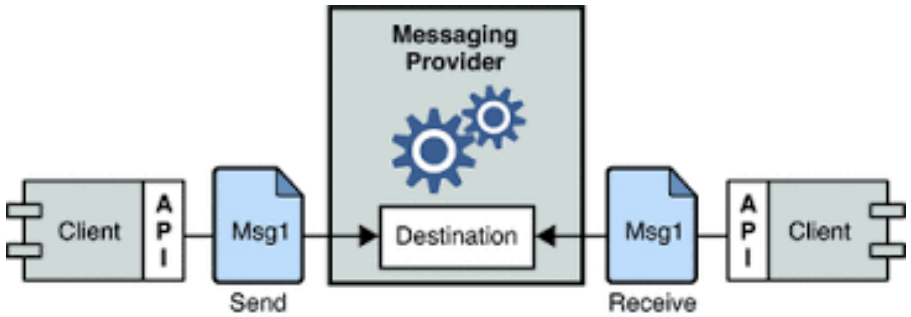
- In general, an ESB should provide the following functionality:
  - Transport Invocation – protocols and data binding
  - Data routing and transformation – Message routing patterns
  - Platform mediation – Language-specific adapters and mapping
  - Messaging – Message oriented middleware patterns
  - Orchestration – Business process coordination
  - QoS – Security, guaranteed delivery, transactions
  - Administration – Monitoring, operational administration
  - Platform agnosticism – Loose coupling, support for disparity
  - Data validation – Schema and/or canonical data validation



Examples of ESBs

Message oriented middleware

- If you don’t need a full ESB, you may still be making use of (and paying for) commercial messaging
- Technically, MOM is any platform that sends and receives messages between distributed applications
- Commercial versions include IBM MQ, Amazon SQS, and Oracle AQ
- This pattern allows for asynchronous processing, and normalization of data exchanges
- Clients connect to a messaging provider, and send and receive messages via that provider



What's It Good For?

- Data normalization makes it a lot easier for disparate applications to trade information
- Just like in web services, heterogeneous systems can agree on a data "contract" and send and receive data in that format
- All the application needs is a client compatible with the messaging provider
- And since messages are allowed to queue up on the message provider, it also makes it very easy to perform asynchronous processing
- Most message providers offer additional benefits, like guaranteed delivery, options for traffic shaping, high availability, and scaling

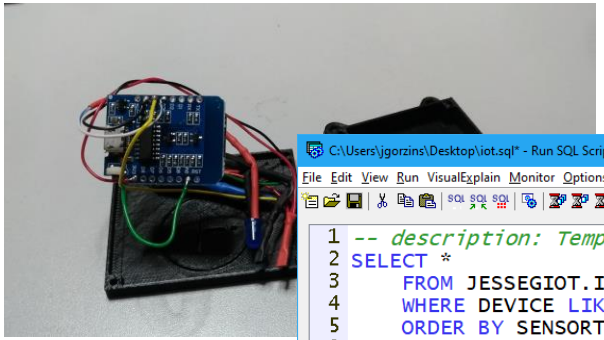
Commercial Messaging Software

- Looking at the role fulfilled by an ESB platform, it is fair to say that these technologies can quickly become deeply embedded in your infrastructure
- In fact, at first blush, many businesses just assume that they are “stuck with” their ESB, messaging, and/or otherwise middleware and integration-related pieces of software
- Commercial vendors understand the importance of this integration power as well as the ”stickiness,” and so these technologies, things like TIBCO, WebSphere MQ, Oracle AQ, etc, tend to be expensive
- Remember, though, that these technologies are necessarily loosely integrated -- the SOA-like recommended architectures that called for them in the first place also calls for modularity and ease of replaceability
- So all you really need to do is make sure that whatever solution you choose to replace your existing commercial ESB or messaging solution can recreate the same endpoint contracts – i.e. the same REST messages, SOAP WSDLs, etc…
- That way, related systems will ideally never even notice that you swapped out the endpoint provider!

Meet: Some Really Good, Really Free Solutions!

- Apache ActiveMQ is a JMS implementation, and can help with both federation of systems and asynchronous processing
- Apache Kafka is a high-throughput streaming event engine suitable for very large datasets
- So…

Use cases beyond the traditional



C:\Users\jgorzins\Desktop\iot.sql* - Run SQL Scripts - common1.iinthecloud.com(Ihost)

File Edit View Run VisualExplain Monitor Options Connection Tools Help

```
1  -- description: Temperature
2  SELECT *
3      FROM JESSEGIOT.IOT_RECORDS
4      WHERE DEVICE LIKE '%temp'
5      ORDER BY SENSORTIME DESC
6      LIMIT 50;
7
8  -- description: Light
9  SELECT *
10     FROM JESSEGIOT.IOT_RECORDS
11     WHERE DEVICE LIKE '%light'
12     ORDER BY SENSORTIME DESC
13     LIMIT 50;
14
```

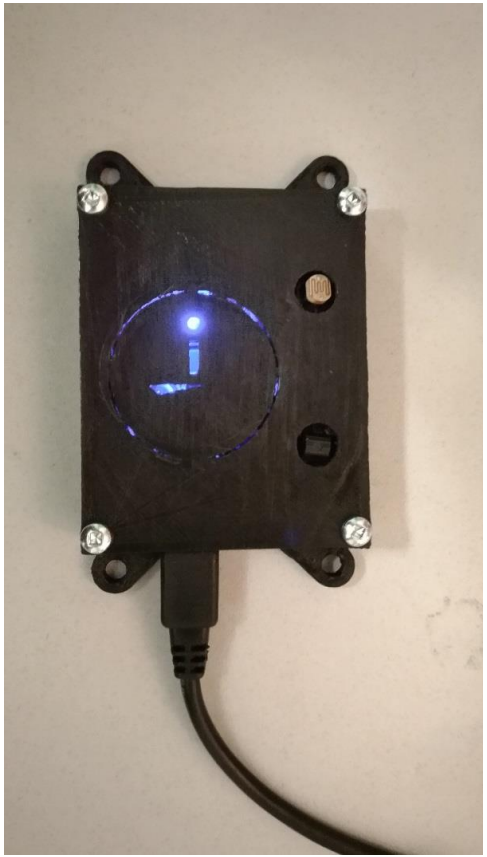
DEVICE	SENSORVALUE	SENSORTIME
factory/1/temp	24.75	2019-04-12 16:07:05.510507
factory/1/temp	24.75	2019-04-12 16:06:35.337323
factory/1/temp	24.75	2019-04-12 16:06:05.127732
factory/1/temp	24.75	2019-04-12 16:05:34.939690
factory/1/temp	24.75	2019-04-12 16:05:04.754581
factory/1/temp	24.75	2019-04-12 16:04:34.556798
factory/1/temp	24.75	2019-04-12 16:04:04.355322

Done: 43 rows retrieved.

Temperature | Light

Messages | Global Variables and Special Registers

Connected to relational database Ihost on common1.iinthecloud.com as TIMMR - 060877/QUSER/QZDASOINIT using JDBC configuration 'Common



What is Kafka?

What's Kafka?

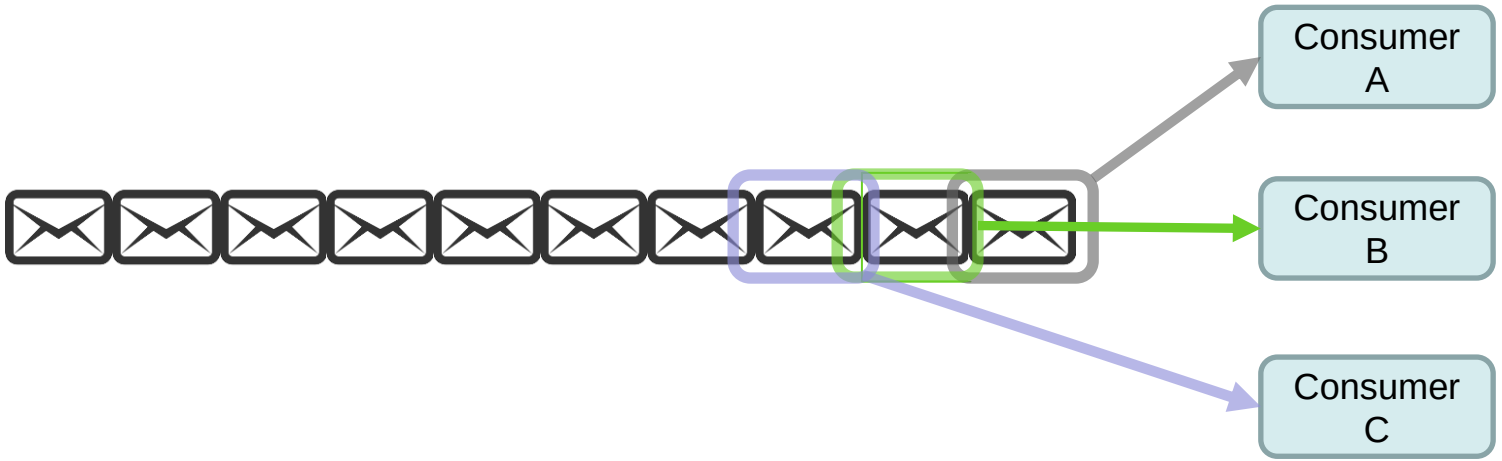
- Where ActiveMQ is a traditional message broker focused on flexible and wide integration of heterogeneous systems, **Kafka is a high-throughput streaming event system** meant for heavy traffic
- Organizations comfortable with both technologies will find that **most of their conceivable messaging needs will be met** by understanding the traditional messaging landscape as well as emerging world of streaming integration
- Both Kafka and ActiveMQ free and open source under the Apache 2.0 license, meaning that they are permissive as well and are **safe for large enterprises** to not only use but also modify if desired
- Kafka was originally conceived at LinkedIn and was built with that kind of scale in mind, so for most businesses in need of a data streaming solution, **Kafka's architecture and ability to scale will be more than enough**
- Kafka has a **wide library of clients** making it easy to integrate with, and although not as many features and traffic shaping patterns are available as in ActiveMQ, Kafka integrates seamlessly with Camel and as such can be folded naturally into a full middleware integration stack

What's Kafka

- Kafka was built to allow downstream analytics and processing platforms to look at data in a way that **allows for correlation, cross referencing, and other complex operations**
- Traditional messaging systems **focus on a single message**, and the quality and mechanisms surrounding the delivery of that message
- Kafka is more about making sure that **large amounts of data**, potentially over a series of time, can be received flexibly and, where necessary, in **historically-tagged chunks** as opposed to just one message at a time
- Kafka does in fact allow for traditional queuing as well as some other patterns, but, its **primary use most closely resembles that of a Topic pattern** in traditional messaging
- A single Topic will have **multiple interested subscribers** receiving streams of data from producer systems
- Kafka virtualizes these Topics by **breaking them into Partitions**, allowing for even greater horizontal scale

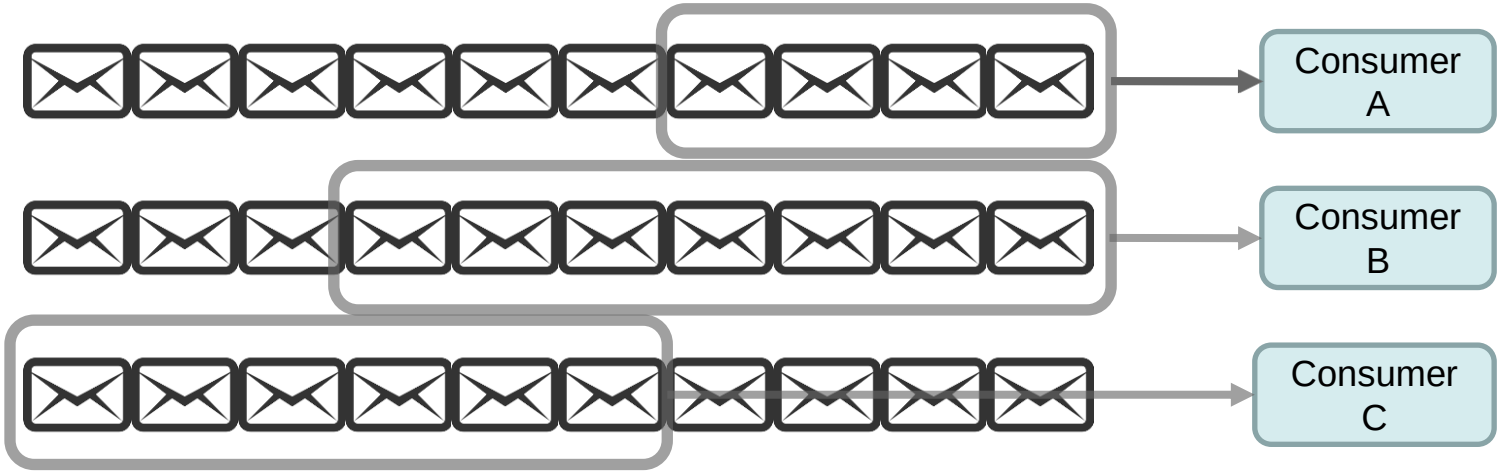
Streaming vs. Traditional Message Queuing

- Primarily, the difference comes down to **how messages are processed** by receiving systems
- In traditional message queueing (not topics), a **single message is processed at a time**, even if that payload contains a lot of data:



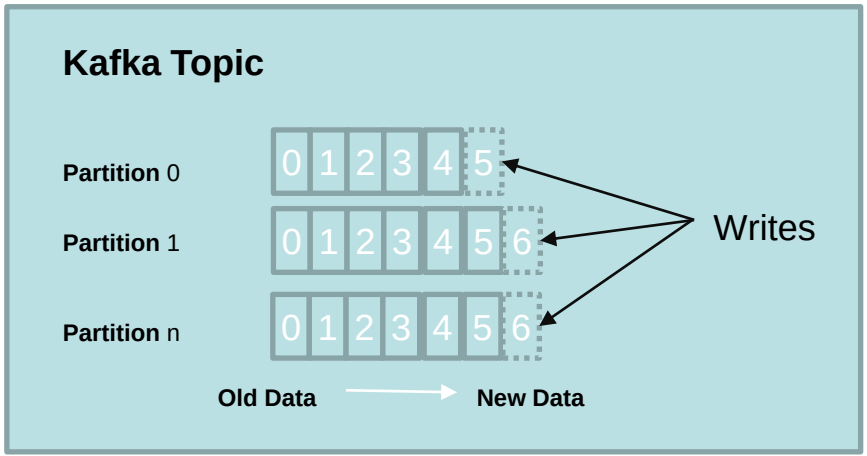
Streaming or Stream Processing

- Streaming differs in that **chunks of data, or series of messages** tend to be processed or at least referenced at a time
- Historical data is usually an option as well, and **consumers can time slice the data** they want to receive



Partitions

- Kafka **stores streamed data in Partitions**, which are on-disk logical groupings of writes from Producing applications
- Notice that “new” writes are **written to the end** of these partitions, which is much different from ActiveMQ, and other traditional FIFO messaging solutions
- Consuming applications will be able to **subscribe to a single partition**
- This allows for **excellent redundancy** in retaining the data and **good load balancing** across storage
- It also provides a straightforward mechanism for **achieving horizontal scale**



Zookeeper

- Apache recognized there was a need for distributed configuration of applications
- Distributed applications, like ones created with ActiveMQ, often require a lot of additonal work to share a common configuration or synchronize local data stores
- All kinds of problems present themselves when dealing with distributed applications in high-traffic environments
  - Race conditions
  - General bugs
  - OS-level limitations
  - File sharing problems
  - Human Error in configuration
- Zookeeper attempts to fix this issue by providing a decentralized network of configuration providers

Zookeeper

- Let's be honest, sharing configuration data dynamically between applications is difficult
- So we tend to slack off in these areas
- We often provide very brittle configuration
- If we want to change or upgrade the way brokers are handled, this usually means a change to every single broker in our arsenal
- Some messaging systems require hundreds of brokers
- Attempting to upgrade all of their configuration by-hand is slow and prone to errors
- Zookeeper takes configuration storage to the next level, by also providing failover and redundancy in its sharing environment
- So a single Zookeeper instance failing will not prevent the entire messaging application from receiving messages

Industry interest in Kafka

10 OUT OF 10

MANUFACTURING

7 OUT OF 10

BANKS

10 OUT OF 10

INSURANCE

8 OUT OF 10

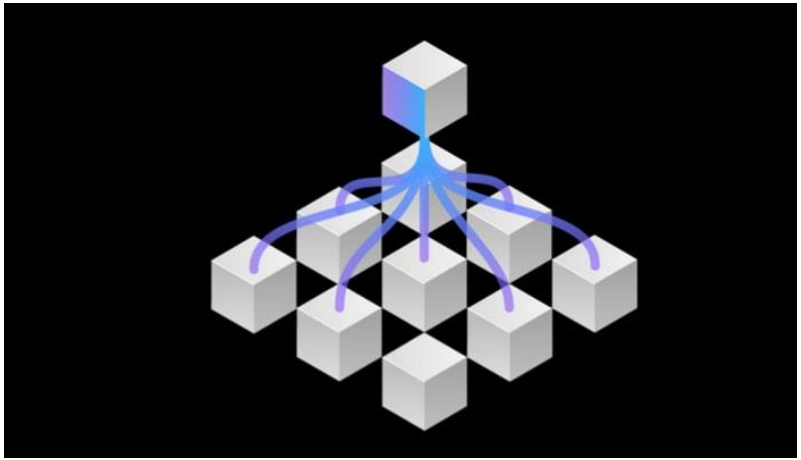
TELECOM

Analysis of key industries using Kafka. (Source: https://kafka.apache.org)

Analysis is based on the 10 largest companies in each sector.

10/10 Largest insurance companies
10/10 Largest manufacturing companies
10/10 Largest information technology and services companies
8/10 Largest telecommunications companies
8/10 Largest transportation companies
7/10 Largest retail companies
7/10 Largest banks and finance companies
6/10 Largest energy and utilities organizations

Hybrid Multicloud



- What needs drove Zookeeper, ActiveMQ, Kafka?
  - Data federation among heterogeneous systems
  - Edge computing
  - Distributed computing
  - Large data workloads

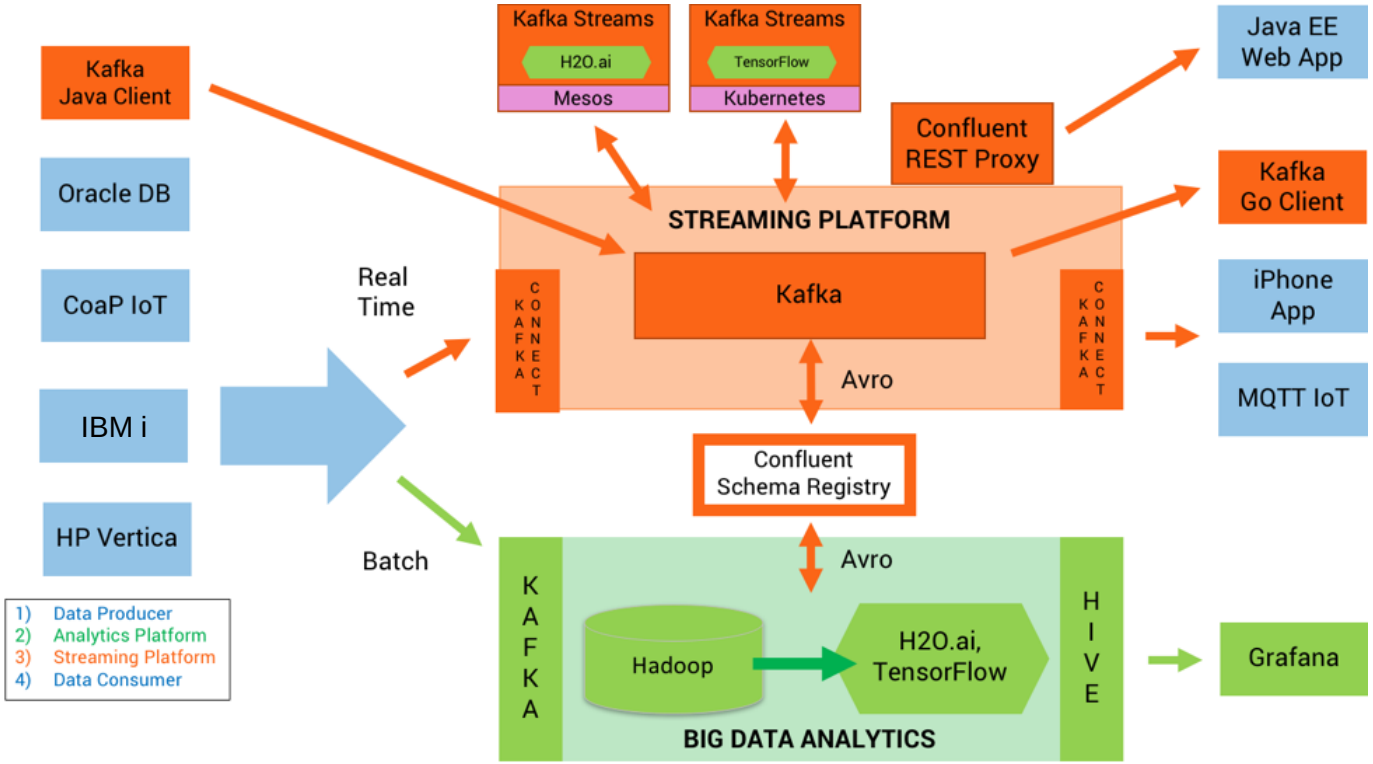


Image source: https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/

Powered by Kafka (https://kafka.apache.org/powered-by)

adidas

adidas uses Kafka as the core of Fast Data Streaming Platform, integrating source systems and enabling teams to implement real-time event processing for monitoring, analytics and reporting solutions.

Spotify

Kafka is used at Spotify as part of their log delivery system.

As part of their Storm stream processing infrastructure, e.g. this and this.

BARCLAYS

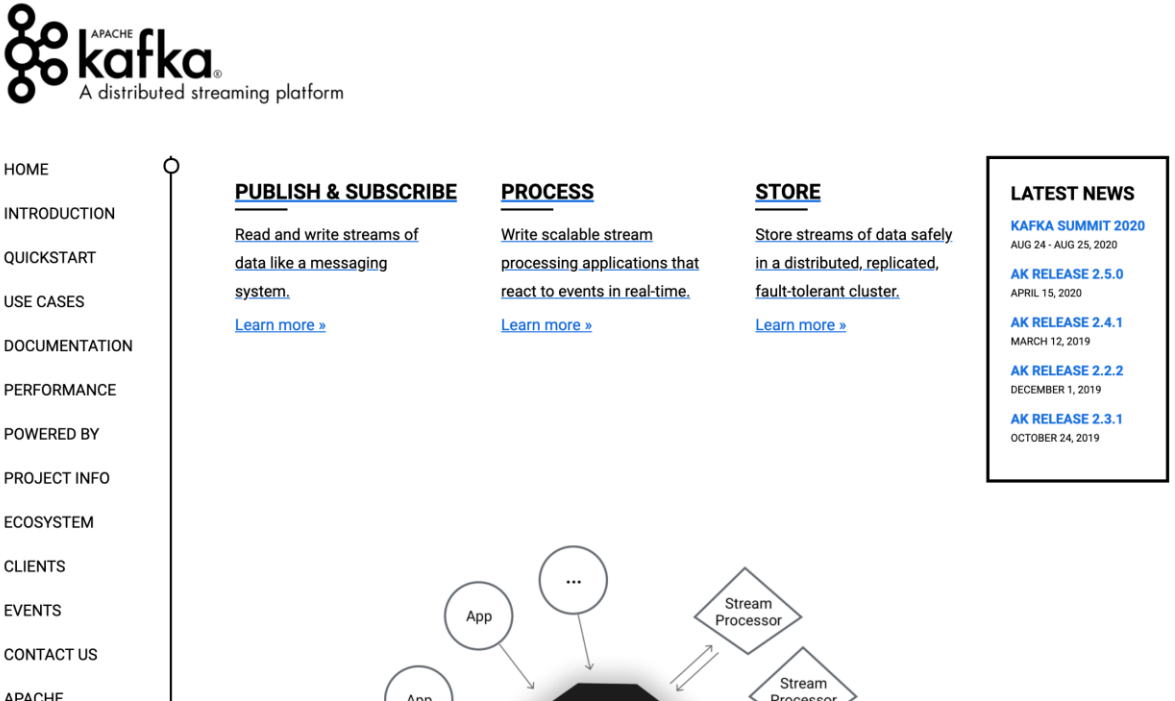
Barclays utilizes Kafka for streaming and analytical information.

LinkedIn

Apache Kafka is used at LinkedIn for activity stream data and operational metrics. This powers various products like LinkedIn Newsfeed, LinkedIn Today in addition to our offline analytics systems like Hadoop.

Obtaining Kafka

- Kafka is maintained and distributed from its top-level community home, https://kafka.apache.org:



Deploying Kafka

- https://ibmi-oss-docs.readthedocs.io/en/latest/kafka/README.html#deploying-kafka-on-ibm-i

- Steps are simple:
  1. Download
  2. Extract
  3. Set up Java environment
  4. Start Zookeeper
  5. Start Kafka

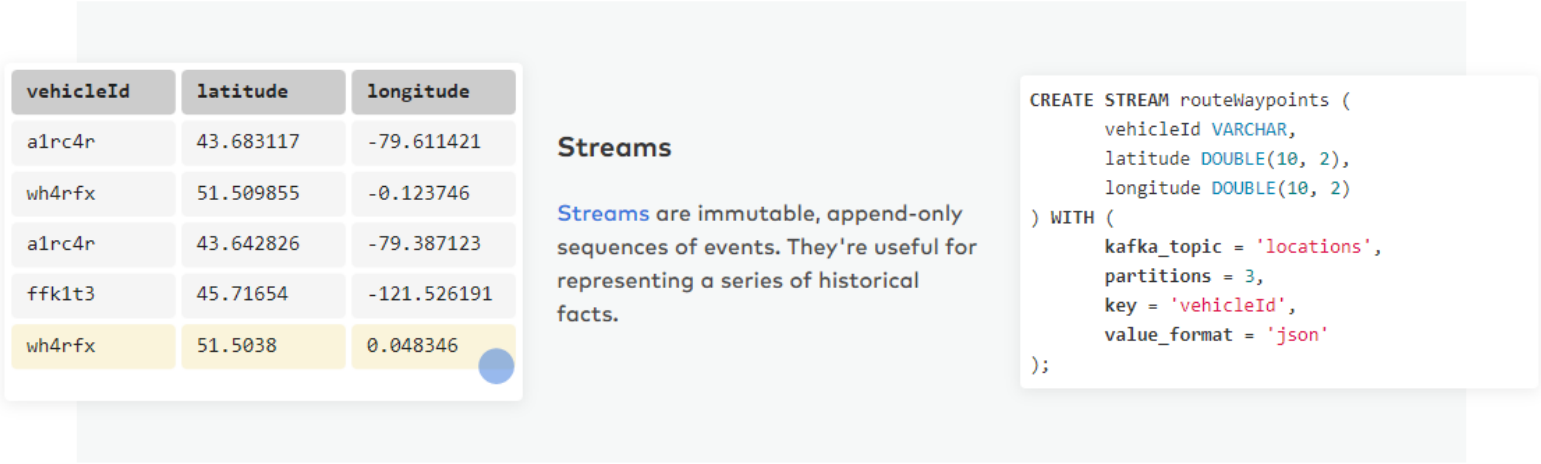
Ways to stream/consume Kafka data?

- https://ibmi-oss-docs.readthedocs.io/en/latest/kafka/README.html
- Db2 Triggers and Apache Camel: stream events in real-time
- Kafka Connect JDBC Source connector: Simple, polling-based technique
- InfoSphere Data Replication and the CDC Replication Engine for Kafka
- Native ILE Kafka client (unsupported): call Kafka functions directly from ILE programs.
- Confluent Platform
    - » ksqlDB, which provides an SQL interface
    - » Kafka REST APIs, which provide a REST interface

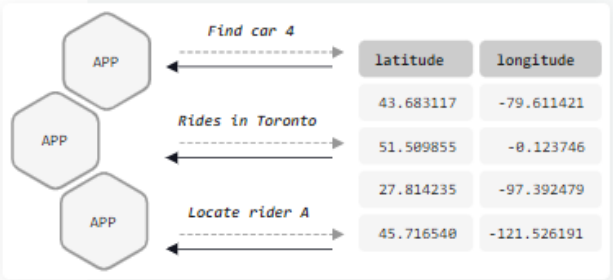
ksqlDB (image credit: ksqldb.io)

- "The database purpose-built for stream processing applications."

- Kafka-native database



ksqlDB queries (image credit: ksqldb.io)



Pull

Pull queries allow you to fetch the current state of a materialized view. Because materialized views are incrementally updated as new events arrive, pull queries run with predictably low latency. They're a great match for request/response flows.

```
SELECT vehicleId,
       latitude,
       longitude
FROM currentCarLocations
WHERE ROWKEY = '6fd0fcdb';
```

ksqlDB queries (image credit: ksqldb.io)

Push

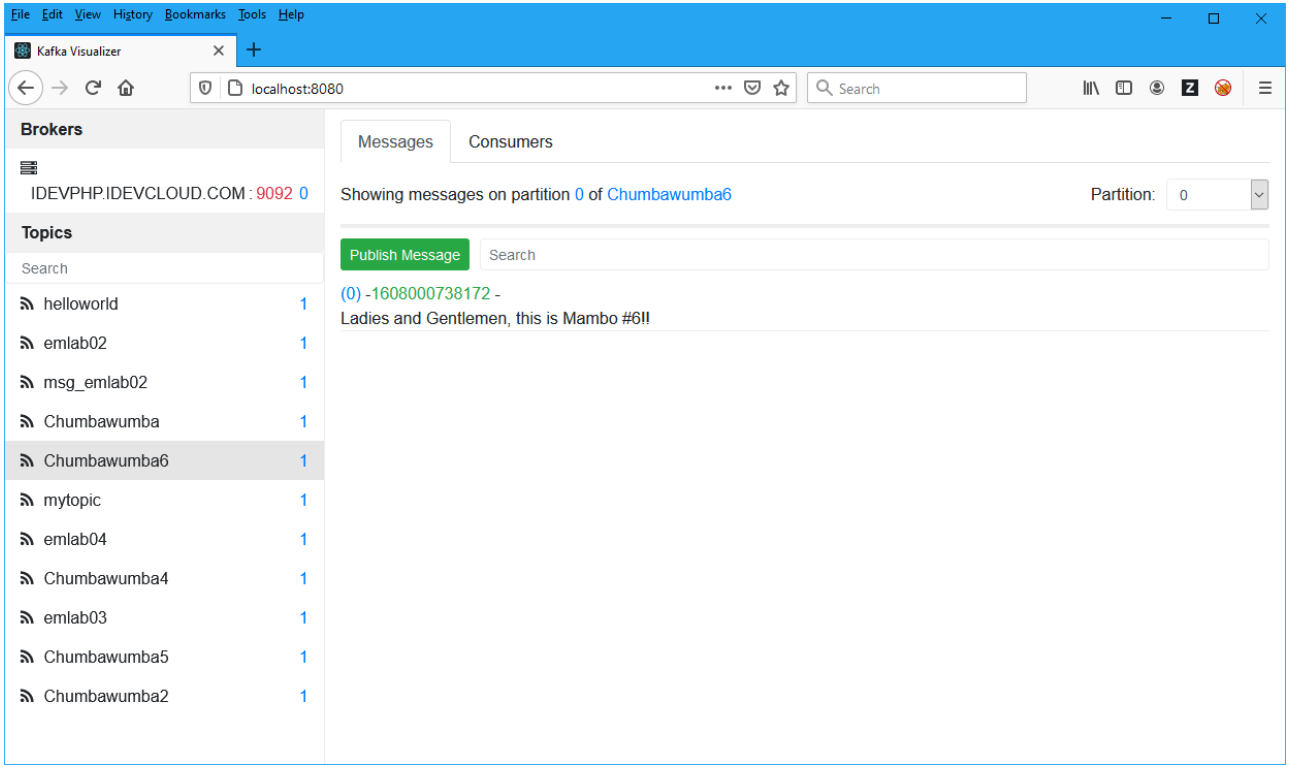
Push queries let you subscribe to a query's result as it changes in real-time. When new events arrive, push queries emit refinements, which allow you to quickly react to new information. They're a perfect fit for asynchronous application flows.

```
SELECT vehicleId,
       latitude,
       longitude
FROM currentCarLocations
WHERE ROWKEY = '6fd0fcdb'
EMIT CHANGES;
```

Kafka Admin/Visualizer tools

- All the necessary stuff is bundled with Kafka itself
- There are plenty of tools out there, both commercial and open source
- One handy open source one we found (screenshot on next slide)
  https://github.com/manasb-uoe/kafka-visualizer
- Fork you can run on IBM i
  https://github.com/ThePrez/kafka-visualizer

G



G

What is Camel?

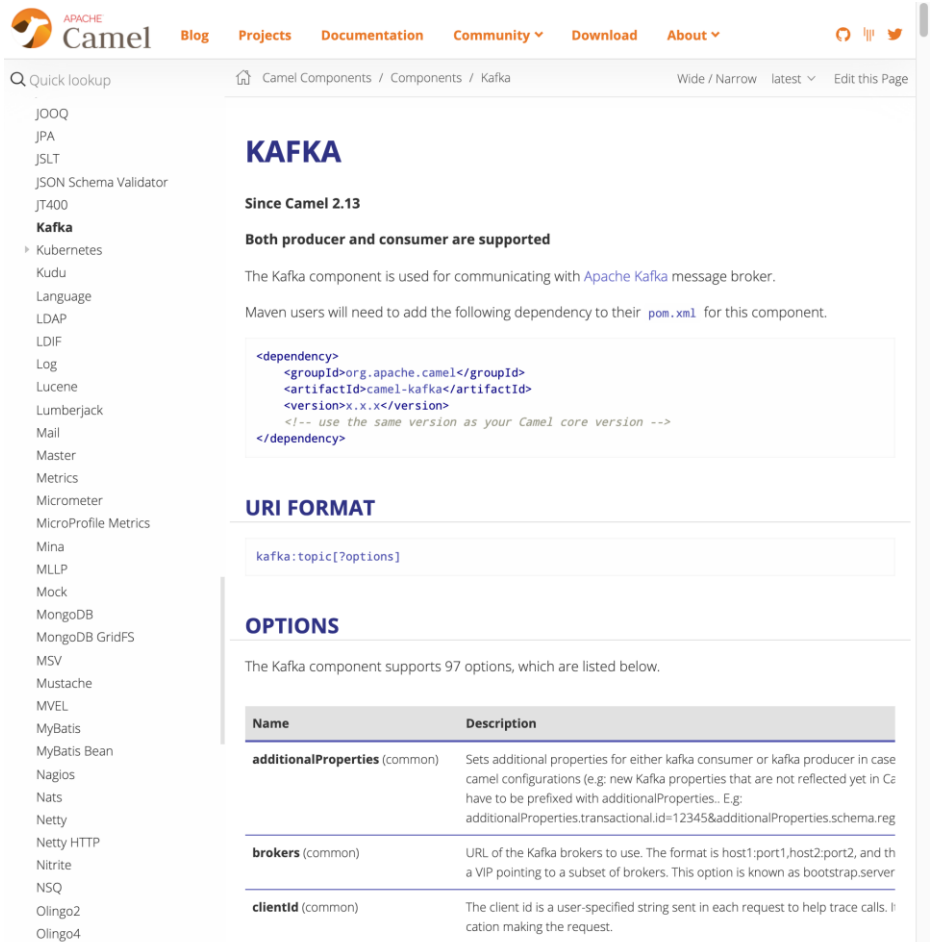
Apache Camel

- Dubbed “the swiss knife of integration”

- Learn about Enterprise Integration Patterns: https://camel.apache.org/components/latest/eips/enterprise-integration-patterns.html

- Has modules/adapters for:
  - Spring Boot
  - OSGi
  - Kafka
  - AWS
  - ActiveMQ
  - MongoDB
  - Facebook/Twitter
  - IoT
  - Google Sheets
  - Git
  - GraphQL
  - REST
  - Nagios
  - PDFs
  - WordPress
  - Twilio
  - TONS more!!
  - …. Including IBM i via JT400! https://camel.apache.org/components/latest/jt400-component.html

Kafka / Camel Integration



- Much like ActiveMQ, **Kafka can integrate seamlessly with Camel**, extending its usability
- Whereas ActiveMQ ships with Camel built-in (though of course it can be decoupled), Kafka integrates with Camel using the **camel-kafka component**
- This component is **provided by the Camel community** and distributed through its normal channels
- **Producer and consumer capabilities** are provided by the component
- It is highly configurable, providing all of the **same options available in the Kafka's Java client library**

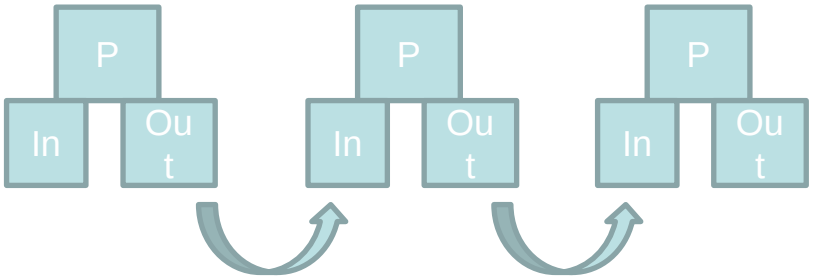
Camel components

camel-ahc-ws
camel-ahc
camel-amqp
camel-apns
camel-asterisk
camel-atmos
camel-atmosphere-websocket
camel-atom
camel-avro
camel-aws
camel-azure
camel-bam
camel-barcode
camel-base64
camel-bean-validator
camel-beanio
camel-beanstalk
camel-bindy
camel-blueprint
camel-bonita
camel-boon
camel-box
camel-braintree
camel-cache
camel-cassandraql
camel-castor
camel-cdi
camel-chronicle
camel-chunk
camel-cm-sms
camel-cmis
camel-coap
camel-cometd
camel-consul
camel-context
camel-core-osgi
camel-core-xml
camel-couchbase
camel-couchdb
camel-crypto
camel-csv
camel-cxf-transport
camel-cxf
camel-digitalocean
camel-disruptor
camel-dns
camel-docker
camel-dozer
camel-drill
camel-dropbox
camel-eclipse
camel-ehcache
camel-ejb
camel-elasticsearch
camel-elasticsearch5
camel-elsql
camel-etcd
camel-eventadmin
camel-exec
camel-facebook
camel-flatpack
camel-flink
camel-fop
camel-freemarker
camel-ftp
camel-ganglia
camel-geocoder
camel-git
camel-github
camel-google-calendar
camel-google-drive
camel-google-mail
camel-google-pubsub
camel-gora
camel-groovy
camel-groovy-dsl
camel-grpc
camel-gson
camel-guava-eventbus
camel-guice
camel-hawtdb
camel-hazelcast
camel-hbase
camel-hdfs
camel-hdfs2
camel-hessian
camel-hipchat
camel-hl7
camel-http-common
camel-http
camel-http4
camel-hystrix
camel-ibatis
camel-ical
camel-ignite
camel-infinispan
camel-influxdb
camel-irc
camel-ironmq
camel-jackson
camel-jacksonxml
camel-jasypt
camel-javaspace
camel-jaxb
camel-jbpm
camel-jcache
camel-jclouds
camel-jcr
camel-jdbc
camel-jetty-common
camel-jetty
camel-jetty9
camel-jgroups
camel-jibx
camel-jing
camel-jira
camel-jms
camel-jmx
camel-johnzon
camel-jolt
camel-josql
camel-jpa
camel-jsch
camel-jsonpath
camel-jt400
camel-juel
camel-jxpath
camel-kafka
camel-kestrel
camel-krati
camel-kubernetes
camel-kura
camel-ldap
camel-leveldb
camel-linkedin
camel-lucene
camel-lumberjack
camel-lzf
camel-mail
camel-metrics
camel-milo
camel-mina
camel-mina2
camel-mllp
camel-mongodb-gridfs
camel-mongodb
camel-mongodb3
camel-mqtt
camel-msv
camel-mustache
camel-mvel
camel-mybatis
camel-nagios
camel-nats
camel-netty-http
camel-netty4
camel-ognl
camel-olingo2
camel-olingo4
camel-openshift
camel-openstack
camel-opentracing
camel-optaplanner
camel-paho
camel-paxlogging
camel-pdf
camel-pgevent
camel-printer
camel-protobuf
camel-pubnub
camel-quartz
camel-quartz2
camel-quickfix
camel-rabbitmq
camel-reactive-streams
camel-rest-swagger
camel-restlet
camel-ribbon
camel-rmi
camel-routebox
camel-rss
camel-ruby
camel-rx
camel-salesforce
camel-sap-netweaver
camel-saxon
camel-scala

Uhm, yeah, the list isn't gonna fit…
The point is there are a lot!

How Does It Work?

- And, you can chain these Exchanges together – just like piping commands through UNIX – and form a Camel Route
- The “Out” message of a previous Exchange becomes the “In” message of a new Exchange:



Interacting with Camel-JT400 Component

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&…`

Db2 Enhancements for Apache Camel

- JSON Publishing Functions provide data in a manner understood by Kafka/ActiveMQ consumers

```
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;
```

- Data Queue Functions allow integration with queues (and therefore Apache Camel) directly from the database

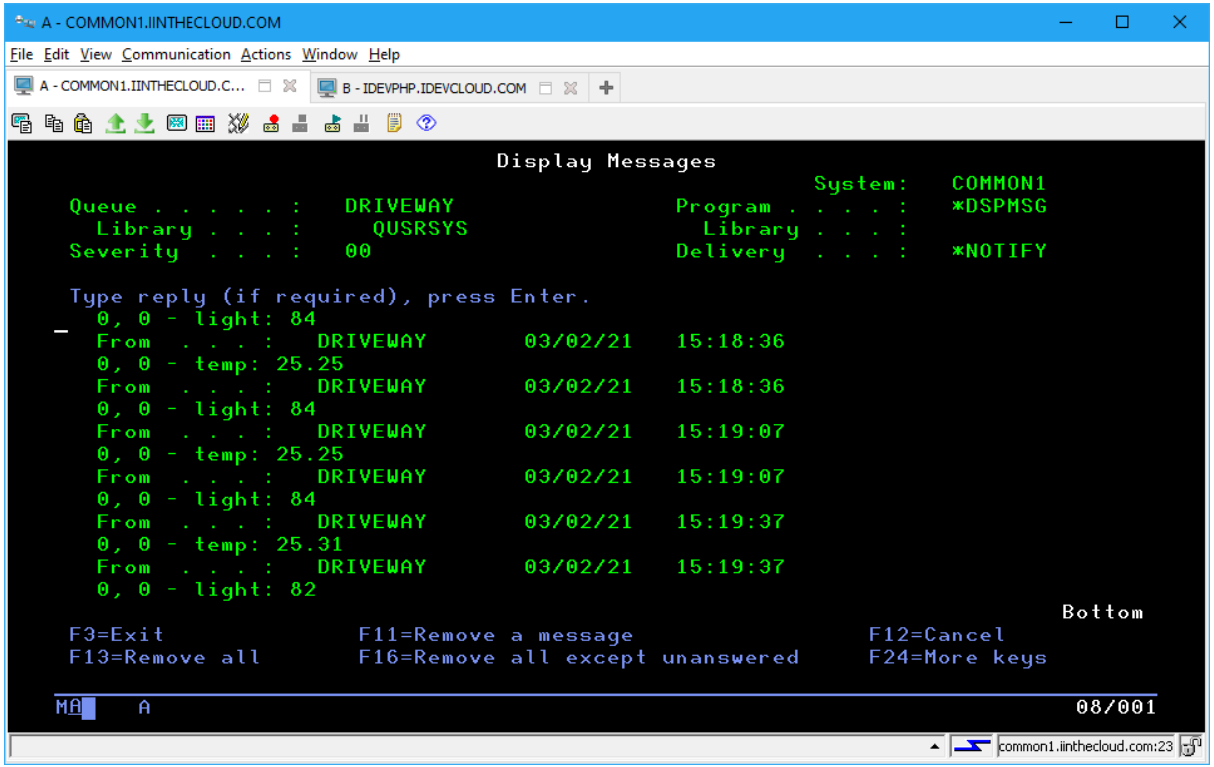
```
call qsys2.send_data_queue_utf8(
    message_data       => scottf.dq_json,
    data_queue         => 'HANDOFF_DQ',
    data_queue_library => 'BANKONOSS');
```

Data Queue to Kafka Bridge

- Same concept, different URIs
- Applied case: stream Db2 transactions to Apache Kafka

```
final String dtaqUri = conf.getDtaQUri(); //something like -> jt400://username:password@localhost/qsys.lib/mylib.lib/myq.DTAQ?keyed=false&format=binary&guiAvailable=false
final String kafkaUri = conf.getKafkaUri(); //something like -> kafka:mytopic?brokers=mybroker:9092
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from(dtaqUri)
        .wireTap("log:msgq_to_email?showAll=true&level=INFO") // This is just for debugging data flowing through the route
        .to(kafkaUri);
    }
});
```

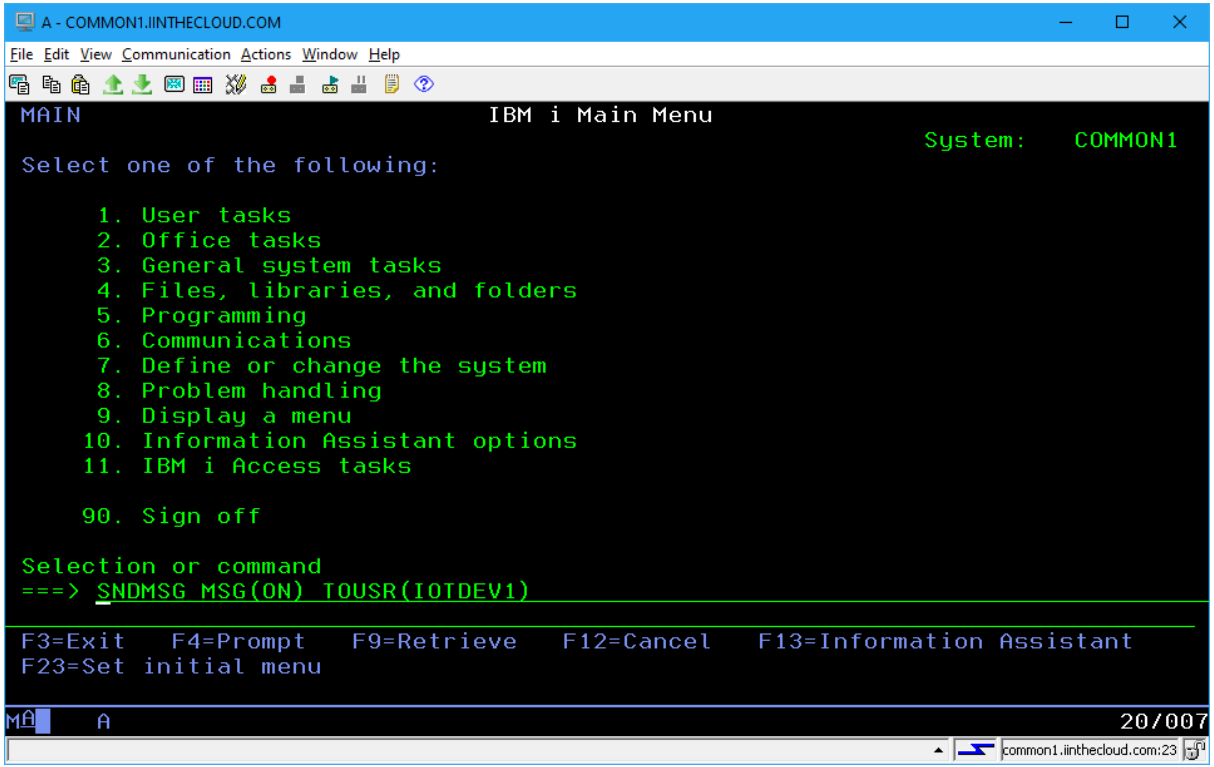
Consume IoT Data? No Problem!



Consume IoT Data? No Problem!

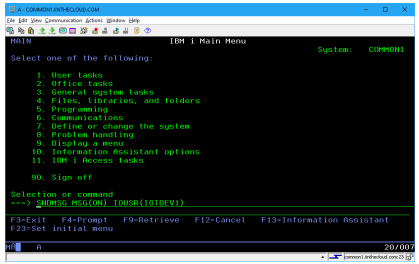
```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("paho:factory/1/light?brokerUrl=ssl://localhost")
        .to("jt400://driveway:xxxxxx@localhost/qsys.lib/QUSRSYS.lib/DRIVEWAY.MSGQ?guiAvailable=false");
    }
});
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("paho:factory/1/temp?brokerUrl=ssl://localhost")
        .to("jt400://driveway:xxxxxx@localhost/qsys.lib/QUSRSYS.lib/DRIVEWAY.MSGQ?&guiAvailable=false");
    }
});
```

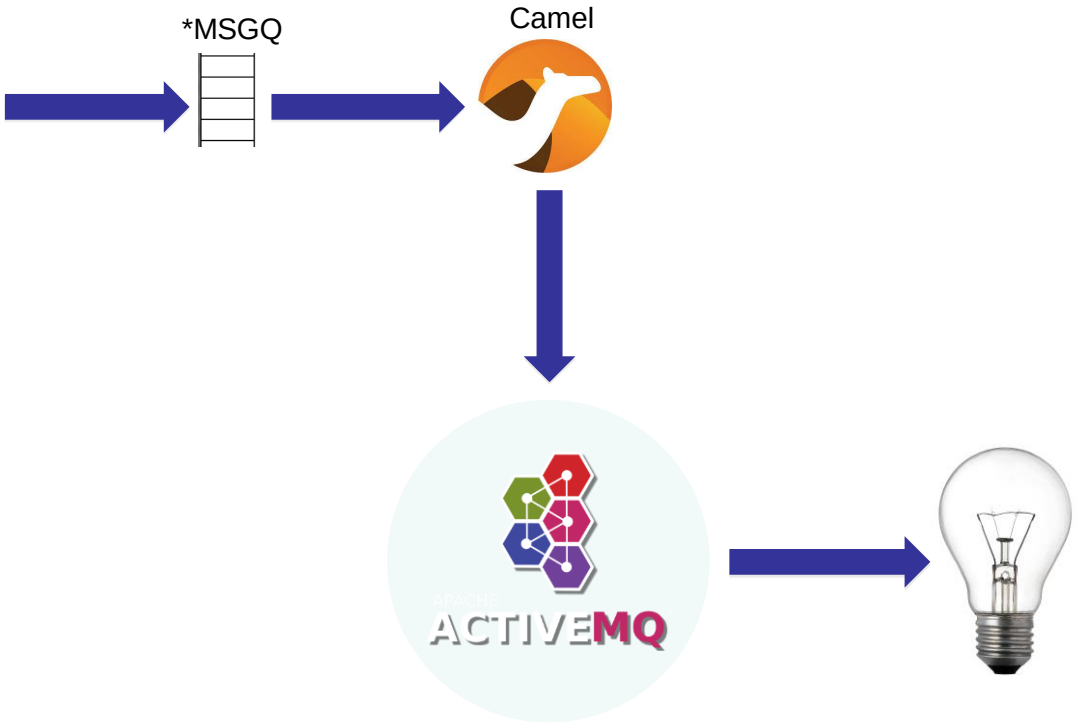
Control IoT Devices? No Problem!



Control IoT Devices? No Problem!

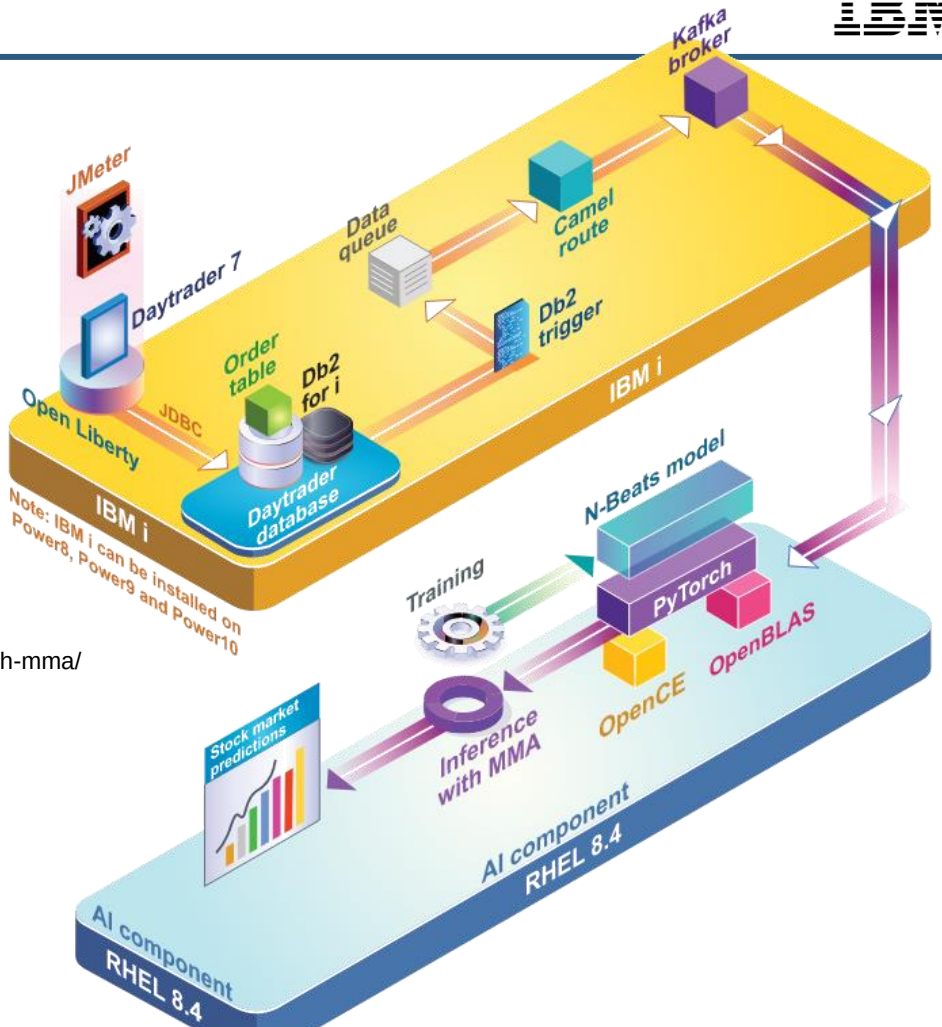
IBM i SNDMSG Command





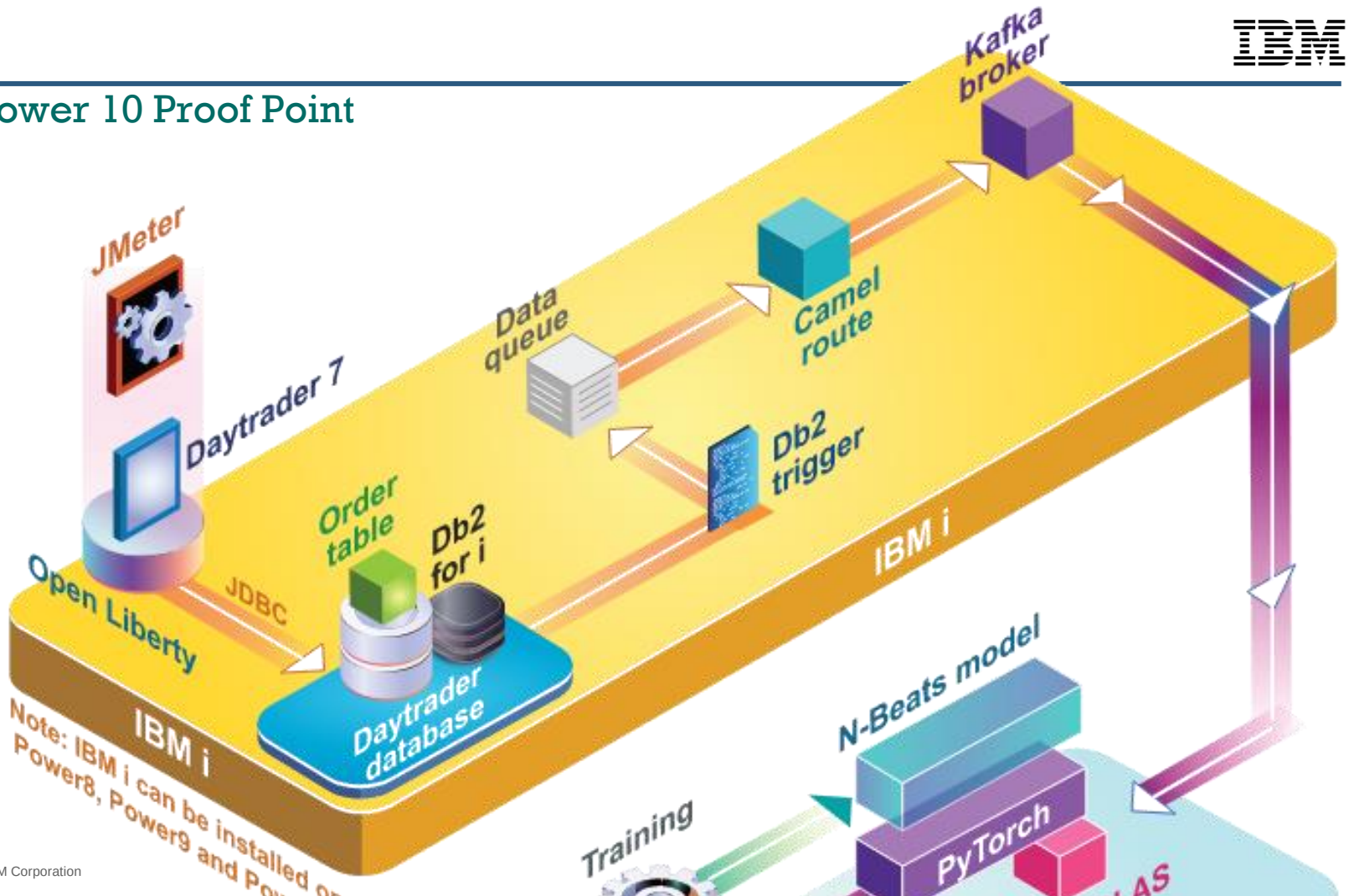
Power 10 Proof point

Power 10 Proof Point



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

Power 10 Proof Point



P10 Proof Point

Closing thoughts

Where to get help?

- Community
  - https://camel.zulipchat.com/
  - https://kafka.apache.org/contact

- Professional Services
  - IBM Systems Lab Services
    - Contact at ibmsls@us.ibm.com or your local Lab Services team
  - IBM Techology Support Services (TSS)
    - Contact jgorzins@us.ibm.com and Randal.Wilson@ibm.com or your local TSS representative

Resources

- IBM i RPMs (RedHat Technology we use for building/distributing)
  - http://ibm.biz/ibmi-rpms
- IBM i Open Source Support
  - http://ibm.biz/ibmi-oss-support
- Jesse Gorzinski's blog
  - http://ibm.biz/open-your-i
  - https://ibmsystemsmag.com/Power-Systems/06/2020/common-open-source-questions-answered
- Open Source Examples
  - http://github.com/IBM/ibmi-oss-examples
- IBM i customer stories
  - http://ibm.biz/ibmistories
- Community chat
  - http://ibm.biz/ibmioss-chat (join at http://ibm.biz/ibmioss-chat-join)
- Jesse
  - jgorzins@us.ibm.com
  - http://twitter.com/IBMJesseG

Kafka and IBM i?

- Yes, of course!
  - It makes sense
  - IBM can help deploy
  - IBM can provide support

The Hybrid Approach

IBM i

World's Best RDBMS

COBOL+RPG

Lowest cost of ownership (TCO)

Reliability, securability, efficiency

Protection of investment

Open Source

Artificial Intelligence

Quantum Computing

Microservices / APIs

DevOps

Internet of Things

Web Technologies

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Revised September 26, 2006

Special notices (cont.)

IBM, the IBM logo, ibm.com AIX, AIX (logo), AIX 5L, AIX 6 (logo), AS/400, BladeCenter, Blue Gene, ClusterProven, DB2, ESCON, i5/OS, i5/OS (logo), IBM Business Partner (logo), IntelliStation, LoadLeveler, Lotus, Lotus Notes, Notes, Operating System/400, OS/400, PartnerLink, PartnerWorld, PowerPC, pSeries, Rational, RISC System/6000, RS/6000, THINK, Tivoli, Tivoli (logo), Tivoli Management Environment, WebSphere, xSeries, z/OS, zSeries, Active Memory, Balanced Warehouse, CacheFlow, Cool Blue, IBM Systems Director VMControl, pureScale, TurboCore, Chiphopper, Cloudscape, DB2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Parallel File System, , GPFS, HACMP, HACMP/6000, HASM, IBM Systems Director Active Energy Manager, iSeries, Micro-Partitioning, POWER, PowerExecutive, PowerVM, PowerVM (logo), PowerHA, Power Architecture, Power Everywhere, Power Family, POWER Hypervisor, Power Systems, Power Systems (logo), Power Systems Software, Power Systems Software (logo), POWER2, POWER3, POWER4, POWER4+, POWER5, POWER5+, POWER6, POWER6+, POWER7, System i, System p, System p5, System Storage, System z, TME 10, Workload Partitions Manager and X-Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries.

A full list of U.S. trademarks owned by IBM may be found at: http://www.**ibm.com**/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
AltiVec is a trademark of Freescale Semiconductor, Inc.
AMD Opteron is a trademark of Advanced Micro Devices, Inc.
InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.
Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries or both.
NetBench is a registered trademark of Ziff Davis Media in the United States, other countries or both.
SPECint, SPECfp, SPECjbb, SPECweb, SPECjAppServer, SPEC OMP, SPECviewperf, SPECapc, SPEChpc, SPECjvm, SPECmail, SPECimap and SPECsfs are trademarks of the Standard Performance Evaluation Corp (SPEC).
The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.
TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).
UNIX is a registered trademark of The Open Group in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

Revised December 2, 2010