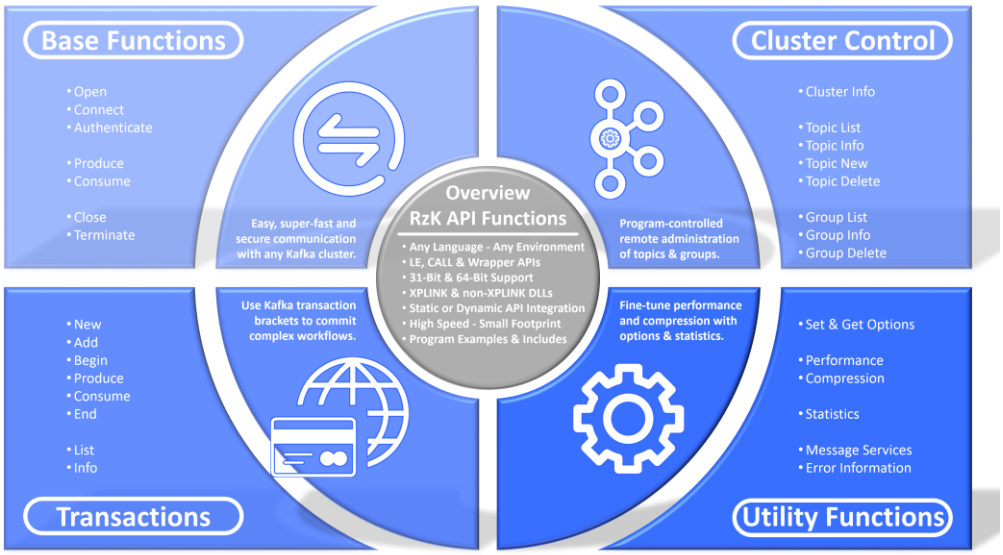
zKafka Connector

Connect to Kafka from any z/OS Application.

Publishing your Mainframe Data to your Kafka Cluster.

IT requirements have picked up on the speed in which the world is changing. And data is the key to your extremely valuable business information. Today data is required to be available anywhere at any time – and still be secure.

Apache Kafka and the mainframe share certain core capabilities: high availability (resilience), scalability and permanent storage. Although Kafka client libraries allow to connect anything from almost anywhere, the connection from z/OS to a Kafka cluster in the open world is quite complex. Until now, there was no easy way to integrate Kafka client libraries.



zKafka Connector (RzK)

- Easy to use
- Use with any language, any environment
- Secure
- Extremely fast
- Kafka transaction support
- Remote administration of cluster
- A set of utilities included

zKafka Connector (RzK) – the Smart Link between the Mainframe and your Kafka Cluster

The zKafka Connector simplifies the way to produce (publish) and consume (read) data from any z/OS application to and from your Kafka cluster. No further program product or middleware is needed.

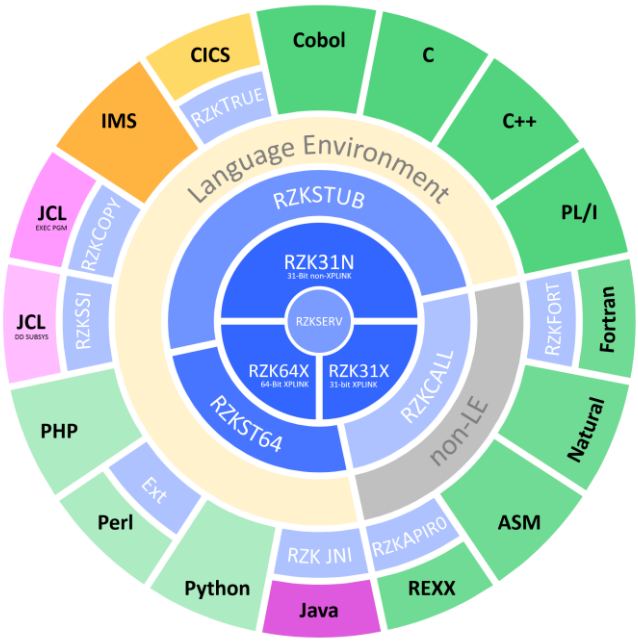
Exchange your data directly between the z/OS environment and your Kafka cluster, wherever this resides.

Direct Kafka Connections for all z/OS Applications.

Features

- Use the standard z/OS CALL API in any z/OS application (any language).
- Administrate topics & groups out of any program with the RzK Cluster Control API.
- Kafka transaction brackets are supported through the RzK Transaction API.
- The zKafka Connector is available for z/OS and for Open Systems.
- Use state-of-the-art security using SSL/TLS encryption and SASL authentication.

- Supported Environments: Batch, TSO, OMVS/USS, CICS, IMS, Natural & RPC, LE & non-LE.
- Languages: Cobol, C, C++, PL/I, Fortran, Natural, ASM, REXX, Java, Python, Perl, PHP.



Comfort

- Complexity hidden in the RzK API:
  Using the RzK is easy, effortless and seamless.
- The RzK manages the complex connections to your Kafka cluster, incl. recovery.
- Use Kafka transaction brackets to commit complex workflows.
- Very small CPU and storage footprint.

```
* COBOL API

CALL "RZKPROD" USING
      BY VALUE    RzkHdl
      BY CONTENT KeyNam BY VALUE KeyLen
      BY CONTENT MsgTxt BY VALUE MsgLen
      BY VALUE    CNV_UTF8
      RETURNING   RzkRet.
```

Flexibility & Variety

- Fine tune performance and compression.
- Use the embedded data conversion between EBCDIC and UTF-8 formats or send binary data.
- Administer program-controlled and remotely your topic and groups.
- Enhance testing with different levels of debug messages and the RzK CLI.
- Produce or consume Kafka messages directly into any dataset with RzKCopy.

```
/* REXX API */

RzkRet = RzkOpen("RzkHdl");
RzkRet = RzkConn(RzkHdl,BrkNam,TopNam);
RzkRet = RzkProd(RzkHdl,KeyNam,MsgTxt,CNV_UTF8);
RzkRet = RzkClse(RzkHdl);

RETURN 0;
```

```
//* Batch Utility
//PRODUCE  EXEC PGM=RZKCOPY
//RZKFILE  DD  DISP=SHR,DSN=TO.PRODUCE
//RZKPARM  DD  *
PRODUCE domain:port topic
KeyNam
UTF-8
/*
```

rvs-Systems GmbH
Alexander-Fleming-Ring 19
D-65428 Rüsselsheim

Email: info@rvs-systems.com
Internet: www.rvs-systems.com