**Kafka client info for task understanding**

Not at all all of this needed. So do not visit every link, unless really into this.

node-rdkafka (NodeJS wrap for the *librdkafka* c/c++ lib)  
  
[https://github.com/edenhill/**librdkafka**](https://github.com/edenhill/librdkafka) Does rd stand for RPM and Debian, or what?

Producer, Consumer, Admin -clients. Written in C/C++,

SSL <https://en.wikipedia.org/wiki/Transport_Layer_Security> SSL 1996 ~~2014/2015~~ => 1999 TSL

SASL <https://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer> , statistics

<https://blizzard.github.io/node-rdkafka/current/>

Zookeeper (Apache)

[https://**zookeeper**.apache.org/](https://zookeeper.apache.org/) "ZooKeeper is a **centralized service for maintaining configuration information**, naming, providing **distributed synchronization**, and providing group services."  
<https://cwiki.apache.org/confluence/display/ZOOKEEPER/ProjectDescription>

Producer  
[https://blizzard.github.io/node-rdkafka/current/**Producer**.html](https://blizzard.github.io/node-rdkafka/current/Producer.html)   
**createWriteStream(conf, topicConf, streamOptions)** - Create a write stream interface for a producer. This stream does not run in object mode. It only takes buffers of data.  
[https://blizzard.github.io/**node-rdkafka**/current/**ProducerStream**.html](https://blizzard.github.io/node-rdkafka/current/ProducerStream.html)   
[https://**nodejs**.org/api/**buffer**.html#static-method-bufferfromobject-offsetorencoding-length](https://nodejs.org/api/buffer.html#static-method-bufferfromobject-offsetorencoding-length)   
[https://**nodejs**.org/api/**stream**.html](https://nodejs.org/api/stream.html)

Consumer  
[https://blizzard.github.io/node-rdkafka/current/**KafkaConsumer**.html](https://blizzard.github.io/node-rdkafka/current/KafkaConsumer.html)  
**new KafkaConsumer(conf, topicConf)** - **conf** object - Key value pairs to configure the consumer  
**topicConf** object - Key value pairs to create a default topic configuratio  
**group.id** - (a consumer group can have many consumers)

**metadata.broker.list** - only for getting metadata (topics, partitions and replicas). The socket connections for sending the actual data will be established based on the broker information returned in the metadata.

**connect(metadataOptions, cb)** - Connect to the broker and receive its metadata. Connects to a broker by establishing the client and fetches its metadata.

**ready** - Ready event. Called when the Client connects successfully  
**data** - Data event. called whenever a message is received.  
**Message** - KafkaConsumer message. This is the representation of a message read from Kafka  
**subscribe(topicsArray)** - Subscribe to an array of topics (synchronously)  
**consume(cb)** - Read messages from Kafka as fast as possible  
[https://blizzard.github.io/node-rdkafka/current/KafkaConsumer.html#~**Message**](https://blizzard.github.io/node-rdkafka/current/KafkaConsumer.html#~Message)Avro specification(Apache) - not needed yet