Consumer threading refactor

Overview

Engineers	Philip Nee Kirk True Lianet Magrans
Approvers	Jason Gustafson guozhang Wang Rajini Sivaram
Objective	Refactor the current `KafkaConsumer` implementation to separate out the API processing in the user thread from the network I/O and heartbeat processing in a dedicated background thread
Status	In progress
Top-level Jira	⚠️ Unable to render Jira issues macro, execution error.

Problem Statement

The current `KafkaConsumer` implementation suffers from a number of issues related to an inconsistent approach to ownership of resources among different threads. Complicated and error-prone synchronization and communication has been applied that makes the code difficult to reason on and has lead to a class of bugs and issues.

Scope

Must have	100% API compatibility with existing `KafkaConsumer` - Test success rate parity with existing `KafkaConsumer` (i.e. no new test failures)
Nice to have	- Simulation testing - Resolution of all the Jira issues with the new-consumer-threading-should-fix label
Not in scope	API-level changes (we don't want to write a KIP)

Contributing

 We'd love to have the community more involved in the project. In order to are some additional onboarding tasks as we start to move information to the Apache Kafka community:

- Join the developer mailing list: https://kafka.apache.org/contact
- Create a Jira account: https://selfserve.apache.org/jira-account.html
- Create a wiki account: https://cwiki.apache.org/confluence/signup.action

Further information

- Consumer rebalance
- Consumer threading refactor design
- Consumer threading refactor release plan
- Consumer threading refactor test plan