Consumer threading refactor test plan

We will first write and use the existing unit tests and integration tests.

We need to ensure the timing of the critical events are happening in the correct sequence. For example: We need to first discover the coordinator, second, commit the offset while pausing the partition for being fetched, revoke the partition, and then continue onto rest of the rebalance process.

We will also ensure the heartbeat interval, and poll interval are respected.

We also need to ensure there's no event lost, and the event should happen in the correct sequence.

Please review https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis and try to assess potential failures.

User Adoption

The refactor should have (almost) no regression or breaking changes upon merge. So user should be able to continue using the new client.

Release Plan

- Support code will exist in parallel from the current code. The support code are:
  - Background thread
  - A new coordinator implementation, AsyncConsumerCoordinator, for example.
  - Events and event executors
- We will create a new KafkaConsumer class first, then have it override the existing one once reach stability