KIP-566: Add rebalance callbacks to ConsumerInterceptor

Status

Current state: *Under Discussion*

Discussion thread: *here*

JIRA: TODO

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

KIP-42 added the `ConsumerInterceptor` interface to support the introspection, and potentially, modification of messages before they are consumed. Many uses of message interception such as metrics collection/monitoring (which was a primary motivator for KIP-42) require the interceptor to keep some sort of per-partition state. Unfortunately, the `ConsumerInterceptor` interface does not provide a way for the interceptor to be made aware of consumer rebalances, making maintenance of such state difficult. To remedy this, we propose having `ConsumerInterceptor` extend `ConsumerRebalanceListener`.

Public Interfaces

We propose augmenting `ConsumerInterceptor` as follows:

org.apache.kafka.clients.consumer.ConsumerInterceptor

```
public interface ConsumerInterceptor<K, V> extends Configurable, ConsumerRebalanceListener, AutoCloseable {

        // Existing methods elided

    @Override
    default void onPartitionsRevoked(Collection<TopicPartition> partitions) { }

    @Override
    default void onPartitionsAssigned(Collection<TopicPartition> partitions) { }
}
```

Note the methods from `ConsumerRebalanceListener` have no-op `default` implementations for backwards compatibility. Note also there is an additional method on `ConsumerRebalanceListener` , `onPartitionsLost()` , which is already a default method.

Proposed Changes

We propose modifying `ConsumerInterceptor` as detailed above, and wiring these up to be called from `ConsumerCoordinator`, similarly to existing interceptor methods.

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
    - None. Existing `ConsumerInterceptor` implementations are still fully compatible due to additional methods having default implementations.
- *If we are changing behavior how will we phase out the older behavior?*
    - No need. Empty default methods are equivalent to not receiving callbacks at all.
- *If we need special migration tools, describe them here.*
    - N/A

- *When will we remove the existing behavior?*
    - N/A

Rejected Alternatives

We could not add these methods and try to infer partition assignment based on which partitions are being committed, which is obtainable through the existing `onCommit` method. But there is no guarantee when or even if consumers commit offsets, and even if they do, it's not reasonable to assume that because a partition was not committed it must have been revoked.

We could not have `ConsumerInterceptor` extend the existing `ConsumerRebalanceListener` interface and instead just add similarly named methods. This allows the interfaces to diverge in the future and it's not clear we would ever want rebalance listeners to have additional callbacks but not also provide them to interceptors.