Assignment 1
Distributed Data Management

Team NoCommentCode

1 Message Passing

We pass the `LargeMessage` via AkkaStreams. The slices of the serialized object are sent to a sink with a backpressure layer.

```
Source<byte[], NotUsed> source = Source.from(messageContents);
source.buffer(BUFFER_SIZE, OverflowStrategy.backpressure())
    .runWith(sink, ActorMaterializer.create(this.context().system()));
```

Listing 1: Stream Setup

2 Stream Sink

We set up our stream to use the `LargeMessageProxy` Actor as sink. When successfully streamed, we end our stream with a `StreamCompleted` message, which holds the references for sender and receiver.

```
Sink<byte[], NotUsed> sink =
    Sink.actorRef(
        receiverProxy,
        new StreamCompleted(this.sender(), receiver));
```

Listing 2: Sink

3 Serialization

We serialize the content of the `LargeMessage` with *Kryo* using a wrapper class and split the bytes to chunks with a size of maximum 1 MB. That set of chunks is then passed to the source.

4 Deserialization

We join all the received chunks when receiving the `StreamCompleted` message. We deserialize that byte encoded object with *Kryo* and pass it to the receiver.