# INGENIOUS DESIGN DECISIONS

IN APACHE KAFKA

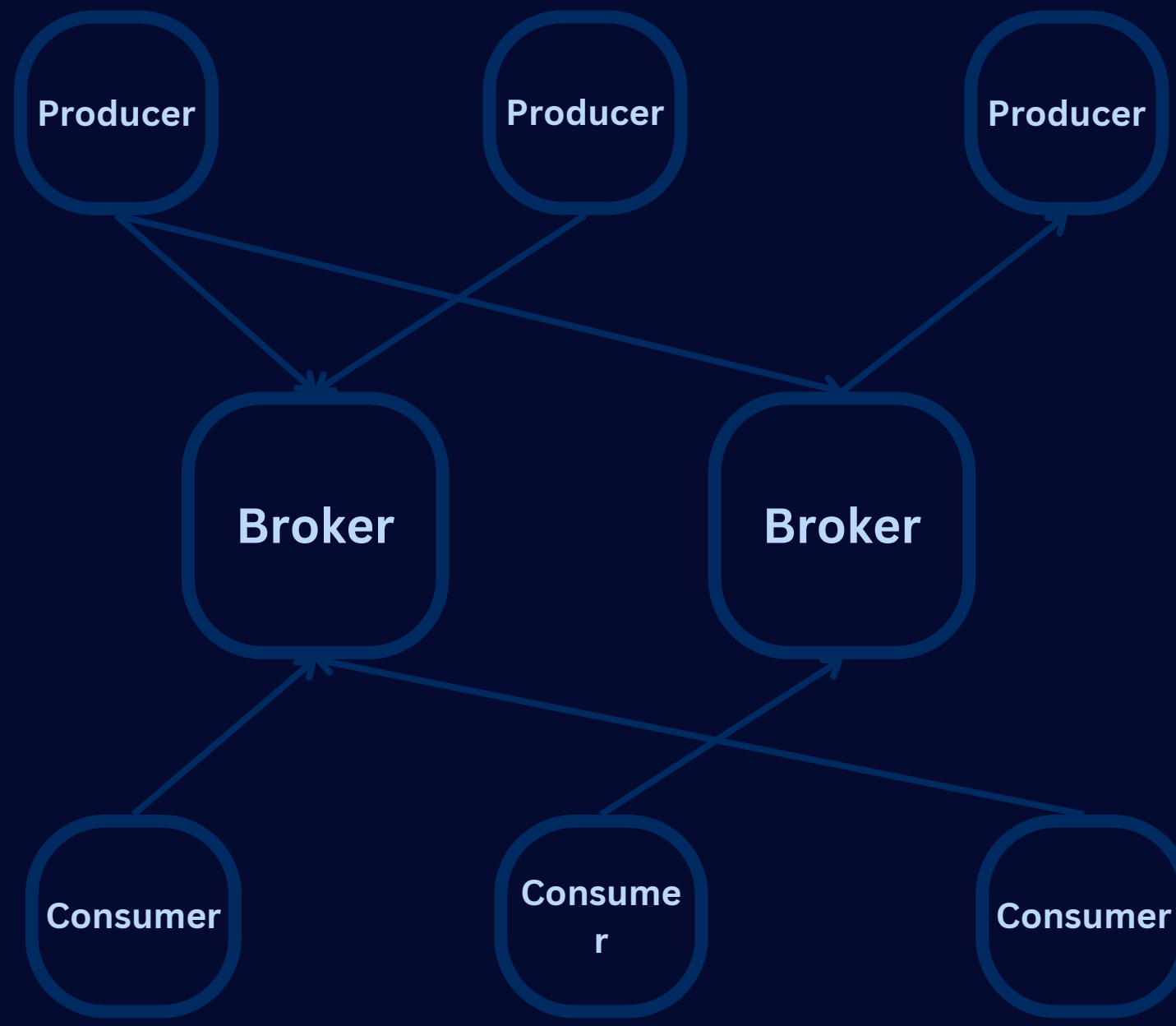
⬇️ Pull-based approach for consuming message

🗄️ Using logical offsets instead of message ids

📤 Using the sendFile API to consume messages

💰 Leveraging system's page cache

# ⬇ Pull-based approach




In a pull-based approach, the consumers decide how they want to consume messages. The consumers send a “pull request” to a broker with the offset from which to start sending messages along with the number of bytes to send.

# ⬇ Pull-based approach

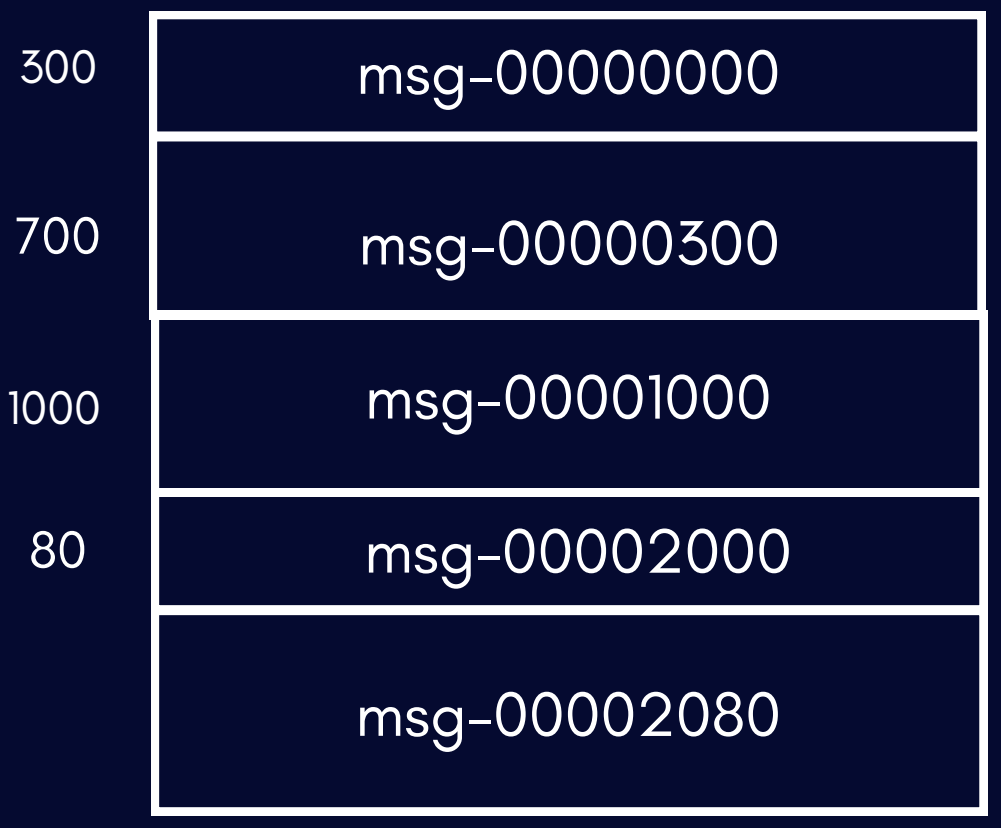
This pull-based approach has some benefits

The consumer is never flooded with messages that it cannot process.

The broker is stateless so even if it restarts, the consumers can immediately continue consuming messages

The consumer can “rewind” back to an old offset and re-consume messages.

# 🗄️ Using logical offsets instead of message ids




Kafka doesn't assign a message id to incoming messages. To get to the next message, the size of the message can be added to the current offset.

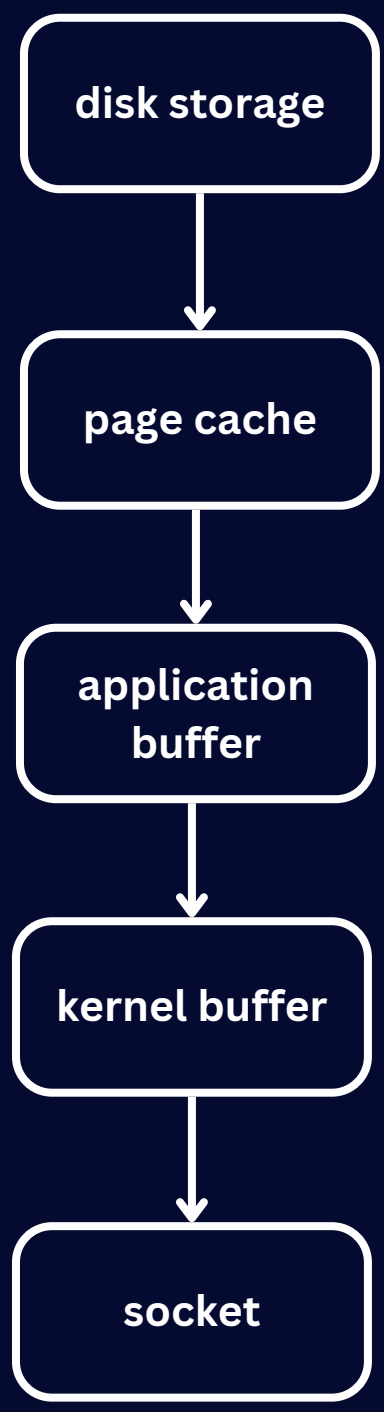
For example, to get to the 2nd message we can add 300 bytes to the offset of the 1st message. To get to the 3rd message we can add 700 bytes to the offset of the 2nd message, so on and so forth.

With this approach, the overhead of maintaining an index-to-address mapping system can be avoided.

# 📤 Using the sendFile API

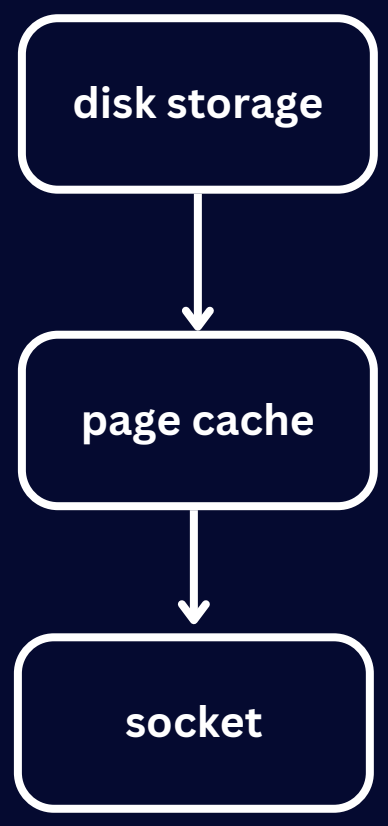
without sendFile

(4 copies, 2 system calls)




with sendFile

(2 copies, 1 system call)




Sending data from the broker to the consumer is optimized using the sendFile API available on Linux and other Unix-like operating systems.

# 💰 Leveraging system's page cache

Kafka doesn't cache any messages. Instead, it relies on the file system-level page cache. This way it avoids double buffering.

This means it has a small memory footprint. This makes it run efficiently in VM-based languages.

This has one advantage that even if the broker process restarts, it still retains warm cache.

Another advantage is that since the consumer lags behind the producer by a small amount, the os level caching mechanisms are very effective.

# Have questions?

Let’s connect.