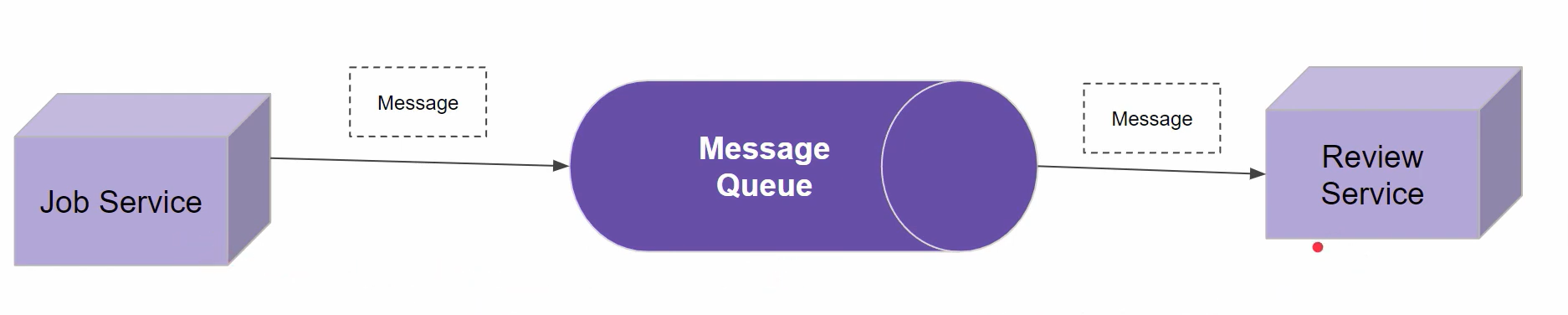
**MESSAGE QUEUE**

1. **Message Queue là gì?**

* A Message Queue is a form of asynchronous, decoupled communication between software components (process, thread or service), enabling reliable message delivery without requiring sender and receiver to interact simultaneously.



1. **Tại sao lại cần Message Queue?**

* Here are the core points:
  + **Asynchronous Communication:** Producers send messages to the queue and continue processing immediately. Consumers retrieve and process messages later, at their own pace. This decouples sender and receiver lifecycles and smooths out spikes in load.
  + **Decoupling of Components:** The sender (producer) and receiver (consumer) only need to agree on the queue name and message format—they don’t need to know about each other’s implementation, uptime, or location.
  + **Reliability & Durability:** Queues are managed by a broker (e.g., RabbitMQ, Apache Kafka, Amazon SQS) which can persist messages to disk, replicate them across nodes, and ensure **at‑least‑once** or **exactly‑once** delivery semantics, even in the face of failures.
  + **Load Leveling:** By buffering messages, a queue can absorb bursts of incoming work, allowing consumers to process at a steady rate without being overwhelmed. This improves overall system resilience and responsiveness.

1. **Use case**

* **Some Use Cases:**
  + **Microservices Communication:** Passing events or commands between services without tight coupling.
  + **Task Offloading:** Background jobs such as email sending, report generation, or video processing.
  + **Event Streaming:** Real‑time analytics pipelines (e.g., using Kafka for high‑throughput event streaming).