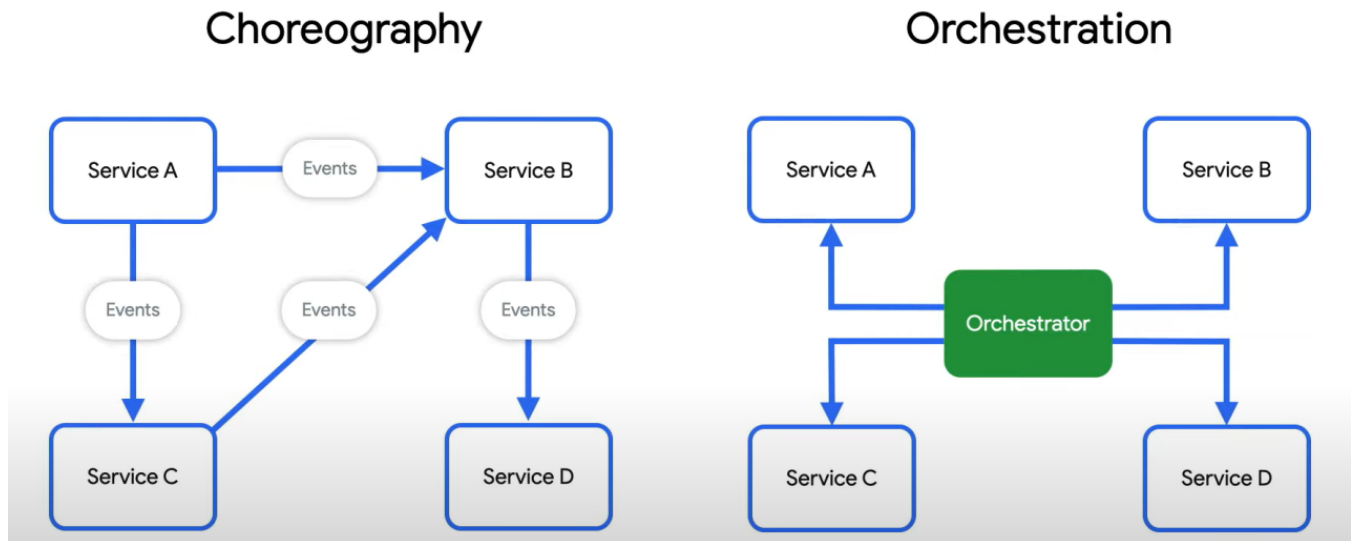# Orchestration vs. Choreography in Saga Pattern

Summary of the comparison between orchestration and choreography in Saga patterns, focusing on their characteristics, use cases, and suitability for finance systems.

Choreography

Service A → Events → Service B

Service A → Events
Events → Service B
Service B → Events → Service D
Events → Service C

Service C   Service D

Orchestration

Service A   Service B

Orchestrator → Service A, Service B, Service C, Service D

Service C   Service D

Key Characteristics

- **Choreography**

  - Simple & asynchronous
  - Loose coupling; decentralized
  - Harder to trace & debug
  - No single point of failure
  - Suitable for scalable, autonomous services

- **Orchestration**

  - Complex & synchronous
  - Centralized control with a single orchestrator
  - Easier to understand; single point of failure
  - Better for managing complex workflows
  - Can delegate control to a service or one of the involved services

When to Use

- **Orchestration**

  - Need centralized control
  - Clear workflow visibility
  - Complex, multi-step processes
  - Precise error handling & rollback
  - Example (Finance): Multi-step payment processing with rollback

- **Choreography**

  - Autonomy & scalability
  - Looser coupling, flexible workflows
  - Ideal for simpler, high-volume systems
  - Resilient against failures
  - More challenging to manage & debug

Suitability for Finance Systems with Synchronous Communication

- **Orchestration** is generally preferred due to:

  - Precise control
  - Transactional consistency
  - Need for auditability & rollback

- **Choreography** may introduce complexities and unpredictable states, making it less ideal unless scalability and loose coupling are prioritized.