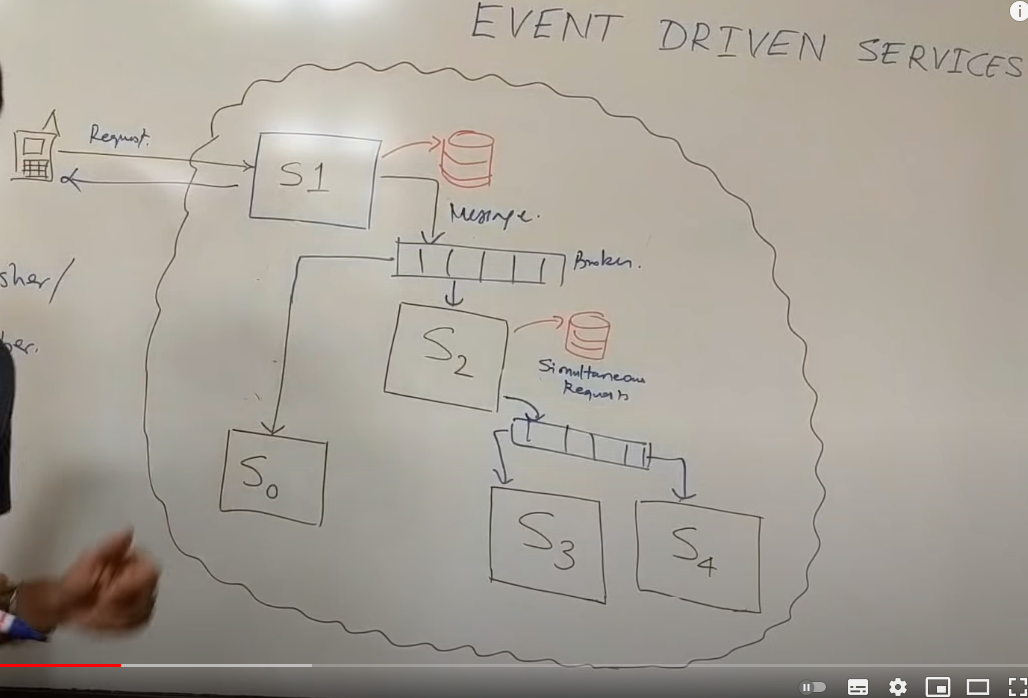
Link video: <https://www.youtube.com/watch?v=FMhbR_kQeHw&t=145s>



Khi microservice S2 bị down . ngay khi S2 hoạt động trở lại thì message broker sẽ gửi lại message cho S2

Microservices benefit from loose data coupling, which is provided by a publish subscribe model. In this model, events are produced by a publishing service and consumed by downstream services. Designing the micro service interactions involves event handling and consistency checks. We look into a pub-sub architecture to evaluate it's advantages and disadvantages compared to a request response architecture. This type of architecture relies on message queues to ensure event passing. An example would be rabbitMQ or Kafka. The architecture is common in real life scenarios and interviews. If there is no strong consistency guarantee to made for transactions, an event model is good to use in microservices. Here are the main advantages:

1) Decouples a system's services.

2) Easily add subscribers and publishers without informing the other.

3) Converts multiple points of failure to single point of failure.

4) Interaction logic can be moved to services/ message broker.

Disadvantages:

1) An extra layer of interaction slows services

2) Cannot be used in systems requiring strong consistency of data

3) Additional cost to team for redesigning, learning and maintaining the message queues.

Financial system

