

Designing the desired application

MOSTAFA RASTGAR

JUNE 2019

Agenda

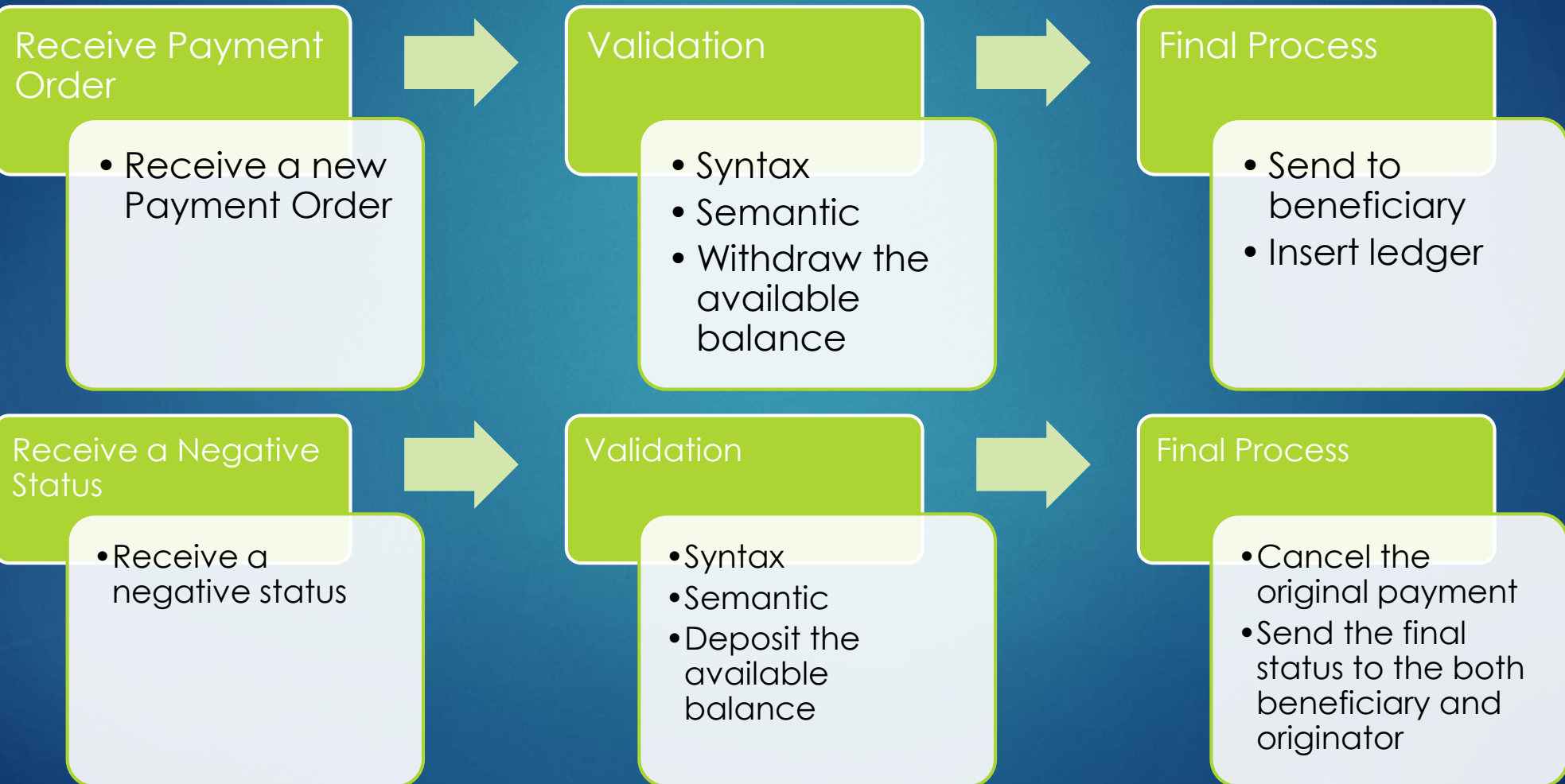
2

- ▶ Review the desired application
- ▶ Decomposition Patterns
- ▶ Proposing a suitable architecture
- ▶ Designing application's microservices

Review the desired application

3

Payment Project



Decomposition Patterns

► Decompose by Business Capability

- Microservices is all about making services loosely coupled, applying the single responsibility principle. However, breaking an application into smaller pieces has to be done logically. How do we decompose an application into small services?
- One strategy is to decompose by business capability. A business capability is something that a business does in order to generate value. The set of capabilities for a given business depend on the type of business. For example, the capabilities of an insurance company typically include sales, marketing, underwriting, claims processing, billing, compliance, etc. Each business capability can be thought of as a service, except it's business-oriented rather than technical.

Decomposition Patterns

5

- ▶ Decompose by Subdomain
 - ▶ Decomposing an application using business capabilities might be a good start, but you will come across so-called "God Classes" which will not be easy to decompose. These classes will be common among multiple services. For example, the Order class will be used in Order Management, Order Taking, Order Delivery, etc. How do we decompose them?
 - ▶ For the "God Classes" issue, DDD (Domain-Driven Design) comes to the rescue. It uses subdomains and bounded context concepts to solve this problem. DDD breaks the whole domain model created for the enterprise into subdomains. Each subdomain will have a model, and the scope of that model will be called the bounded context. Each microservice will be developed around the bounded context.
 - ▶ **Note:** Identifying subdomains is not an easy task. It requires an understanding of the business. Like business capabilities, subdomains are identified by analyzing the business and its organizational structure and identifying the different areas of expertise.

Decomposition Patterns

6

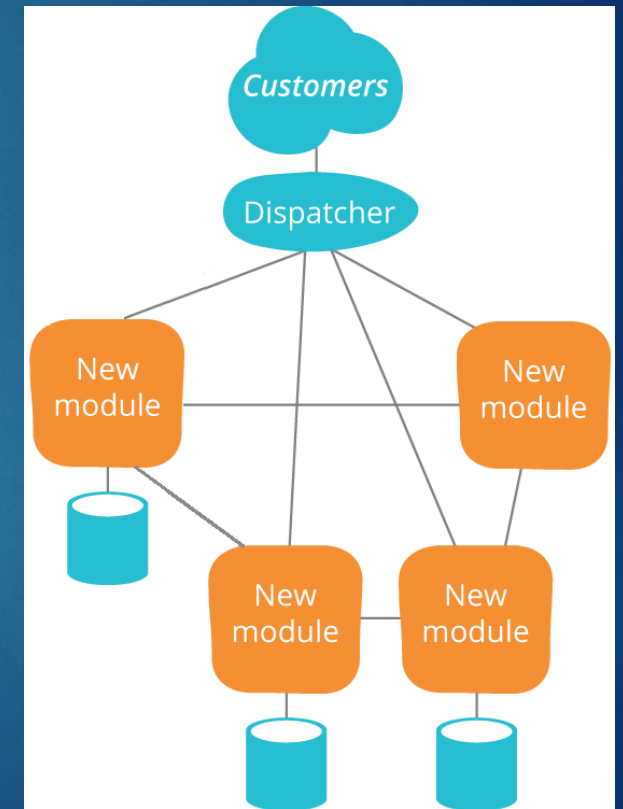
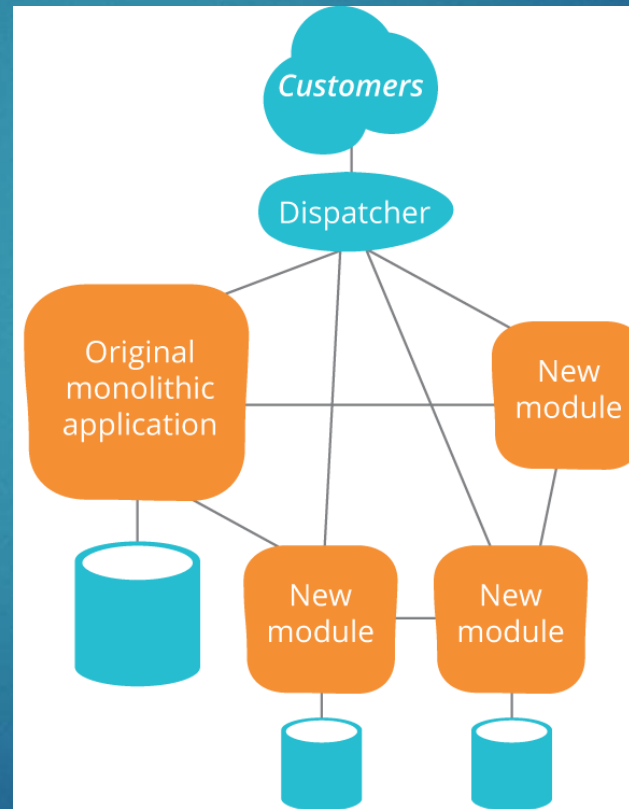
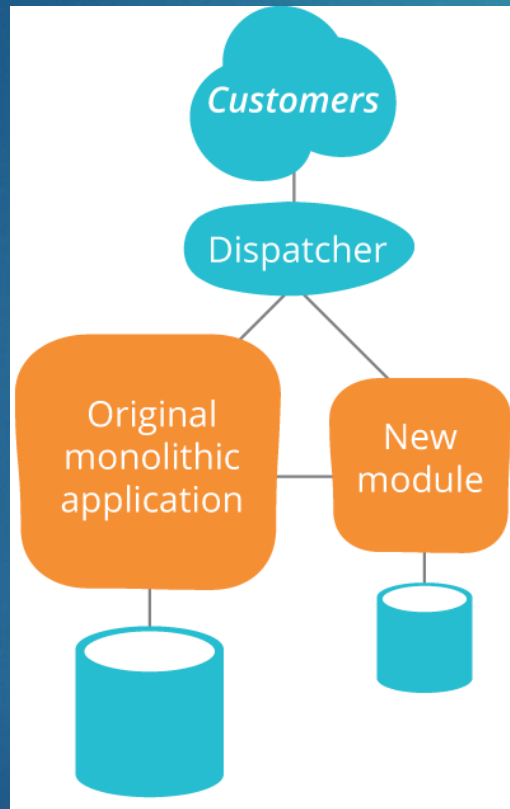
► Strangler Pattern

- So far, the design patterns we talked about were decomposing applications for greenfield, but 80% of the work we do is with brownfield applications, which are big, monolithic applications. Applying all the above design patterns to them will be difficult because breaking them into smaller pieces at the same time it's being used live is a big task.
- The Strangler pattern comes to the rescue. The Strangler pattern is based on an analogy to a vine that strangles a tree that it's wrapped around.

Decomposition Patterns

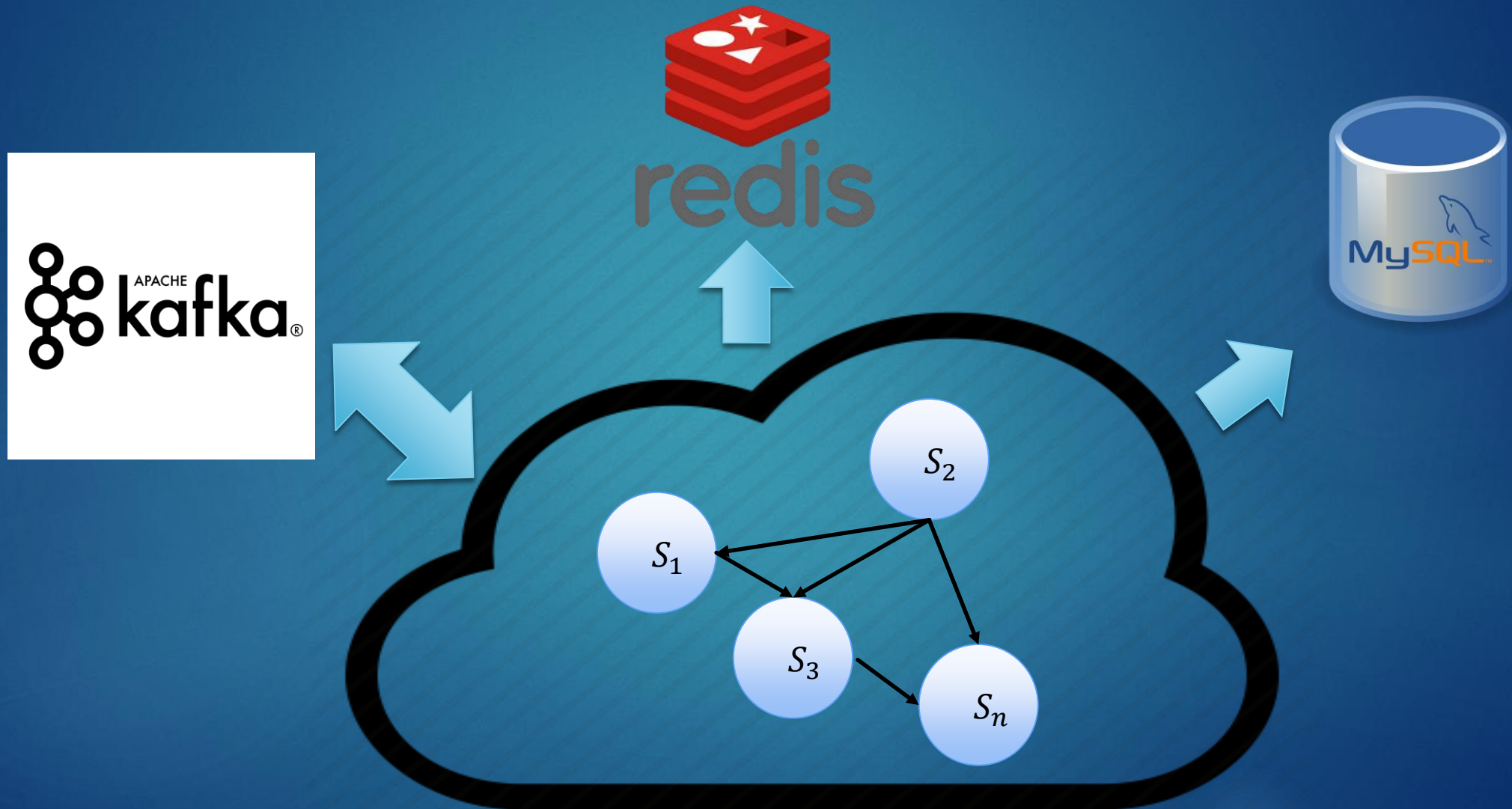
7

► Strangler Pattern schematic diagram



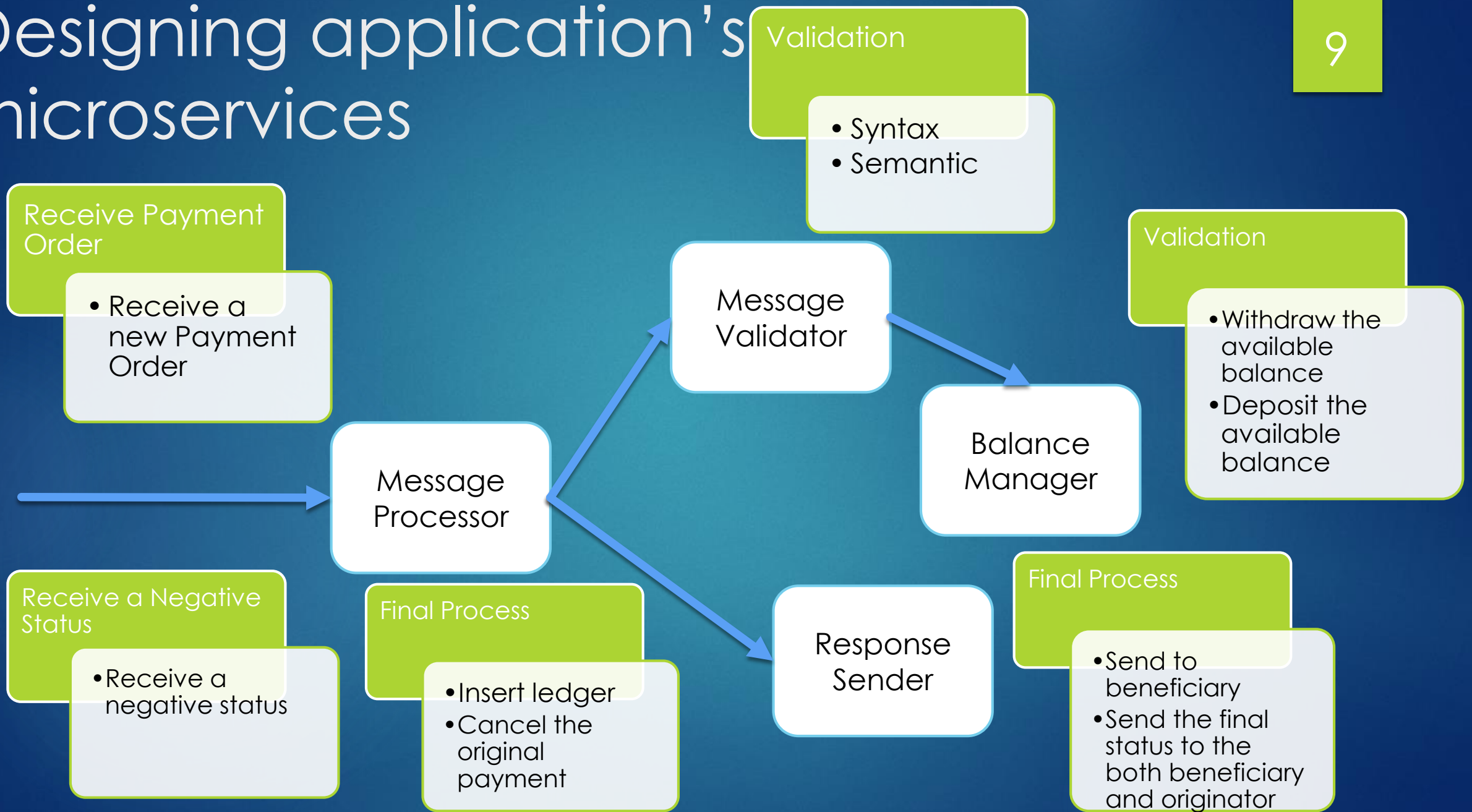
Proposing a suitable architecture

8



Designing application's microservices

9



Designing application's microservices

10

- ▶ Kafka topics:
 - ▶ balance-check
 - ▶ 32 partitions
 - ▶ Balance-check-result
 - ▶ 32 partitions
 - ▶ bank1-out
 - ▶ SLA-based number of partitions
 - ▶ ...
 - ▶ bankn-out
 - ▶ SLA-based number of partitions

