MICROSERVICES

CONTENTS:

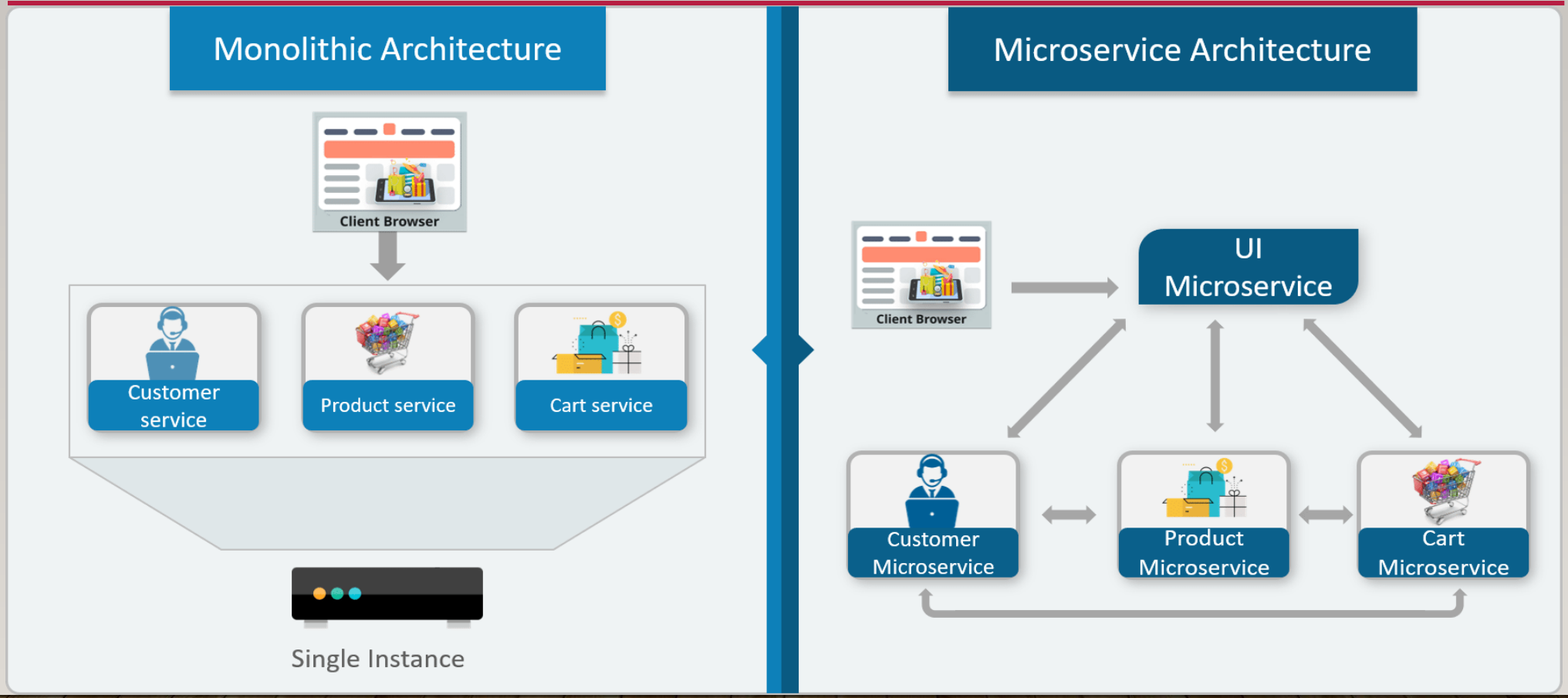
- Before Microservices?
- What are Microservices?
- Why Microservices?
- Types of microservices.
- Microservices architecture.

BEFORE MICROSERVICES?

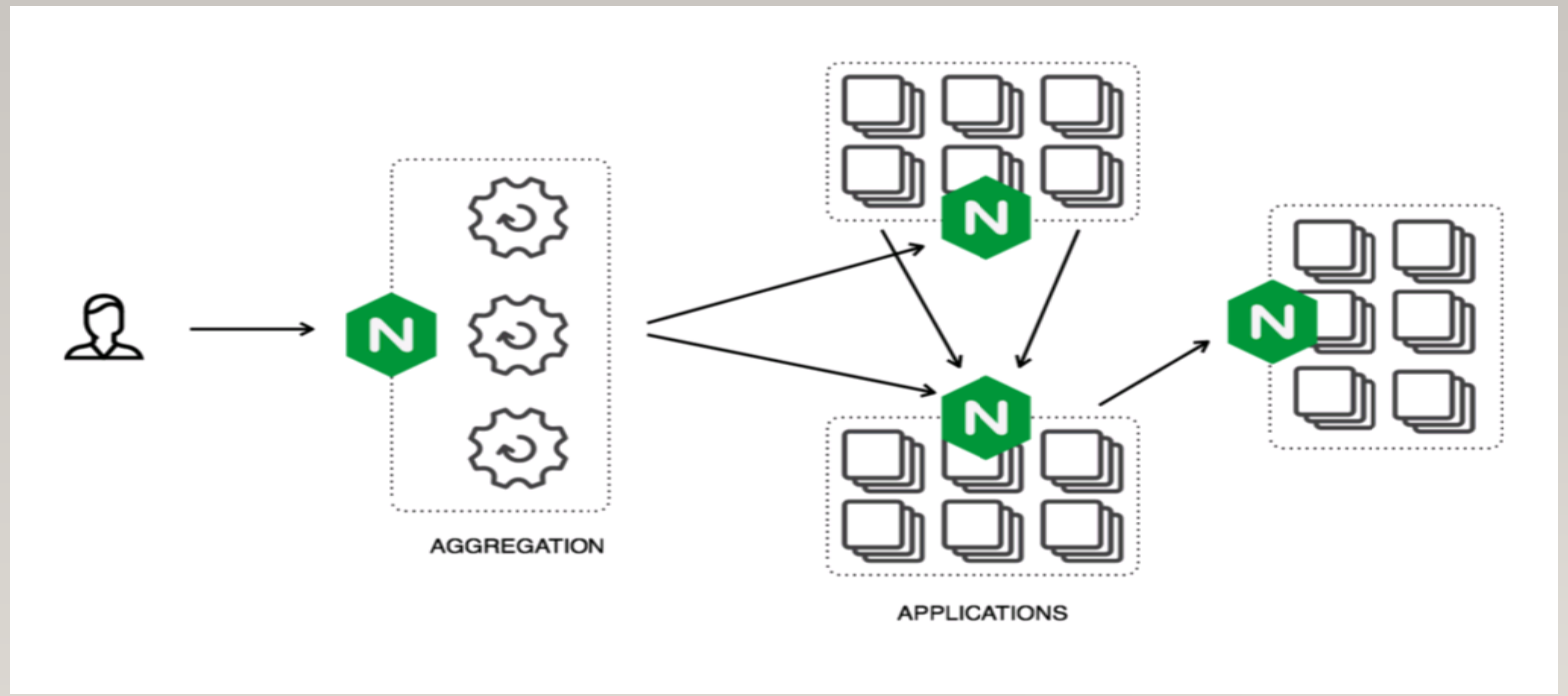
- Prior to microservices, we had the monolithic architecture and service-oriented architecture that reigned over the world of enterprise development.
- **Benefits of Monolithic Architecture:**
  1. Simple to develop.
  2. Simple to test. For example you can implement end-to-end testing by simply launching the application and testing the UI with Selenium.
  3. Simple to deploy. You just must copy the packaged application to a server.
  4. Simple to scale horizontally by running multiple copies behind a load balancer.

- **Drawbacks of Monolithic Architecture:**
  1. This simple approach has a limitation in size and complexity.
  2. Application is too large and complex to fully understand and made changes fast and correctly.
  3. The size of the application can slow down the start-up time.
  4. You must redeploy the entire application on each update.
  5. Impact of a change is usually not very well understood which leads to do extensive manual testing.
  6. Monolithic applications can also be difficult to scale when different modules have conflicting resource requirements.
  7. Another problem with monolithic applications is reliability. Bug in any module (e.g. memory leak) can potentially bring down the entire process. Moreover, since all instances of the application are identical, that bug will impact the availability of the entire application.
  8. Monolithic applications has a barrier to adopting new technologies. Since changes in frameworks or languages will affect an entire application it is extremely expensive in both time and cost.

WHAT ARE MICROSERVICES?



- Microservices has the power to bring together those old monolithic applications in one seamless efficient application that is both secure and flexible.
- Each microservice is a distinct unit within the software development project, with its own codebase, infrastructure, and database.



WHY MICROSERVICES?

- **Benefits of Microservices Architecture:**
  1. It tackles the problem of complexity by decomposing application into a set of manageable services which are much faster to develop, and much easier to understand and maintain.
  2. It enables each service to be developed independently by a team that is focused on that service.
  3. It reduces barrier of adopting new technologies since the developers are free to choose whatever technologies make sense for their service and not bounded to the choices made at the start of the project.
  4. Microservice architecture enables each microservice to be deployed independently. As a result, it makes continuous deployment possible for complex applications.
  5. Microservice architecture enables each microservice to be deployed independently. As a result, it makes continuous deployment possible for complex applications.

- **Drawbacks of Microservices Architecture:**
  1. Inter-process communication mechanism leading to fallacies in Distributed computing.
  2. Partitioned database architecture creating more challenges to the developers.
  3. Carefully plan and coordinate the rollout of changes to each of the services.
  4. Each runtime instance need to be configured, deployed, scaled, and monitored. In addition, you will also need to implement a service discovery mechanism.

TYPES OF MICROSERVICES

1. Stateless:
   - Does not maintain the session state between requests.
   - If any service within the application is removed, it would not affect the processing logic for that service, and it would become a part of the overall application.
   - recommended If using a distributed system.
2. Stateful:
   - Maintain a service request state by storing session information in the code.
   - Use if there are cases where session information needs to be stored.

MICROSERVICES ARCHITECTURE