# **Application architecture**

## class diagramA close up of a map Description automatically generated

## Sequence diagram for placing an Order

A screenshot of a cell phone

Description automatically generated

# How to store state

## Client/ Browser

### Advantages

* Applications that store state in the client such as React and Angular are highly responsive
* It’s easy to scale applications that store state in the client

### Disadvantages

* It’s easy to lose information in case the browser/ client crashes
* It cannot be used as a reliable source to store data for long periods of time

## Server

### Advantages

* Information is secure and it’s not easy to lose the information.

### Disadvantages

* Its less responsive since state has to be updated in the server all the time
* Its less responsive compared to applications that’s store state in the client/ browser

## Database

### Advantages

* Can be used as a reliable source to store data required after a long period of time.
* It’s easy to do scaling compared to when we store state in the server
* There is reliability on the information when state is stored in the server

### Disadvantages

* High latency and thus low performance
* Its is less responsive.

# Integration possibilities

## RMI - Remote Method Calls

RMI allows an object residing in one java virtual machine to access/invoke an object running on another java virtual machine.

### Advantages

* Server-side implementation can be changed without knowledge of client side

### Disadvantages

* RMI requires that serialization and deserialization of objects which increases overhead costs
* RMI works for only Java to java object calls

### Usage

* Not good to be used

## Messaging (JMS – Java messaging services)

The communication is in form of a message between different services on the network.

### Advantages

* Asynchronous communication: JMS uses asynchronous communication by using a queue for the messages and thus this can lead to high throughput
* High reliability: Messages sent will always be delivered to the consumer without loss of data
* Loose coupling between systems instead of the systems sharing a common database
* JMS API is very easy
* JMS does efficient load balancing in case of very many messages

### Disadvantages

* There is a need for a middle ware for the messaging service
* The middle ware can be a single point of failure

### Usage

* Can be applied in cases where we need the communication to be asynchronous

## SOAP- Simple Object Access Protocol

### Advantages

* Has a standard HTTP protocol that makes it easier to communicate across firewalls and proxies.
* SOAP is highly secure and highly standardized

### Disadvantages

* Has a poor performance compared to REST.
* It is more complex.
* SOAP is less flexible

### Usage

* Where we need more security

## REST - Representational State Transfer

### Advantages

* Supports greater variety of data formats such as XML, JSON, HTTP among others
* Gives better support for browser clients
* Faster and uses less bandwidth

### Disadvantages

* Less secure

### Usage

* Everywhere

## Serialized objects over HTTP

### Advantages

* Its easy
* Supports secure java computing

### Disadvantages

* Works for only Java to java
* It has large overheads and cannot be used for very large objects

### Usage

* Used for java to Java applications

## Database integration

### Advantages

* Its easy

### Disadvantages

* High coupling

### Usage

* Not advisable

## File based integration

### Advantages

* Its easy
* Its reliable

### Disadvantages

* Scalability is hard
* High coupling

### Usage

* Transferring large files

# Distributed systems

## Advantages of distributed systems

* Improved performance since each system can be run with its own independent resources thus improving the performance of the overall system
* Easy scaling: Since the processes are distributed, it’s easy to scale each process depending on the need and requirements of that particular process
* Transparency: Middleware improves transparency because processes communicating to each other need not to be aware of each other
* It is easy to use shared resources while using distributed systems.

## Disadvantages

* Its more complex to plan and implement distributed systems
* Distributed services are more expensive due to the fact that each process needs its own resources in terms of hardware
* Distributed systems present security threats.
* Fault isolation in distributed systems is also harder.
* There are a lot of network calls in distributed systems and this can also impact the performance of distributed systems.