# Monolithic vs. Service Based Architecture

Made by:

Frederik Andersen

Janus Stivang

Julius Krüger

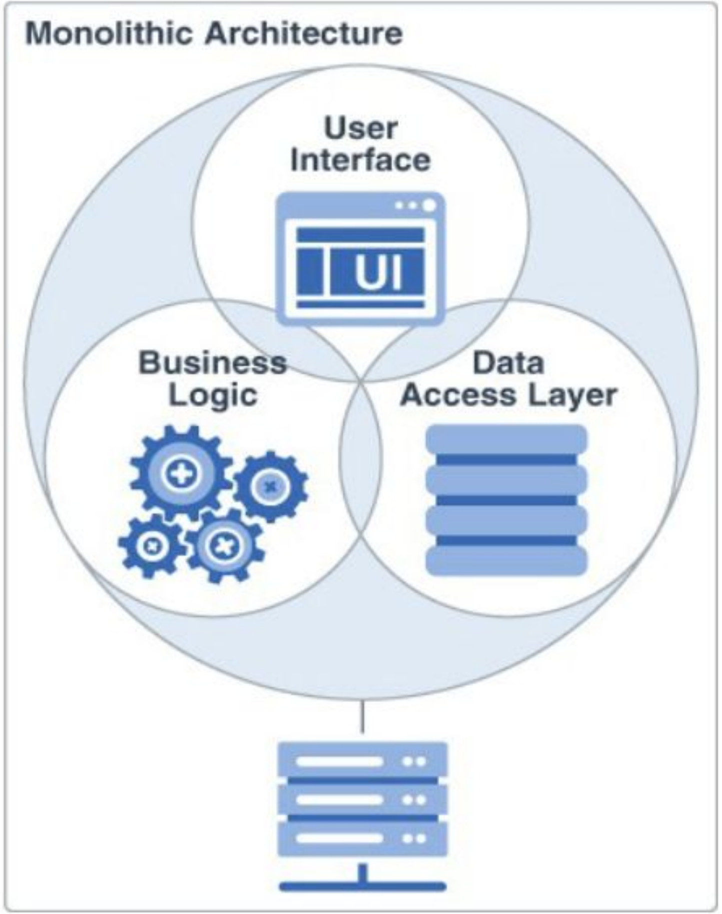
# Definition - Monolithic Architecture

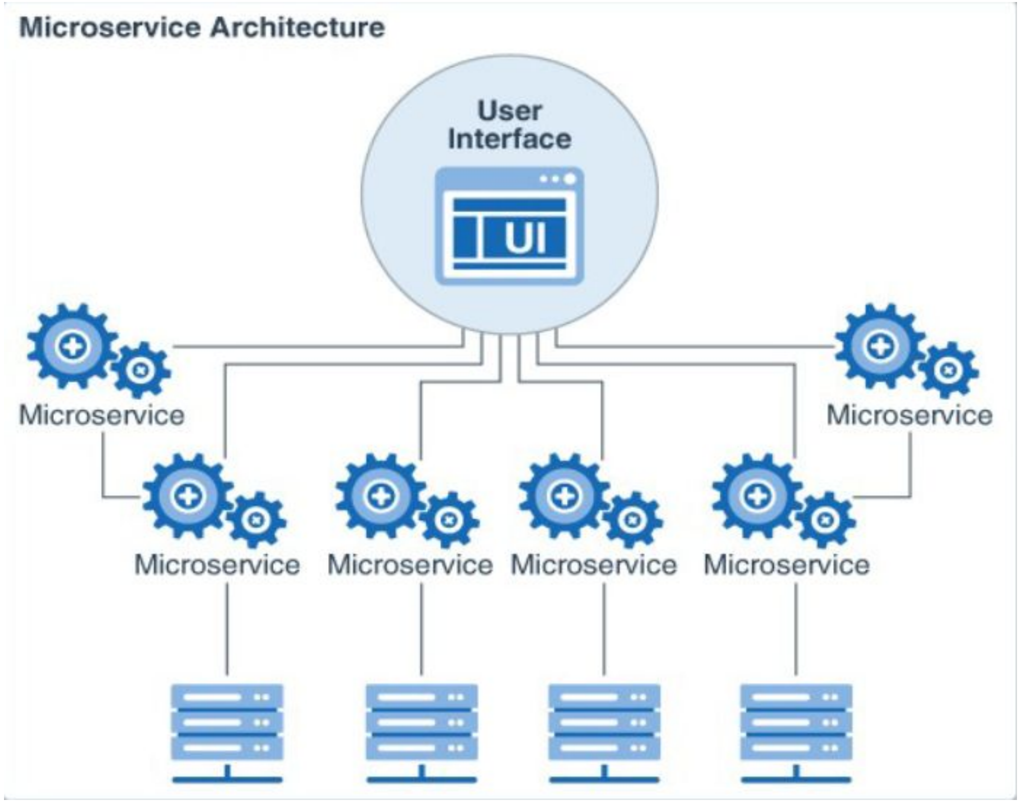
A unified codebase

- Example: An application where the back end, front end, and database are all within the same codebase.

Deployment

- Applications with a monolithic architecture are deployed to the same web server.


Monolithic Architecture
User Interface
UI
Business Logic
Data Access Layer


Microservice Architecture
User Interface
UI
Microservice
Microservice
Microservice
Microservice
Microservice
Microservice
Microservice

# Pros - Monolithic Architecture

- Simpler to develop and deploy - more quickly create, test, and launch applications.

- Performance - Components in a monolith typically share memory which is faster.

- Simplicity in onboarding new team members. - The source code is located in one place.

- The main function of the application is to be profitable. - Proof of Concept to verify the application in the real world.

# Cons - Monolithic Architecture

- fault tolerance. - if a tiny bit breaks the entire project goes down.

- Code ownership cannot be used. - *Flight Service*, another — for *Billing Service*. One team can affect another.

- Slow speed of development. CI/CD- monolith that contains a lot of services - tests that are executed for each Pull Request.

# Definition - Service Based Architecture

Separation of concern

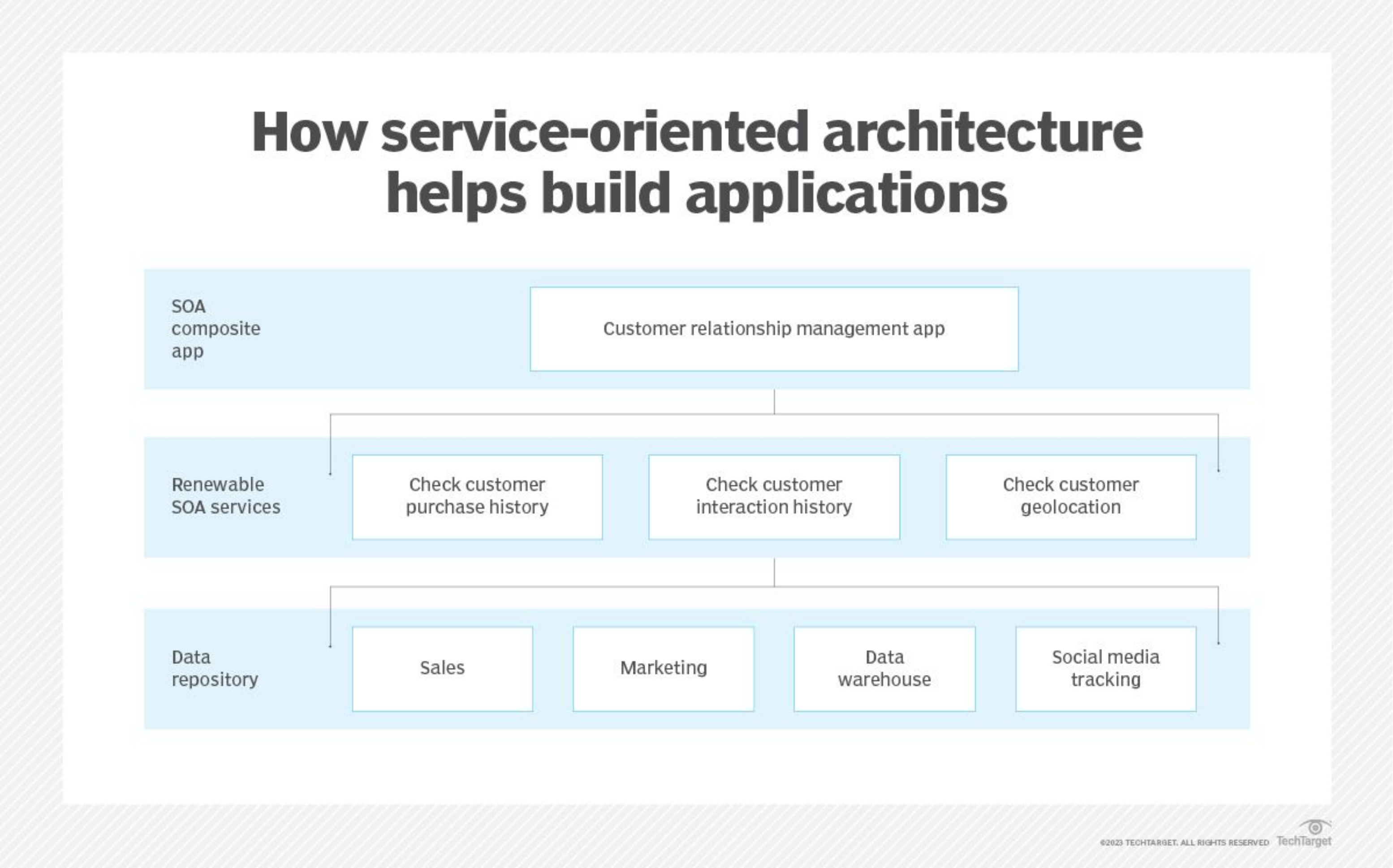
- Distributions of functionalities and components to separate, deployable services that can communicate using APIs
- These services can then be used in a composite application or another service

Reusability

- Each service can be reused in other applications or services.

Self contained

- Loose coupling
- Independent from other services



# How service-oriented architecture helps build applications

SOA composite app

Customer relationship management app

Renewable SOA services

Check customer purchase history

Check customer interaction history

Check customer geolocation

Data repository

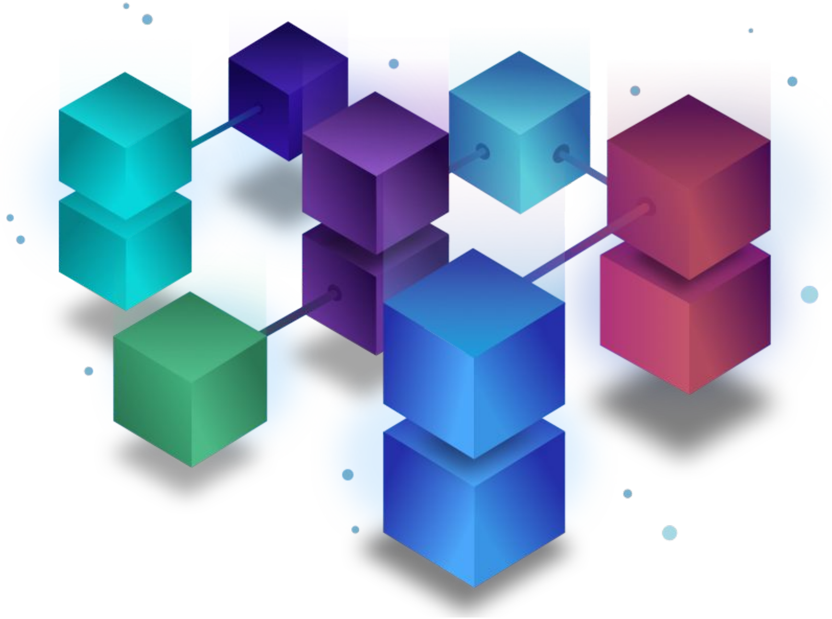
Sales

Marketing

Data warehouse

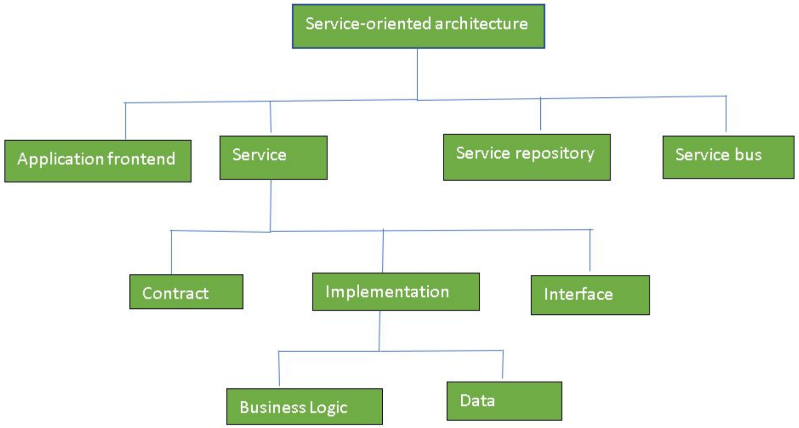
Social media tracking

# Components of Service Oriented Architecture



# 1. Service

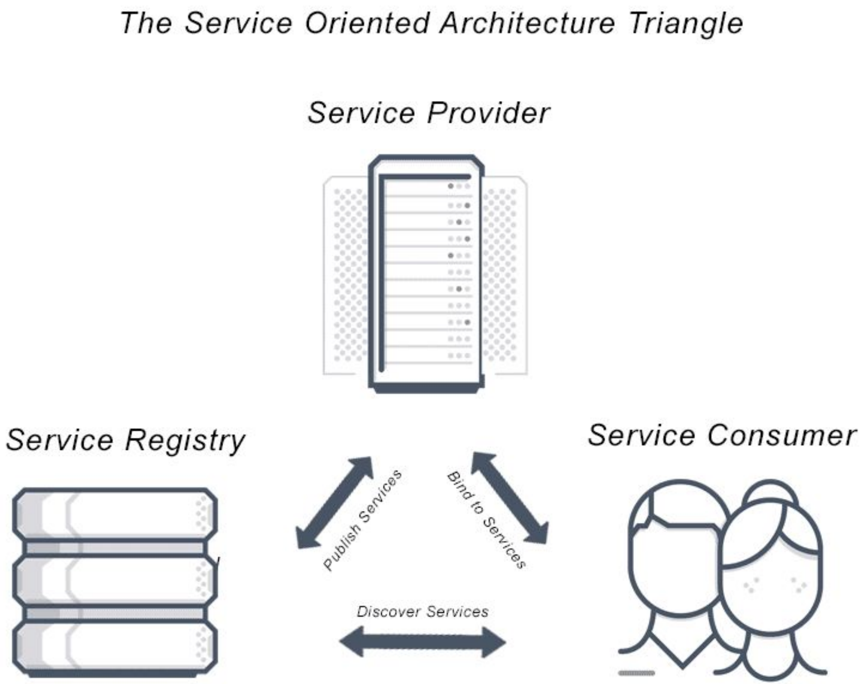
- Service Implementation (the code)
- Service Contract (describes parameters such as cost, and quality)
- Service Interface (communication)



# 2. Service Provider

- Creates and provides the services to the service registry

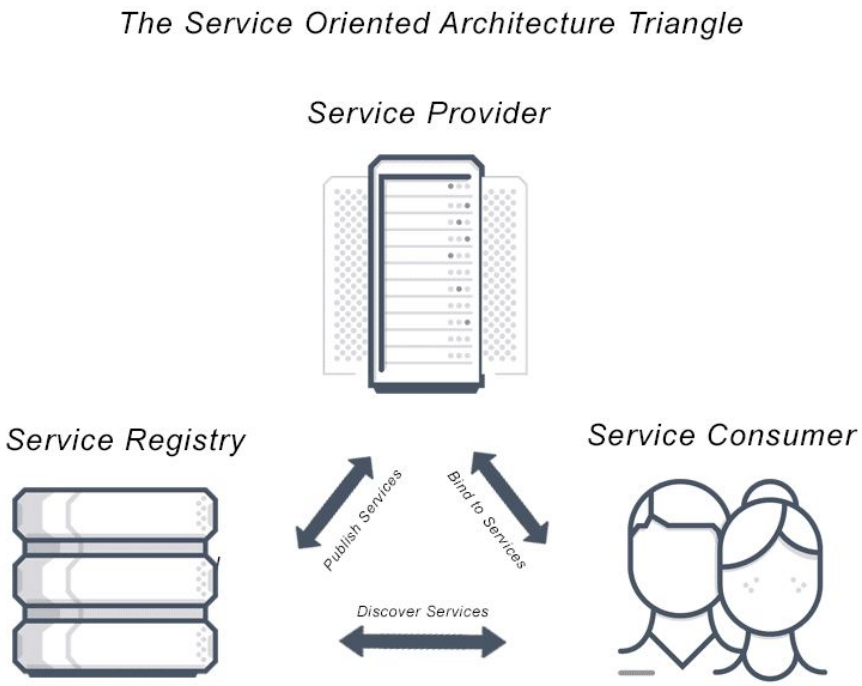
- Selfmade or third-party services



# 3. Service Consumer

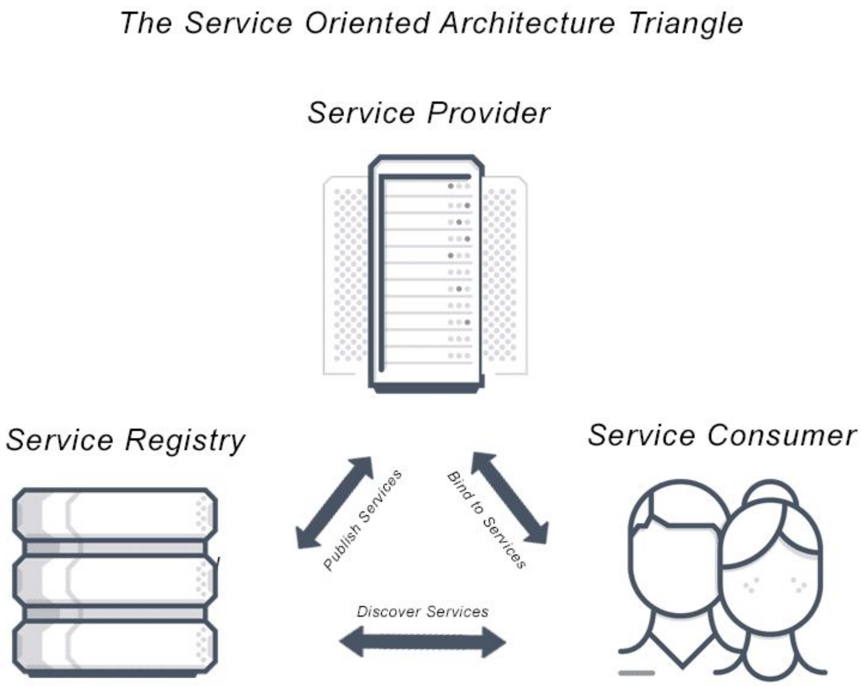
- Interacts and uses the services provided by the service provider

- Can be applications, other services, etc..



# 4. Service Registry

- Also known as the service repository
- Contains service descriptions and relevant information about a collection of services
- Service Documentation



# Pros - Service Based Architecture

Reusable code

Ease of maintenance

- Each service can be easily modified as standalone code not affecting other services (unlike monolithic)

Loose coupled code is more testable

Scalability (services run on multiple servers)

# Cons - Service Based Architecture

Implementation of SOA requires a large initial investment

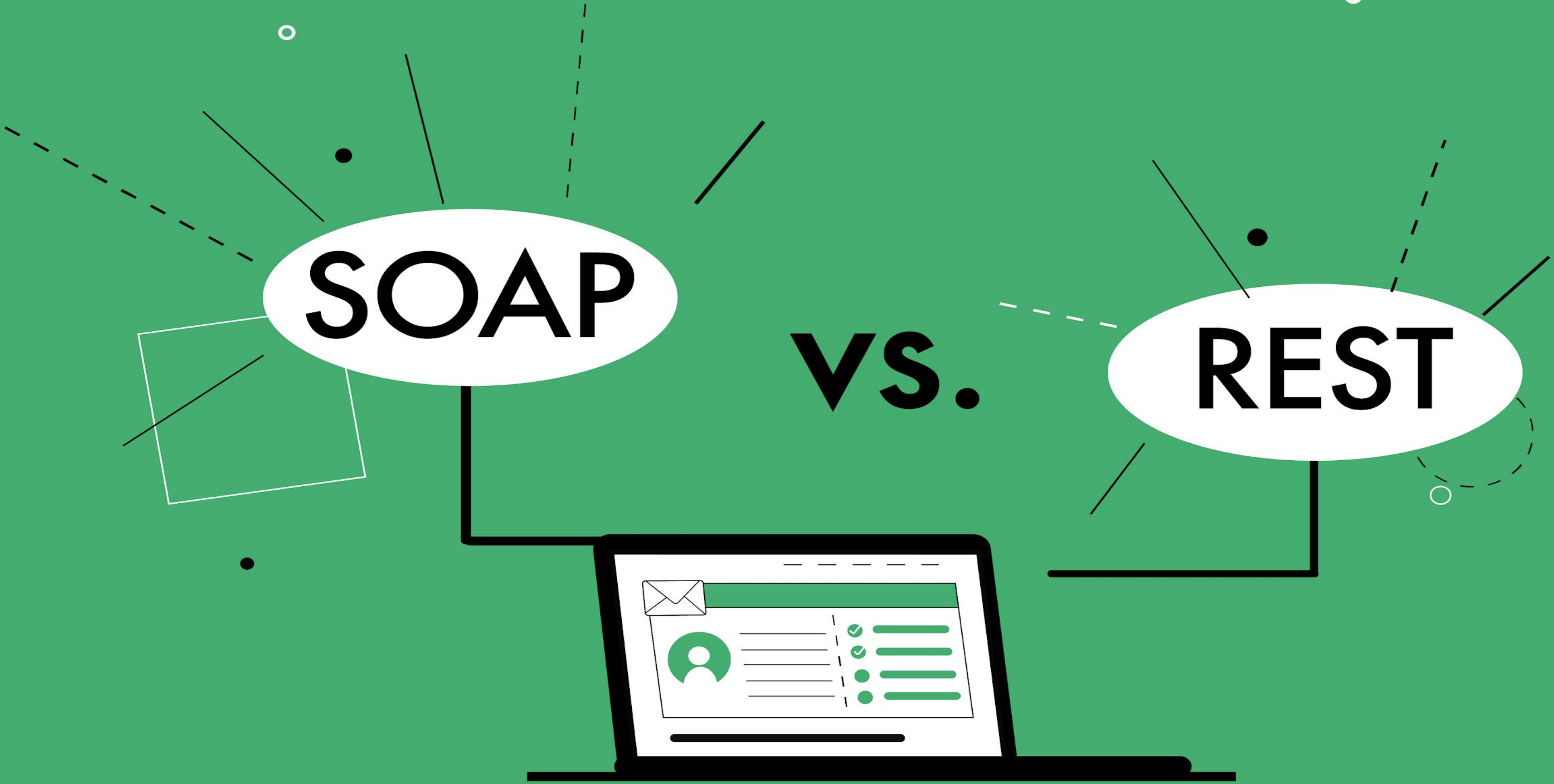
Managing multiple services can be complicated (and the communication between them)

# Common usage of Service Based Architecture

- GPS
- Health Tracking
- Weather forecasts
- Etc...

# Services and Protocols

- Introduction to Services and Protocols
- The Foundation of Web-Based Communication
- Serving as Guidelines for Interaction



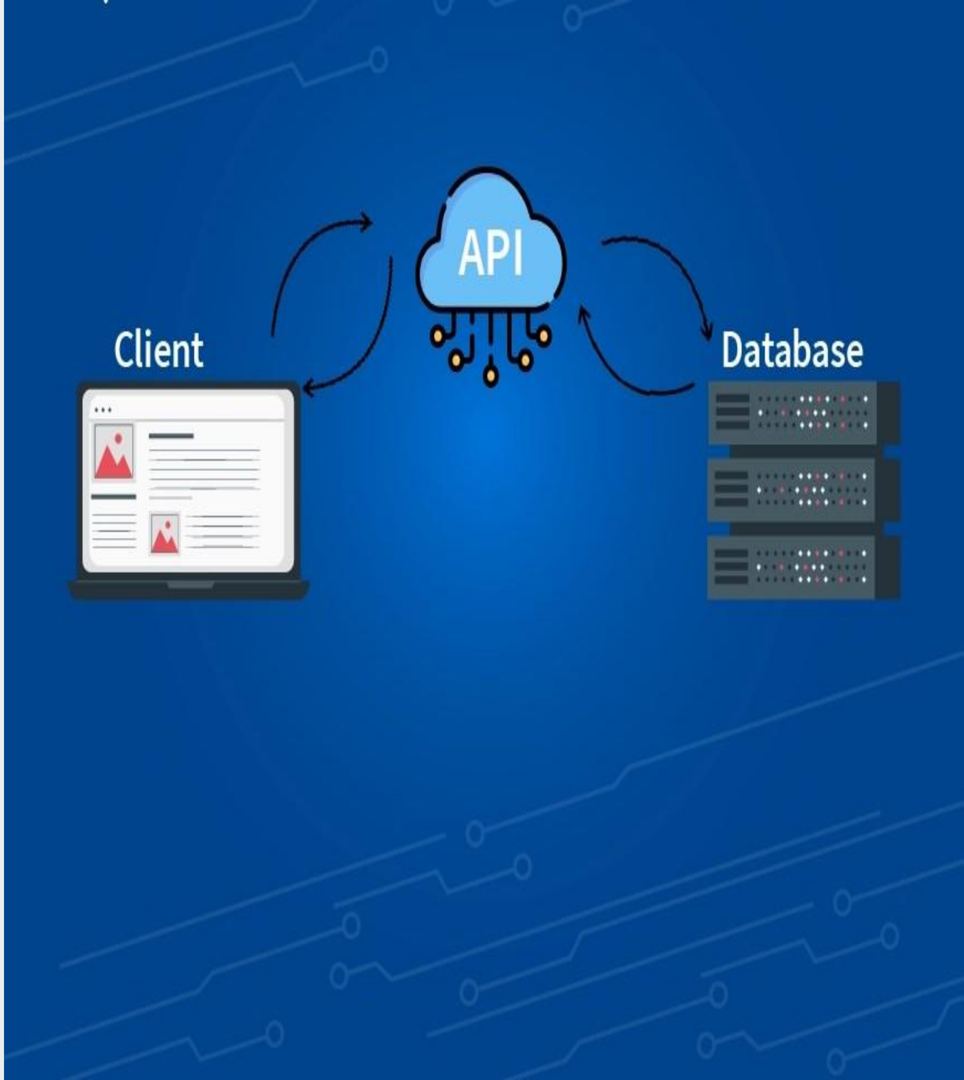
SOAP vs. REST

# SOAP -
# (Simple Object Access Protocol)

- Protocol for Web Services
- XML-based Message Protocol
- Functionality through Operations

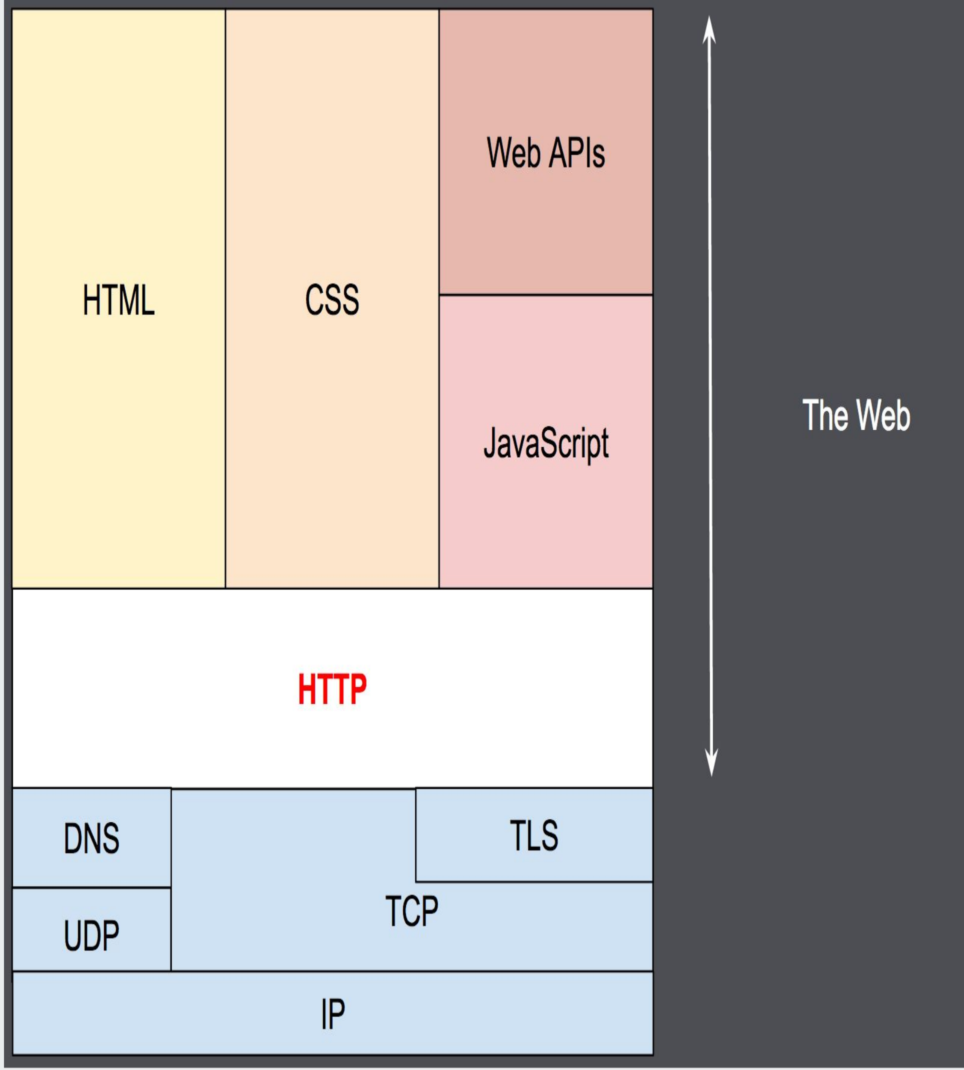
# REST (Representational State Transfer)

- Architectural Style, Not a Protocol
- Uses Standard HTTP Methods (GET, POST, PUT, DELETE)
- Stateless Communication



# HTTP (HyperText Transfer Protocol)

- Application Protocol for Distributed Systems
- Foundation of Data Communication for the Web
- Stateless Protocol



# WSDL -
# (Web Services Description Language)

- XML-Based Format for Describing Web Services
- Details on Available Functions, Data Types, & Protocols
- Used in Conjunction with SOAP