

WEB SERVICES

Presented by: Four Loop



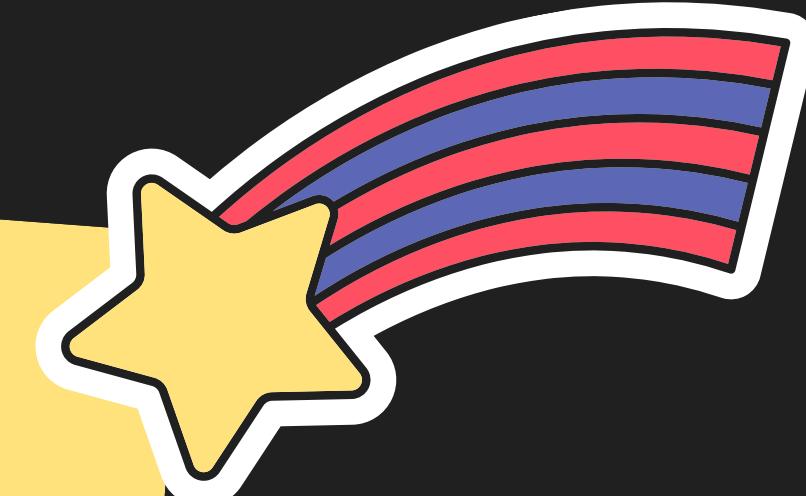


WEB SERVICES

Are a technology that facilitates communication and data exchange between software applications over the internet. Clients use the World Wide Web to find and use various web services.

ICE BREAKER

WORD



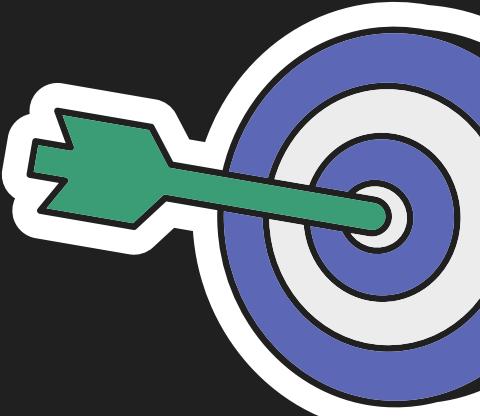
PUZZLE



Rearrange the letters



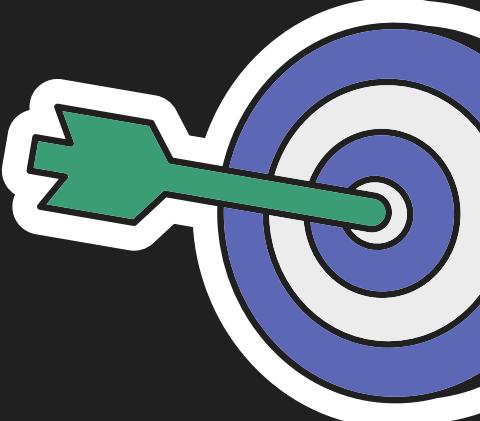
ERSCVIE





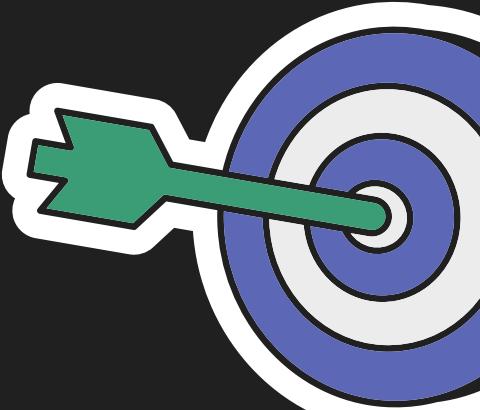
ANSWER

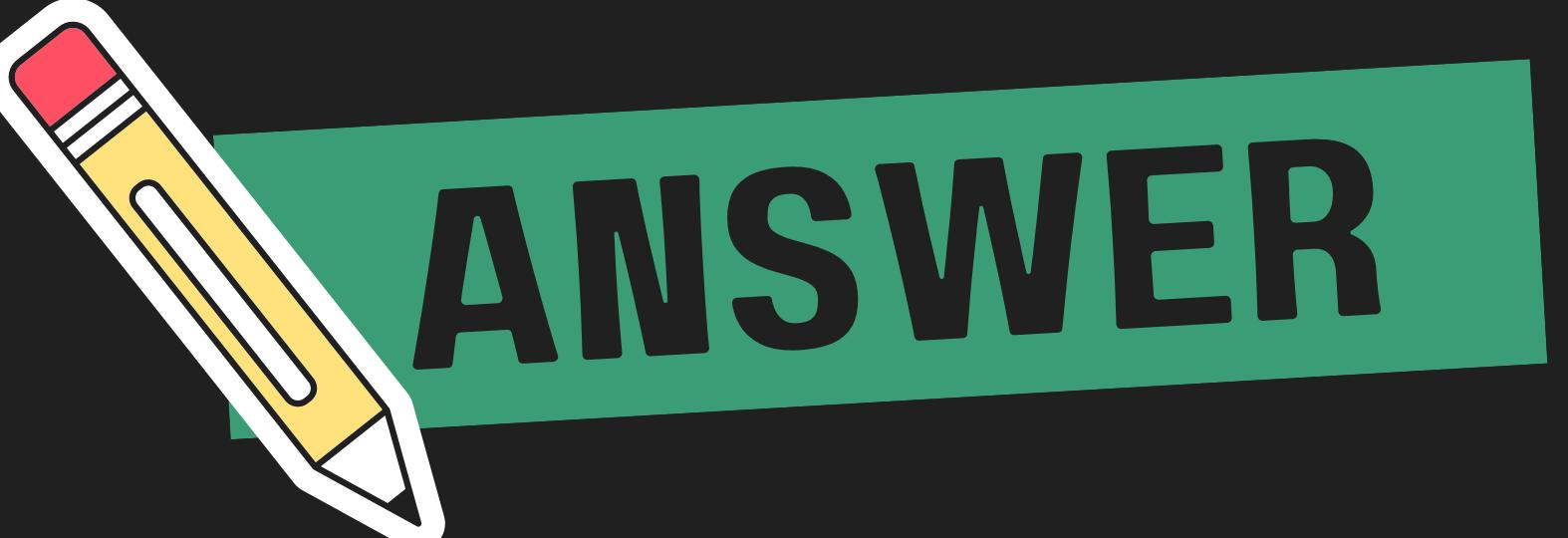
SERVICE





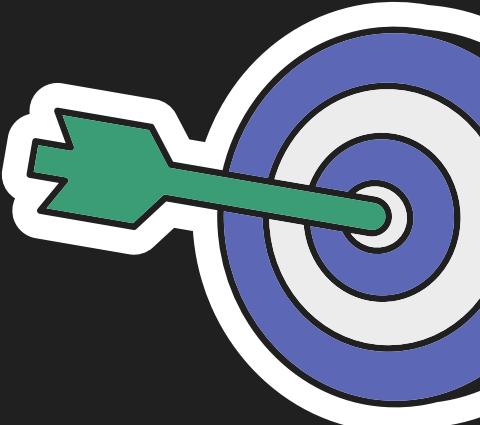
NEDPIOPT





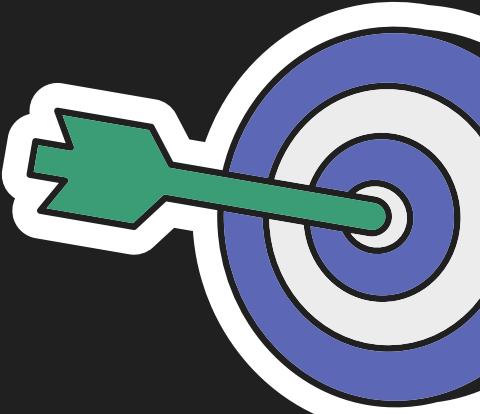
ANSWER

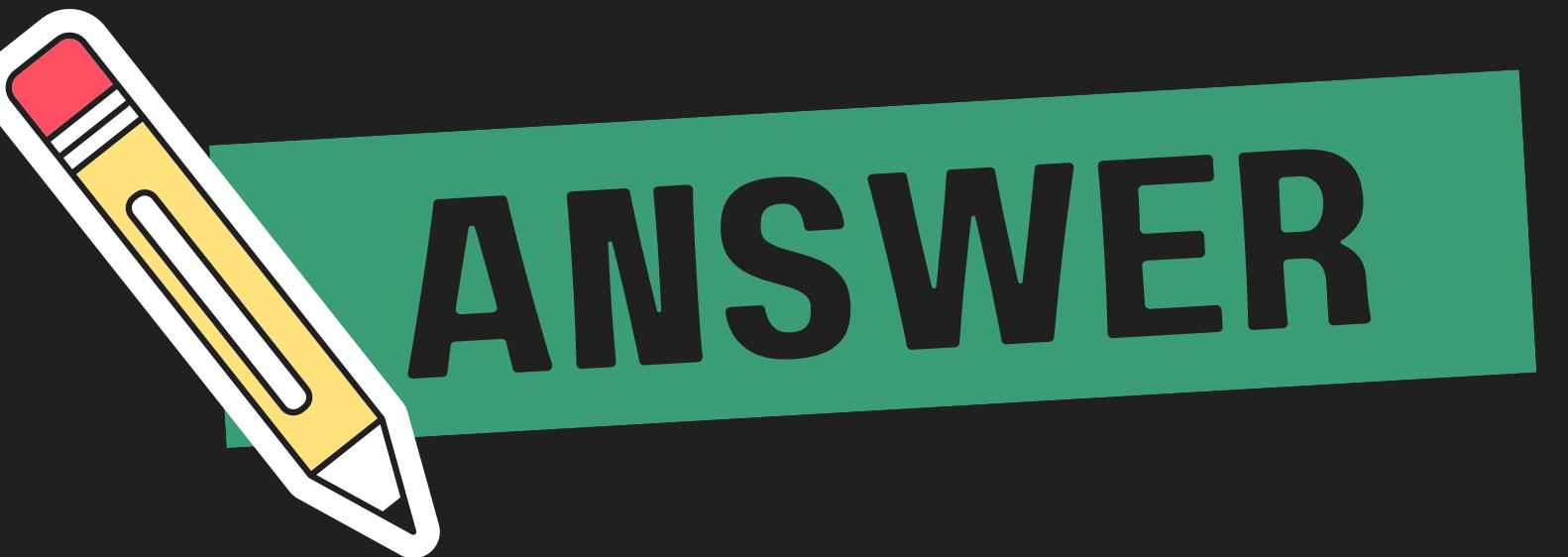
ENDPOINT



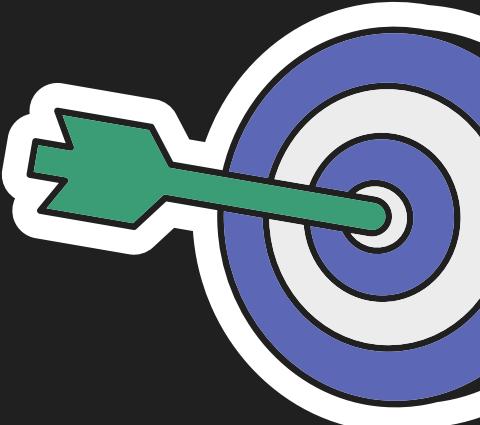


NNECOTOC



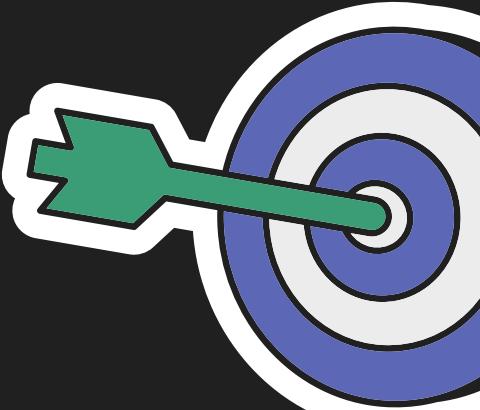


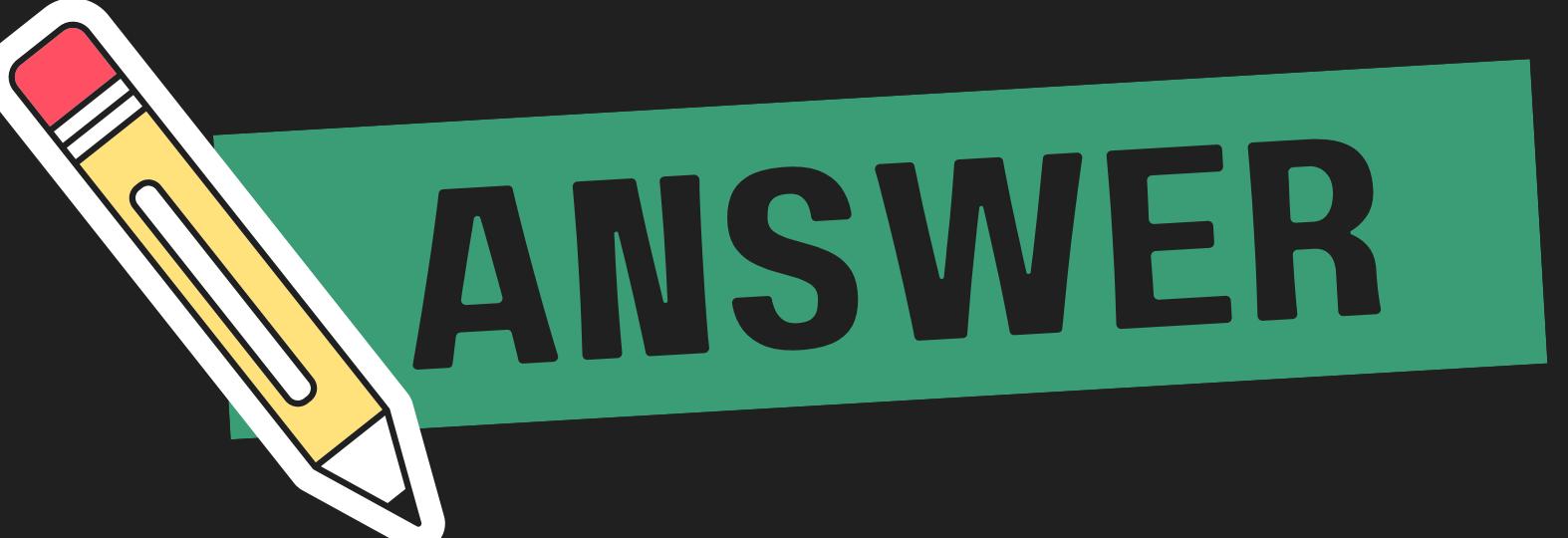
CONNECTOR





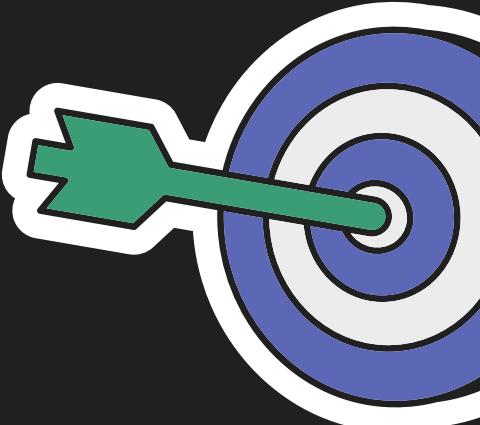
SEDABAAT





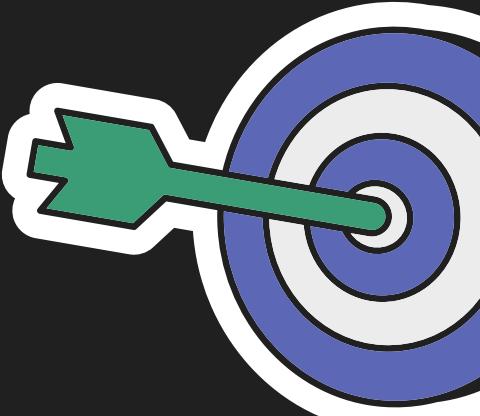
ANSWER

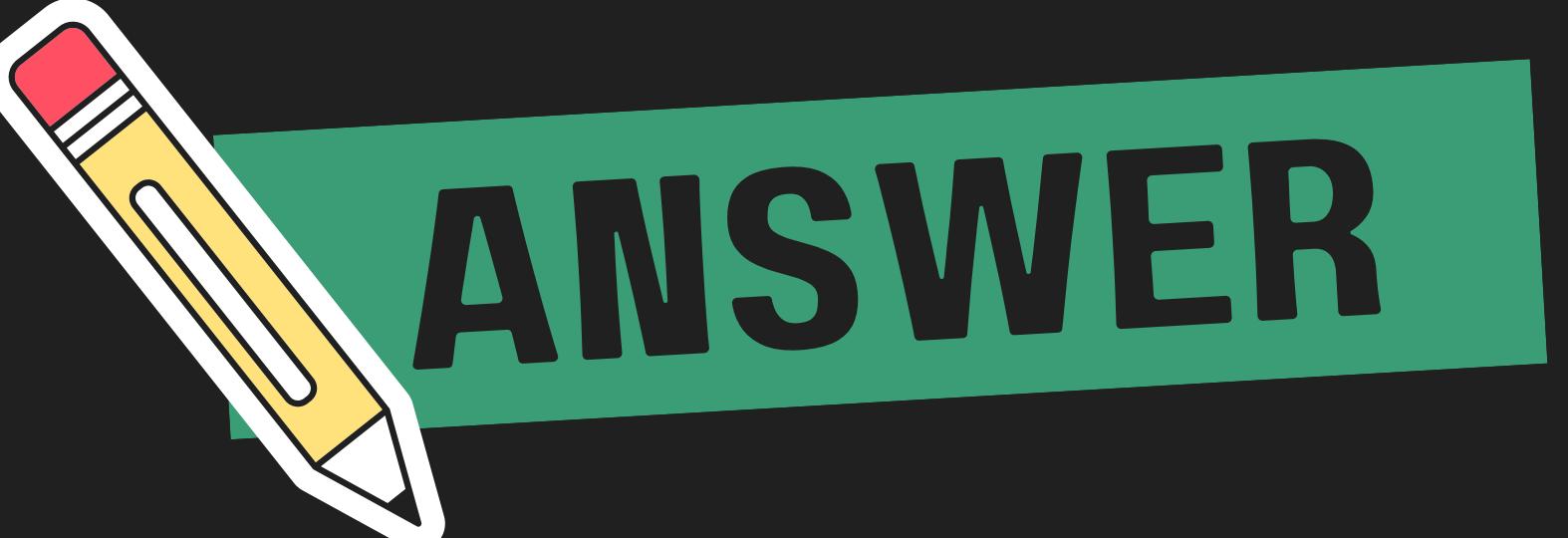
DATABASE





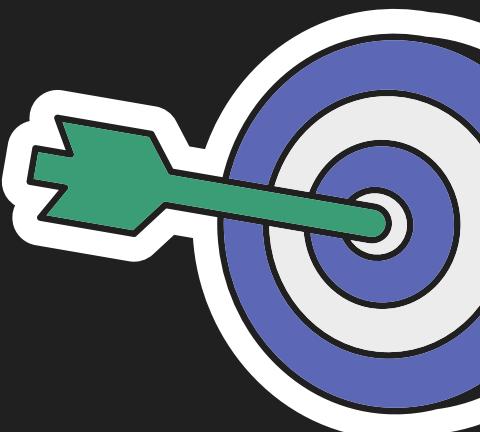
RINTEETN





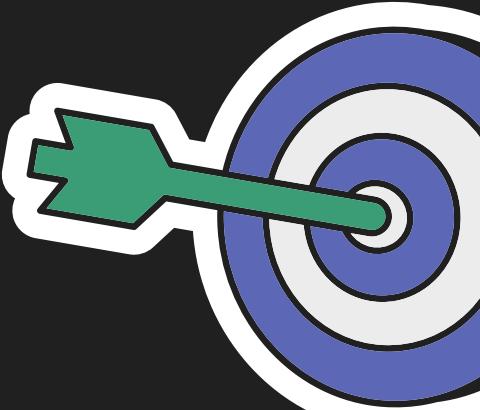
ANSWER

INTERNET





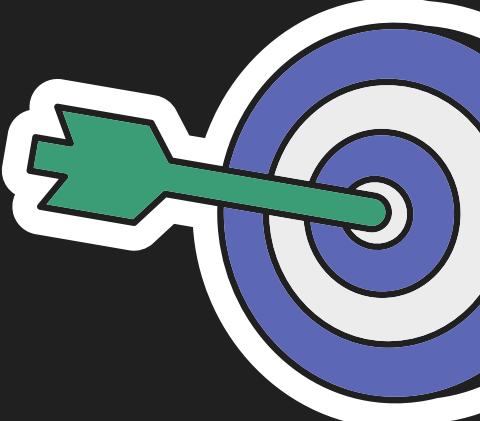
EYTTPRP00





ANSWER

PROTOTYPE





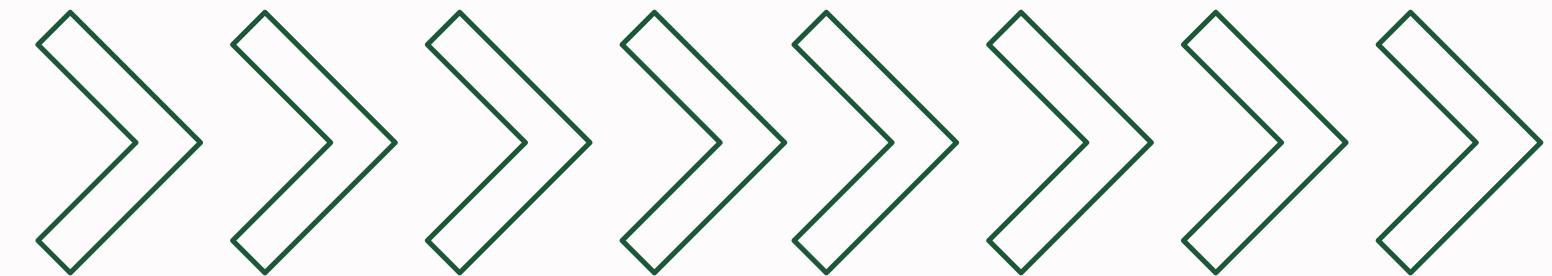
FOUR LOOP |

WEB SERVICES

01



WHAT IS WEB SERVICES



- is a software module that is intended to carry out a specific set of functions.
- a set of open protocols and standards that allow data to be exchanged between different applications or systems.

FUNCTIONS OF WEB SERVICES

- 01  It's possible to access it via the internet or intranet networks.
- 02  XML messaging protocol that is standardized.
- 03  Operating system or programming language independent.
- 04  Using the XML standard, it is self-describing.
- 05  A simple location approach can be used to locate it.

✗ ✗ ✗ ✗

FOUR LOOP I

COMPONENTS OF A WEB SERVICES



✗ ✗ ✗ ✗



FOUR LOOP |

SOAP

Simple Object Access Protocol

SOAP is a transport-independent messaging protocol that utilizes XML-based messages for communication.

Its structure follows a pattern with XML documents attached to messages, ensuring standardized communication over HTTP, the standard web protocol.



FOUR LOOP |

UDDI

Universal Description, Discovery, and Integration

UDDI serves as a standard for specifying, publishing, and discovering online services.

Acting as a repository for WSDL files, UDDI enables client applications to discover web services, like how a telephone directory holds information, ensuring clients can locate and understand services offered by service providers.



FOUR LOOP |

WSDL

Web Services Description Language

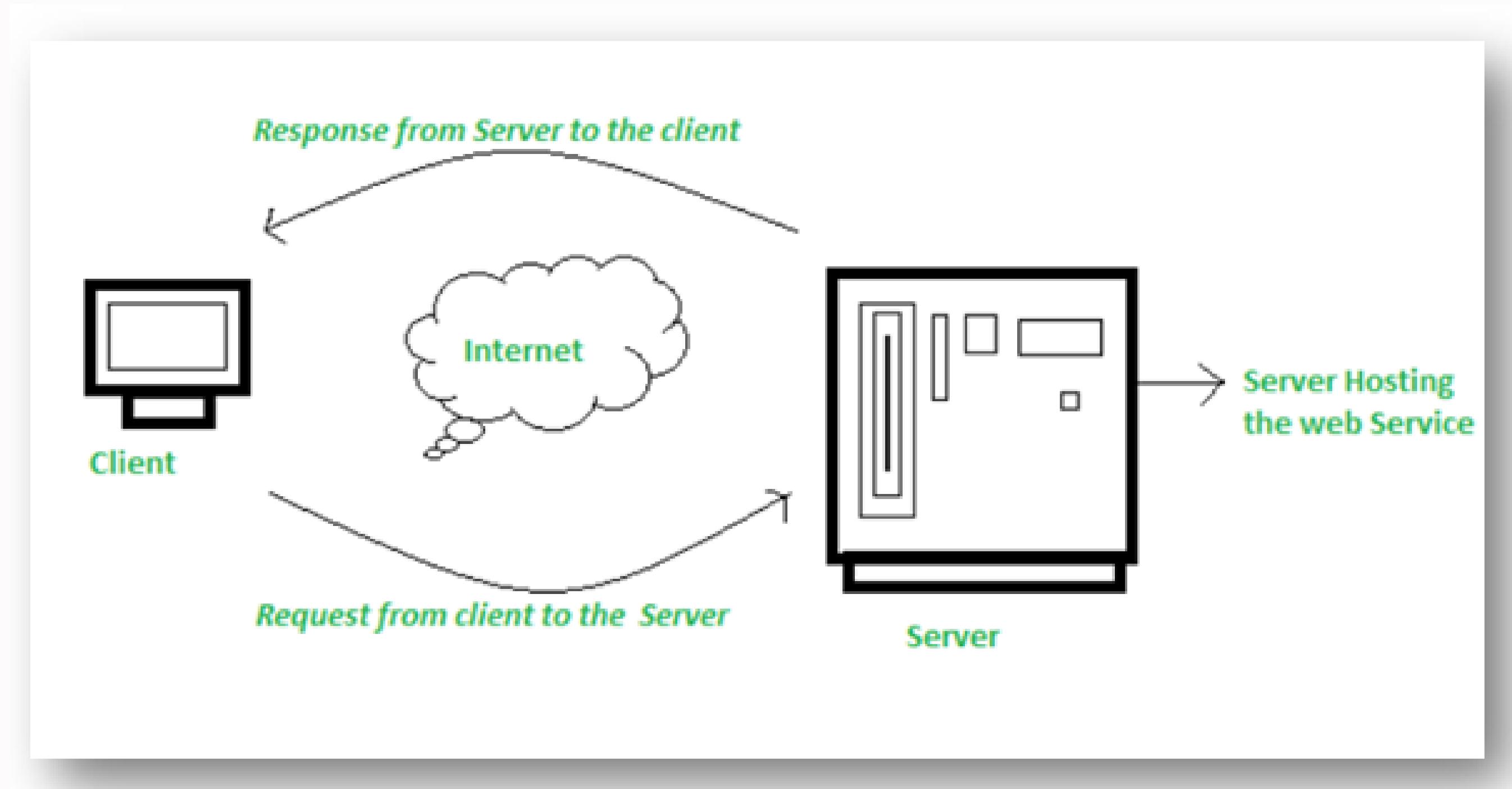
WSDL is crucial for web service discovery and understanding.

This XML-based language provides a description of the web service, including its location and functionality.

By utilizing WSDL files, client applications can effectively locate and invoke the correct web service.

| FOUR LOOP

HOW DOES IT WORK?



FEATURES OF WEB SERVICES

XML Based

Information representation and record transportation layers use XML.

No need for networking, operating system, or platform binding.

Enhances interoperability at the middle level.

Loosely Coupled

Clients are not directly linked to service providers.

Changes in the service provider's interface don't impact user interaction.

Enables easier integration and makes software systems more manageable.

Synchronous or Asynchronous

Synchronous: Client is blocked until the service completes its operation.

Asynchronous: Client can invoke a task and continue with other tasks.

Supports loosely linked systems through asynchronous capabilities.

FEATURES OF WEB SERVICES

Coarse-Grained

Object-oriented systems often use fine-grained methods.

Web services promote coarse-grained services at the corporate level.

Facilitates the development of services with access to sufficient business logic.

Supports Remote Procedural Call (RPC)

Consumers can call procedures, functions, and methods on remote objects.

Supports input and output frameworks exposed by remote systems.

Integrates with JavaBeans (EJBs) and .NET Components for component development.

Supports Document Exchanges

Leverages XML's simplicity in communicating with data and complex entities.

Facilitates the exchange of documents from simple addresses to complex entities.

Aids in integration through the simple exchange of documents.

ADVANTAGES OF WEB SERVICES

01

Business Functions
Exposed over the
Internet

02

Interoperability

03

Communication
with Low Cost

04

Standard Protocol

05

Reusability

02

SUPPORTING INTERACTIONS

FOUR LOOP I



SUPPORTING INTERACTIONS



An interaction can start a web service and send data to it from an application.

The interaction can then display data returned from the web service and save this data to the application database.



TO IMPLEMENT INTERACTIONS REQUIRES KNOWLEDGE OF:

01

XML SCHEMAS

03

The integration
framework

WEB SERVICES

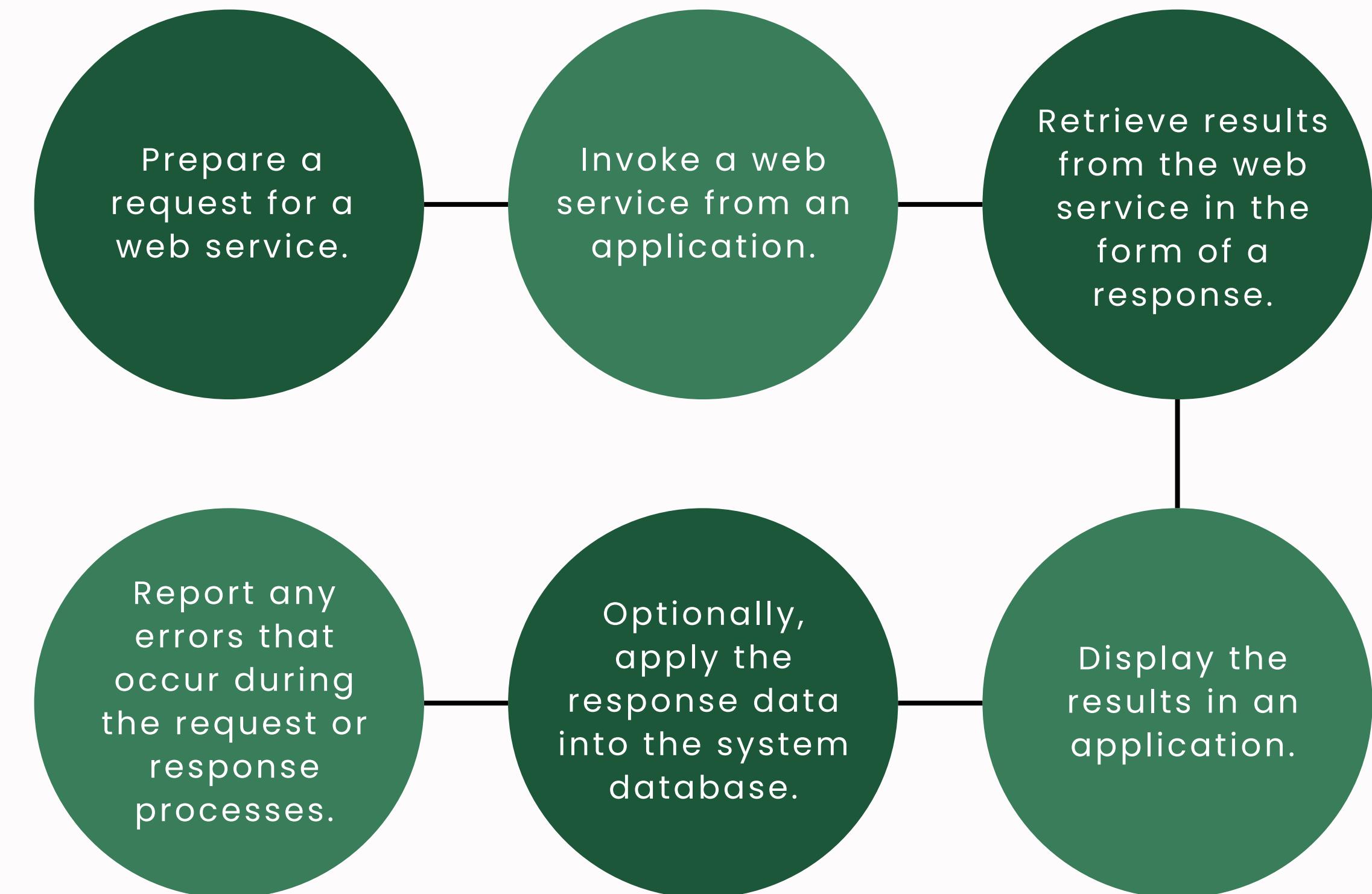
02

Customizing applications
and application user
interfaces

04



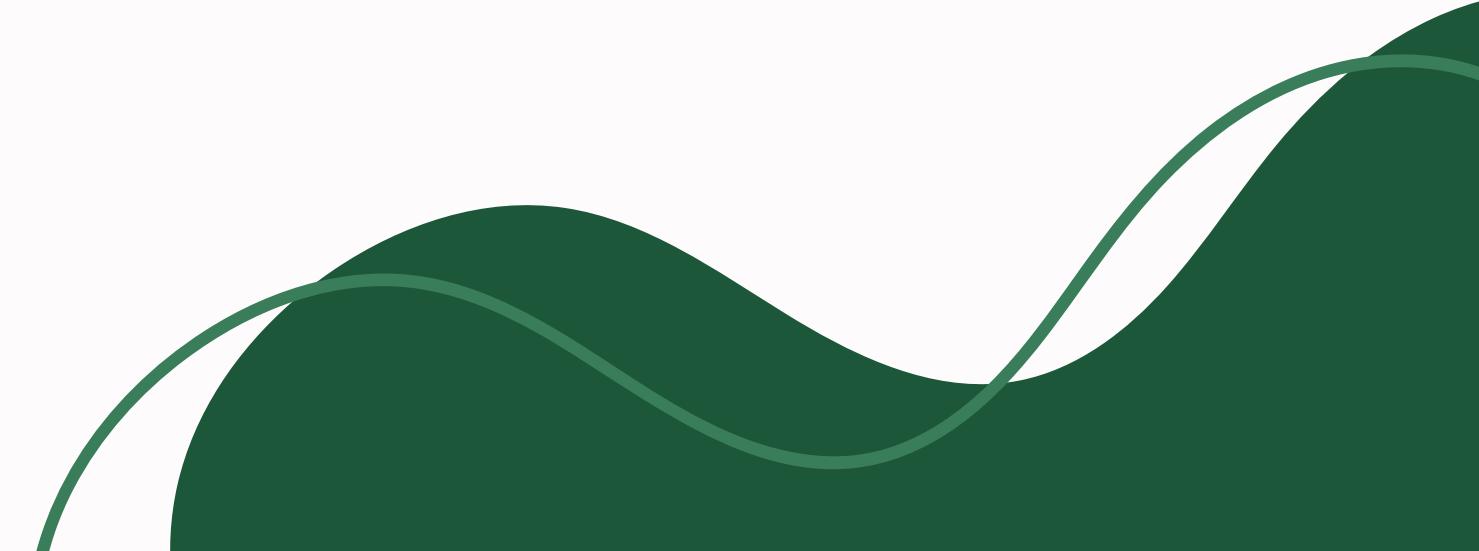
INTERACTION PROCESSES



TWO APPLICATIONS ARE PROVIDED TO HELP YOU CREATE AND MANAGE INTERACTIONS.

- Create Interaction Application
- Interactions Application

AFTER YOU CREATE AN INTERACTION, USERS CAN PERFORM THE FOLLOWING:

- Start the interaction.
 - View and, if configured, change the parameters of the request to the web service.
 - Invoke the web service by sending the request.
 - Optionally, view data received from the web service and commit this data to the database.
- 



FOUR LOOP |

SERVICE-ORIENTED ARCHITECTURE (SOA)

03



SERVICE-ORIENTED ARCHITECTURE (SOA)

SOA is an architectural style for building software applications that use services available in a network such as the web. It promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services.

A service is an implementation of a well-defined business functionality, and such services can then be consumed by clients in different applications or business processes.



BENEFITS OF SOA

01

Faster time to
market

02

Efficient
maintenance

03

Greater
adaptability





BASIC PRINCIPLES OF SOA



FOUR LOOP |

Loose Coupling

Services in SOA should be loosely coupled, having as little dependency as possible on external resources such as data models or information systems.

Interoperability

Each service in SOA includes description documents that specify the functionality of the service and the related terms and conditions.

Granularity

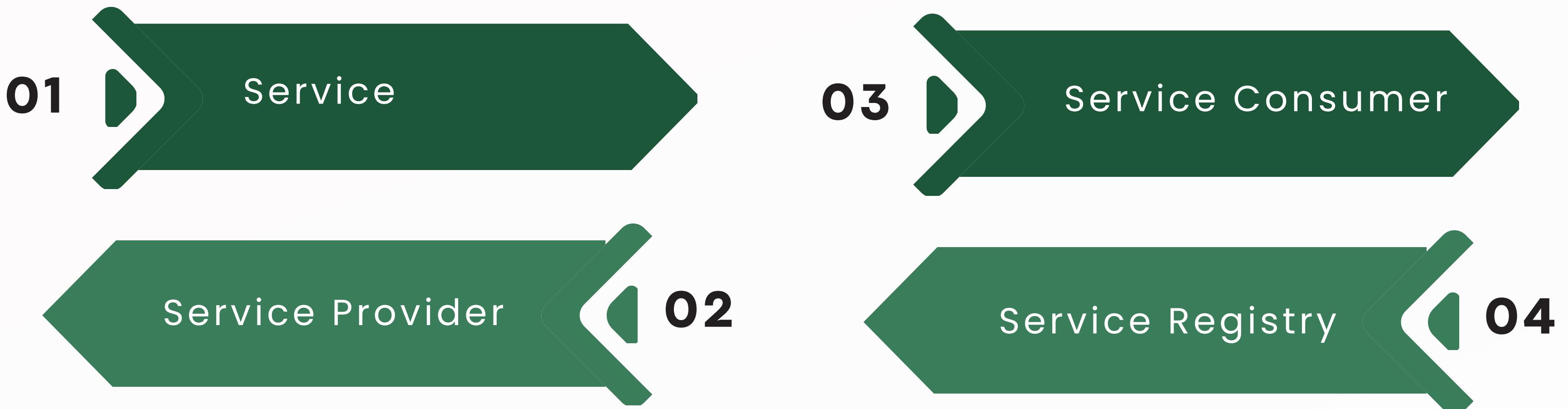
Services in SOA should have an appropriate size and scope, ideally packing one discrete business function per service.

Abstraction

Clients or service users in SOA need not know the service's code logic or implementation details.

COMPONENTS OF SOA

There are four main components in service-oriented architecture (SOA).





COMPONENTS OF SOA

Services are the basic building blocks of SOA. Individually, each service has three main features.

Service Implementation

The service implementation is the code that builds the logic for performing the specific service function, such as user authentication or bill calculation.

Service Contract

The service contract defines the nature of the service and its associated terms and conditions, such as the prerequisites for using the service, service cost, and quality of service provided.

Service Interface

In SOA, other services or systems communicate with a service through its service interface. The interface defines how you can invoke the service to perform activities or exchange data. It reduces dependencies between services and the service requester.

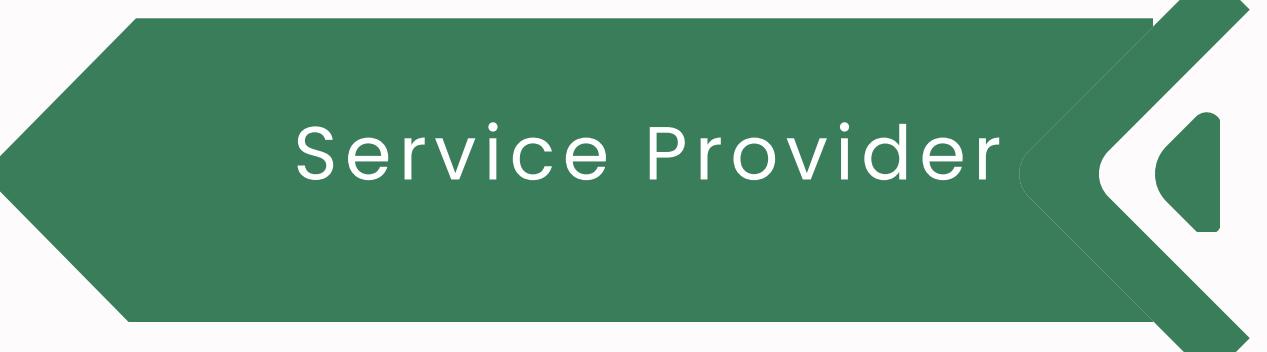
COMPONENTS OF SOA

03

Service Consumer

A service registry, or service repository, is a network-accessible directory of available services. It stores service description documents from service providers. The description documents contain information about the service and how to communicate with it.

Service Provider

02

03

Service Consumer

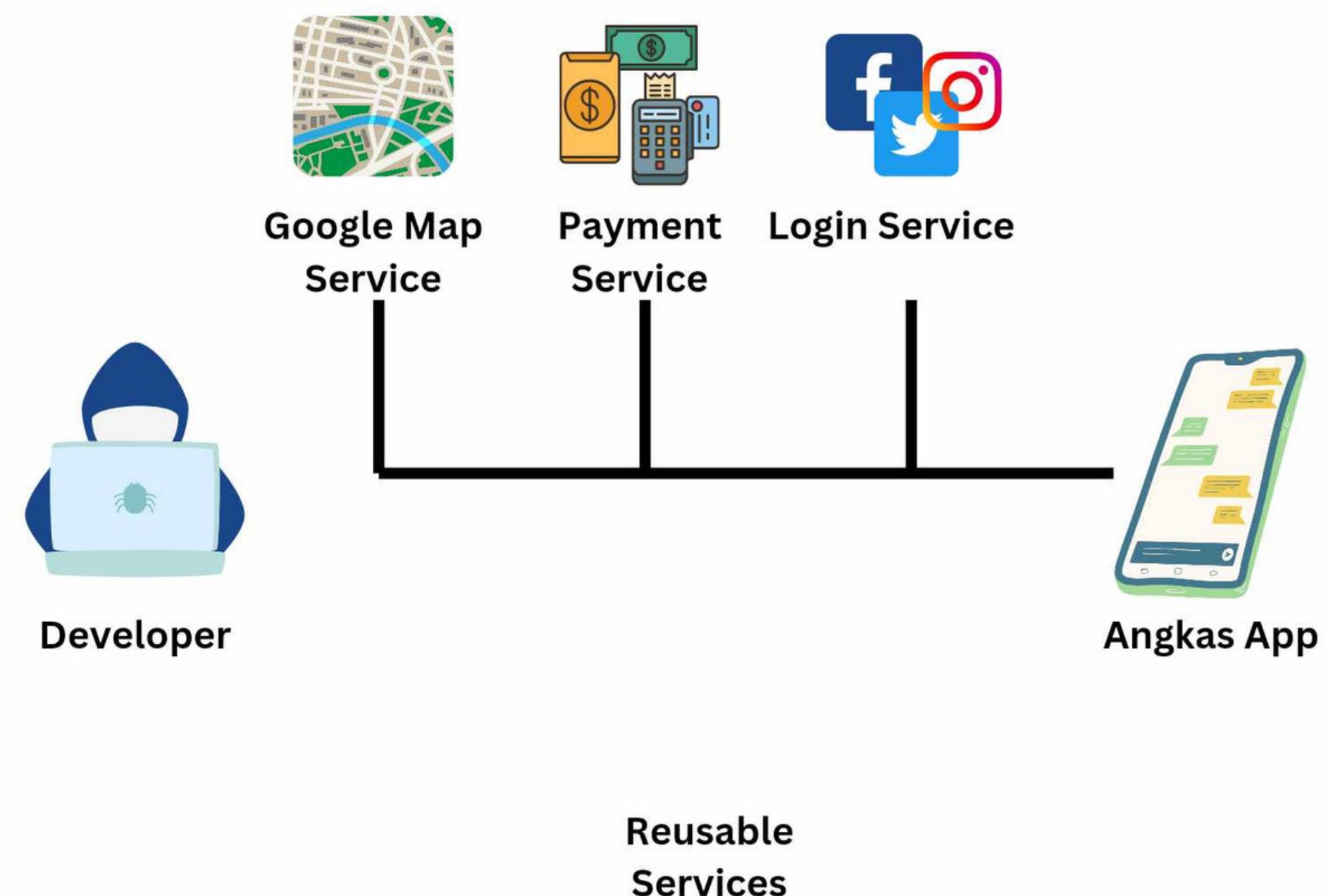
The service consumer requests the service provider to run a specific service. It can be an entire system, application, or other service.

Service Registry

04

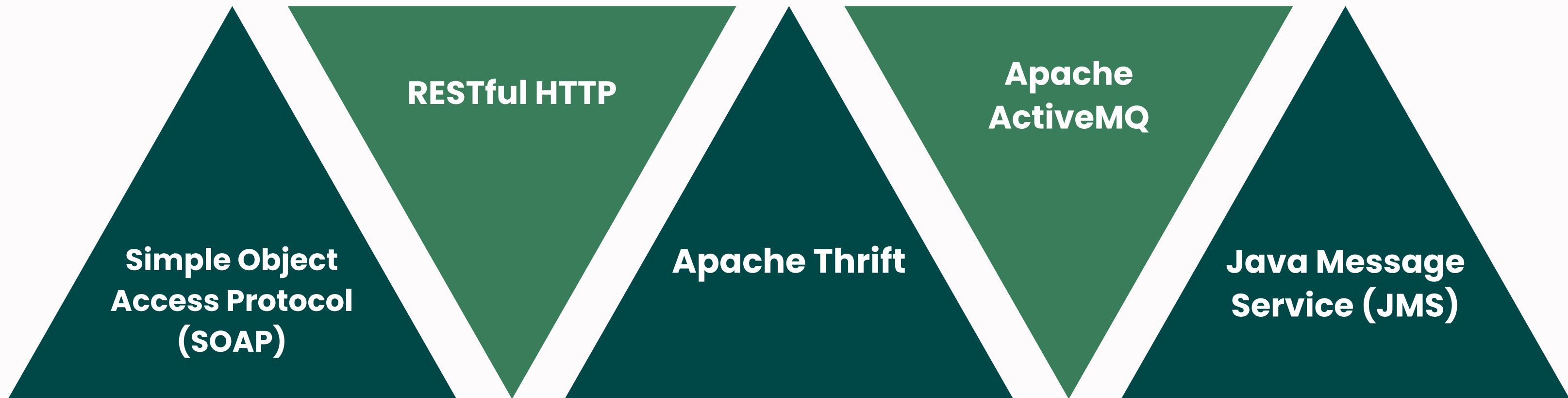
| FOUR LOOP

HOW DOES SOA WORK?



FOUR LOOP |

COMMUNICATION PROTOCOLS



× × × ×



FOUR LOOP |

SIMPLE OBJECT ACCESS PROTOCOL (SOAP)

04



SIMPLE OBJECT ACCESS PROTOCOL (SOAP)

SOAP is an XML-based protocol for accessing web services over HTTP. It has some specification which could be used across all applications. SOAP is known as the Simple Object Access Protocol, SOAP is a protocol or in other words is a definition of how web services talk to each other or talk to client applications that invoke them.

SOAP was developed as an intermediate language so that applications built on various programming languages could talk easily to each other and avoid the extreme development effort.



BRIEF HISTORY

SOAP architecture was initially developed by Microsoft, IBM, and others in the early 2000s. Since then, it has been accepted as an industry standard for web services. The initial version of SOAP was released in 1998 and was designed to provide a standard way for applications to exchange structured data over the internet. SOAP was based on XML (Extensible Markup Language) and used the HTTP protocol for transport.

ADVANTAGES OF SOAP

01

Message-Level
Flexibility

02

Protocol
Independence

03

Built-In Security



SOAP BUILDING BLOCKS

The components of a SOAP message



ENVELOPE

The outermost element wraps all other elements in the SOAP message SOAP Envelope. It identifies the SOAP protocol version number, specifies the encoding style, and provides a mechanism to pass along instructions for processing unknown elements.

SOAP BUILDING BLOCKS

The components of a SOAP message

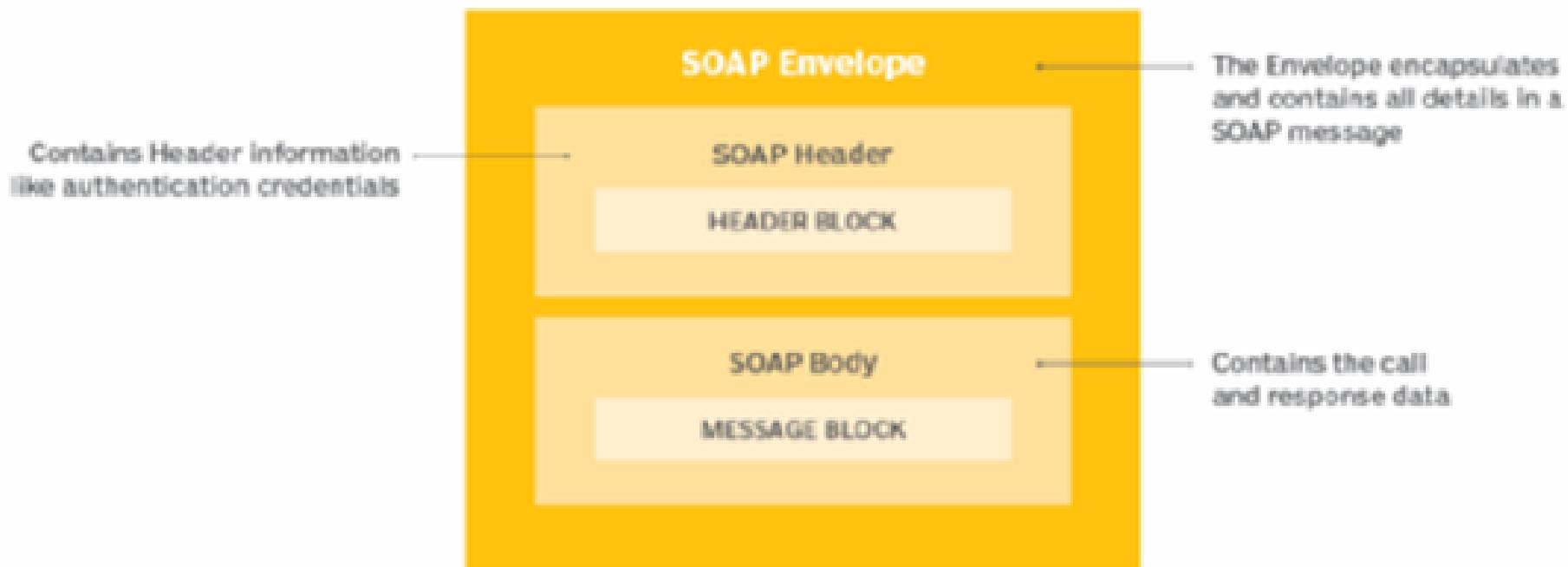


HEADER

Contains information about how the SOAP message should be processed, including how to interpret any parts that appear in the body. The header element is placed before any data elements in the message.

SOAP BUILDING BLOCKS

The components of a SOAP message



BODY

Contains an ordered sequence of body parts that together provide all of the necessary information for processing the message.

SOAP BUILDING BLOCKS

The components of a SOAP message



FAULT

A fault is an optional element that contains information about a problem encountered during processing. The content of the fault element depends on the type of fault code attribute specified, as defined by the SOAP specification.

SOAP MESSAGE STRUCTURE



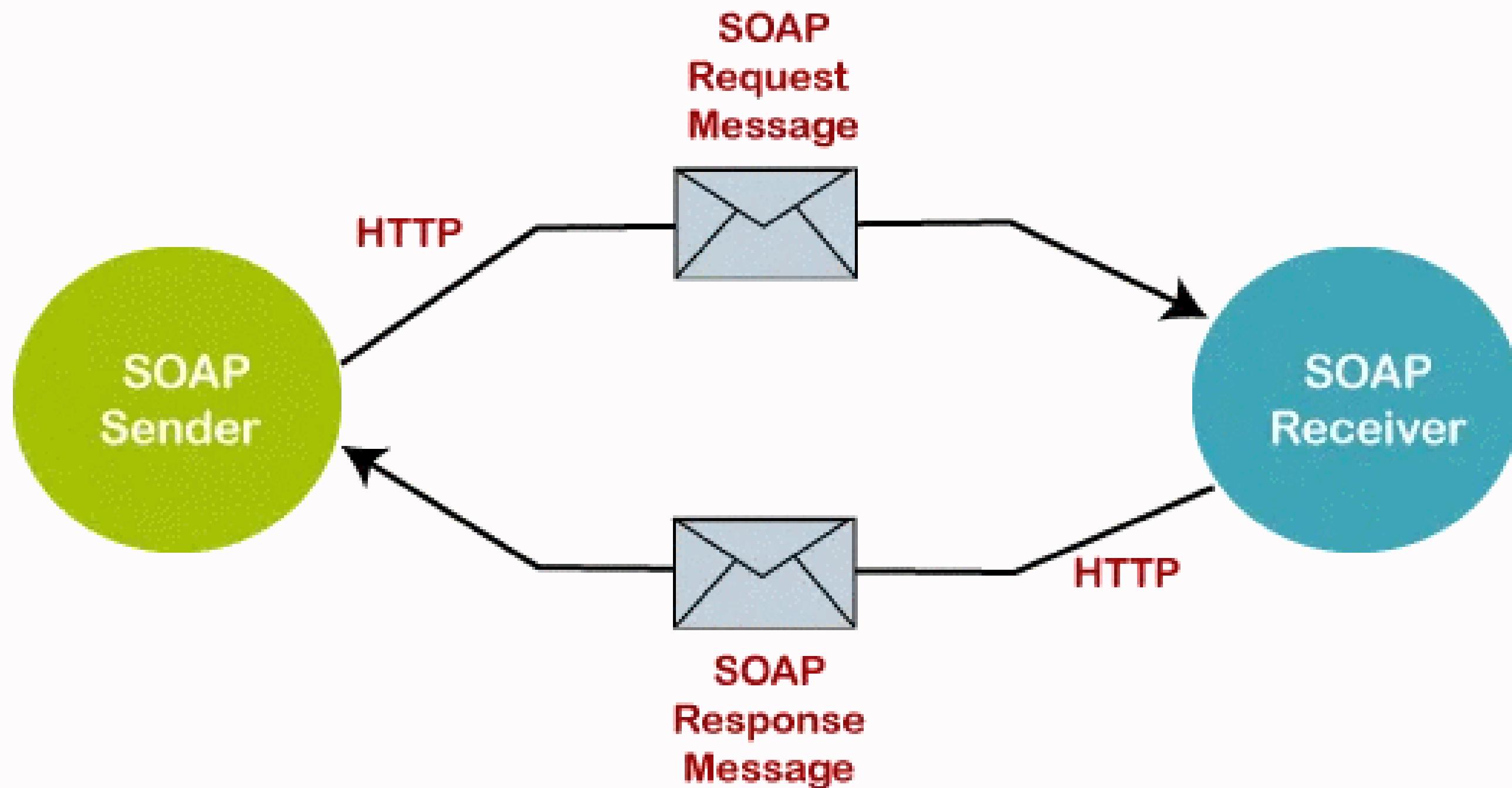


SOAP COMMUNICATION MODEL

All communication by SOAP is done via the HTTP protocol. Prior to SOAP, a lot of web services used the standard RPC (Remote Procedure Call) style for communication. This was the simplest type of communication, but it had a lot of limitations.

| FOUR LOOP

HOW DOES SOAP WORK?





FOUR LOOP I

WEB SERVICES DESCRIPTION LANGUAGE (WSDL)

05



WEB SERVICES DESCRIPTION LANGUAGE (WSDL)



WSDL, or Web Services Description Language, is an XML-based language used to describe the functionalities offered by a web service.

KEY COMPONENTS OF WSDL

01

TYPES

02

MESSAGE

03

PORT TYPE

04

BINDING

05

SERVICE

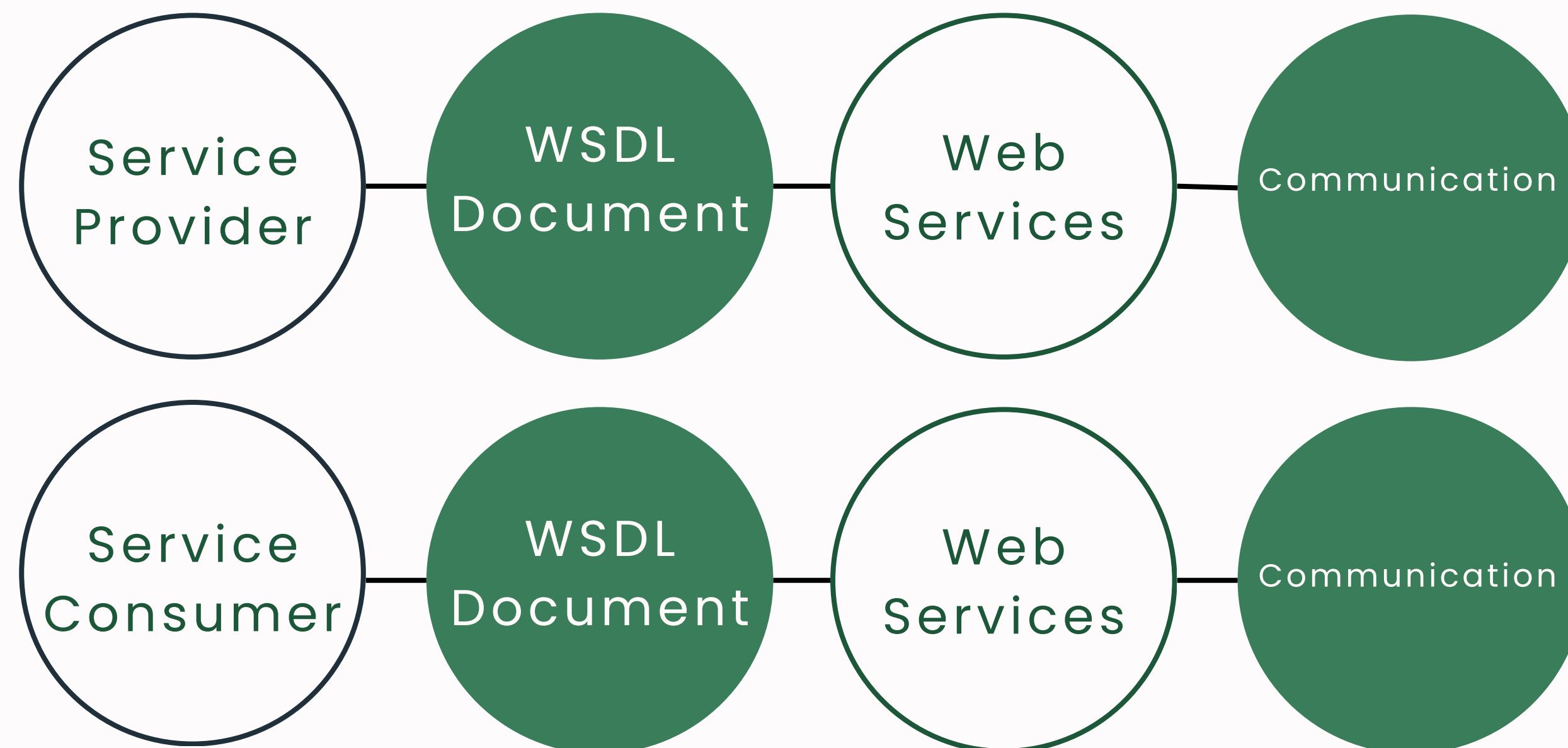
WSDL EXAMPLE

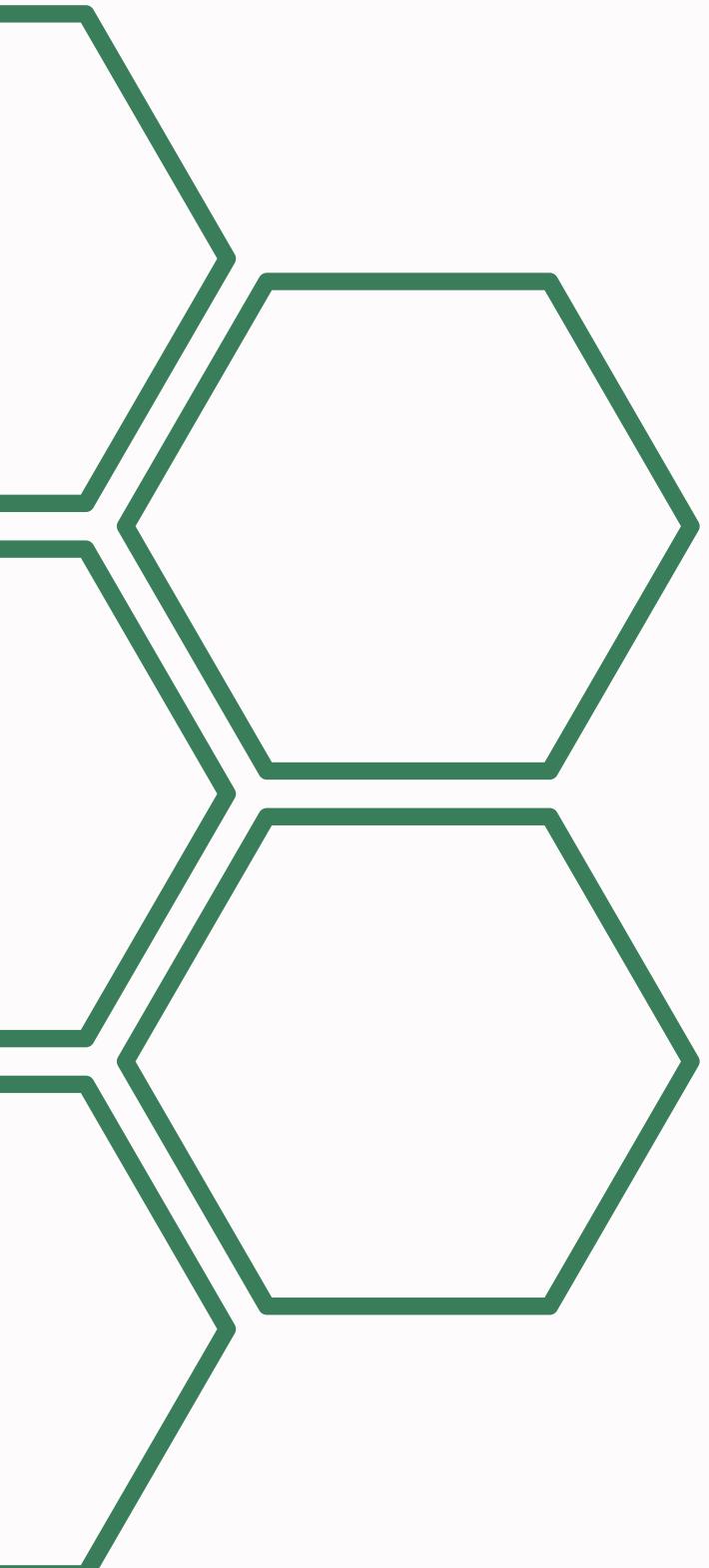
```
<definitions>  
  
  <types>  
    data type definitions.....  
  </types>  
  
  <message>  
    definition of the data being communicated....  
  </message>  
  
  <portType>  
    set of operations.....  
  </portType>  
  
  <binding>  
    protocol and data format specification....  
  </binding>  
  
</definitions>
```

FOUR LOOP |

| FOUR LOOP

HOW DOES WSDL WORK?





BENEFITS OF WSDL

01

Interoperability

02

Standardization

03

Documentation





FOUR LOOP I

UNIVERSAL
DESCRIPTION
DISCOVERY AND
INTEGRATION (UDDI)

06



UNIVERSAL DESCRIPTION, DISCOVERY, AND INTEGRATION (UDDI)

UDDI, or Universal Description, Discovery, and Integration, is a specification that defines a standard way for businesses to publish and discover information about web services.



Publish Services

Businesses can register their web services with detailed information about their functionality, interfaces, and other relevant details.

Discover Services

Other businesses can search and discover available services based on their needs and requirements

Integrate Services

Once a service is discovered, businesses can seamlessly integrate it into their applications or processes

PURPOSE OF UDDI

KEY COMPONENTS OF UDDI

01

White Node
(White Pages)

02

Yellow Node
(Yellow Pages)

03

Green Node
(Green Pages)

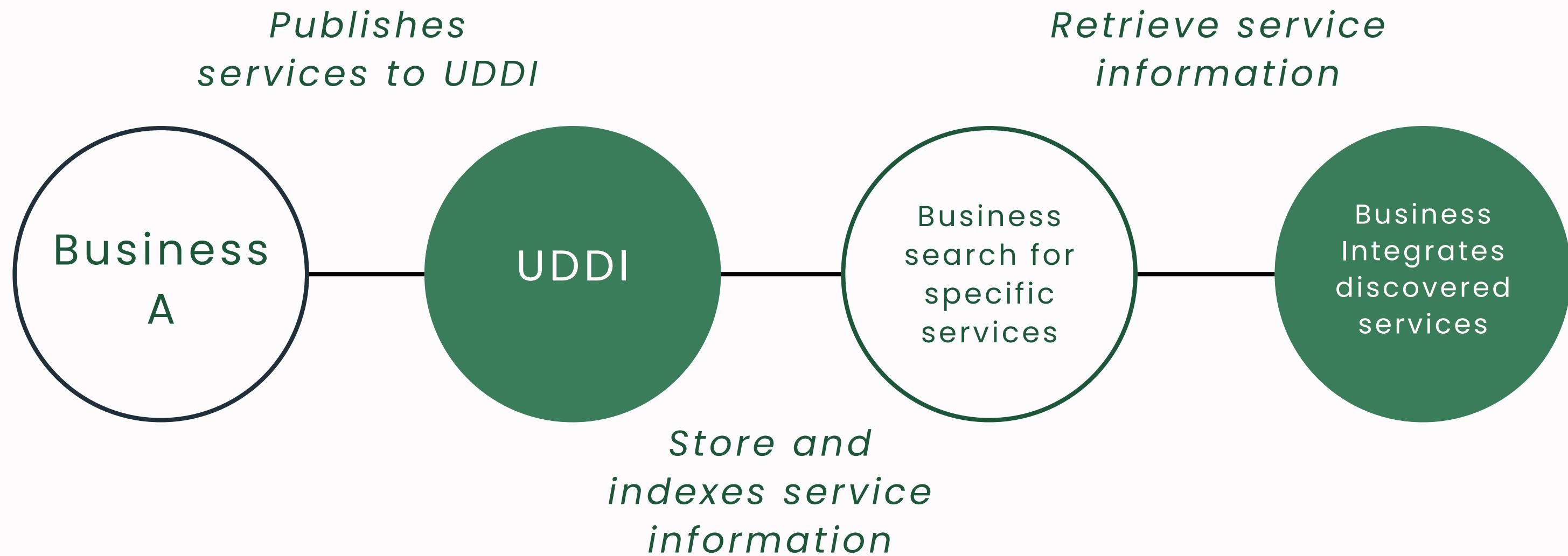


UDDI EXAMPLE

```
1 <businessEntity>
2   <name>Company XYZ</name>
3   <businessServices>
4     <businessService>
5       <name>Service A</name>
6       <bindingTemplates>
7         <bindingTemplate>
8           <accessPoint>https://servicea.com</accessPoint>
9         </bindingTemplate>
10        </bindingTemplates>
11      </businessService>
12    </businessServices>
13  </businessEntity>
```

| FOUR LOOP

HOW DOES UDDI WORK?





FOUR LOOP I

BENEFITS OF UDDI

Interoperability

Standardization

Collaboration



07

REST API

FOUR LOOP |



WHAT IS AN API?

Stands for Application Programming Interface

A software intermediary that allows two applications
to talk to each other.

APIs are an accessible way to extract and share data
within and across organizations.



TWO TYPES OF API

REST API (Representational State Transfer)

- Built on top of HTTP
- Using Standard HTTP Methods
- GET, POST, PUT DELETE, etc.

WEB API

- on top of ASP.NET platform
- Primarily POST, GET, PUT, DELETE



INTRODUCTION TO REST API

REST or **Representational State Transfer**, is an **architectural style** for creating web services. Unlike some popular beliefs, REST is **not** a Tool, Protocol, or Library. Instead, it is a way of accessing web services that **use less processing power**. REST APIs are based on the REST architecture and are sometimes called **RESTful APIs**.



INTRODUCTION TO REST API

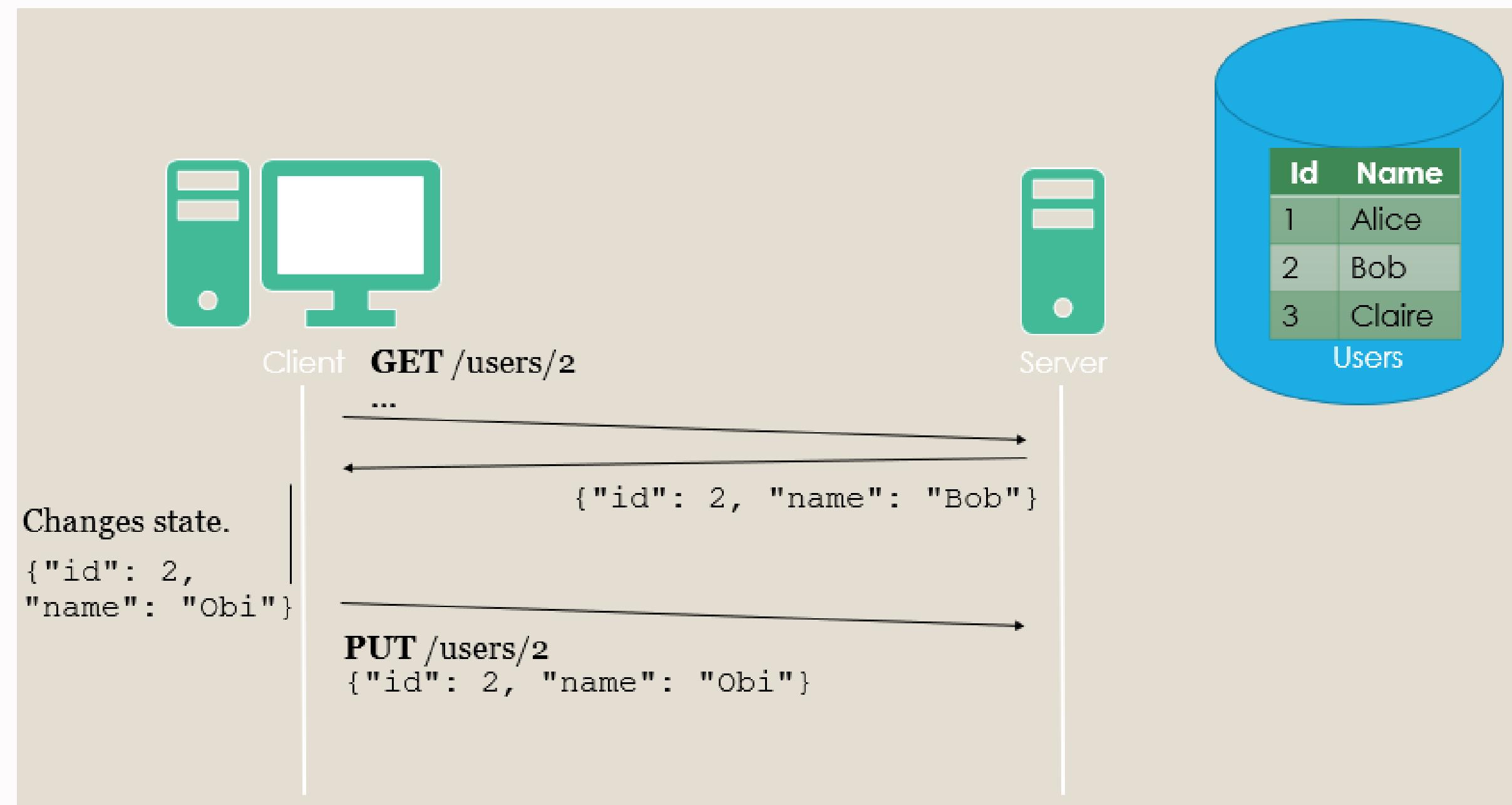
Used to **access** resources such as **HTML pages, Images** and **otherData**.

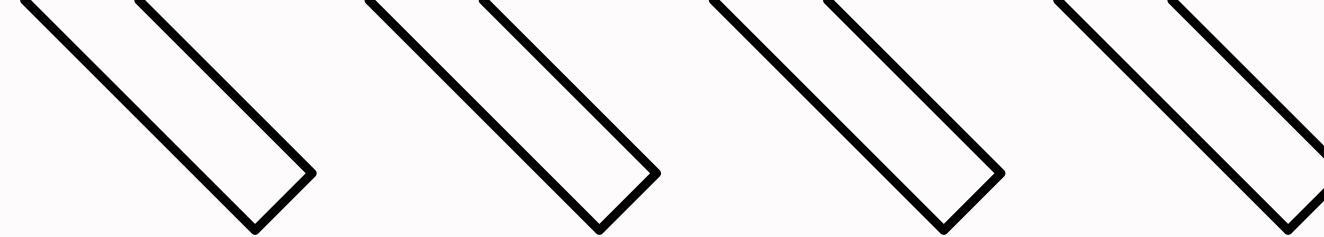
Language- and Platform-independent.

Uses simple HTTP methods

- **POST**: Create a record.
- **GET**: Read a record.
- **PUT**: Update a record.
- **DELETE**: Delete a record.

WHAT DOES REST MEAN?





REST EXAMPLE- GET



FOUR LOOP |

A server with information about users.

The GET method is used to retrieve resources.

- *GET /users*
- *GET /users/2*
- *GET /users/pages/1*
- *GET /users/gender/female*
- *GET /users/age/18*
- *GET /users/???*
- *GET /users/2/name*
- *GET /users/2/pets*



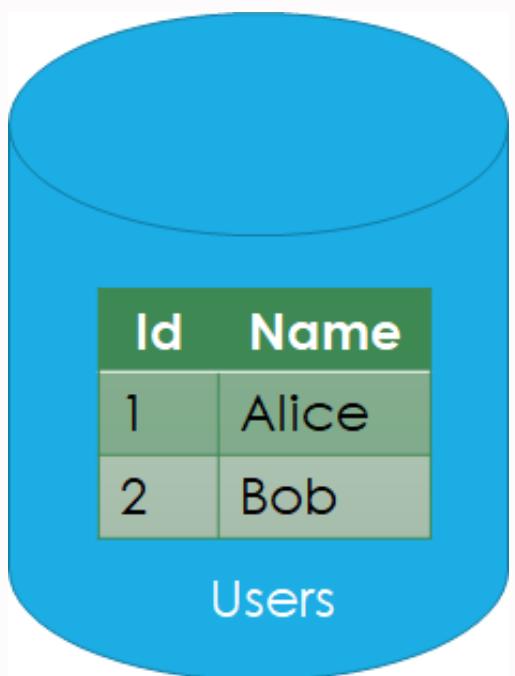


FOUR LOOP |

REST EXAMPLE-GET

```
GET /users HTTP/1.1  
Host: the-website.com  
Accept: application/json
```

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Content-Length: 66
```



A server with information about users.
The GET method is used to retrieve resources.
Specified by the Accept header!

```
[  
  {"id": 1, "name": "Alice"},  
  {"id": 2, "name": "Bob"}]  
]
```

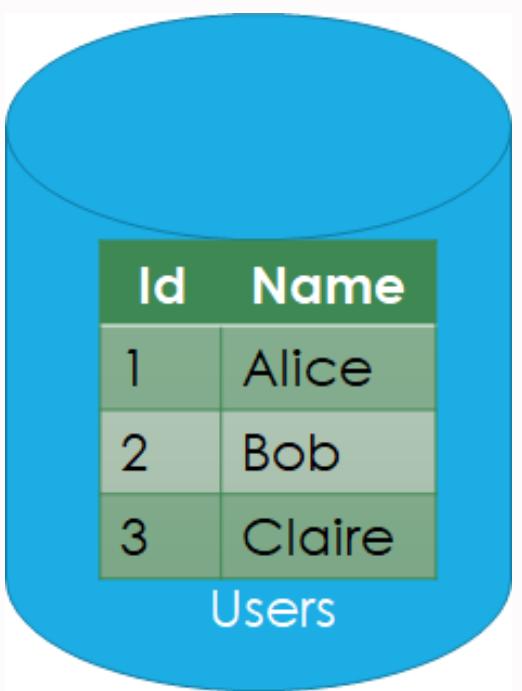




FOUR LOOP |

REST EXAMPLE- POST

```
HTTP/1.1 201 Created  
Location: /users/3  
Content-Type: application/json  
Content-Length: 28  
  
{"id": 3, "name": "Claire"}
```



A server with information about users.
The POST method is used to create resources.

```
POST /users HTTP/1.1  
Host: the-website.com  
Accept: application/json  
Content-Type: application/xml  
Content-Length: 49
```

```
<user>  
  <name>Claire</name>  
</user>
```

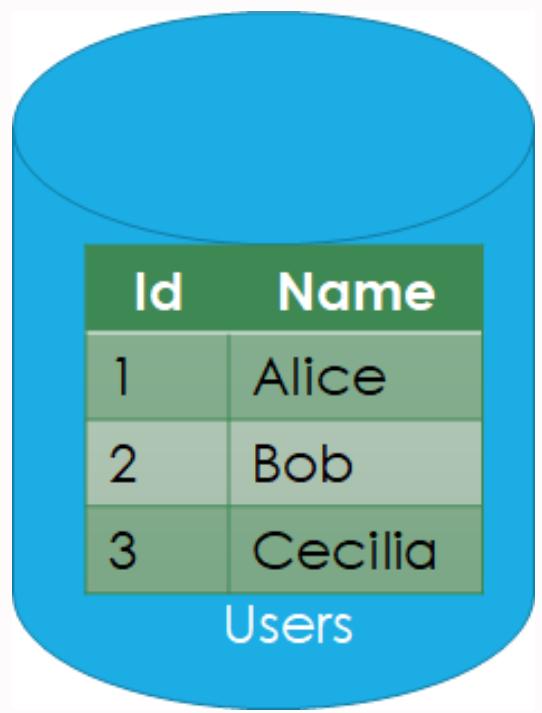




FOUR LOOP |

REST EXAMPLE-PUT

HTTP/1.1 204 No Content



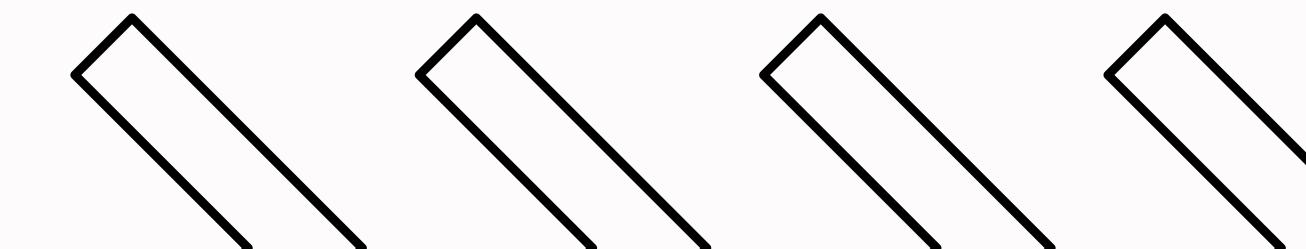
A server with information about users.

The PUT method is used to update an entire resources.

PUT can also be used to create a resource if you know which URI it should have in advance

```
PUT /users/3 HTTP/1.1
Host: the-website.com
Content-Type: application/xml
Content-Length: 52

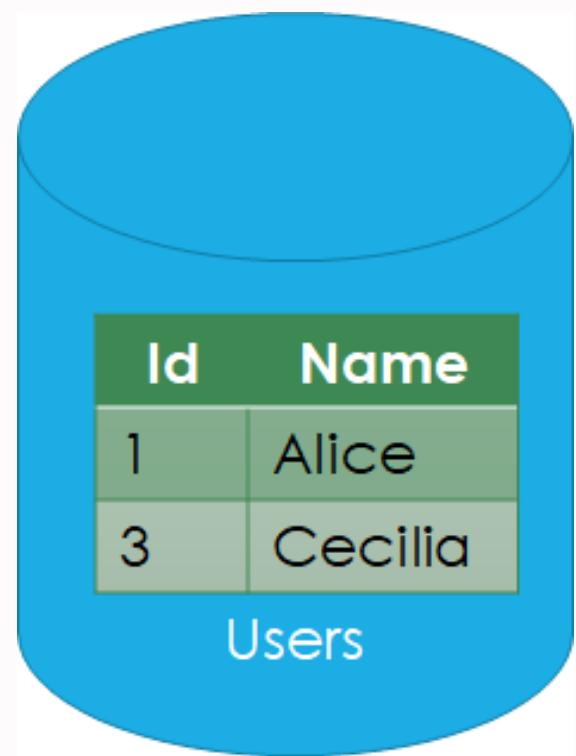
<user>
  <id>3</id>
  <name>Cecilia</name>
</user>
```





FOUR LOOP |

REST EXAMPLE-DELETE

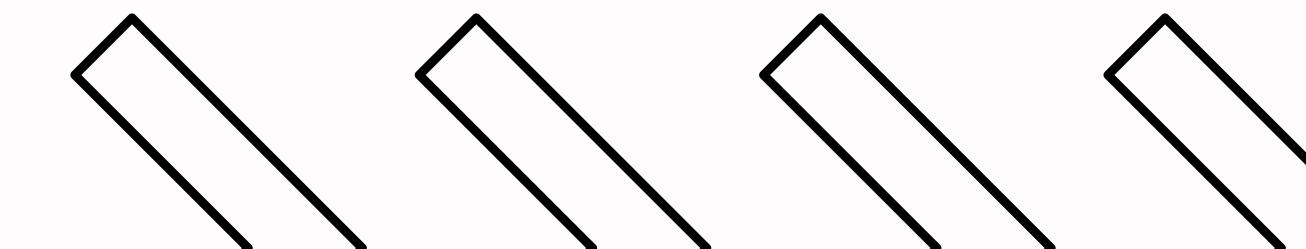


A server with information about users.
The DELETE method is used to DELETE a resource.

DELETE /users/2 HTTP/1.1

Host: the-website.com

HTTP/1.1 204 No Content

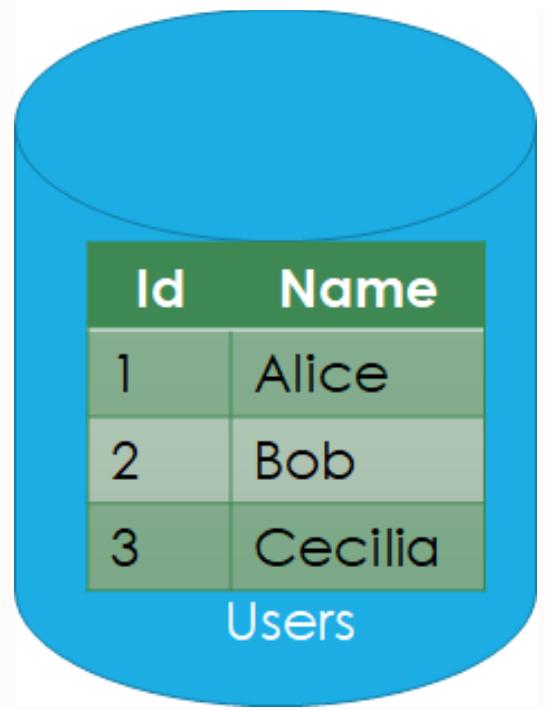




FOUR LOOP |

REST EXAMPLE-ERROR

```
GET /users/999 HTTP/1.1  
Host: the-website.com  
Accept: application/json
```



A server with information about users.

What if something is wrong?
Use the HTTP status codes to Indicate success or Failure.
Optionally include error messages in the response Body.

```
HTTP/1.1 404 Not Found
```

HTTP STATUS CODE:
1xx - Informational
2xx - Success
3xx - Redirection
4xx - Client Error
5xx - Server Error



DESIGNING A REST API

Make it easy as possible to use by other developers

FACEBOOK

Always return 200 OK.

GET /v2.7/{user-id}

GET /v2.7/{post-id}

GET /v2.7/{user-id}/friends

GET /v2.7/{object-id}/likes

TWITTER

Only use GET and POST.

GET /1.1/users/show.json?

user_id=2244994945

POST /1.1/favorites/destroy.json?

id=243138128959913986

ADVANTAGES OF REST API



FOUR LOOP I

Simplicity and Ease of use

- a.Simple and Easy to use
- b.Uses standard HTTP Method
- c.Follow a straightforward request-response model
- d.Resources are identified using URI (Uniform Resource Identifier)

Scalability

- a.The ability of the system to handle increasing number of Users, Requests or Resources
- b.Stateless Communication
- c.Horizontal Scaling

Flexibility

- a.Content Negotiation
- b.Support for Multiple Representations

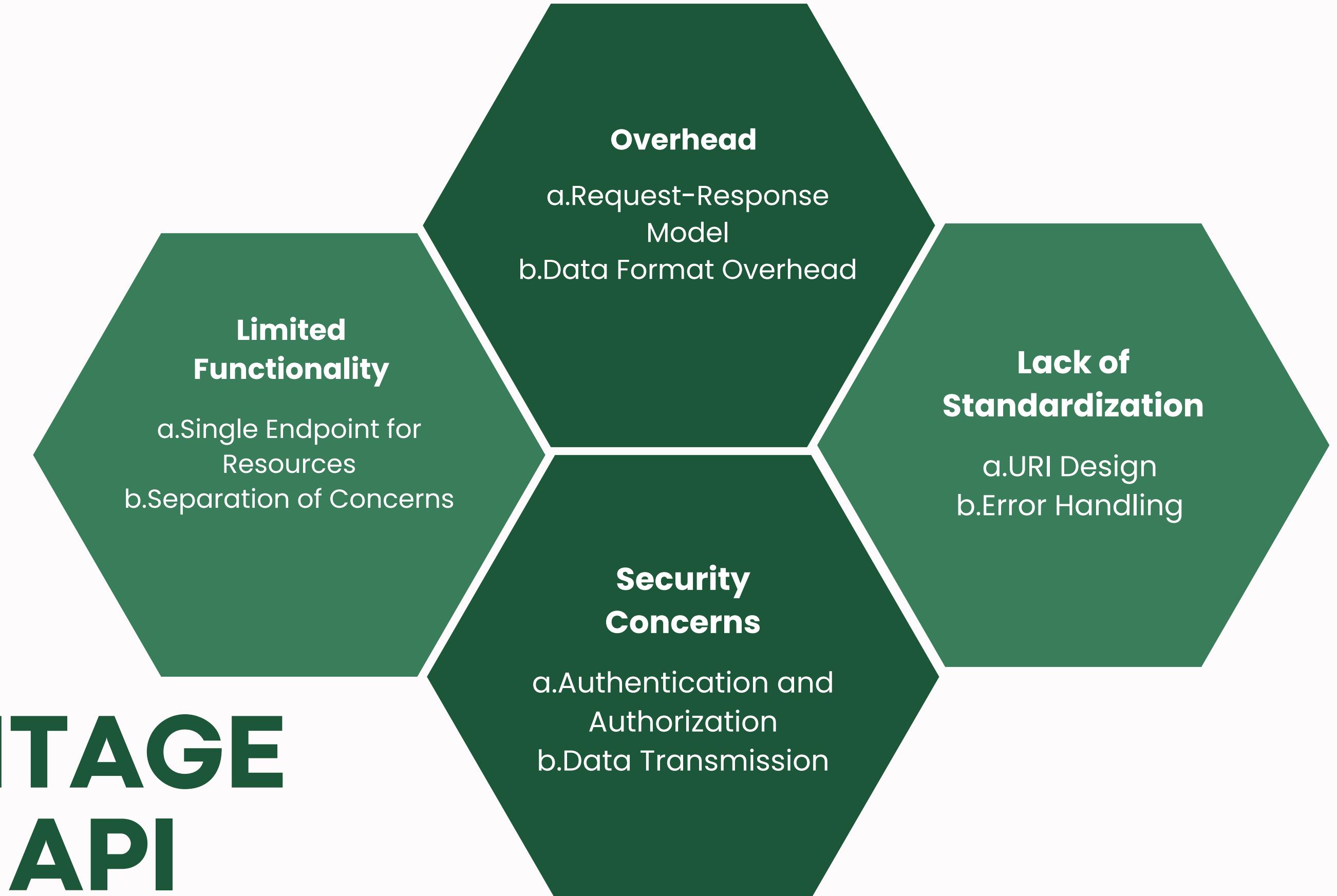
Statelessness

- a.No Session State

DISADVANTAGE OF REST API



FOUR LOOP |





FOUR LOOP |

APPLICATION INTEGRATION

08



APPLICATION INTEGRATION

Application integration is the process of enabling independently designed applications to work together. Commonly required capabilities include:

- Keeping separate copies of data (in independently designed applications) consistent
- Orchestrating the integrated flow of multiple activities performed by disparate applications
- Providing access to data and functionality from independently designed applications through what appears to be a single user interface or application service.

APPLICATION INTEGRATION CONCEPTS



FOUR LOOP |

Components for Successful Integration

Application Programming Interface (API)

Definition: Set of functions specifying how software components interact.
Facilitates quick and easy access to functionalities of other software.

Data Mapping

Specifies information exchange.
Example: Mapping form fields to corresponding datasets.

Events and Actions

Events trigger actions (e.g., payment received).
Actions can be standard or application-specific.

BENEFITS OF APPLICATION INTEGRATION

01

Organizational
Benefits

- Synchronization of data across various clouds.
- Deployment of integration runtimes within multiple clouds.

02

Operational
Benefits

- Access any data anywhere.
- Resolve 'endpoint individuality'.
- Let integrators focus on integration.

APPLICATION INTEGRATION USE CASES

01



Banking

Integration of customer accounts, loan applications, and back-end systems with a mobile app.

Connect devices to monitor production line aspects and integrate with other systems.

Manufacturing

02

03



Healthcare

Integrate hospital patient records with an electronic health record (EHR) system.



MISSION-CRITICAL SYSTEMS INTEGRATION

ERP Systems

- Hub for all business activities.
- Streamlining and automating mission-critical processes.

CRM Platforms

- Maximizing productivity and efficiency.
- Automation of sales, marketing, customer support, and product development functions.

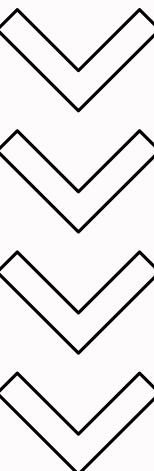
APPLICATION VS. DATA INTEGRATION

APPLICATION INTEGRATION

Links multiple applications at a functional level.
Application data linked in near real-time for dynamic and adaptable applications.

DATA INTEGRATION

Locating and retrieving information from various sources.
Batch-based data processing for analysis over time.





APPLICATION INTEGRATION STYLES

API-led Integration

- Expose and discover business IT assets rapidly and securely.

Service-Oriented Architecture (SOA)

- Integrating applications using a communication layer or 'bus'.
- Orchestrating integration flows triggered by events in one application to the next.