

Skillbox

REST

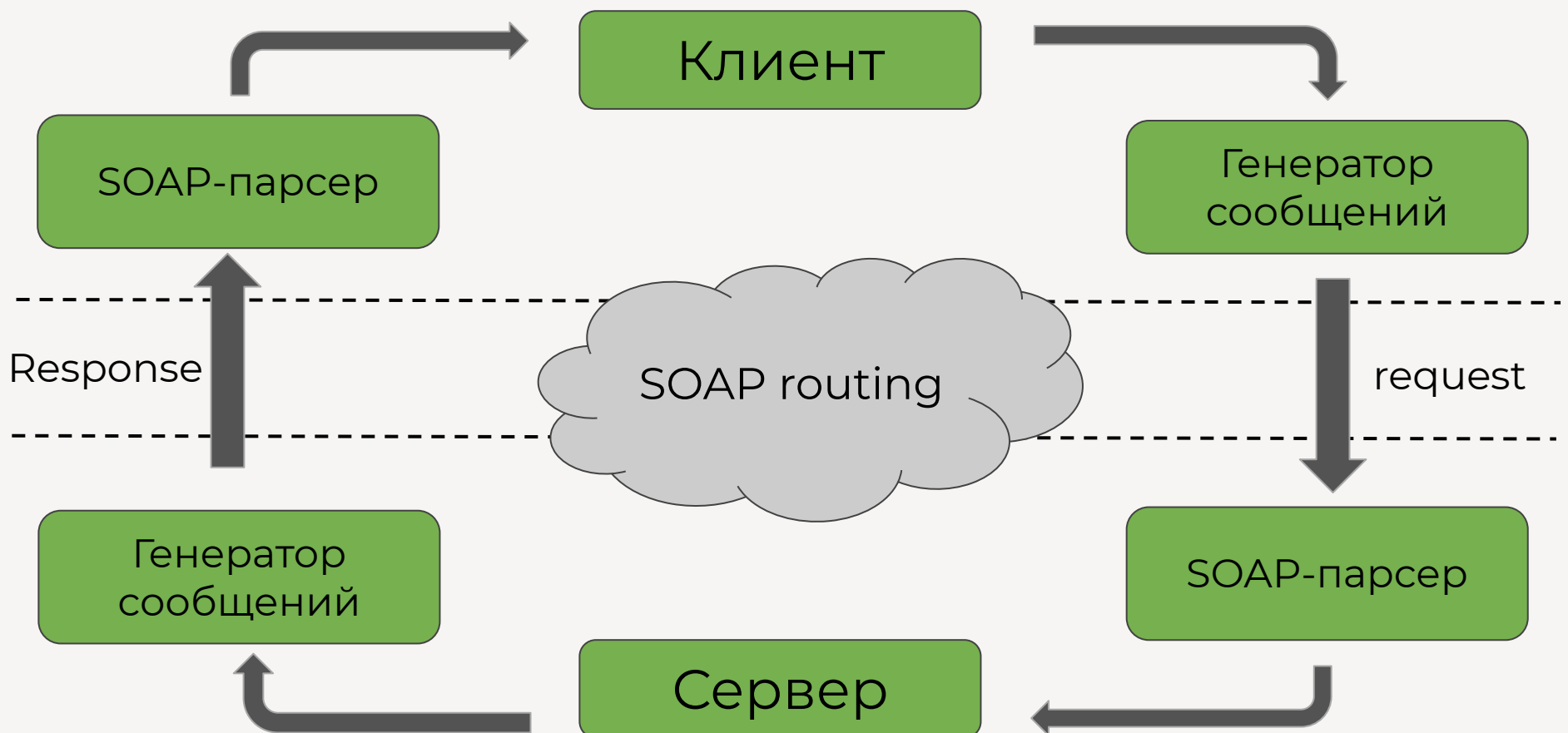
Кирилл Кошаев

Java-разработчик в «Газпром информ»

SOAP

Simple Object Access Protocol — протокол обмена структурированными сообщениями в распределенной вычислительной среде.

Предназначен для реализации удаленного вызова процедур (RPC)



SOAP-message

Тело SOAP-сообщения представляет собой xml-структуру, состоящую из 4 элементов: Envelope, Header, Body и Fault

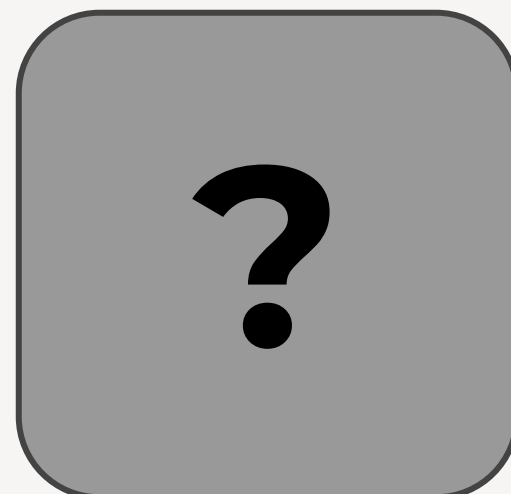
```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

    <soap:Header>
        ....
    </soap:Header>

    <soap:Body>
        ...
        <soap:Fault>
            ...
        </soap:Fault>
        ...
    </soap:Body>

</soap:Envelope>
```

Альтернативы SOAP



Появление REST

Рой Филдинг — американский ученый в области computer science и один из основных авторов спецификации HTTP, а также создатель REST. Помимо этого, является соучредителем проекта Apache HTTP Server.



Что такое REST ?

REST — это архитектурный стиль, определяющий набор согласованных ограничений и принципов, которых необходимо придерживаться при создании слабосвязанных приложений, работающих в сети по HTTP-протоколу.

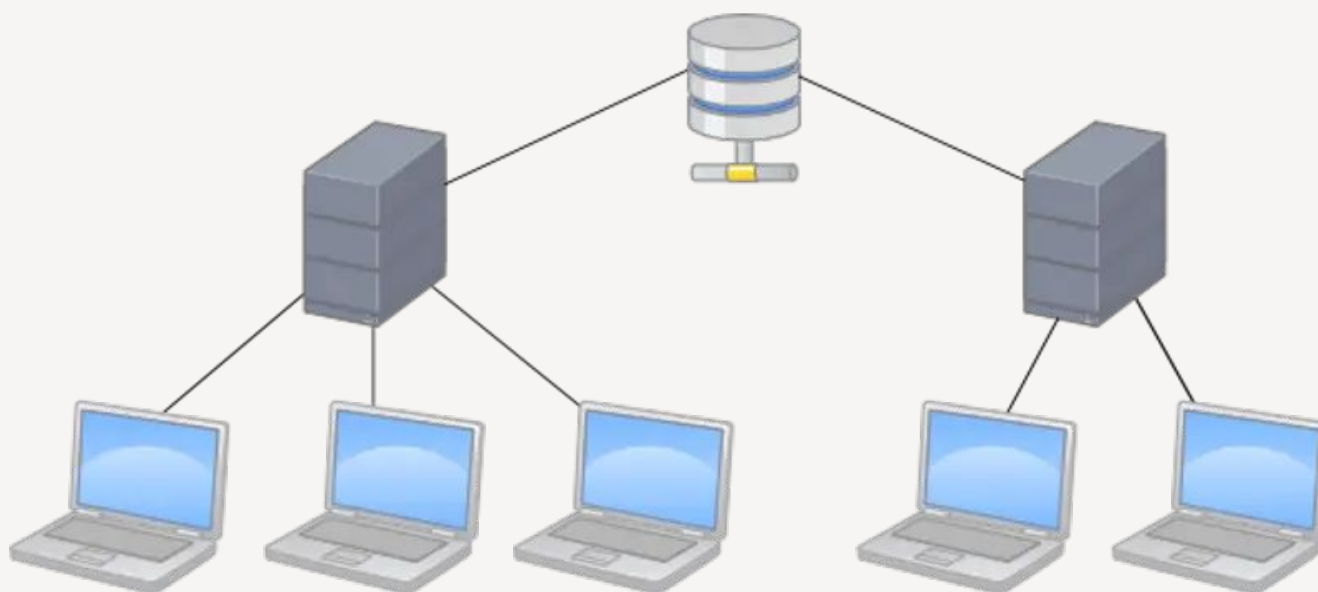
- Клиент-серверная архитектура
- Отсутствие состояния
- Кэшируемость
- Единообразие интерфейсов
- Многослойность системы
- Код по запросу



{ REST }

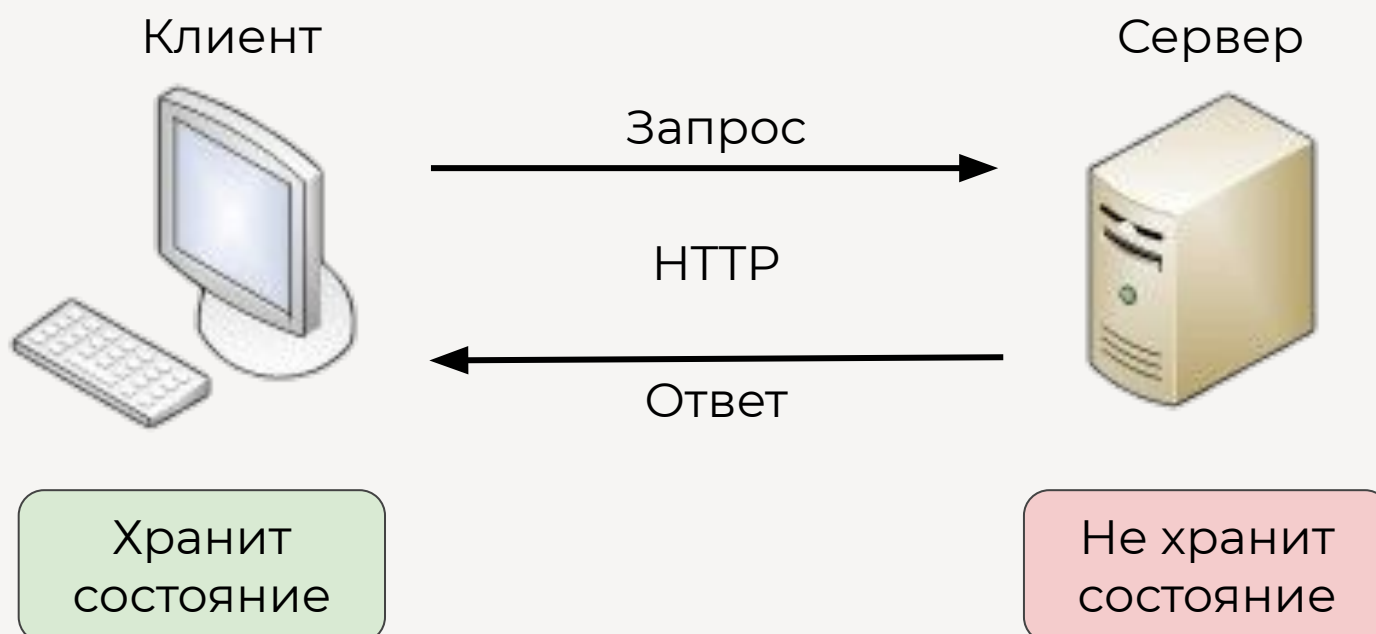
Принцип клиент-серверной архитектуры

REST-приложение должно придерживаться клиент-серверной архитектуры.



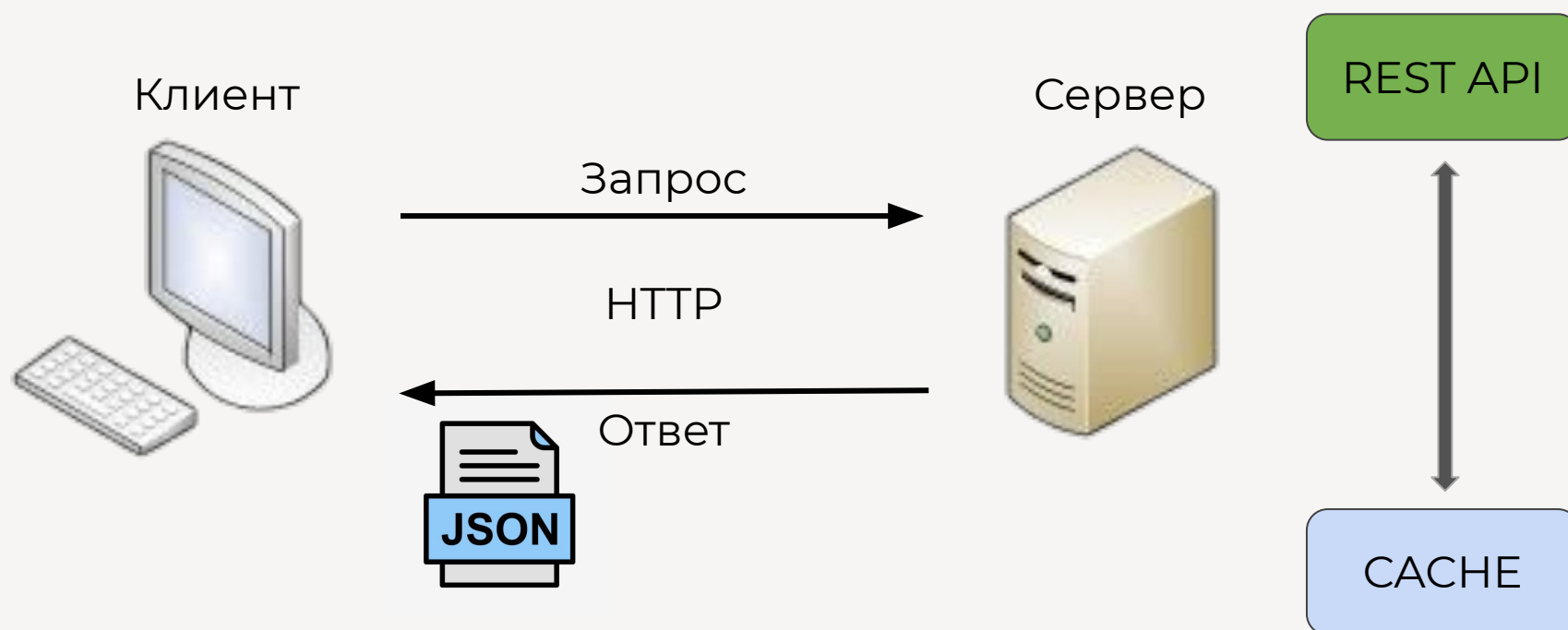
Отсутствие состояния

Сервер не должен хранить какую-либо информацию о клиенте на своей стороне. При этом хранение состояния сессии и подобного происходит на стороне клиента и каждый раз, когда у него возникает необходимость обновить текущее состояние, он должен инициировать отправку соответствующего запроса на сервер.



Кэшируемость

Данные, содержащиеся в ответе сервера на запрос клиента, должны быть явно или неявно отмечены как кэшируемые или некэшируемые. Если ответы кэшируются, то кэш клиента получает право повторно использовать данные этих ответов для их использования при последующих эквивалентных запросах.



Принцип единообразия интерфейса

REST-приложения должны использовать и предоставлять унифицированный интерфейс доступа к своим ресурсам.

- Идентификация ресурсов в запросе
- Манипулирование ресурсами через представления
- Достаточная информативность передаваемых сообщений
- Гипермедиа как двигатель состояния приложения (HATEOAS)

Многослойность системы

Принцип многослойности предлагает использовать многоуровневую системную архитектуру, в которой API-интерфейсы развертываются, скажем, на сервере *A*, данные хранятся на сервере *B*, а запросы проходят аутентификацию, например, на сервере *C*. При этом клиент, обычно, не может определить, подключен ли он напрямую к конечному серверу или к посреднику по пути.



Слой 1

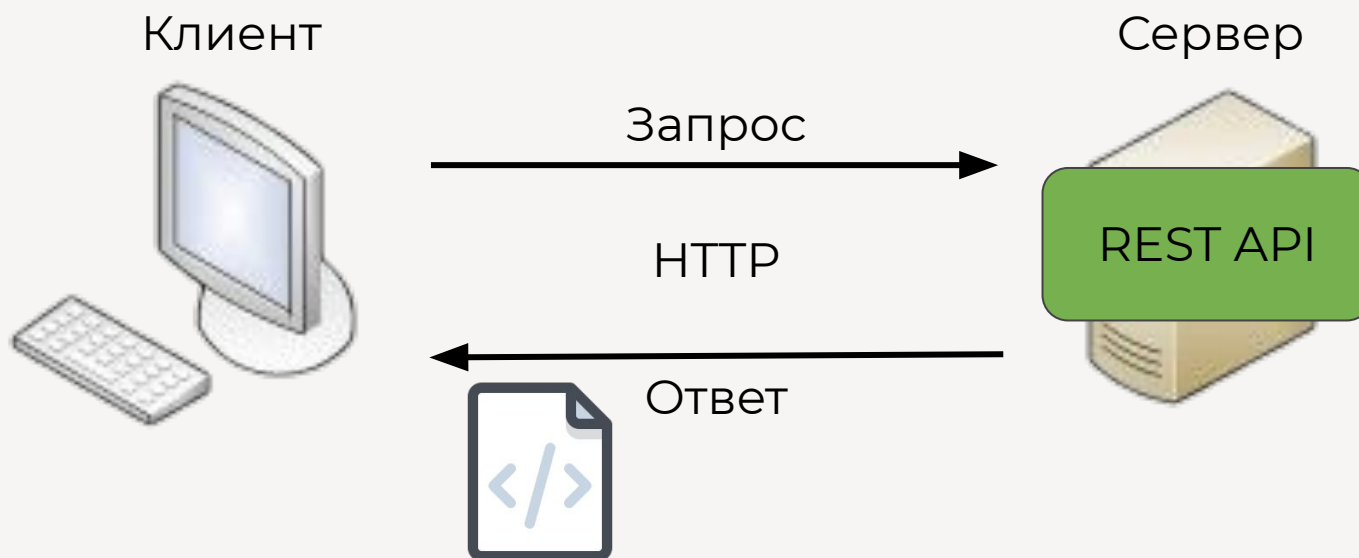
Слой 2

...

Слой 3

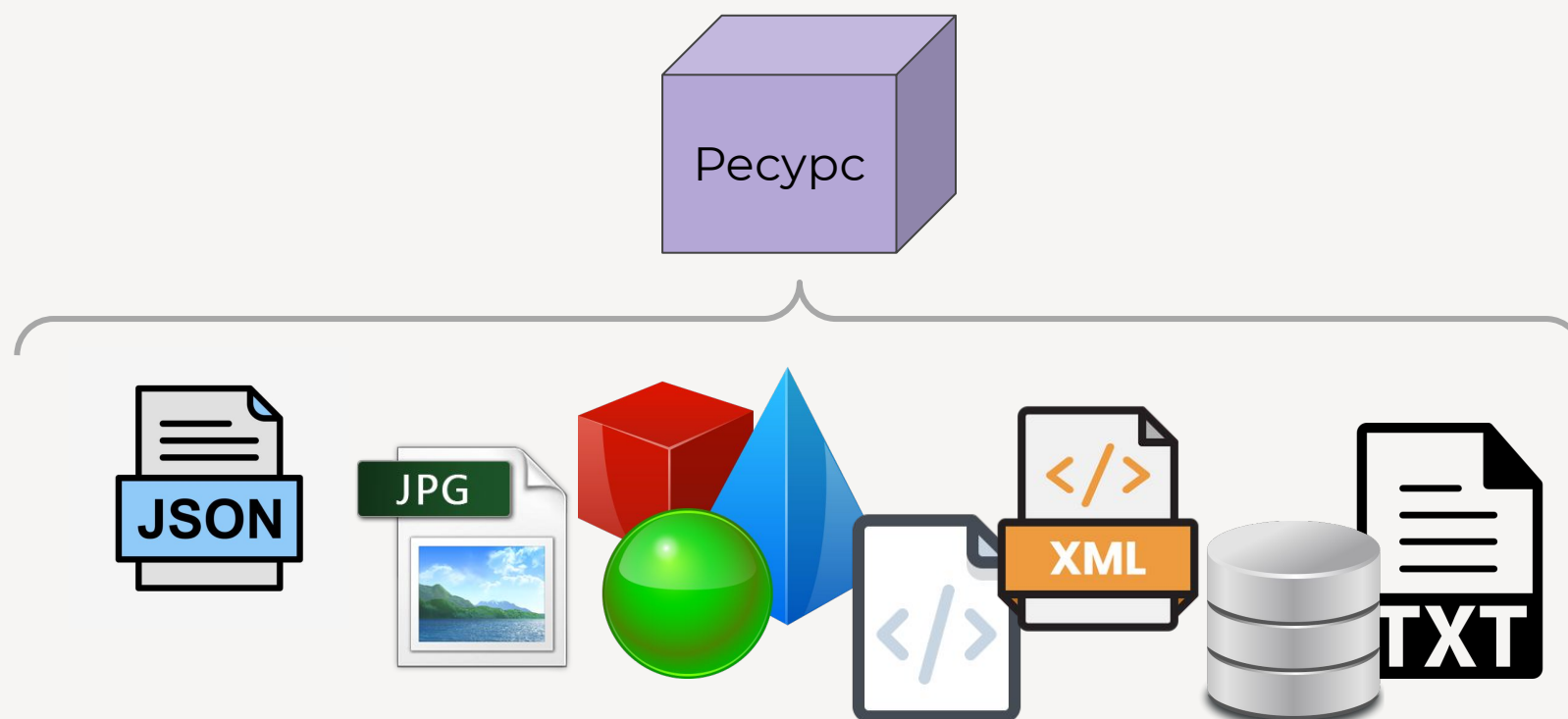
Код по запросу

Принцип кода по запросу не является обязательным для достижения RESTful. В большинстве случаев вы будете отправлять статические представления ресурсов в форме XML или JSON. Но если понадобится, то вы можете возвращать в качестве ответов даже исполняемый код.



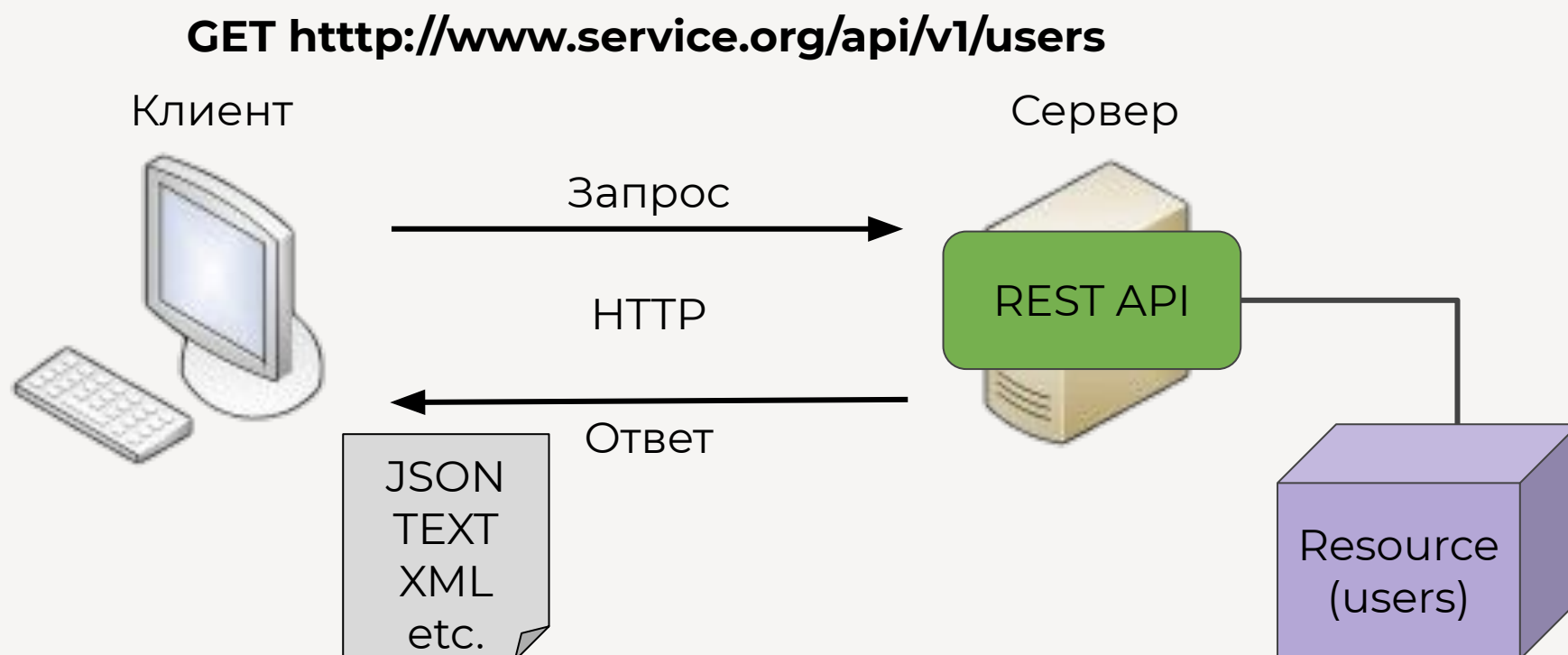
REST resource

Ключевой абстракцией информации в REST является понятие ресурса. Любая информация, которая может быть наделена именем, может быть ресурсом: документ или изображение, сервис, например, «погода в городе N», коллекция других ресурсов, не виртуальный объект, например, автомобиль и тому подобное.



Строительные блоки REST API

Основными строительными блоками REST API являются ресурсы и их URI-и, HTTP-методы и заголовки, а также параметры запросов и коды ответов .



**Спасибо
за внимание!**