



Komunikacja międzysystemowa

REST API

06.12.2017

Karol Buler

0 czym będzie?

- 0 komunikacji ogólnie
- Application programming interface (API)
- Wybrane metody komunikacji
- REST API
 - JavaScript Object Notation (JSON)
 - Protokół HTTP – wybrane zagadnienia
 - CRUD
- SwaggerUI – live demo



To nie jest...

...wszystko

...jedyne słuszne rozwiązanie





O komunikacji ogólnej

Komunikacja

- ruch środków lokomocji między odległymi od siebie miejscami; też: drogi, szlaki i środki lokomocji
- przekazywanie i odbieranie informacji w bezpośrednim kontakcie z drugą osobą
- możliwość przedostania się z jednego pomieszczenia lub miejsca do drugiego
- przepływ informacji między urządzeniami, np. telefonami lub komputerami

Przykład psa?



Przykład psa?

pogłaszcz();



Przykład psa?

pogłaszcz();

nakarm();



Przykład psa?

pobawSie();

pogłaszcz();

nakarm();



Przykład psa?

pobawSie();

pogłaszcz();

nakarm();



ugryź();

Interfejs

Interfejs

updateUser()

getUser()

addUser()

Serwer



Interfejs

updateUser()

getUser()

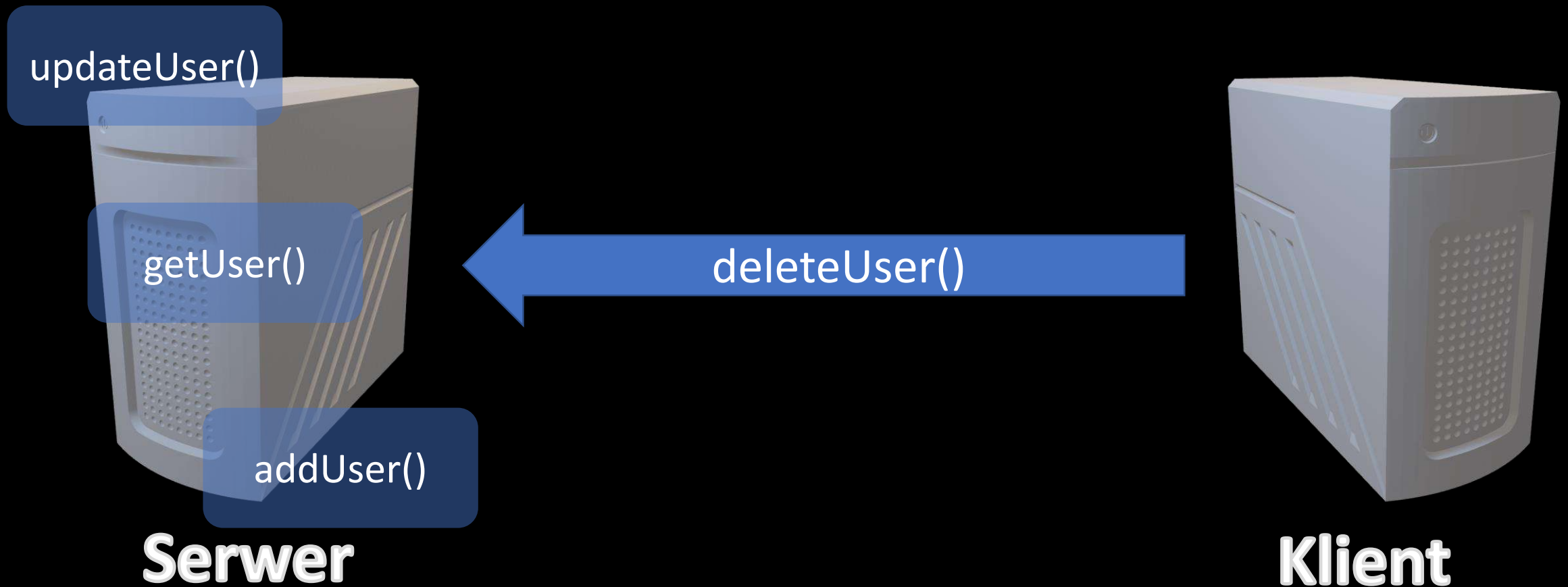
addUser()

Serwer

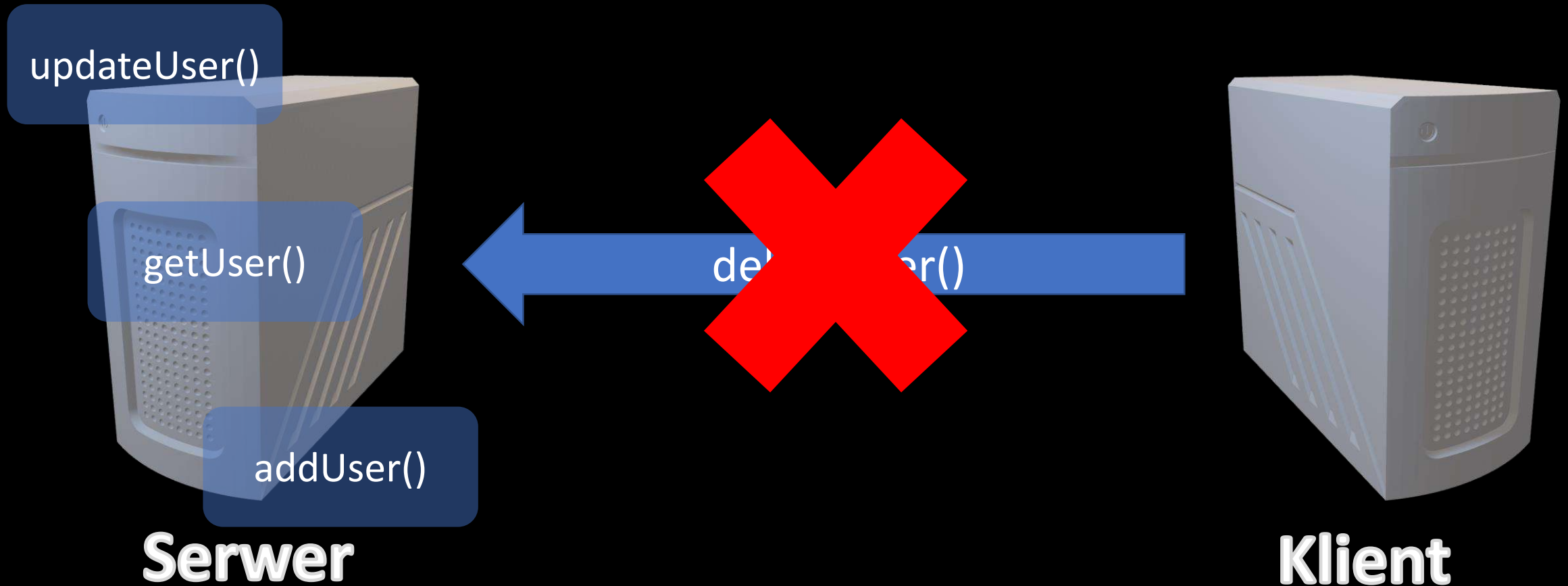


Klient

Interfejs



Interfejs





Application programming interface (API)

API

- Programiści wiedzą jakich metod mogą użyć do komunikacji z innym systemem
- Dzięki API aplikacje „rozmawiają” między sobą bez żadnej wiedzy ani ingerencji użytkownika
- API dostarcza informacje nie tylko o możliwości wykonania danej operacji, ale też o sposobie w jaki ją wykonać, oraz o rezultacie jej wykonania
- NIE jest interfejsem użytkownika, a interfejsem „software-to-software”



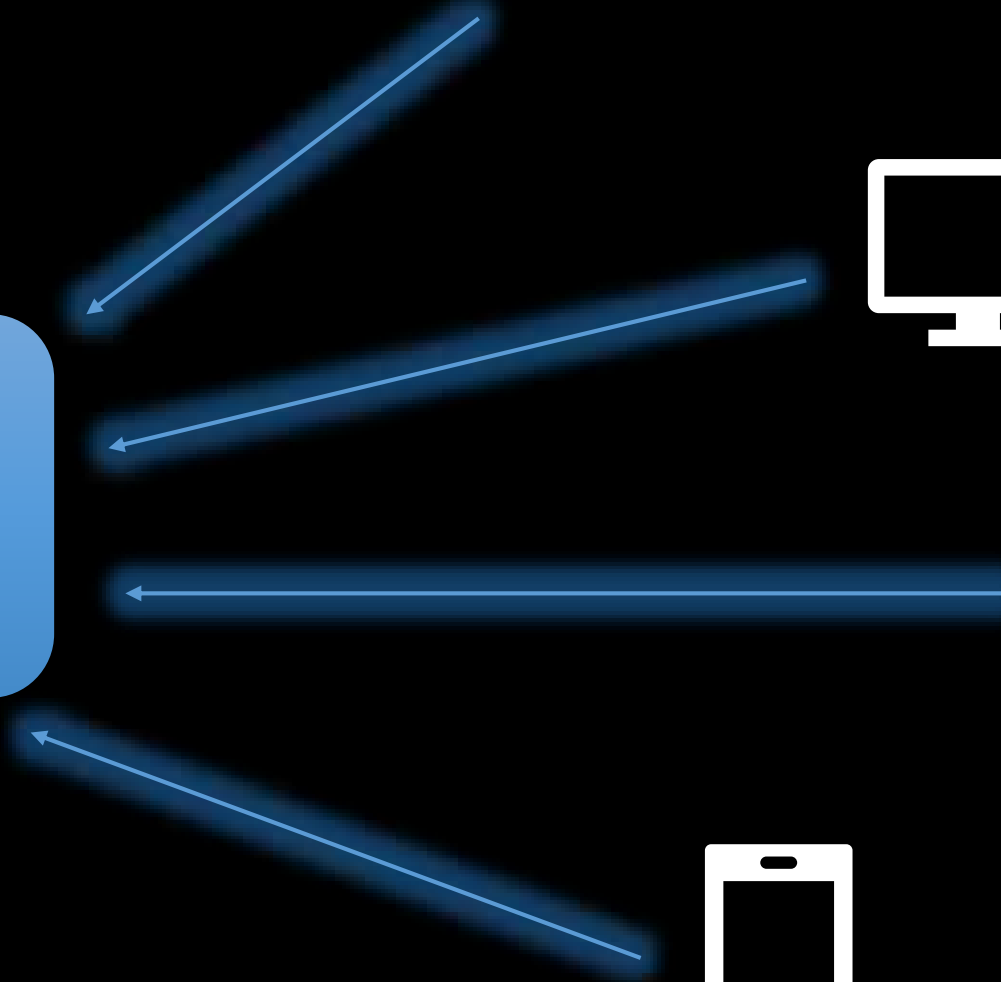
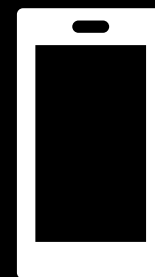
API c.d.



Server



Klient

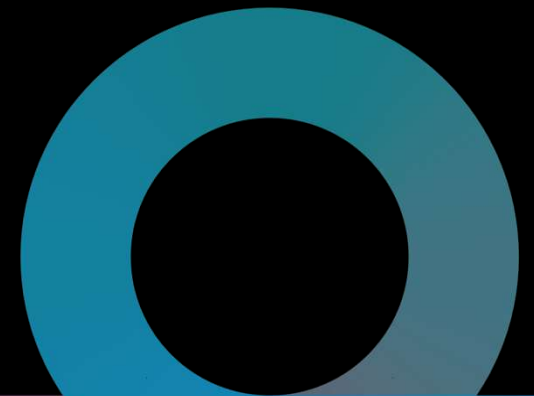
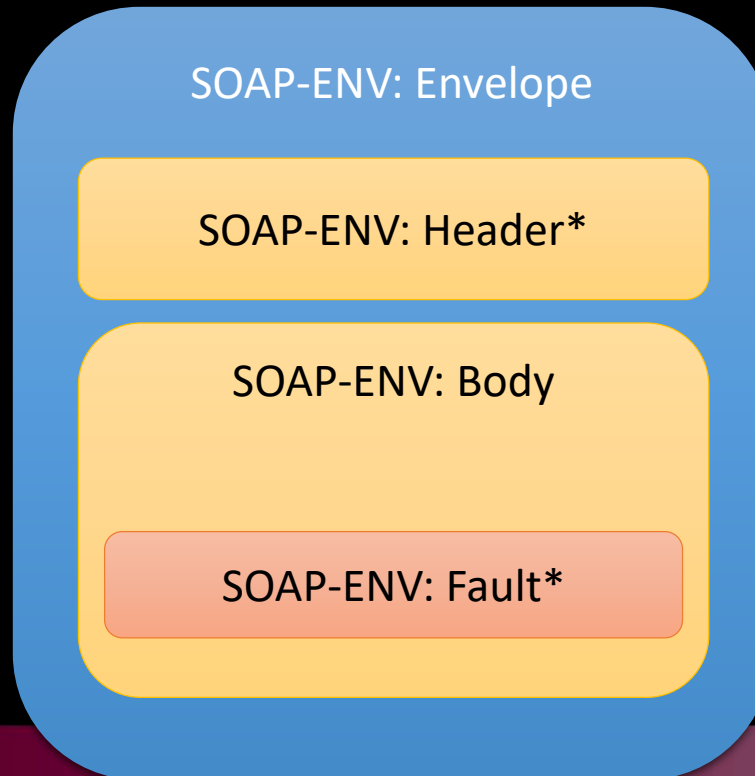




Metody komunikacji

SOAP - Simple Object Access Protocol

- wymiana wiadomości opartych o język znaczników XML
- transmisja najczęściej odbywa się po HTTP (nie tylko)
- jasno opisana budowa



SOAP c.d.

```
<?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:Body>
    <m:GetPrice xmlns:m=
      "https://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>

</soap:Envelope>
```

Request

```
<?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:Body>
    <m:GetPriceResponse xmlns:m=
      "https://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
    </m:GetPriceResponse>
  </soap:Body>

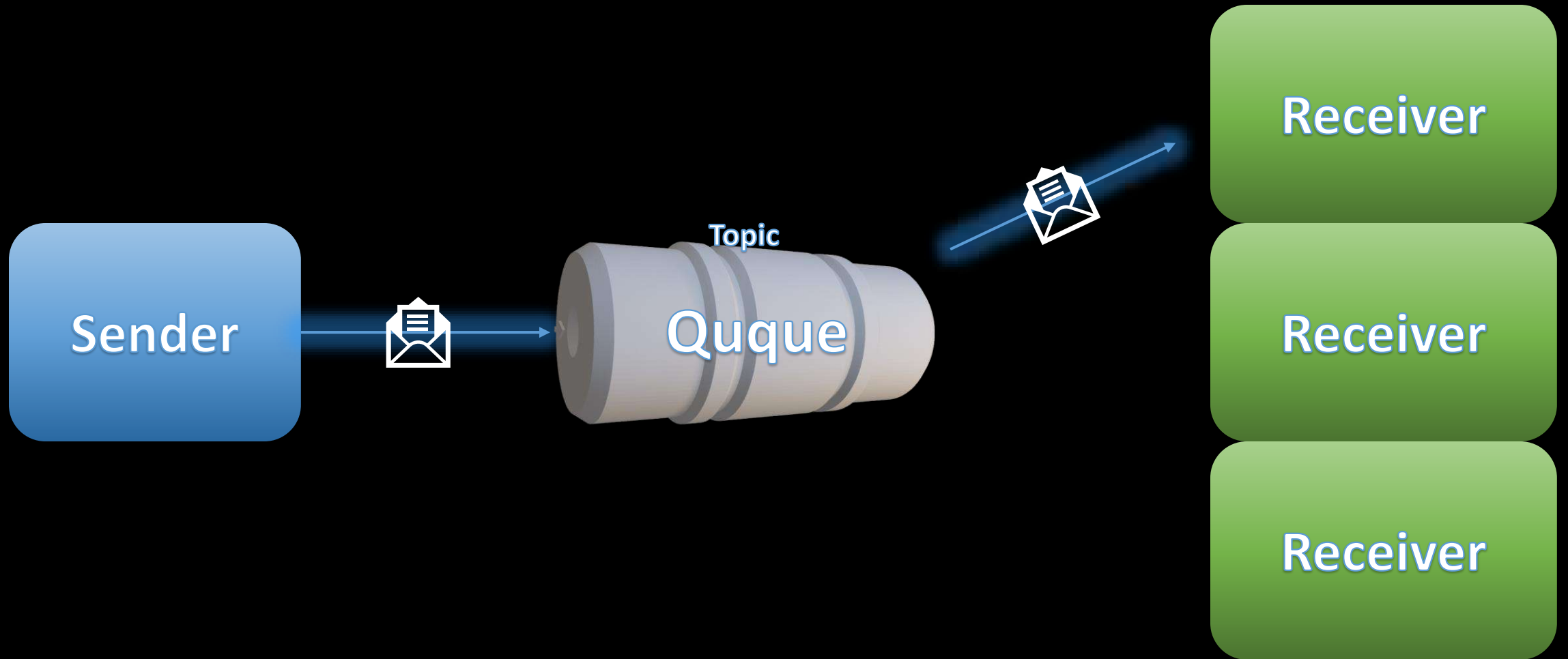
</soap:Envelope>
```

Response

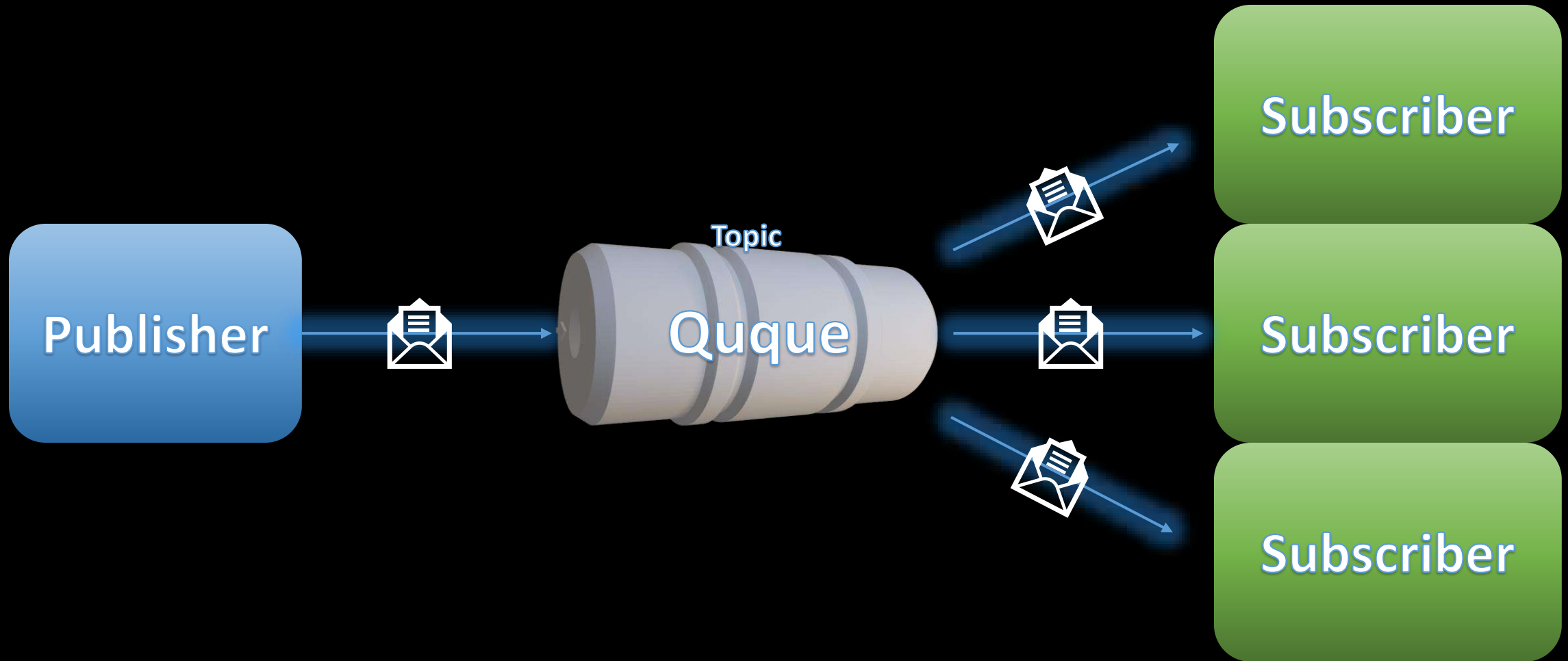
Message Broker
= Kolejka



Message Broker - Kolejka Point-to-Point



Message Broker - Kolejka Publish/Subscribe



Message Broker - popularne

MQTT





REST API

REST

- Nie jest protokołem !!
- Zbiór dobrych praktyk tworzenia architektury aplikacji rozproszonych (wzorzec)
- Najczęściej kojarzony z HTTP, choć nie jest z nim ściśle związany
- Nie jest standardem, lecz zbiorem ograniczeń.
- API napisane wg zasad REST nazywane RESTful API

REST - Zasady

Zasoby	Wyeksponowane przez łatwo zrozumiałe identyfikatory katalogów (URI)
Obiekty	Reprezentowane i przekazywane w postaci JSON
Zapytania	Wykorzystują jawnie metody HTTP (GET, POST, PUT, DELETE)
Interakcje	Są całkowicie bezstanowe. Kontekst klienta nie jest przechowywany na serwerze między żądaniami.

REST – CRUD, URI(sy), HTTP

CRUD – create, retrieve, update, delete

CRUD	HTTP	URI
create	POST	/users
retrieve	GET	/users /users/{userId}
update	PUT	/users/{userId}
delete	DELETE	/users/{userId}

JSON – JavaScript Object Notation

- Bardzo lekki, tekstowy, niezależny od języka format wymiany danych
- Bardzo dobrze sprawdza się w opisie i serializacji danych strukturalnych
- Obsługuje 4 typy prymitywne:
 - String
 - Number
 - Boolean
 - Null
- Oraz 2 typy strukturalne:
 - Object (JSON Object)
 - Array

JSON – JavaScript Object Notation

```
<?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:Body>
    <m:GetPrice xmlns:m=
      "https://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>

</soap:Envelope>
```

XML

```
{
  "envelope": {
    "body": {
      "getPrice": {
        "item": "Apples"
      }
    }
  }
}
```

JSON

JSON – JavaScript Object Notation

```
<?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:Body><m:GetPrice xmlns:m="https:  
//www.w3schools.com/prices"><m:Item>Apples</m:It  
em></m:GetPrice></soap:Body></soap:Envelope>
```

XML

```
{"envelope":{"body":{"getPrice":{"item":"Apples"  
}}}}
```

JSON

JSON – JavaScript Object Notation

```
{
  "users": [
    {
      "firstname": "John",
      "lastname": "Smith",
      "age": 24,
      "sex": "MALE",
      "hobbies": [
        "cars",
        "music"
      ],
      "boss": {
        "firstname": "Susan",
        "lastname": "Doe",
        "age": 27,
        "sex": "FEMALE"
      }
    },
    {
      "firstname": "Susan",
      "lastname": "Doe",
      "age": 27,
      "sex": "FEMALE",
      "hobbies": [
        "flowers",
        "music"
      ],
      "boss": {}
    }
  ]
}
```



Swagger UI



Swagger?



Dziękuję!

adbglobal.com