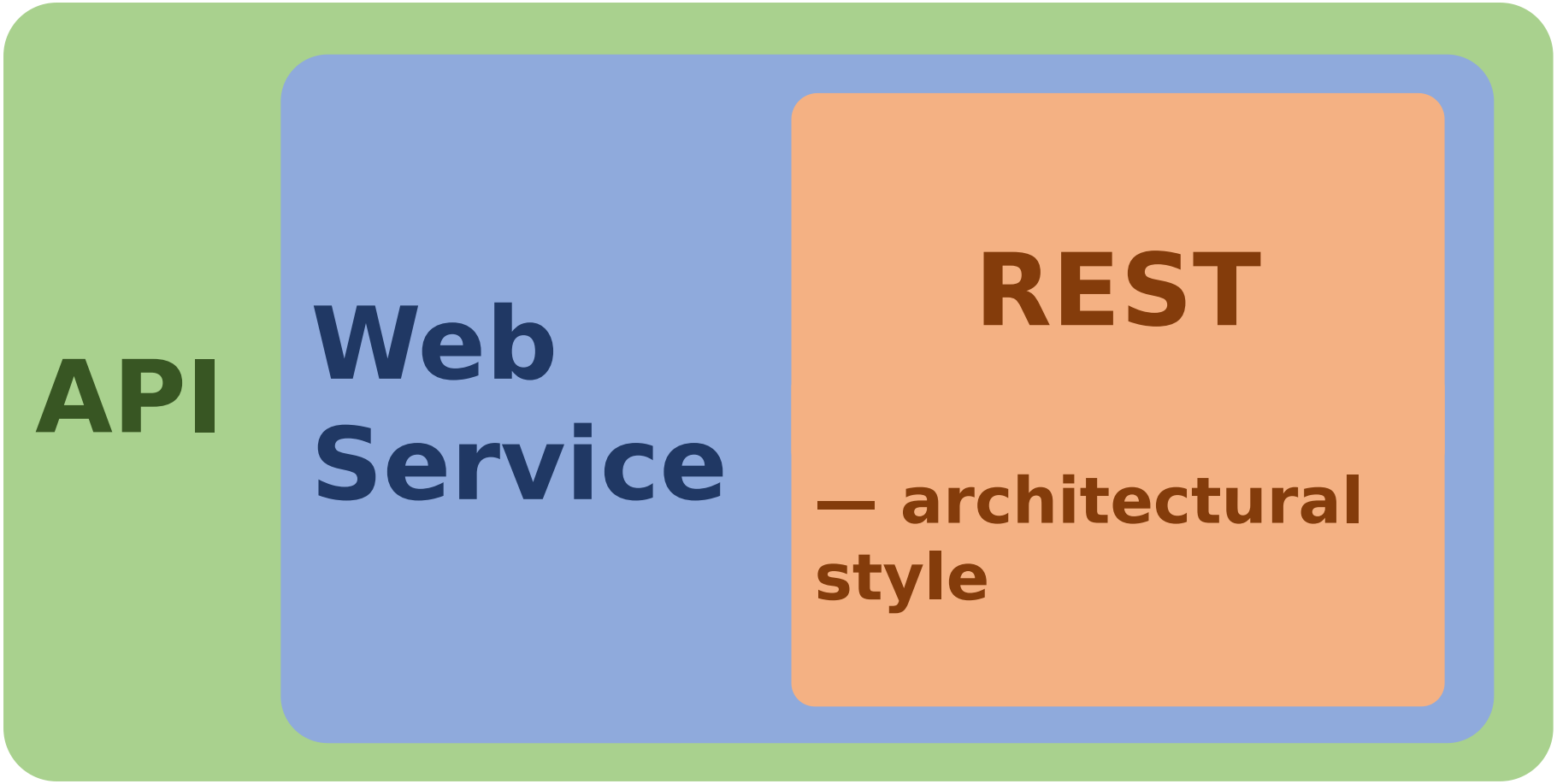
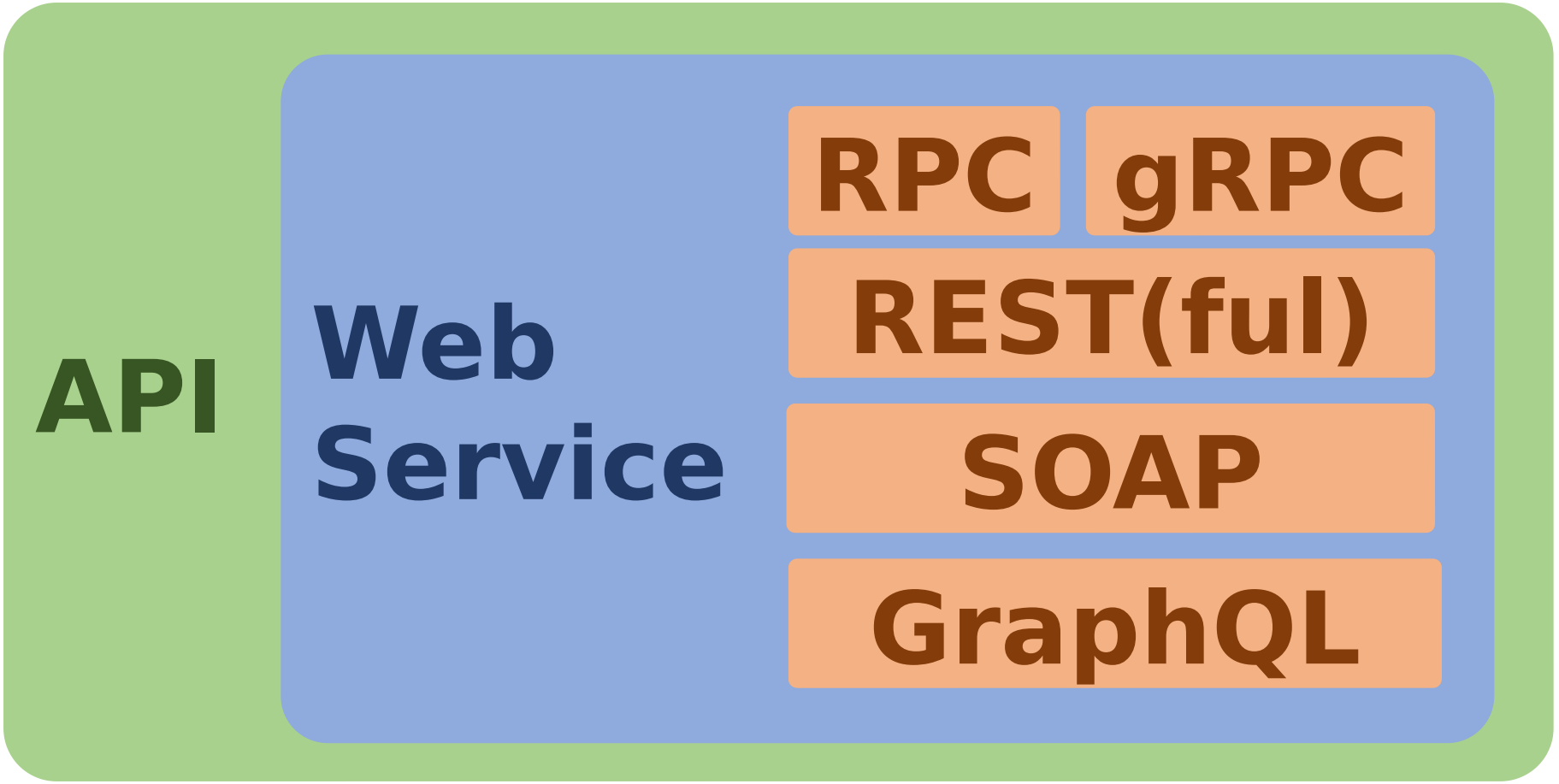
API 101

API — interface published by your app so that other apps could interact with it

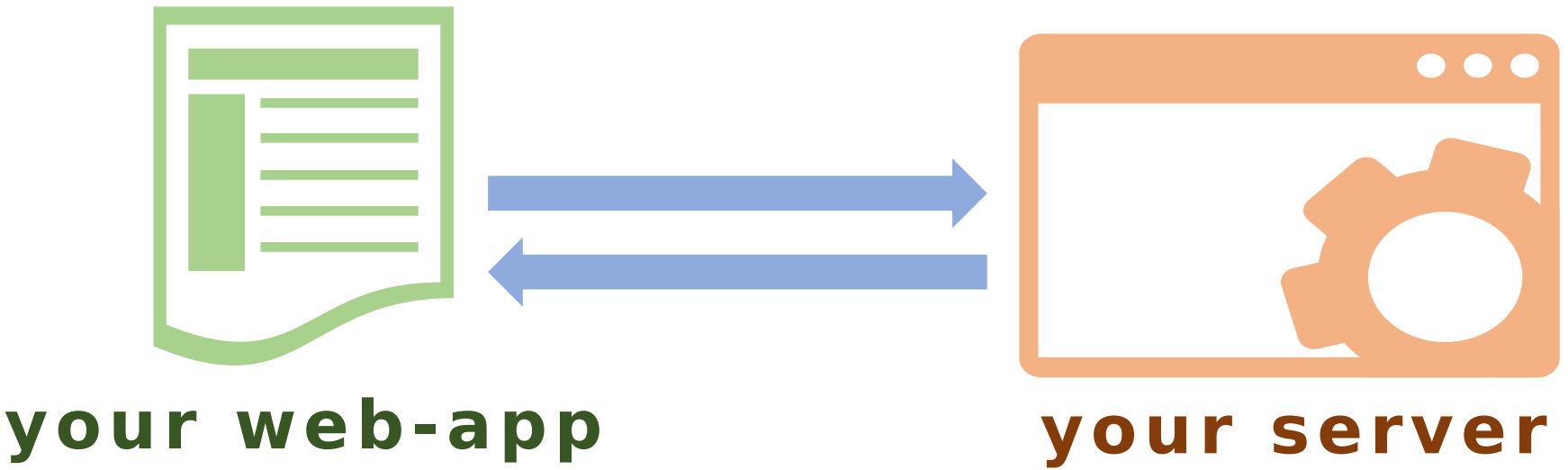

API
Web Service
— API with web-level interactions


API
Web Service
REST
— architectural style

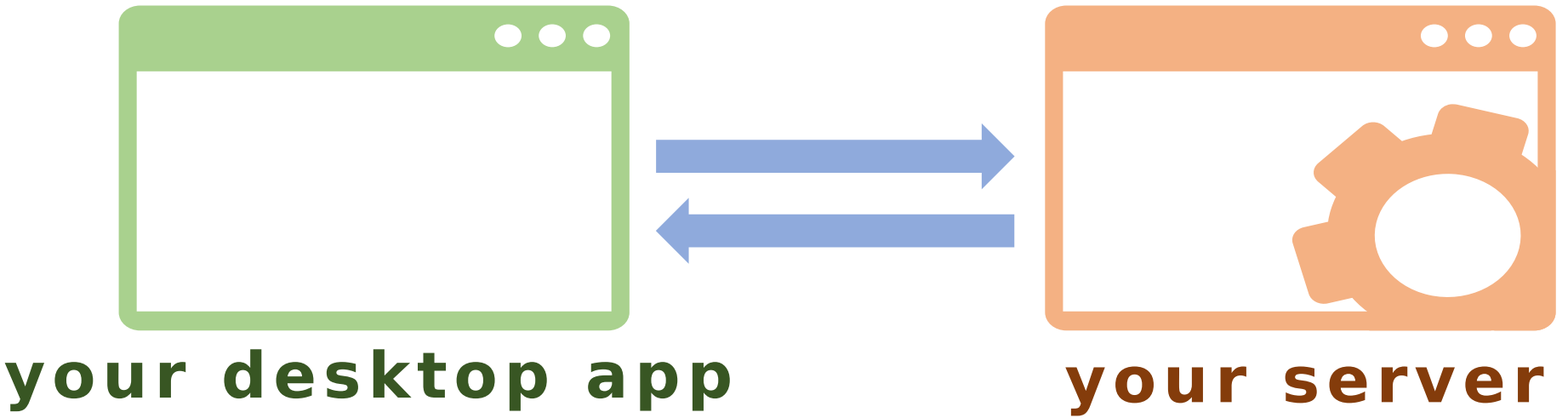

API
Web Service
RPC
gRPC
REST(ful)
SOAP
GraphQL

Typical interactions

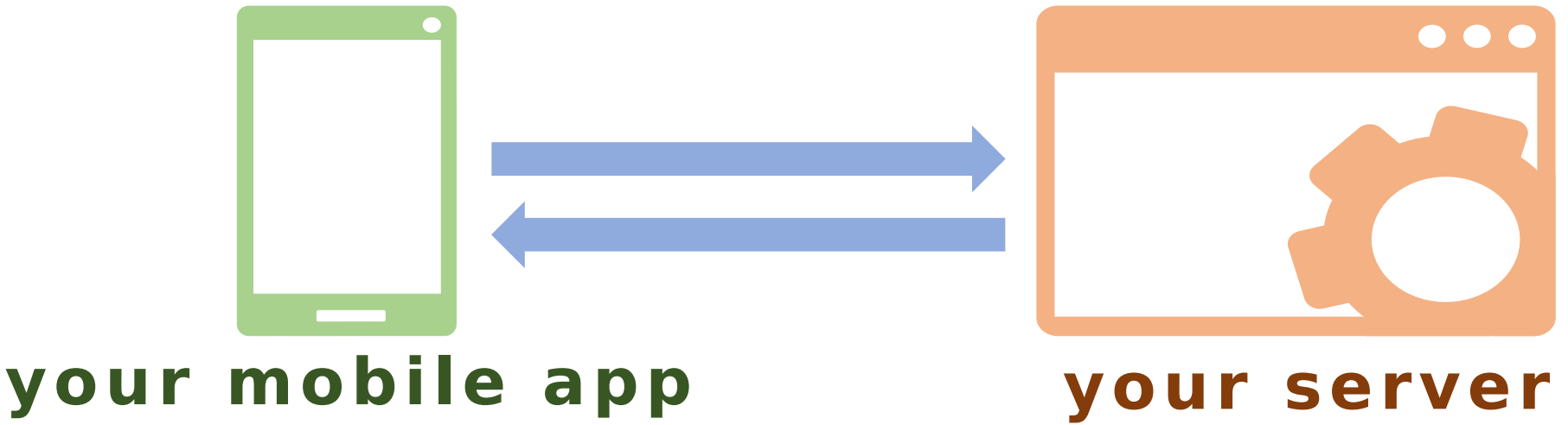
Internal



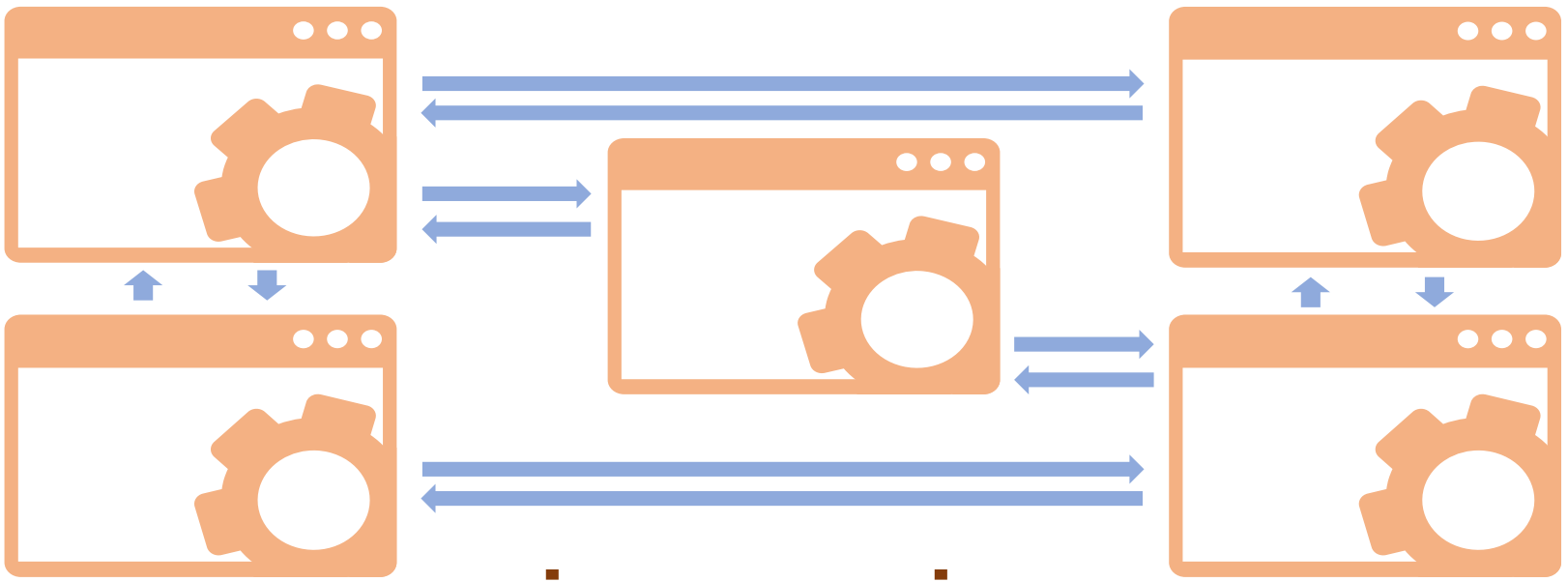
Internal



Internal



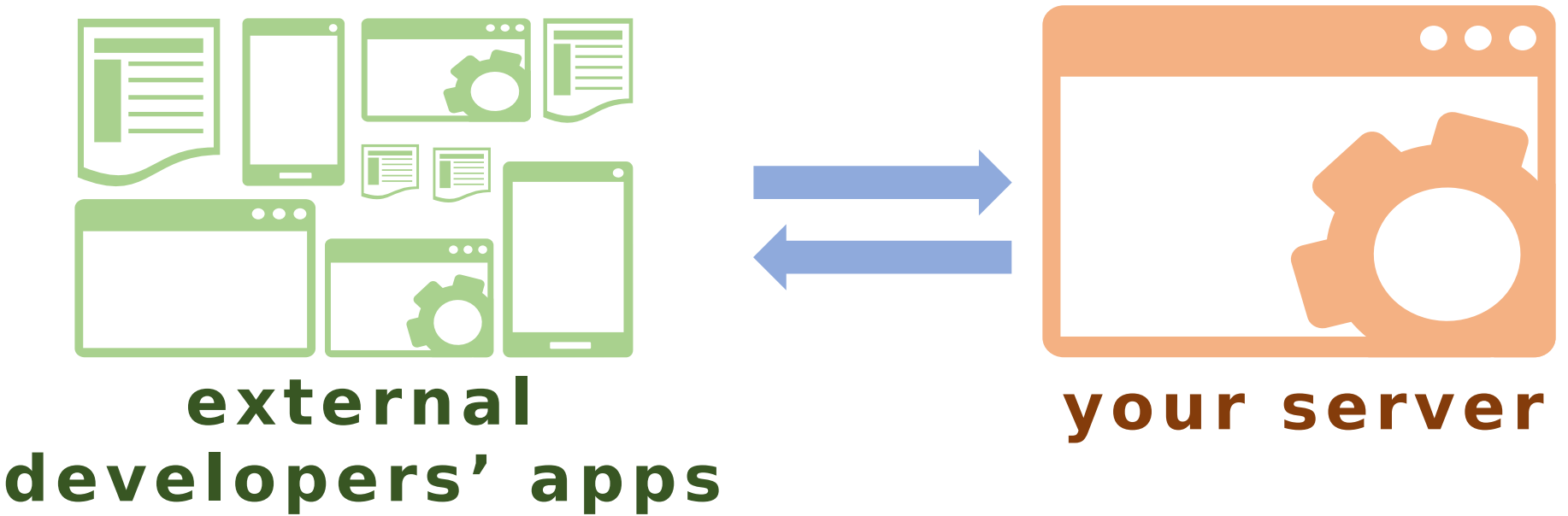
Internal



microservices

(consumer/producer)

External



REST bits

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

REST bits: resource

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

REST bits: query parameters

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

REST bits: HTTP methods

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

HTTP methods

- GET — gimme collection or item
- POST — create
- PUT — update (fully replace) or create
- PATCH — partial update (two types)
- DELETE — delete ;)
- HEAD — like GET without body (item exists?)

**idempotence*

REST bits: headers

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

REST bits: body

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

REST bits: status code

Request

```
POST /api/v1.5/group/1/chat?key=API-KEY&format=json
HTTP/1.1
Host: example.com
Content-Type: text/plain

Some test message
```

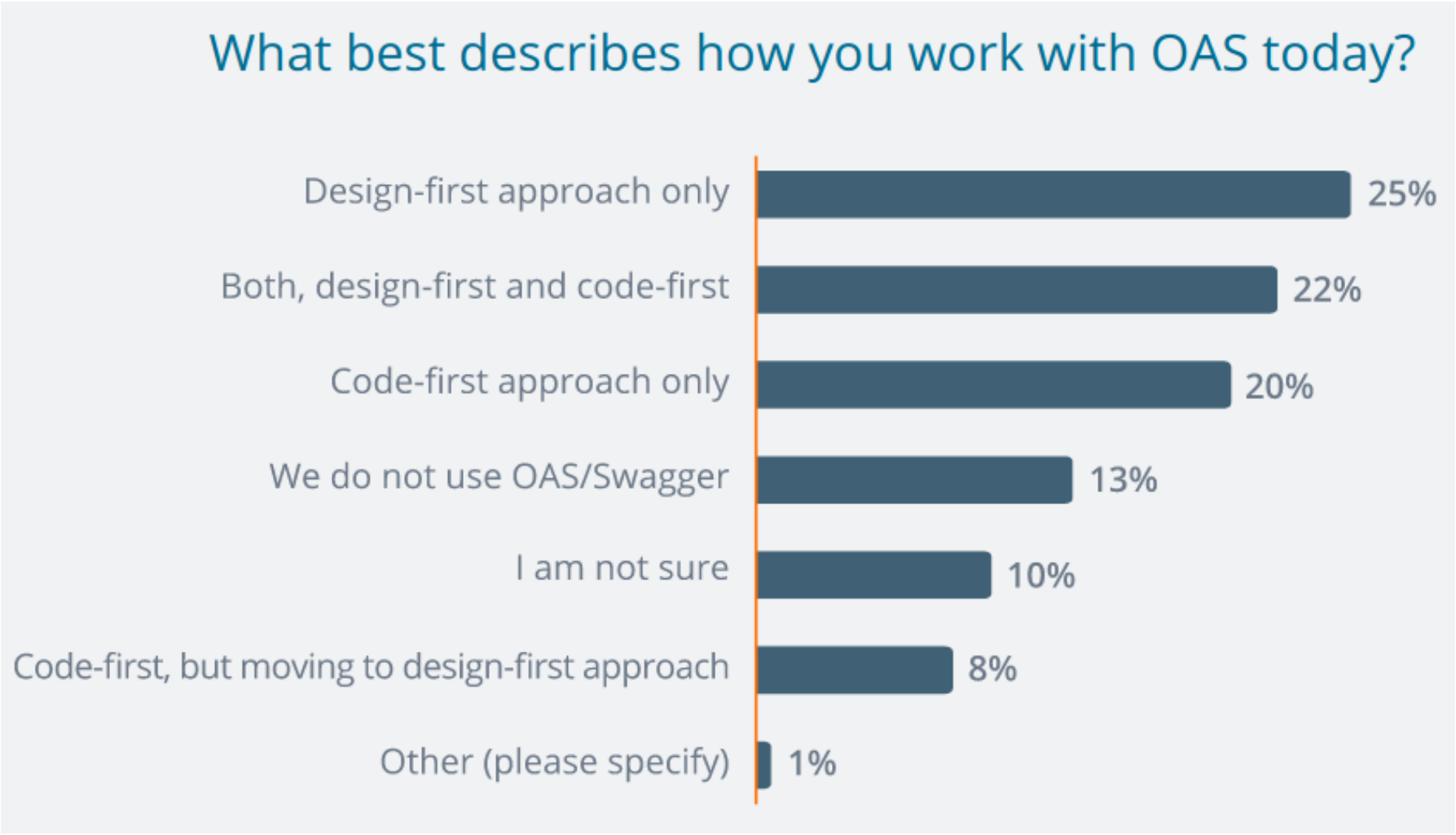
Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{ "code" : 200, "messageID": 15 }
```

Design!

- Code-first VS Design-first
- Description formats (OpenAPI, RAML)
- Code generation
- Validation



State of API, 2019

Tools

- SoapUI
- JMeter
- Runscope
- Postman
- Insomnia
- SwaggerUI
- curl / wget
- ...
- code