Building HTTP Servers

Bogdan Sucaciu

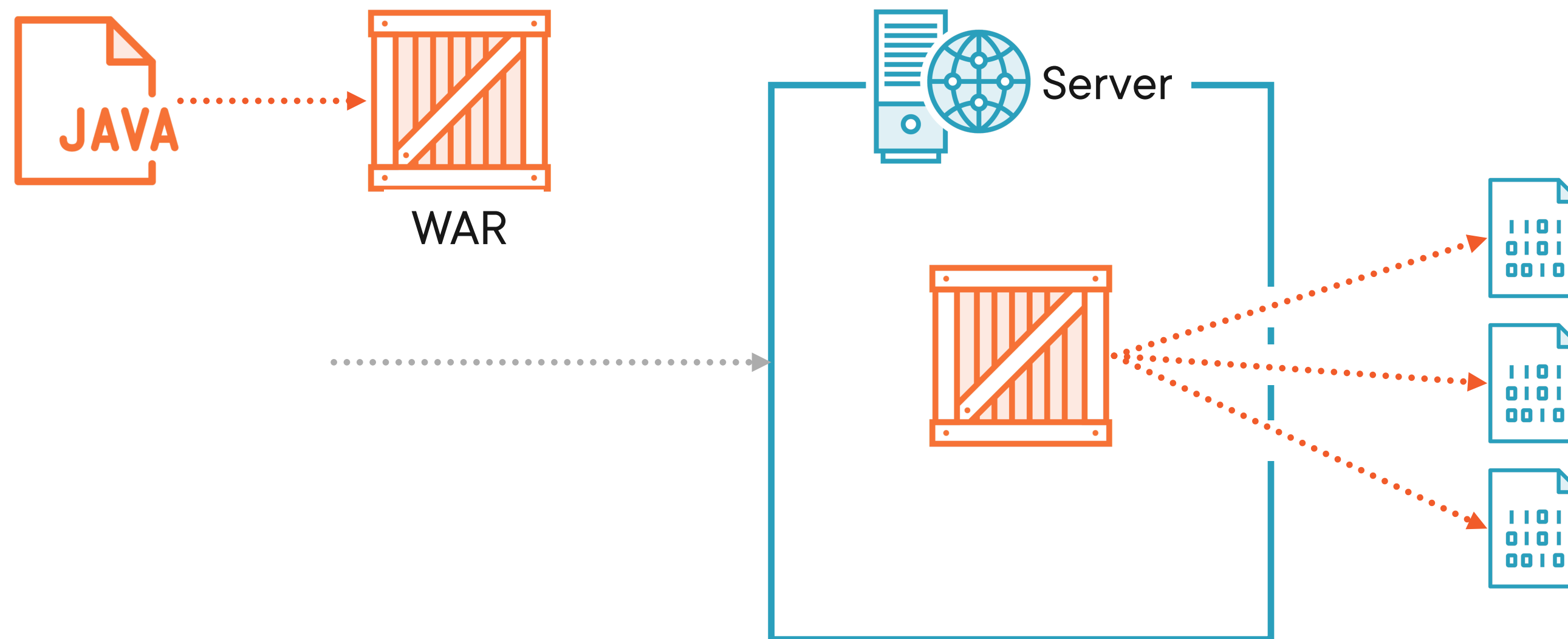Tech Lead

@bsucaciu   bsucaciu.com

Client-server Communication

Machine   Client   Request   Response   Server   Machine

Web Server

JAVA

WAR
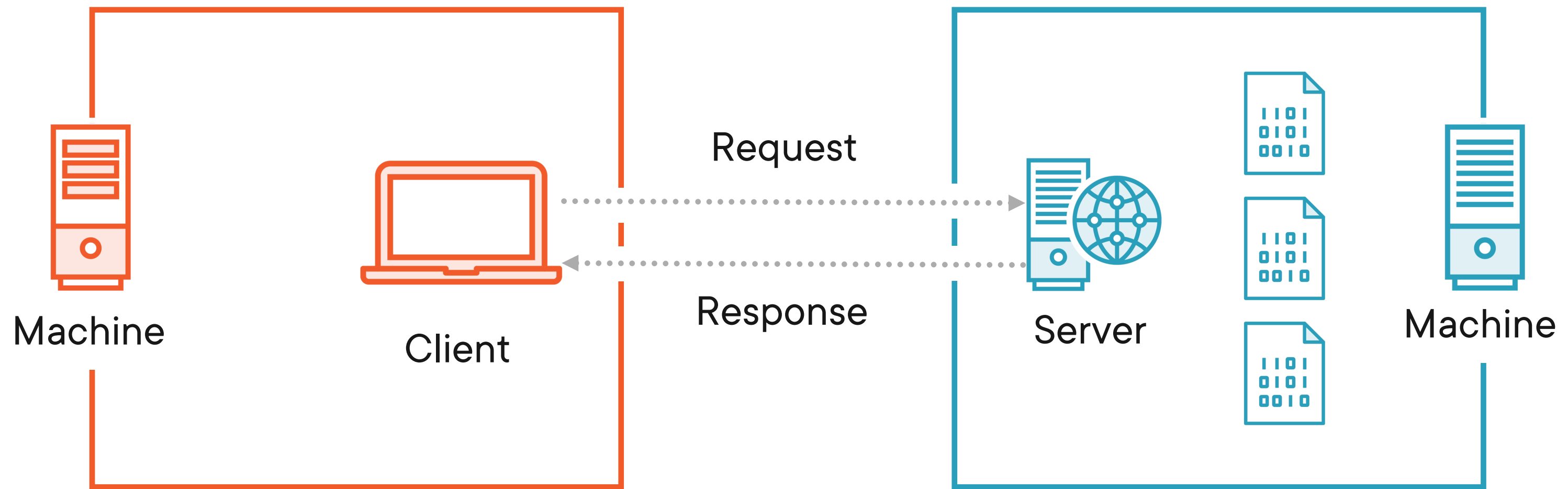
Server

Web Servers

Apache Tomcat	**Jetty**	**Apache TomEE**
WebSphere	**Glassfish**	**WildFly**

Embedded Server

HTTP

GET

POST

PUT

PATCH

DELETE

HEAD

OPTIONS

TRACE

@Controller

HTTP Paths

GET

POST

PUT

PATCH

DELETE

HEAD

OPTIONS

TRACE

@Controller

/ recipes

/ recipes

/ recipes / {recipe}

/ recipes / {recipe}

HTTP Endpoints

GET

POST

PUT

PATCH

DELETE

HEAD

OPTIONS

TRACE

@Controller

/ recipes ? q=chicken

/ recipes

/ recipes / {recipe}
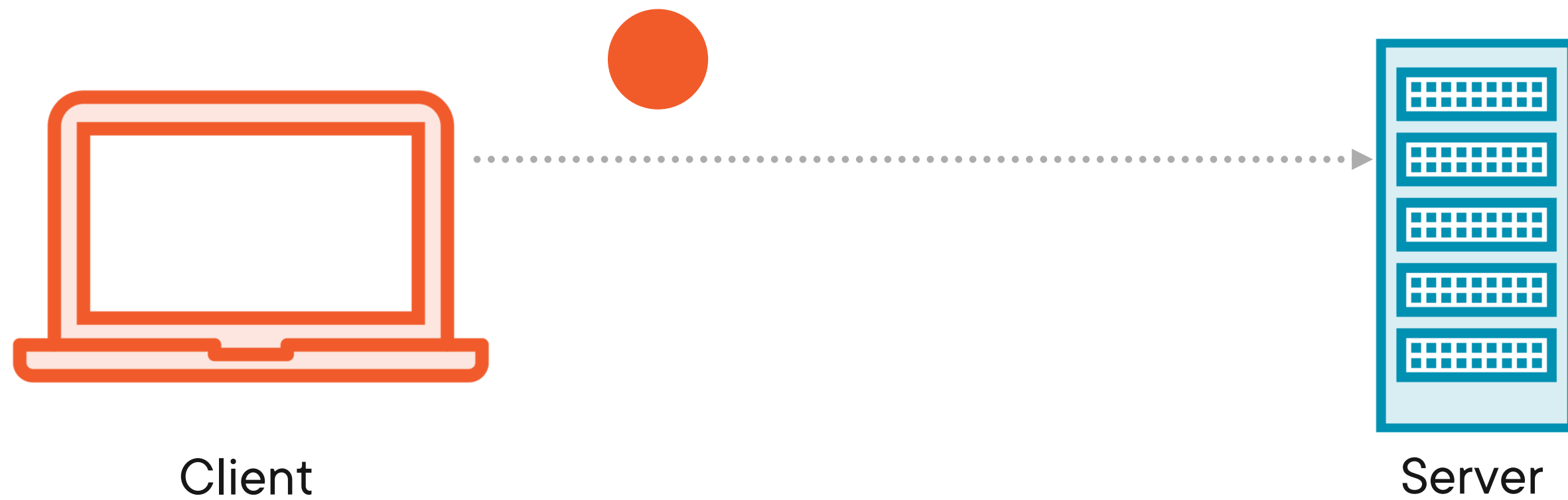
/ recipes / {recipe}
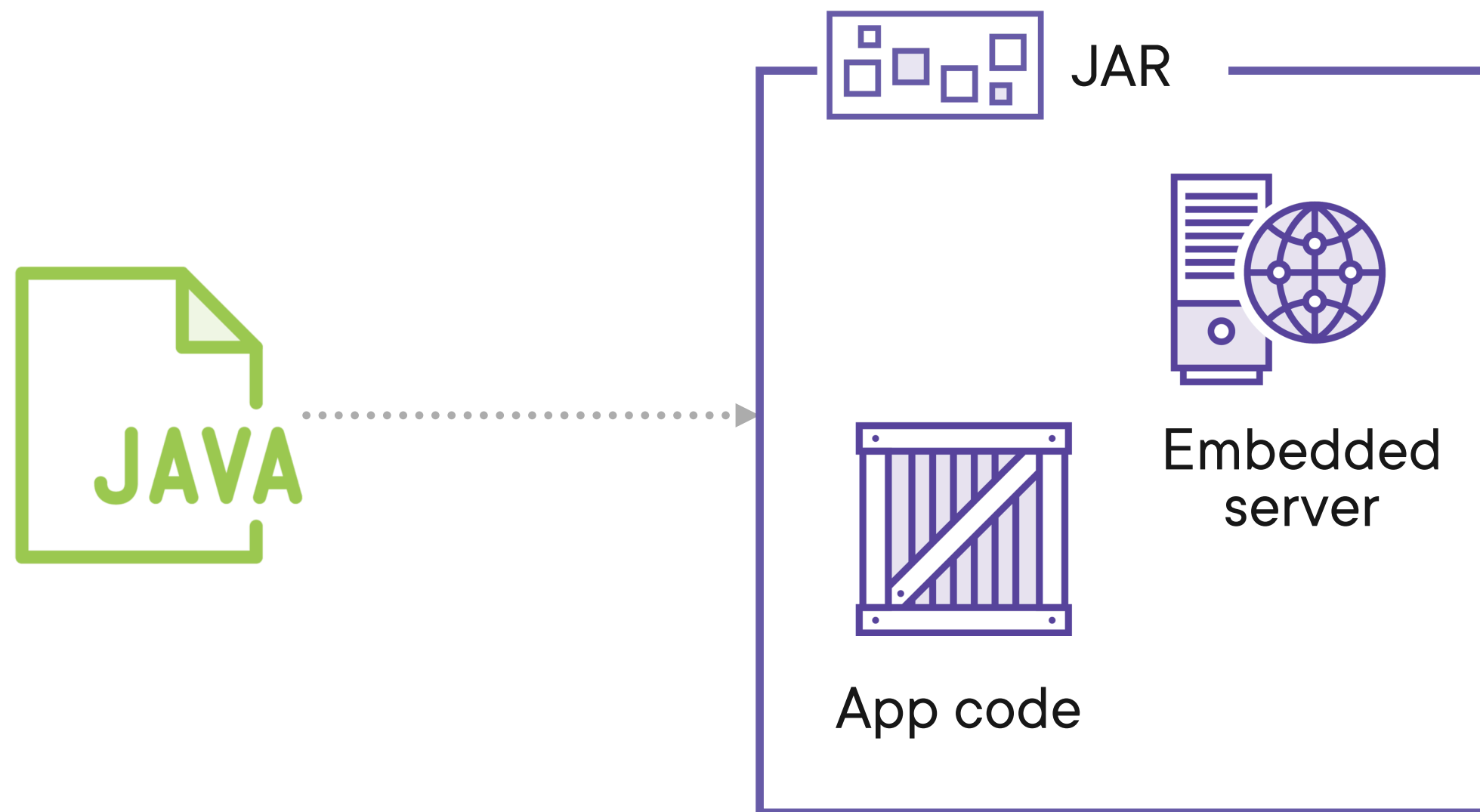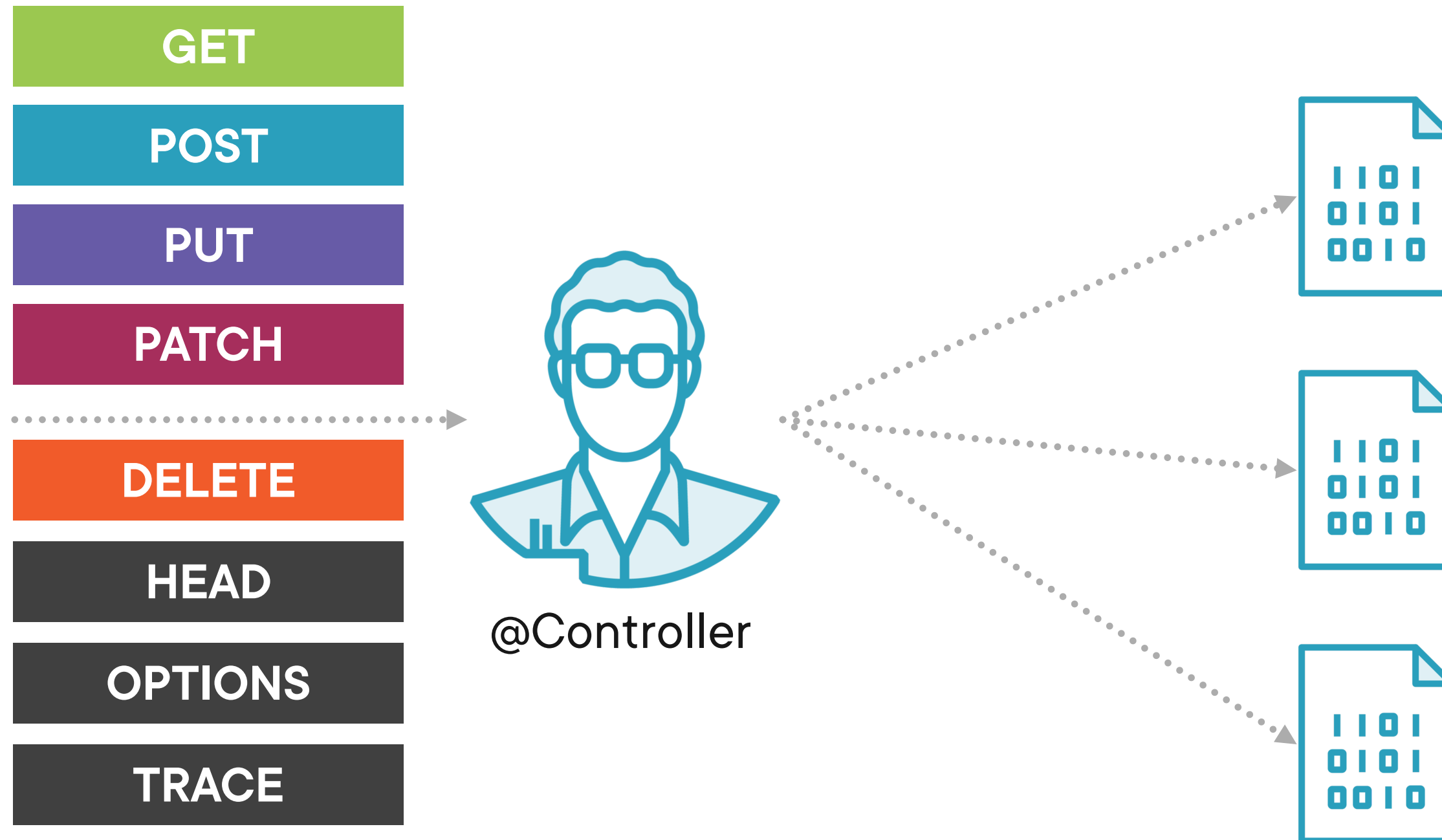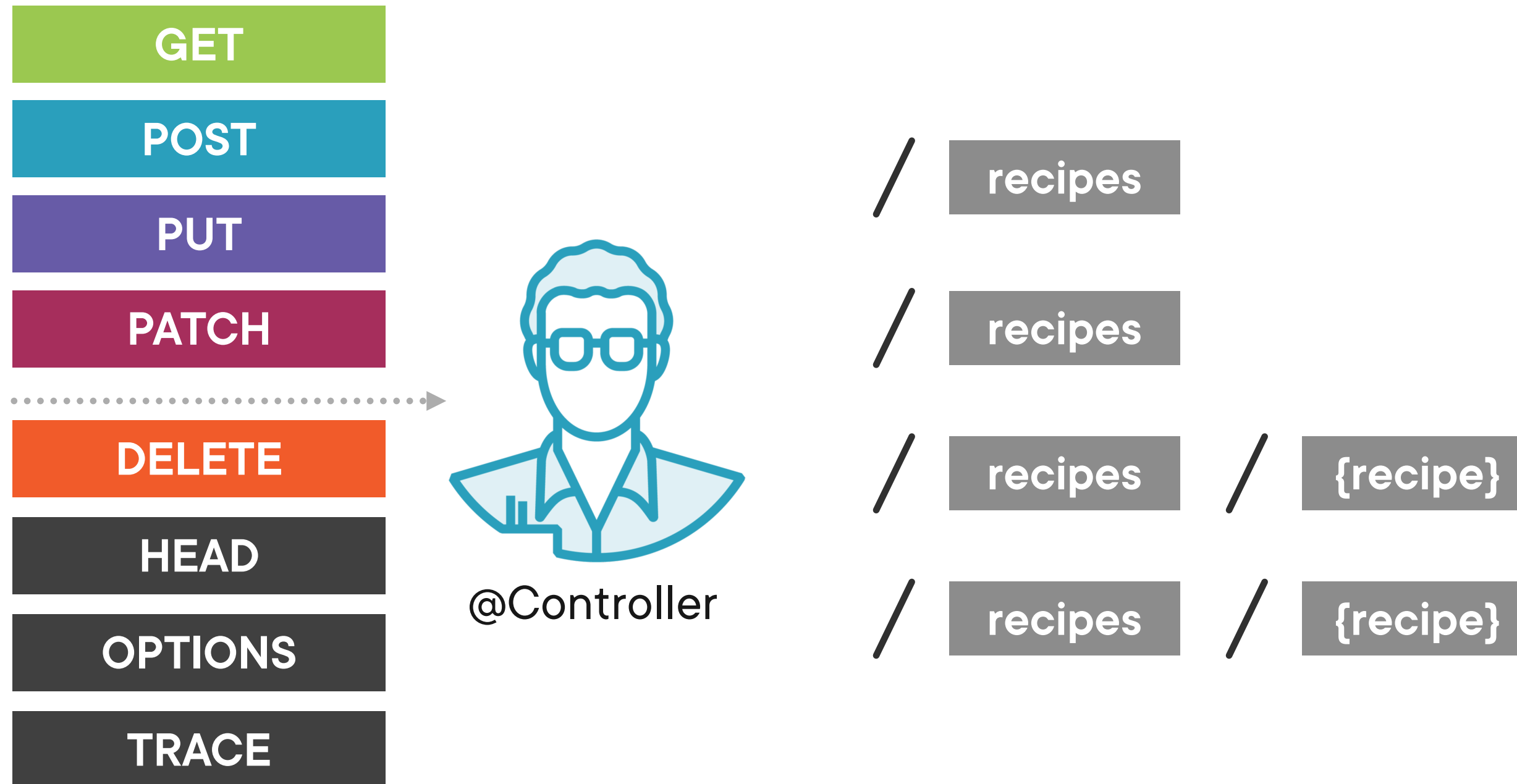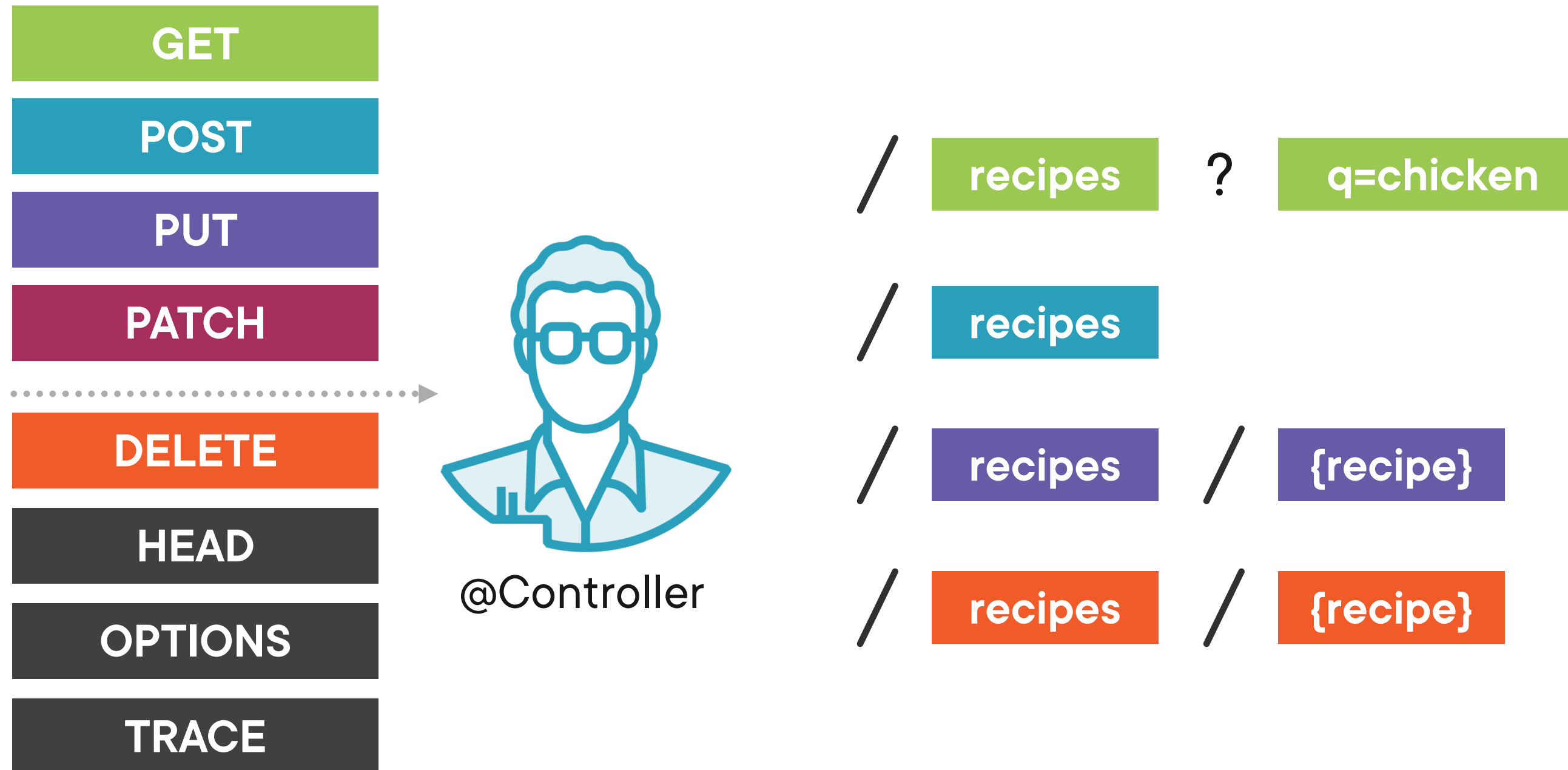
Demo

Set up Micronaut HTTP server

Create a controller

HTTP Request and Response

HTTP Request

Client

Server

HTTP Request

Request Headers
Content-type: application/json
Accept: application/json

Request Body
{
** "recipe": ...**
}

Client

Server
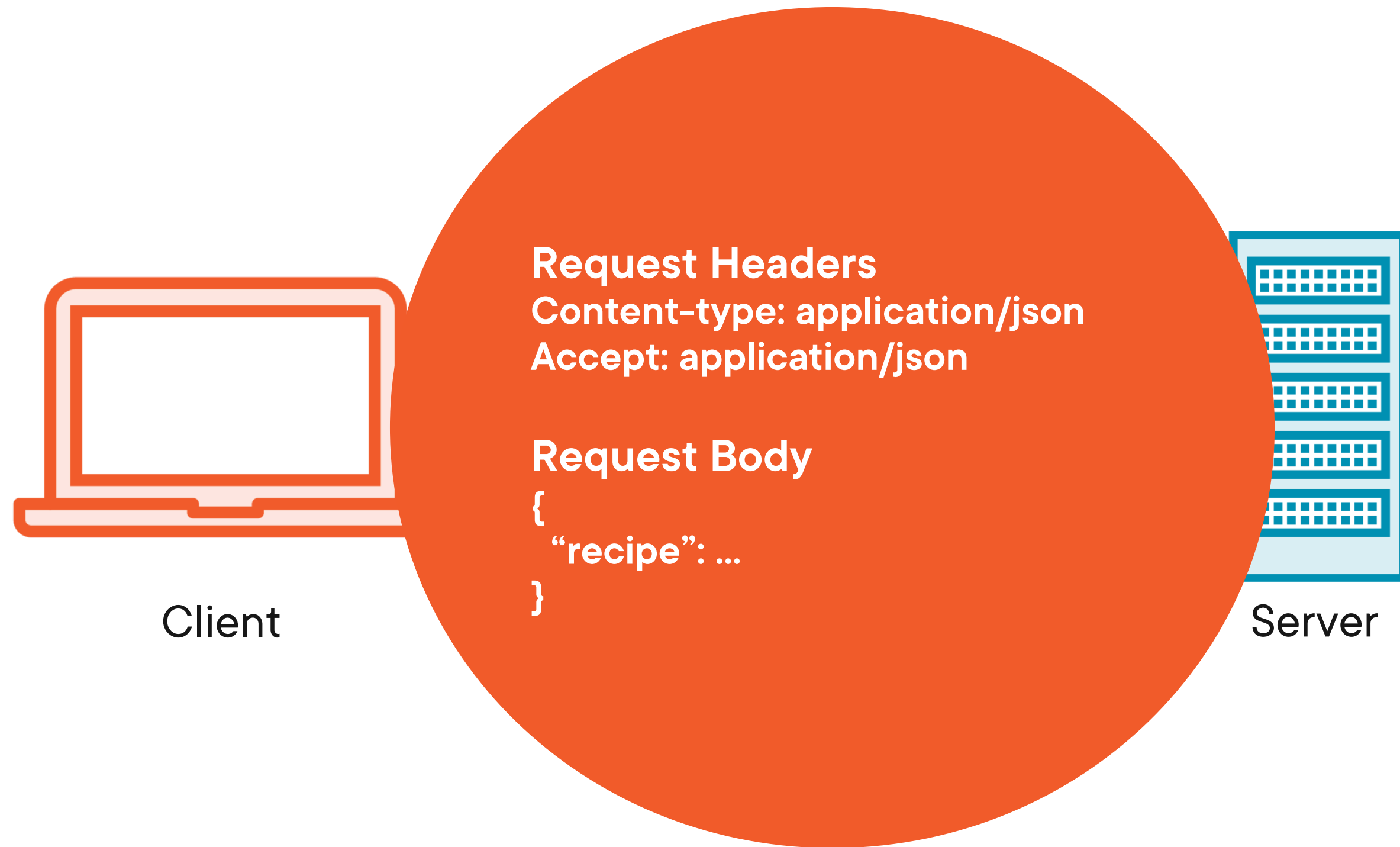
HTTP Response

HTTP Response

Client

Server

Response Headers
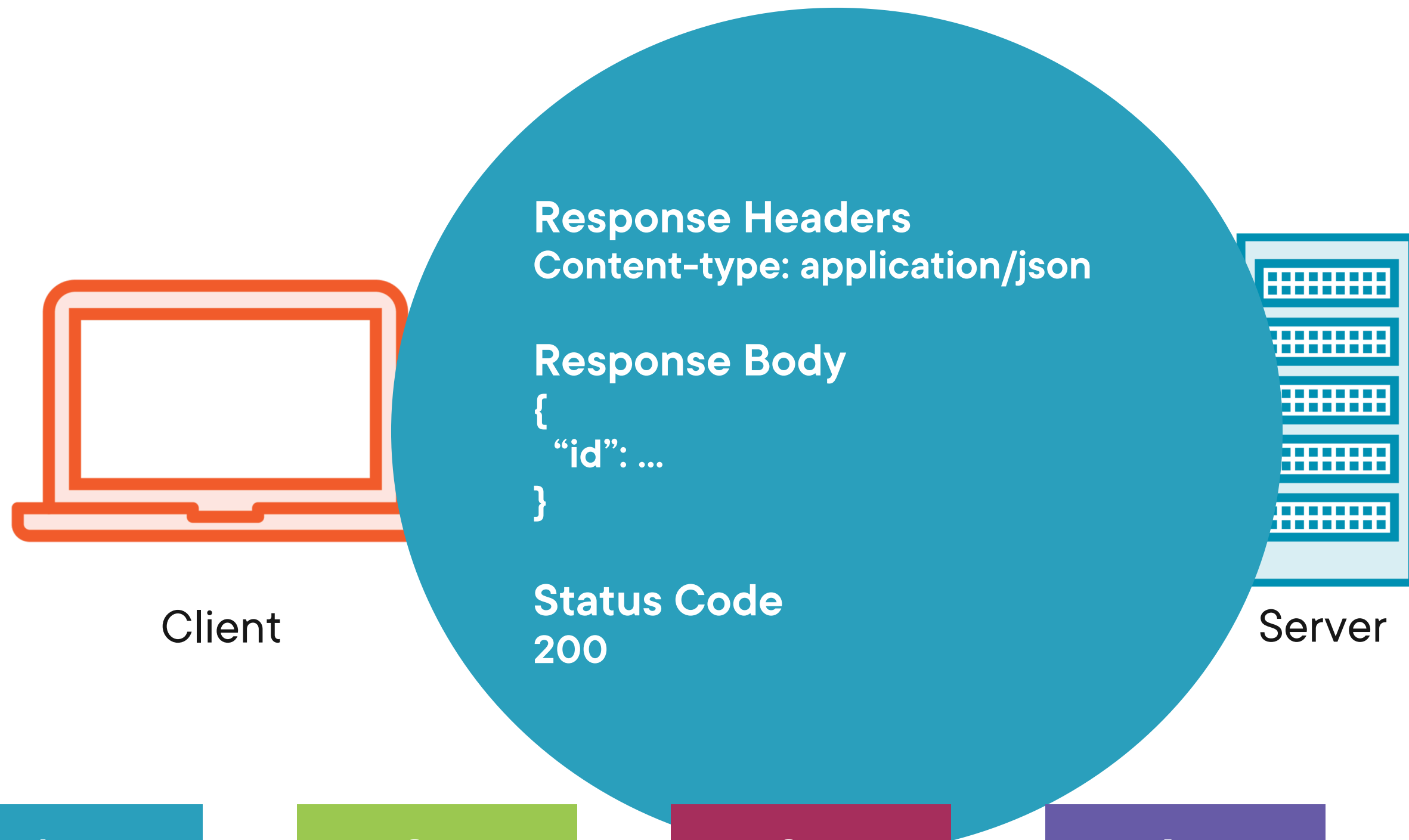Content-type: application/json

Response Body
{
  "id": ...
}

Status Code
200
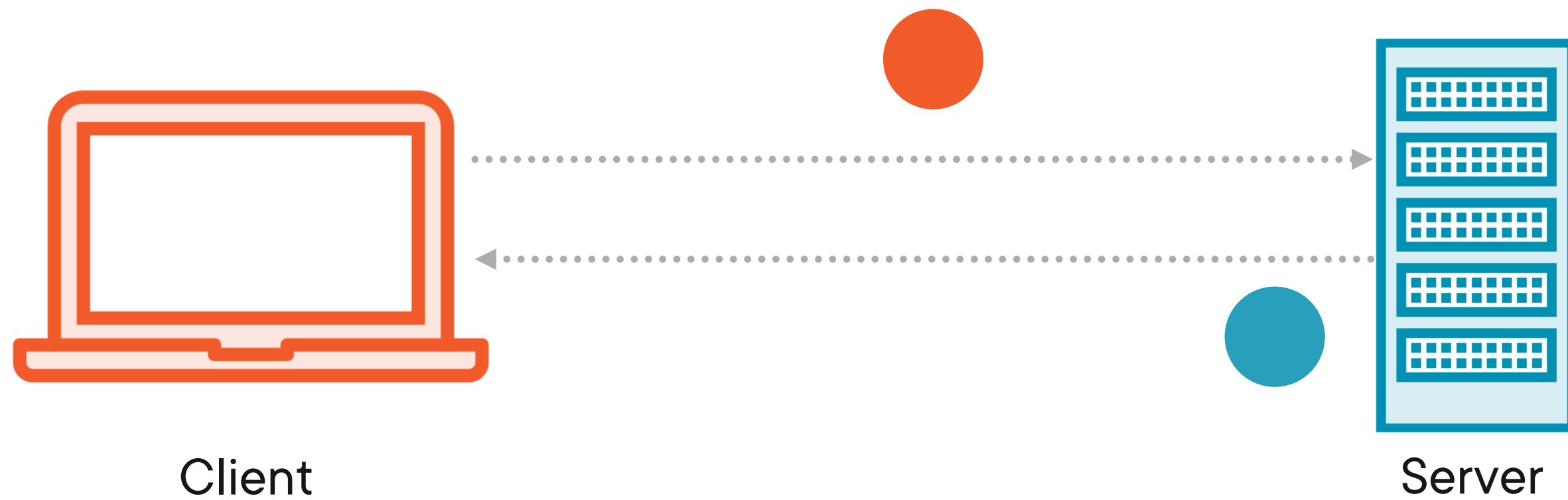
| 1xx | 2xx | 3xx | 4xx | 5xx |

HTTP Response
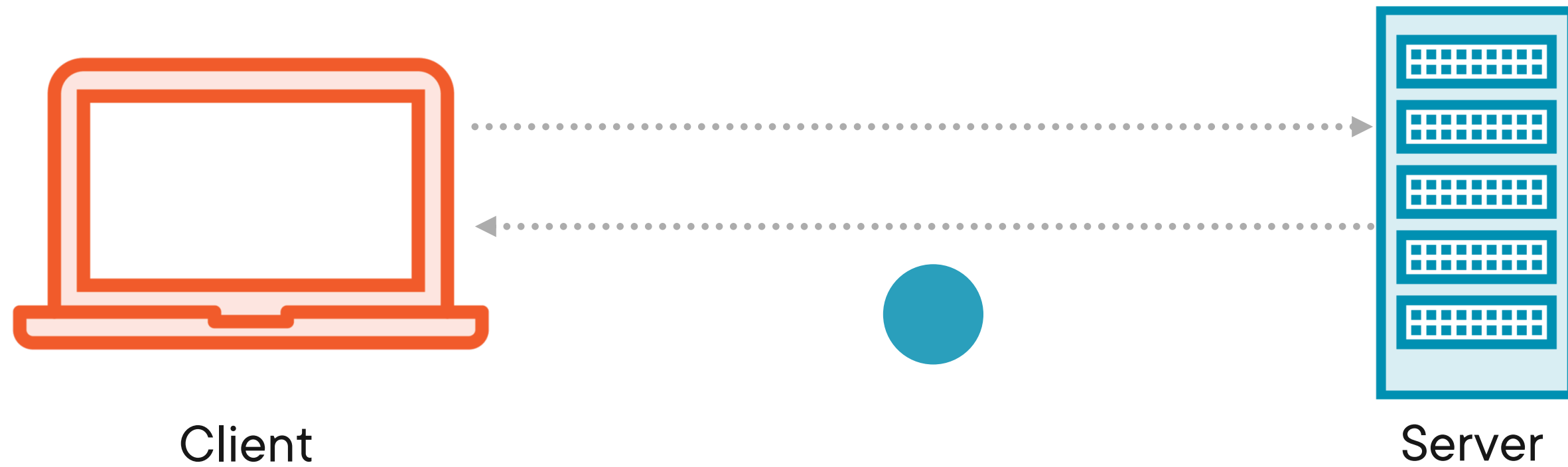
Demo

Add request body

Add response body

Explore headers

Error Handling

Error Handling

Application

Global

@Controller

Local

@Controller

@Controller
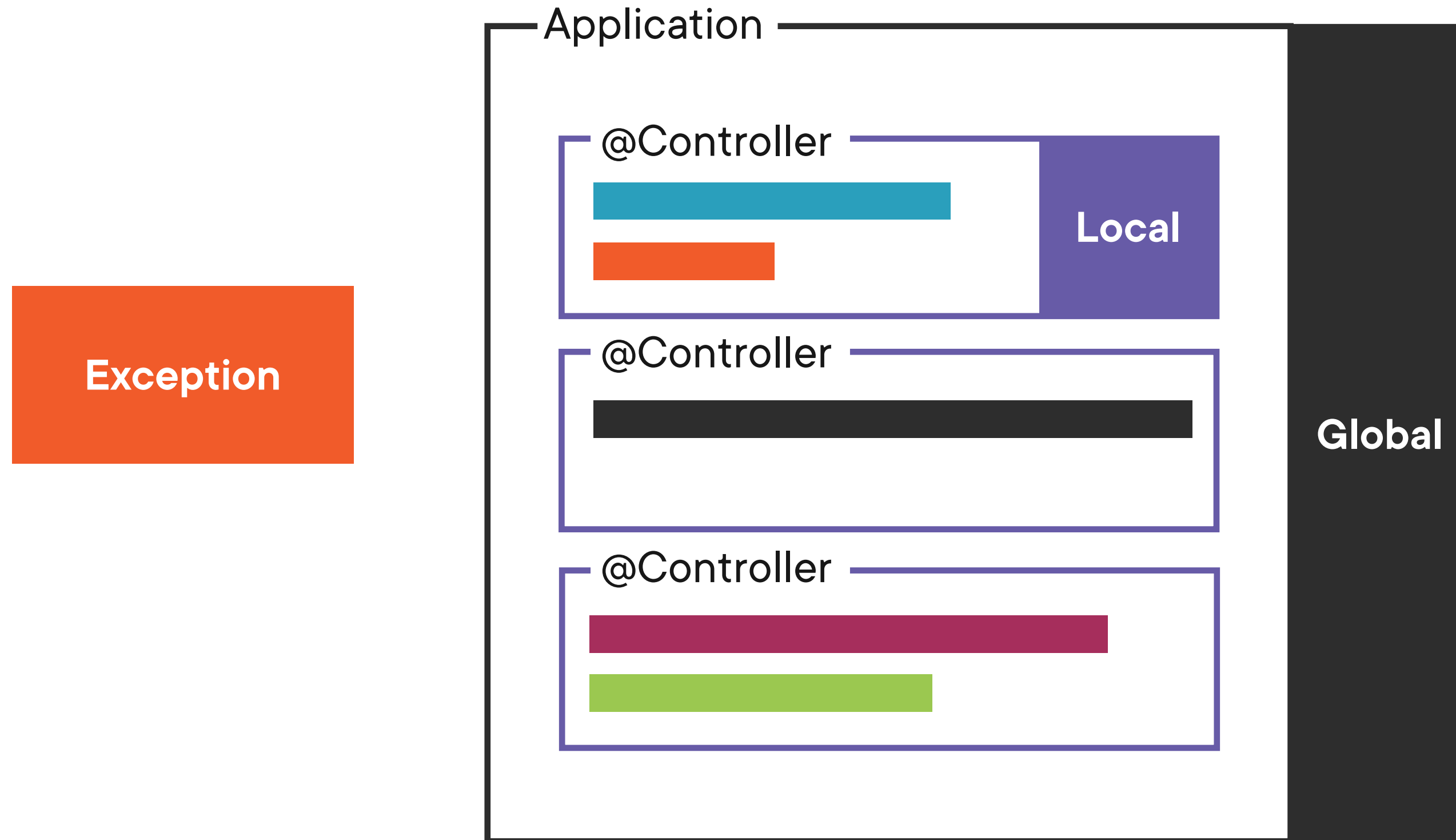
Exception
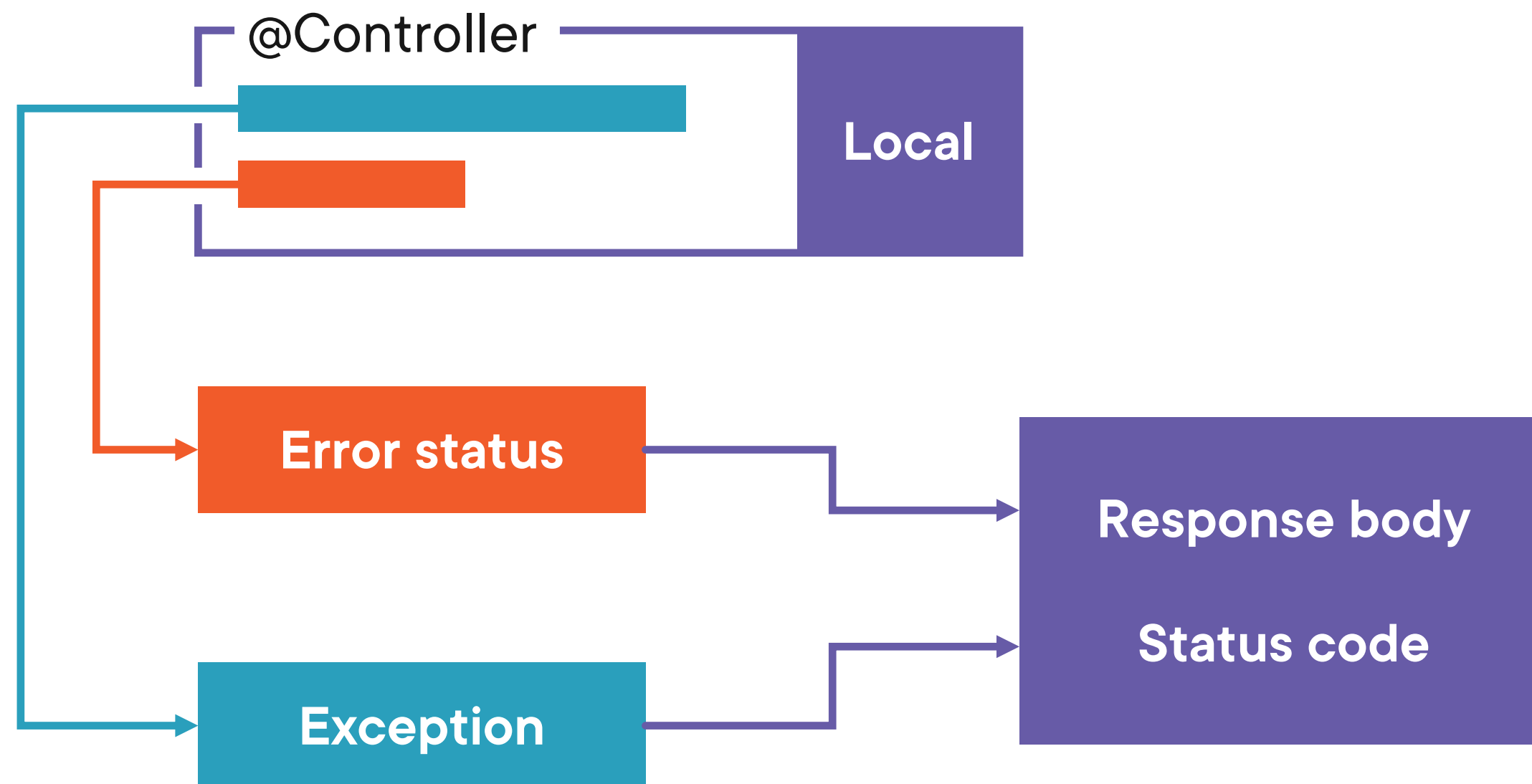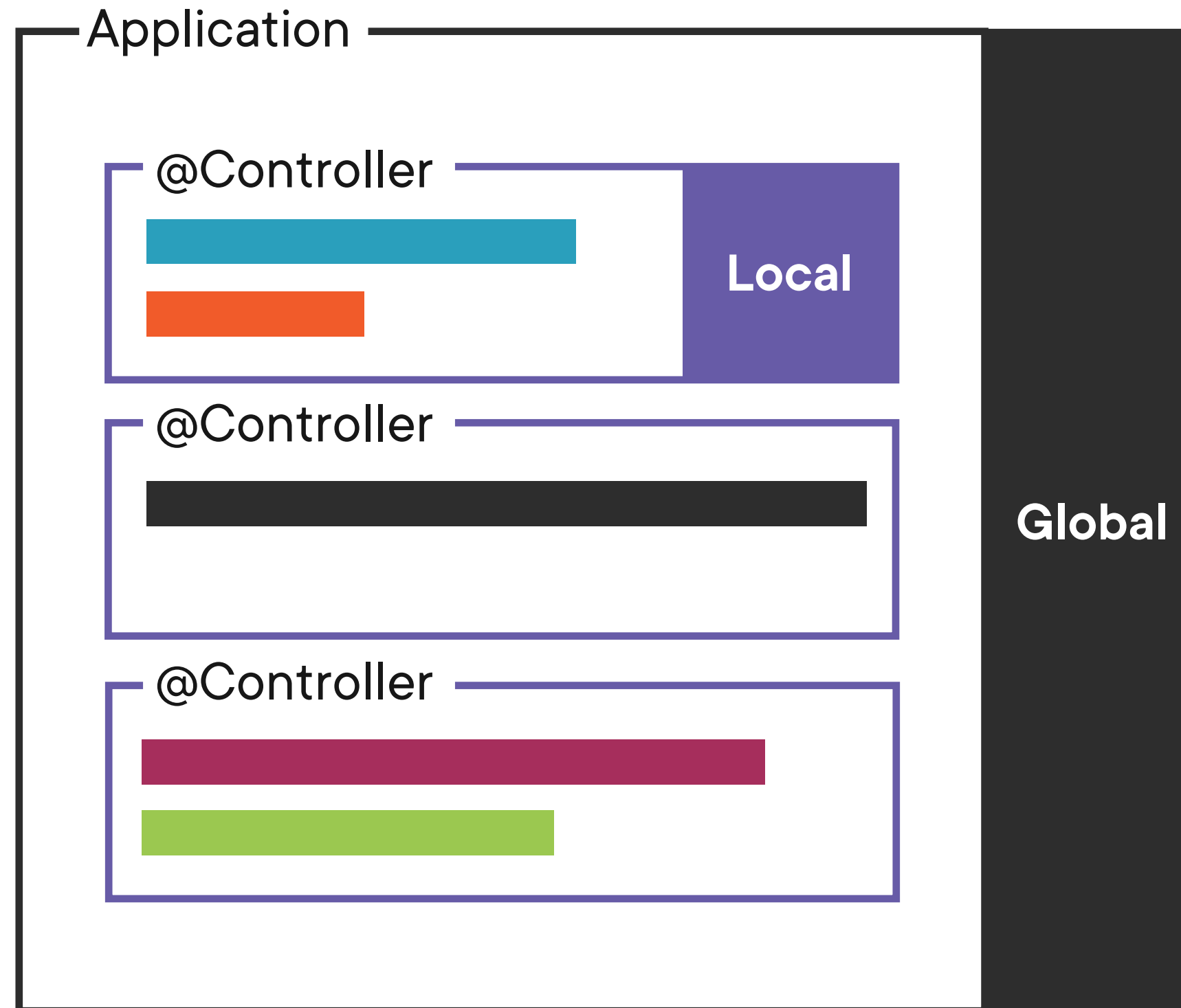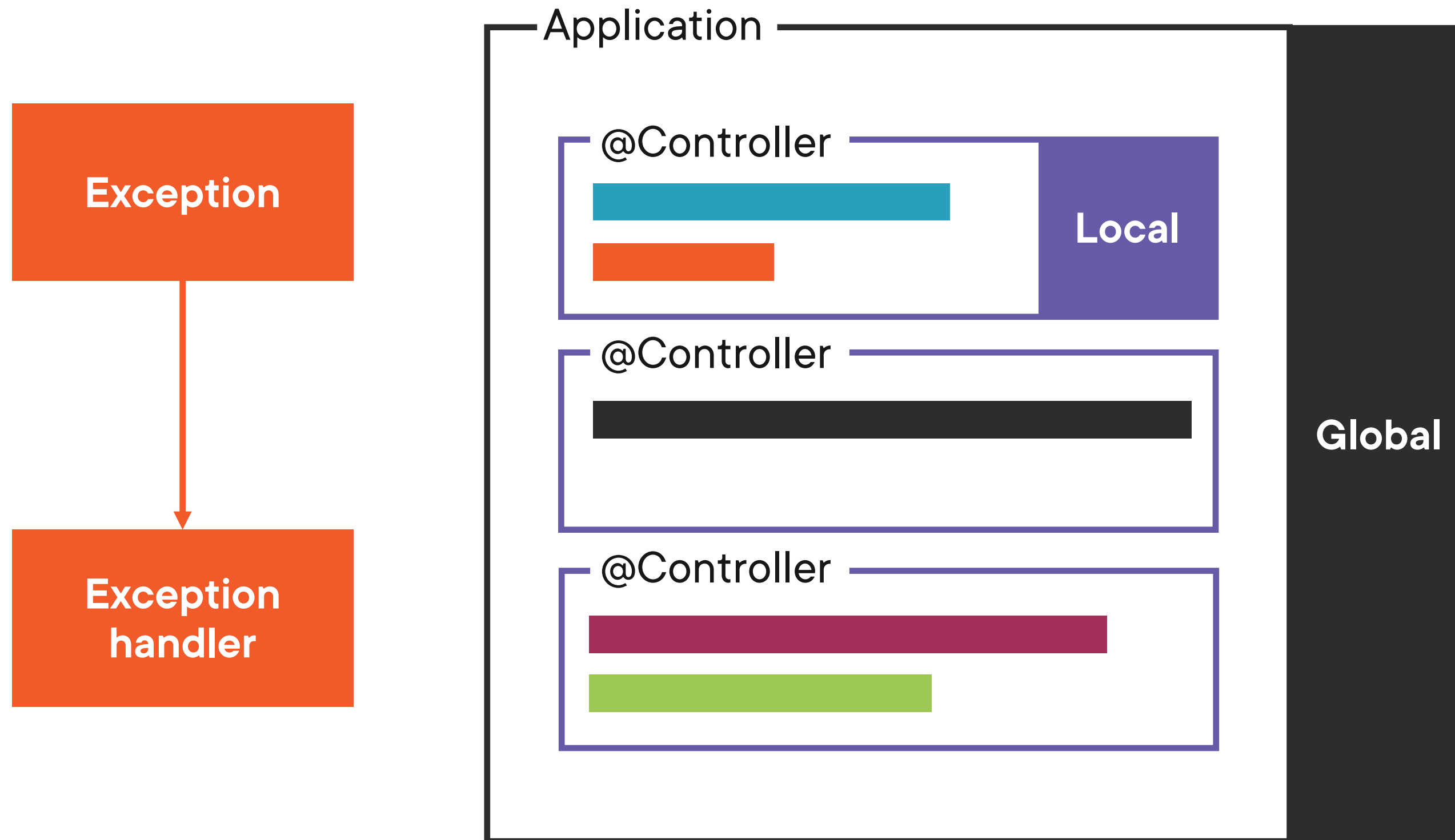
Local Error Handling

Global Error Handling

Exception Handlers

Demo

Local error handling

Global error handling

Exception handler

API Versioning

API Versioning

@Version("1")

@Get("/sensors/status")

Request body

Response body

Demo

API versioning

Management Endpoints

Built-in Endpoints

Endpoint	URI
Beans	/beans
Health	/health
Info	/info
Loggers	/loggers
Metrics	/metrics
Refresh	/refresh

Endpoint	URI
Routes	/routes
Environment	/env
ThreadDump	/threaddump
Caches	/caches
ServerStop	/stop

Demo

Explore management endpoints

Up Next:
Using HTTP Clients