microrestjs

A framework to develop microservices

Carlos Lozano Sánchez

- Entrepreneur
- Computer Engineer
- Software Engineer

Problem

NodeJS provides a powerful platform for building fast and scalable network applications.

BUT, developers do not have tools to build and deploy systematically microservices.

Overview of microrestjs

- Approach to build and deploy microservices
- Specification to describe microservices
- Framework to develop microservices
- Core services to deploy common functionality

Service-oriented Architecture

"Service-Oriented Architecture (SOA) is a paradigm for organizing and utilizing **distributed capabilities** that may be under the control of different **ownership** domains."

OASIS, SOA Reference Model

Services

"A **service is a mechanism to** enable **access** to one or more **capabilities**, where the access is provided using a **prescribed interface** and is exercised consistent with **constraints and policies** as specified by the **service description**."

OASIS, SOA Reference Model

Microservices Architectural Style

"The microservices architectural style is an approach to develop a **single application** as a **suite of** small **services**, each running in its **own process** and **communicating with** lightweight **mechanisms**."

James Lewis & Martin Fowler

SOA vs. Microservices Architectures

"In theory there is no difference. In practice **SOA** was applied to expose and reuse **business services at enterprise level**. **Microservices architectures** are used to structure **individual applications**."

Adam Bien

Reinventing the wheel

- Distributed capabilities
- Services
- Interfaces
- Services descriptions
- Communication mechanisms
- Ownership

Service Design Principles

1. Reusability
2. Formal contract
3. Loose coupling
4. Abstraction
5. Composability
6. Autonomy
7. Statelessness
8. Discoverability

Overview of microrestjs

- **Approach to build and deploy microservices**
- **Specification to describe microservices**
- Framework to develop microservices
- Core services to deploy common functionality

Approach of microrestjs

Approach: Definition Phase

Approach: Definition Phase

- **Goal:** Describe the service and its operations
- **How:** Creating a JSON file with the description
- **Format:** Microrestjs Service Description Specification
- **Principles:** Formal contract, Loose Coupling, Abstraction, Composability, Autonomy and Discoverability

Approach: Definition Phase

```
{
   “microrestSpecification”: 1,
   “info”: {
      “name”: “example-helloworld”,
      “version”: “0.0.1”,
      “api”: 1,
      ...
   },
   “config”: {
      “location”: “directory”
      “dependiencies”: { ... }
   },
```

Approach: Definition Phase

```
"operations": {
   "greet": {
      "request": {
         "path": "/greet/:username",
         "method": "GET",
         ...
      },
      "responses": { ... },
      "errors": { ... }
   }
}
}
```

Approach: Development Phase

Approach: Development Phase

- **Goal:** Build the logic of the service
- **How:** Creating a JavaScript file with the functionality
- **Format:** Service Description
- **Principles:** Composability, Autonomy, Statelessness

Approach: Development Phase

```
function greet(request, response) {
  var greetBody = {
     greet: “Hello ” + request.params.username
  }

  response.status(200).json(greetBody).end();
}
```

Approach: Deployment Phase

Approach: Deployment Phase

- **Goal:** Deploy the services in a distributed context
- **How:** Running an instance of microrestjs with the service description and service functionality.
- **Principles:** Discoverability

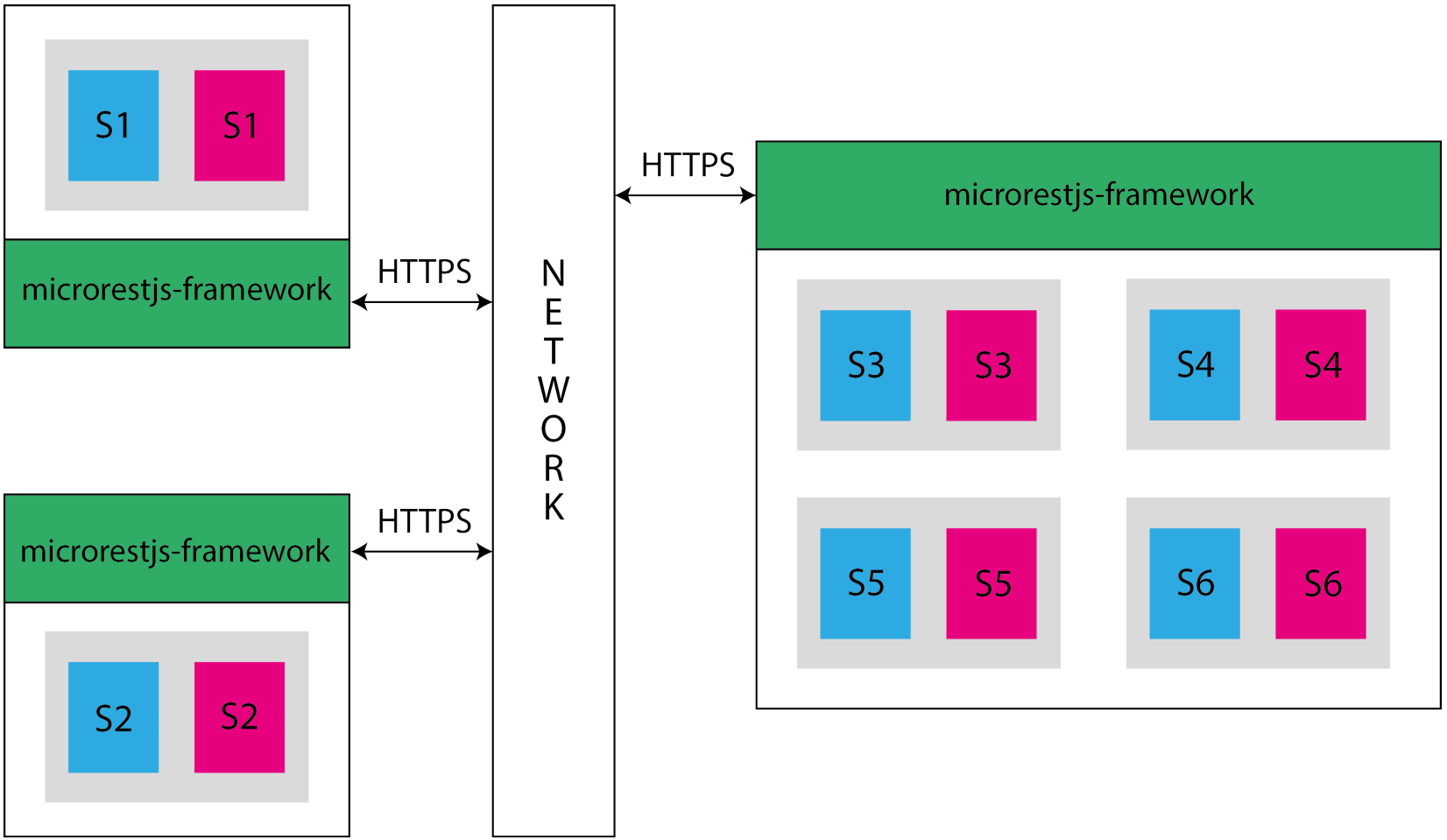
Approach: Deployment Phase

1. Configure microrestjs
2. Save the service description and functionality
3. Execute: *node Launcher.js*

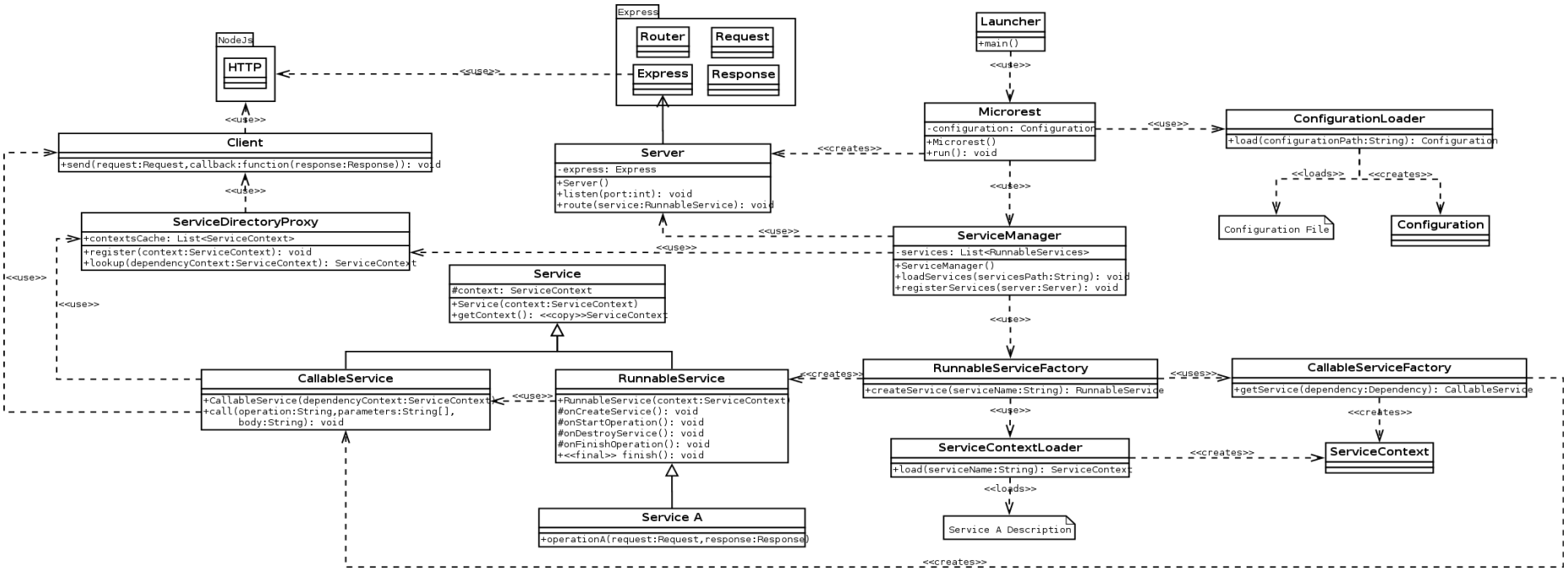
Overview of microrestjs

- Approach to build and deploy microservices
- Specification to describe microservices
- **Framework to develop microservices**
- Core services to deploy common functionality

Framework: Architecture

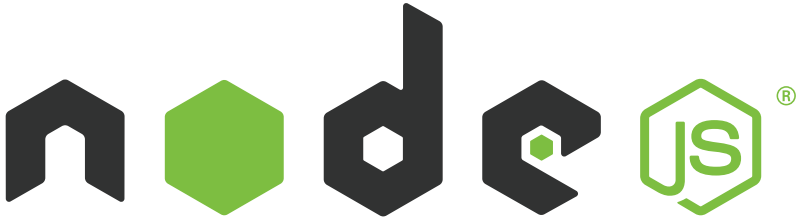


Framework: Architecture



Framework: Technologies

express



OpenSSL

{JSON}

Framework: More and More

- Universally accessible
- RESTful API
- Standard web technologies (HTTP)
- Secure connections
- Location transparency

OPEN SOURCE – MIT License

Overview of microrestjs

- Approach to build and deploy microservices
- Specification to describe microservices
- Framework to develop microservices
- **Core services to deploy common functionality**

Advantages and Drawbacks

Dynamic load without compilation

Decrease in development time

Increase in documentation quality

Production ready without lock-in

Not a silver bullet, not for everything

Weakly typed programming

Strict development approach

No standard descriptions

Conclusion

WSDL

SOAP

HTTPS

NodeJS

SOA

Microservices

JSON

RESTful API

Web

Conclusion

WSDL

SOAP

HTTPS

NodeJS

SOA

Microservices

JSON

RESTful API

Web

microrestjs