Platform for designing, developing and deploying microservices

Carlos Lozano Sánchez

December 15, 2015

PRESENTATION

20 min

Problem

Software systems are becoming more **powerful** and **complex**, resulting in the necessity of using **new approaches** for their development.

Microrestjs

(Microservice architecture)

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

Overview of Microrestjs

- Methodology to build microservices
- Specification to describe microservices
- Framework to develop/deploy microservices
- Core services to provide common functionality

Overview of Microrestjs

- **Methodology to build microservices**
- **Specification to describe microservices**
- Framework to develop/deploy microservices
- Core services to provide common functionality

Microrestjs Methodology

Methodology: Design Phase

Methodology: Design Phase

- **Task:** Describe the microservice and its operations
- **How:** Creating a JSON file with the description
- **Tool:** Microrestjs Service Description Specification

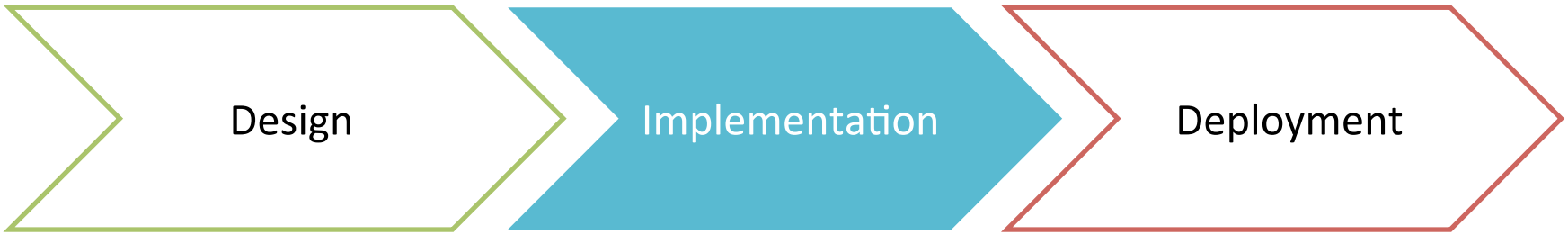
Methodology: Design Phase

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

Methodology: Design Phase

```
  “security”: { ... }
  “operations”: {
      “search”: {
         “request”: {
            “path”: “/:target”,
            “method”: “GET”,
            ...
         },
         “responses”: { ... },
         “errors”: { ... }
      }
   }
}
```

Methodology: Implementation Phase



Methodology: Implementation Phase

- **Task:** Implement the logic of the microservice
- **How:** Creating a JavaScript file with the functionality
- **Tool:** Service Description and Microrestjs Framework

Methodology: Implementation Phase

```
function search(request, response, sendFunction) {
   var target = request.getPathParameter(“target”);

   ... (the items are searched)

   response.setStatus(200).setBody(items);
   sendFunction();
}
```

Methodology: Deployment Phase

Methodology: Deployment Phase

- **Task:** Deploy the microservice and make them accessible from the network

- **How:** Running an instance of Microrestjs with the service description and the functionality.

- **Tool:** Microrestjs Framework

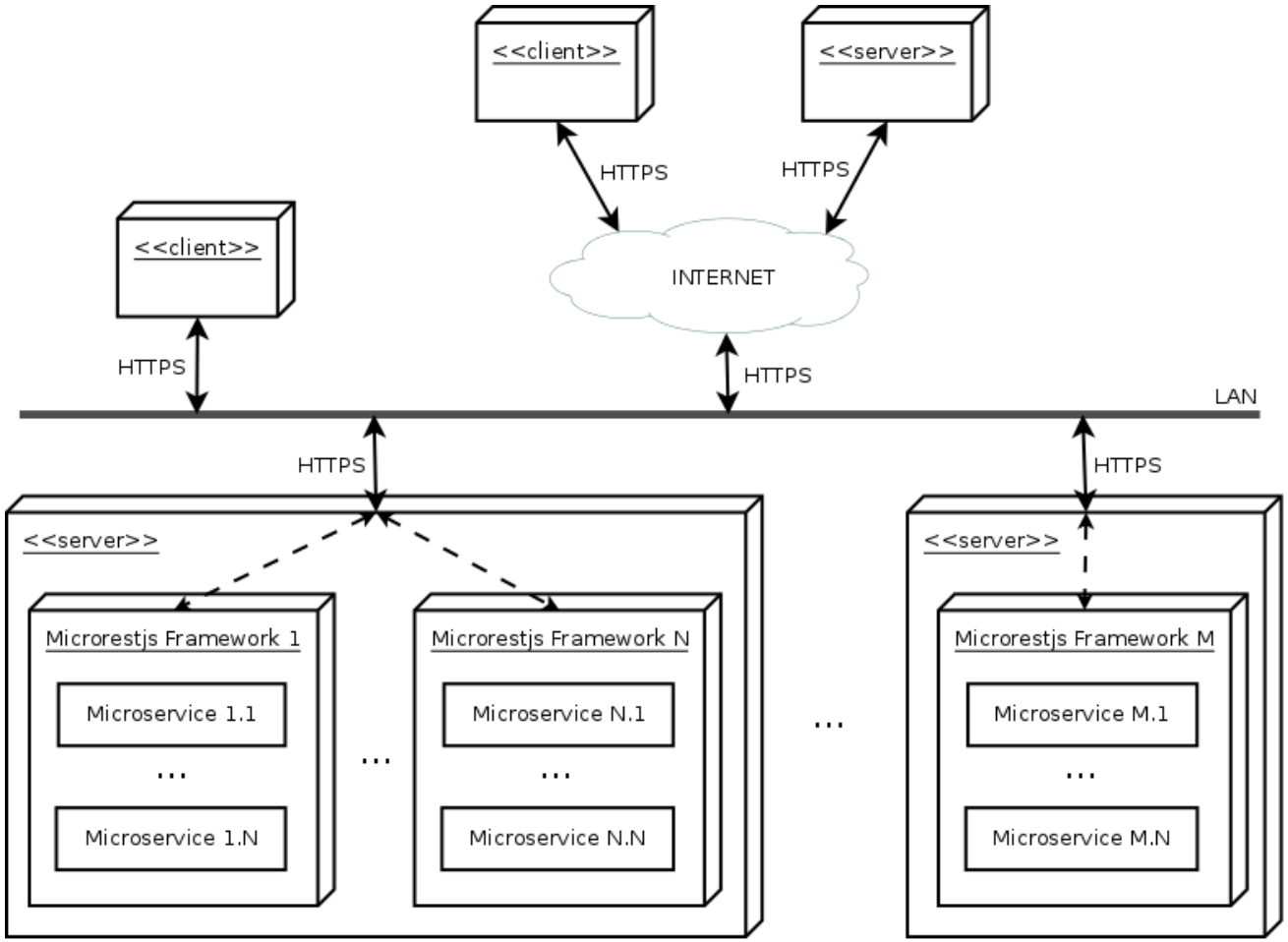
Methodology: Deployment Phase

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

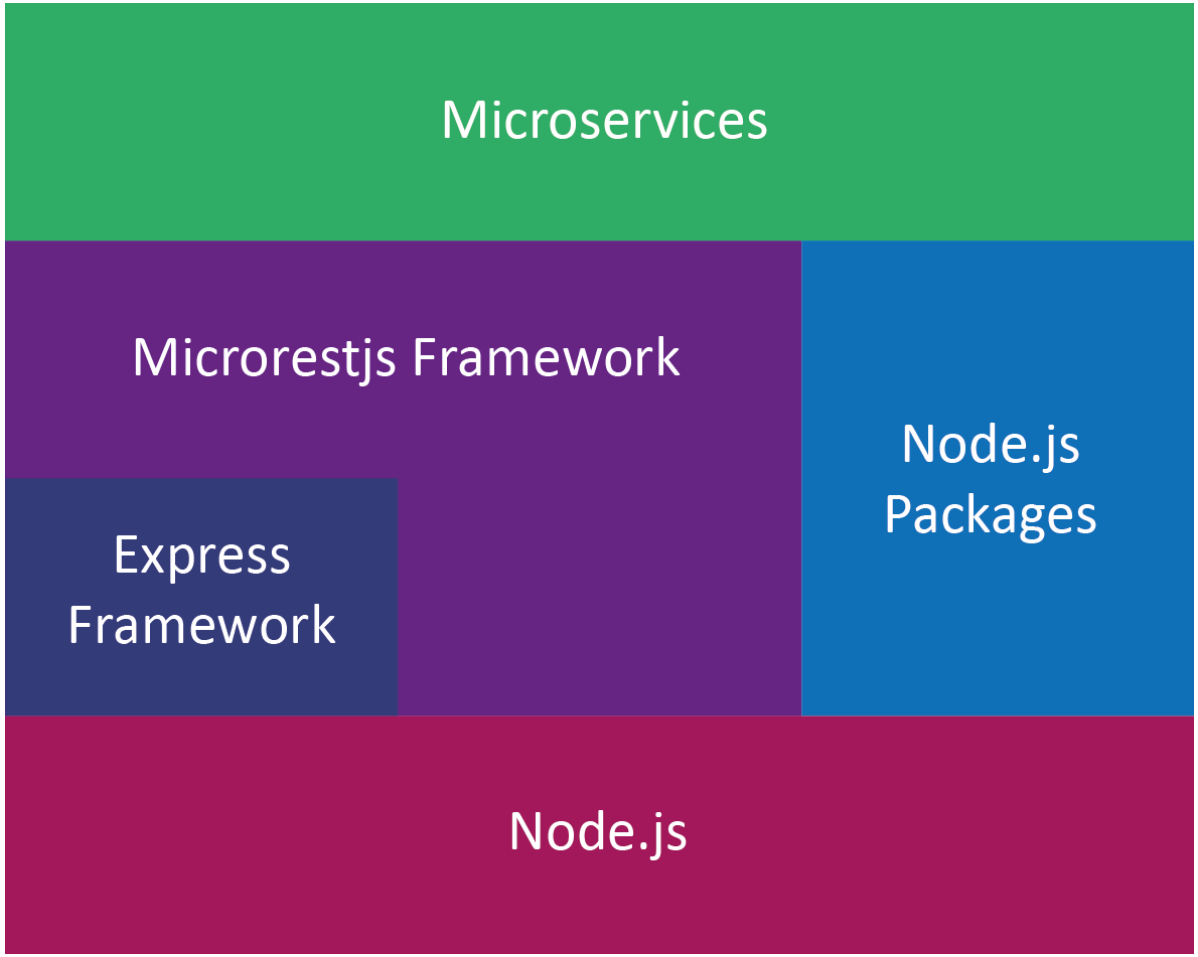
Overview of Microrestjs

- Methodology to build microservices
- Specification to describe microservices
- **Framework to develop/deploy microservices**
- Core services to provide common functionality

Framework: Architecture



Framework: Architecture



Framework: Characteristics

- Influenced by previous technologies
- SOA- and REST-compliant microservices
- Standard Web technologies (HTTP)
- Secure network communications (TLS)
- Uniform interfaces for interaction
- Emphasis on automation
- No technical lock-in

Framework: Properties

- Simplicity
- Scalability
- Interoperability
- Security
- Reusability
- Discoverability
- Uniformity
- Flexibility
- Adaptability
- Extensibility
- Maintainability
- Readability

Overview of Microrestjs

- Methodology to build microservices
- Specification to describe microservices
- Framework to develop/deploy microservices
- **Core services to provide common functionality**

Core Services

- Service directory
- Authentication and Authorization Services
  - None
  - Basic
  - (Token)

Advantages and Drawbacks

Improve overall quality

Reduce software complexity, costs, and risks

Dynamic deployment without compilation

Open source environment

Not a silver bullet, not for everything

Weakly typed programming

No standard descriptions

New environment

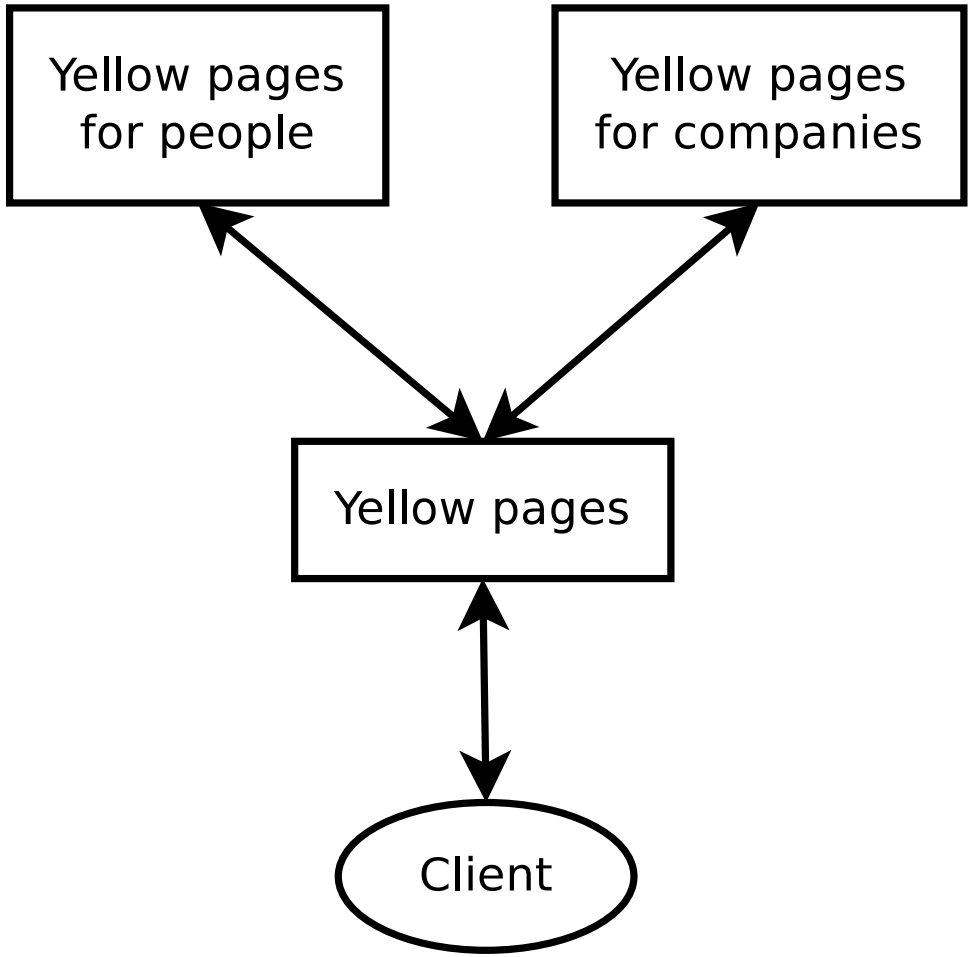
Conclusion

Microrestjs is an innovative platform that aims to be the basis of the future technologies.

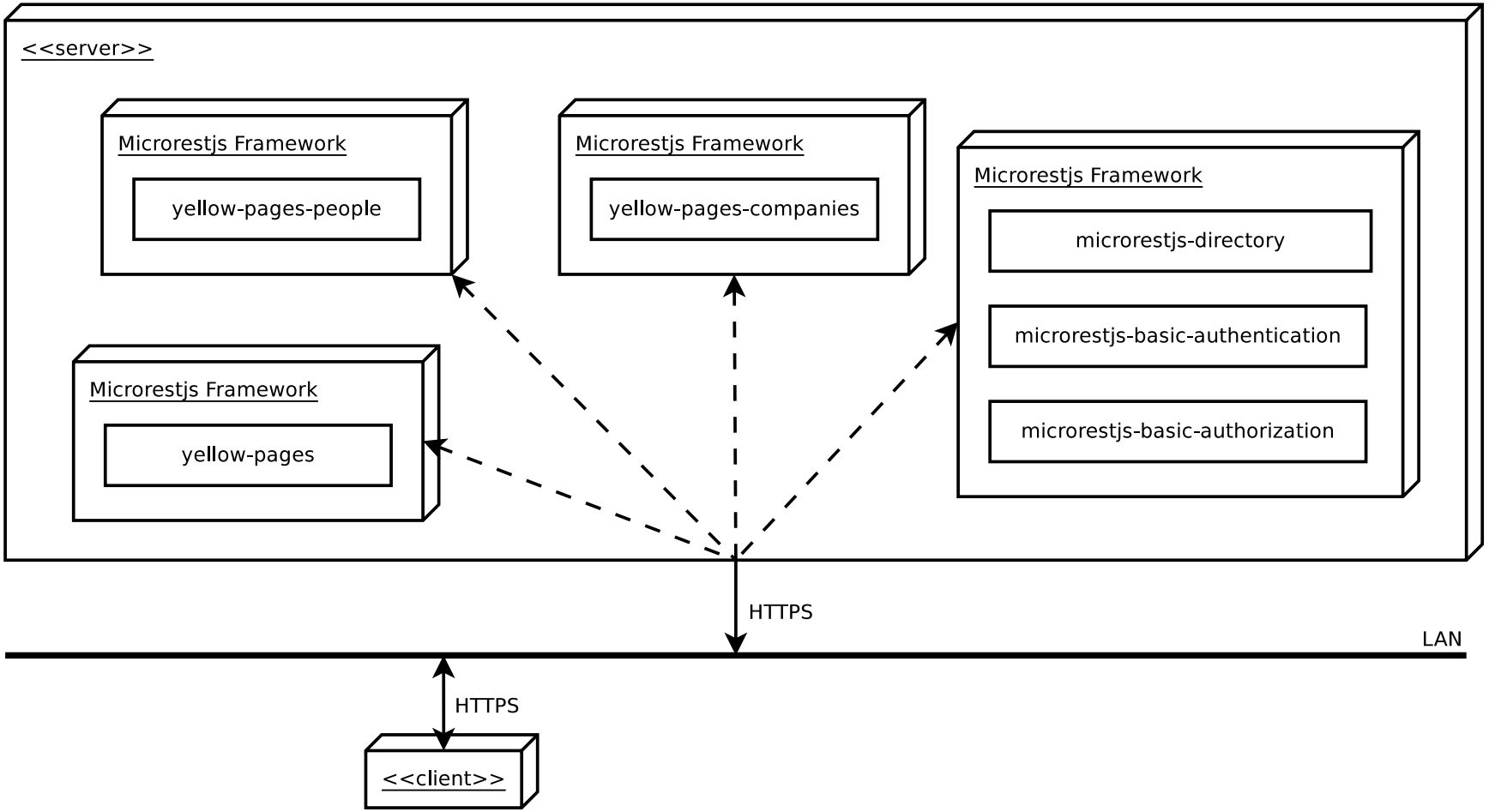
DEMO

5 min

Yellow Pages: Logical View



Yellow Pages: Deployment View



Yellow Pages: Summary

- 3 microservices working together
- Search and filter functionality
- Secure network communications
- Authentication and authorization

< 120 lines of code

Conclusion

Microrestjs is an innovative platform that aims to be the basis of the future technologies.