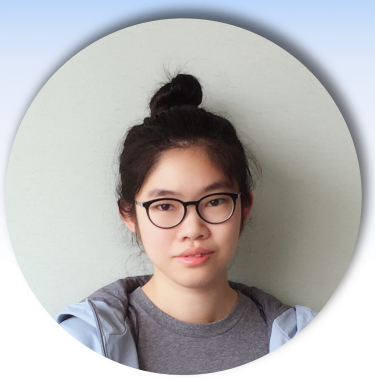
LoopBack

Write Scalable and Extensible Node.js Applications using LoopBack 4

LoopBack Toronto Team

Diana Lau
dhmlau@ca.ibm.com
@dhmlau



Janny Hou
juehou@ca.ibm.com
@jannyHou

Dominique Emond
dremond@ca.ibm.com
@emonddr

Agnes Lin
agneslin.lin@ibm.com
@agnes512

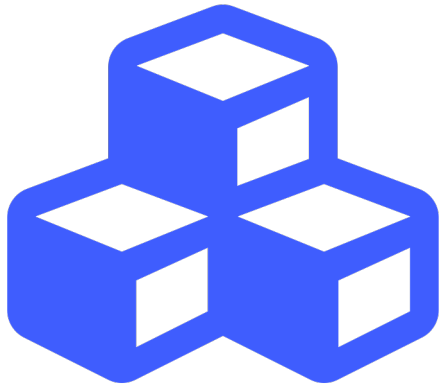
LoopBack

Let's Run a Few Polls 

- Do you work on Node.js applications, frameworks, or modules?
- Do you think your Node.js project is large scale?
- What are the characteristics of a large scale Node.js project?

A Large Scale Node.js Application

# of Teams,
Developers, Users



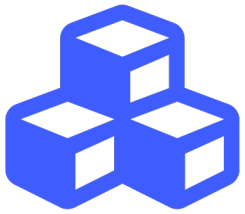
# of NPM Packages,
GitHub Repos

# of Years & Releases of
Development and Maintenance

LoopBack – a large scale Node.js framework

12,000+ GitHub Stars

8 Full-time Maintainers
+ Many Community Contributors



50+ NPM Modules

Created 5+ Years Ago

150,000+ Monthly Downloads

2 Major Releases (v3 and v4)

LoopBack

BEST IN
API MIDDLEWARE

PRESENTED TO
LoopBack

COMPANY
IBM

THE 2019
API AWARDS

presented at

{API:WORLD}

OCT 8-10, 2019 | SAN JOSE, CA

Challenges We Addressed

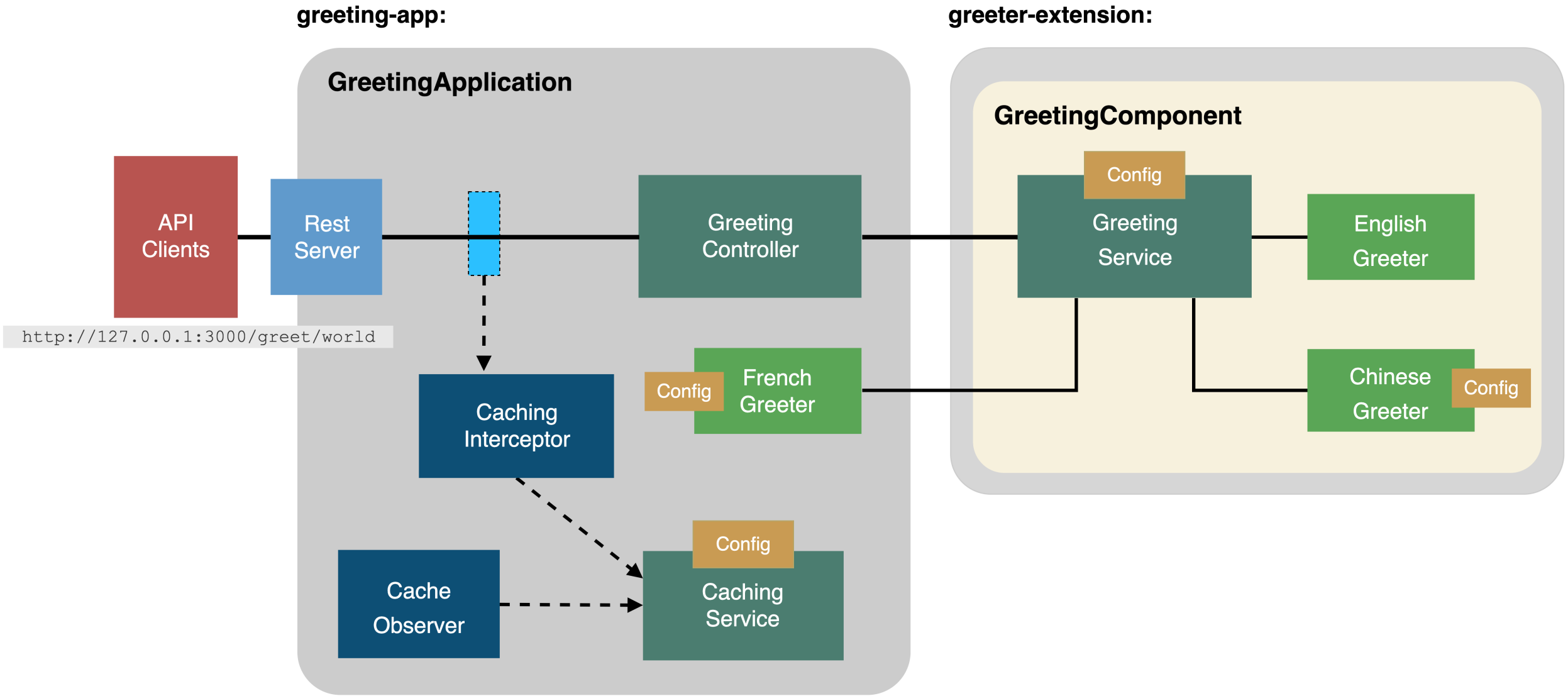
Customize the behavior of certain features

Extend the capabilities of your framework/application

Decompose a sequence of processing flow into actions

Compose a set of actions into a meaningful flow/sequence

Application Scenario



Part 0
Before We Get Started

+ Install Node.js version `8.9` or higher

+ Set up development environment

+ Install LoopBack 4 CLI

```
$ npm i -g @loopback/cli
```

+ Clone our git repository

```
$ git clone https://github.com/strongloop/cascon2019.git
$ cd cascon2019
```

>> The code you need for the workshop are all in the instruction:
https://github.com/strongloop/cascon2019

Part 1

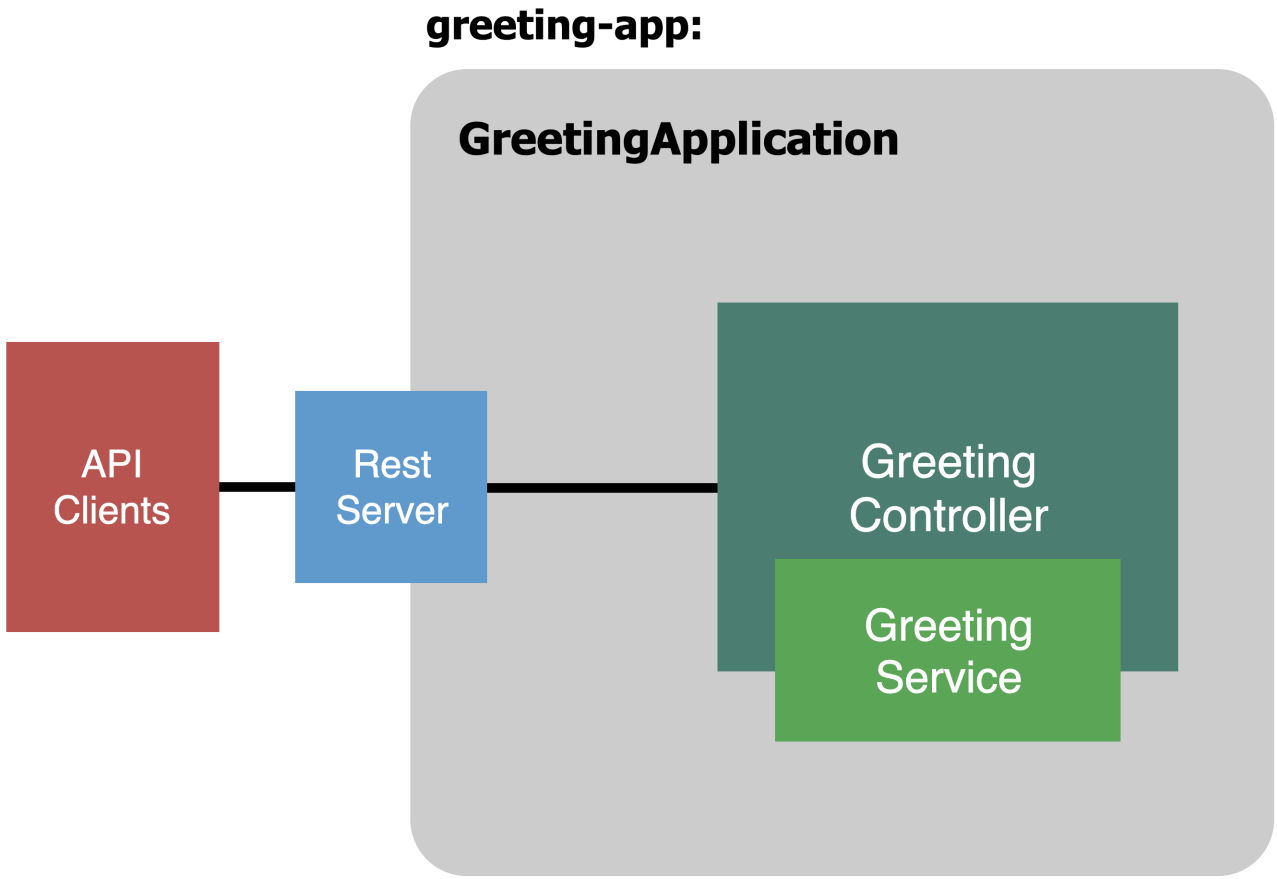
Enter the following command to start get started:

```
$ git checkout master
$ cd greeting-app
$ npm i
```

Part 1
Simple Application

Simple

One Module


greeting-app:
GreetingApplication
API
Clients
Rest
Server
Greeting
Controller
Greeting
Service

Part 2

If you haven't finished the previous step,
enter the following commands to start on next part

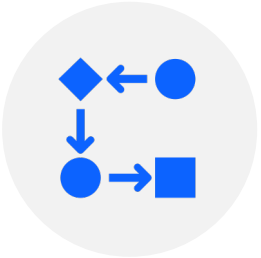
```
$ git stash
$ git checkout workshop-part1-completed
$ npm i
```

Part 2
Decouple the Application

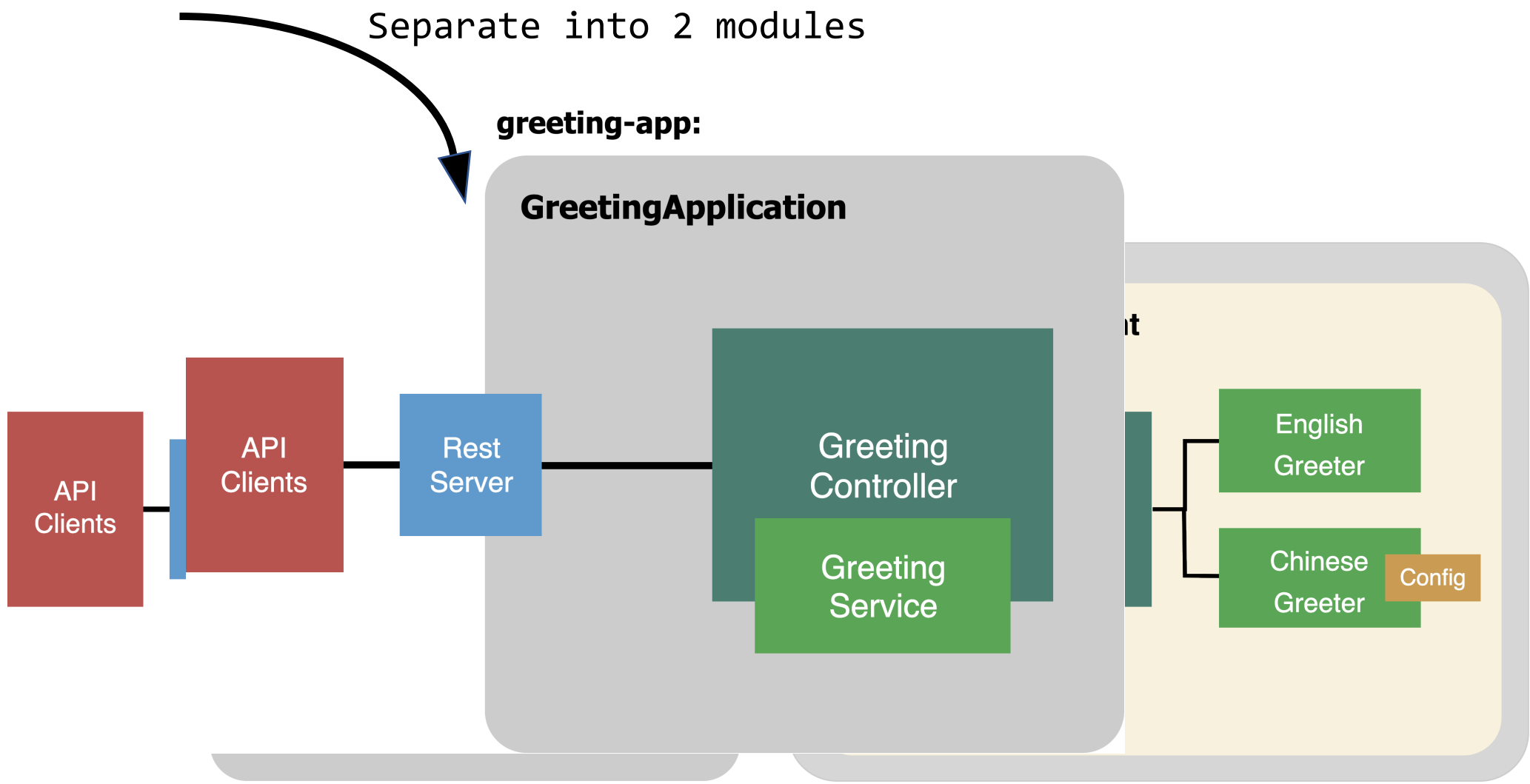
Component

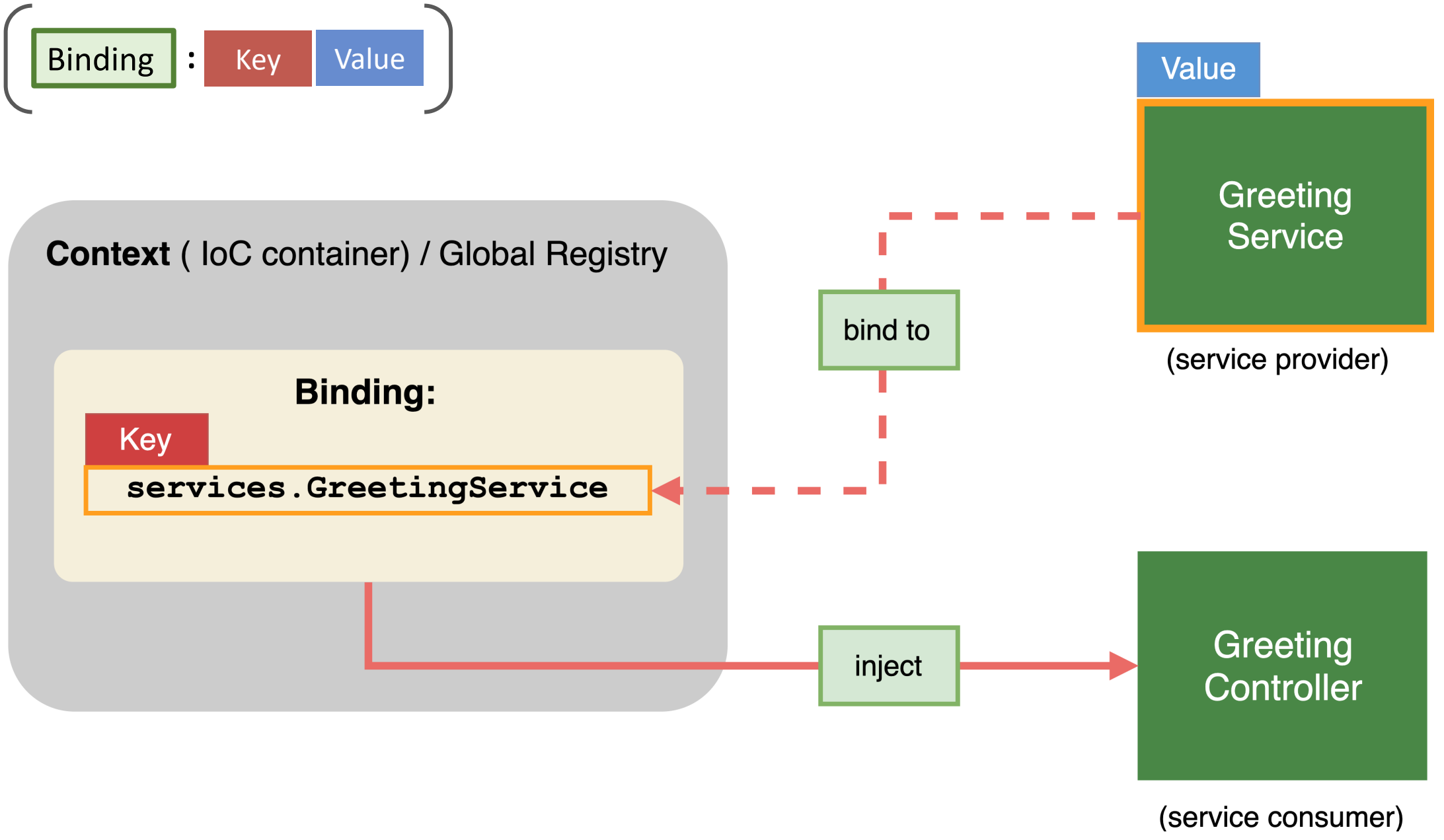
Dependency Injection

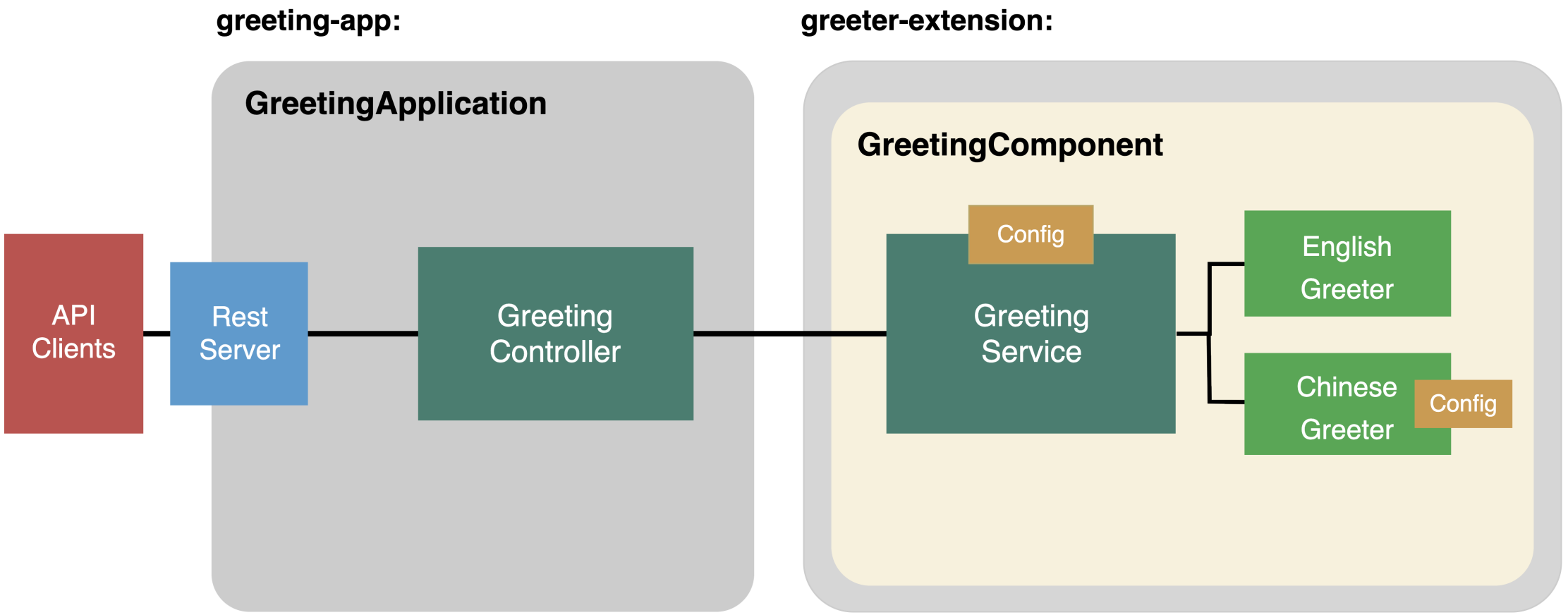
Binding



Inversion of Control


Separate into 2 modules
greeting-app:
GreetingApplication
API Clients
API Clients
Rest Server
Greeting Controller
Greeting Service
English Greeter
Chinese Greeter
Config


Binding
:
Key
Value
Value
Greeting Service
(service provider)
Context (IoC container) / Global Registry
bind to
Binding:
Key
services.GreetingService
inject
Greeting Controller
(service consumer)


greeting-app:
GreetingApplication
API Clients
Rest Server
Greeting Controller
greeter-extension:
GreetingComponent
Config
Greeting Service
English Greeter
Chinese Greeter
Config

Part 3

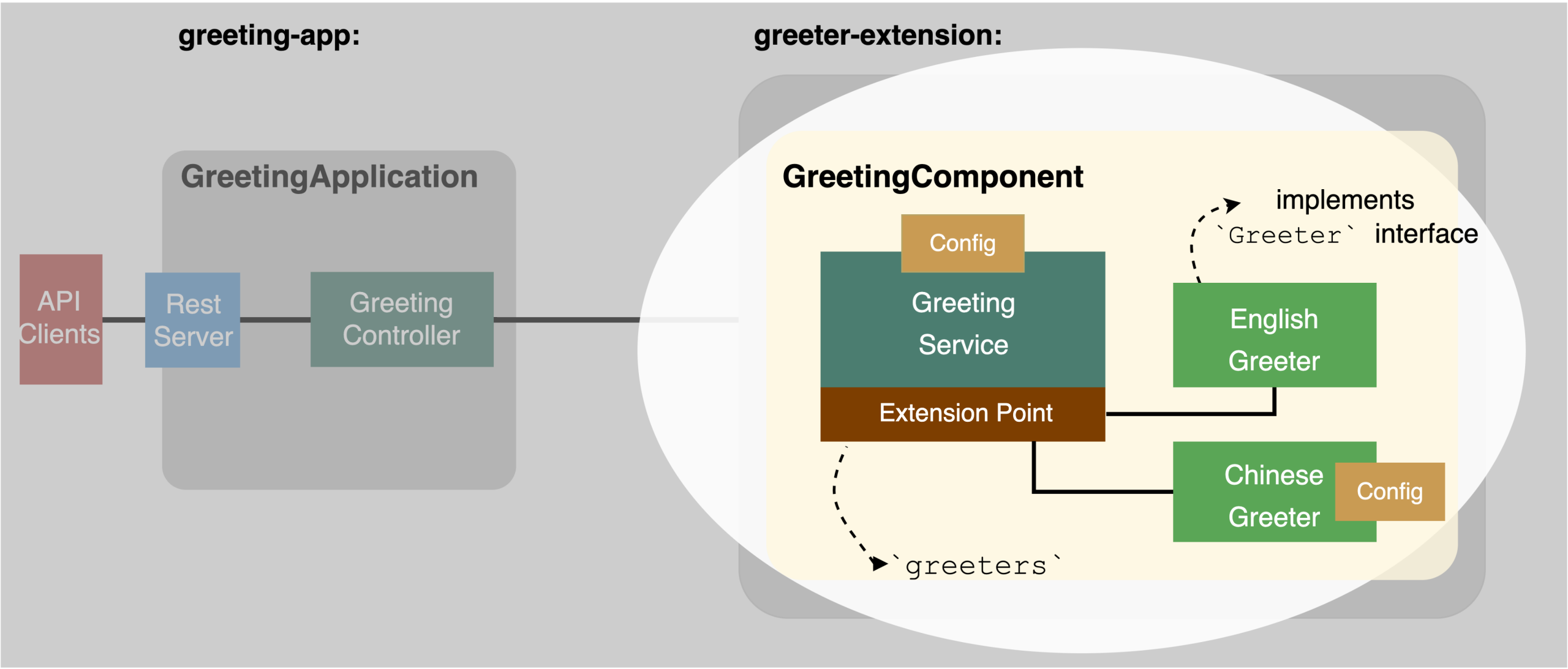
If you haven't finished the previous step,
enter the following commands to start on next part

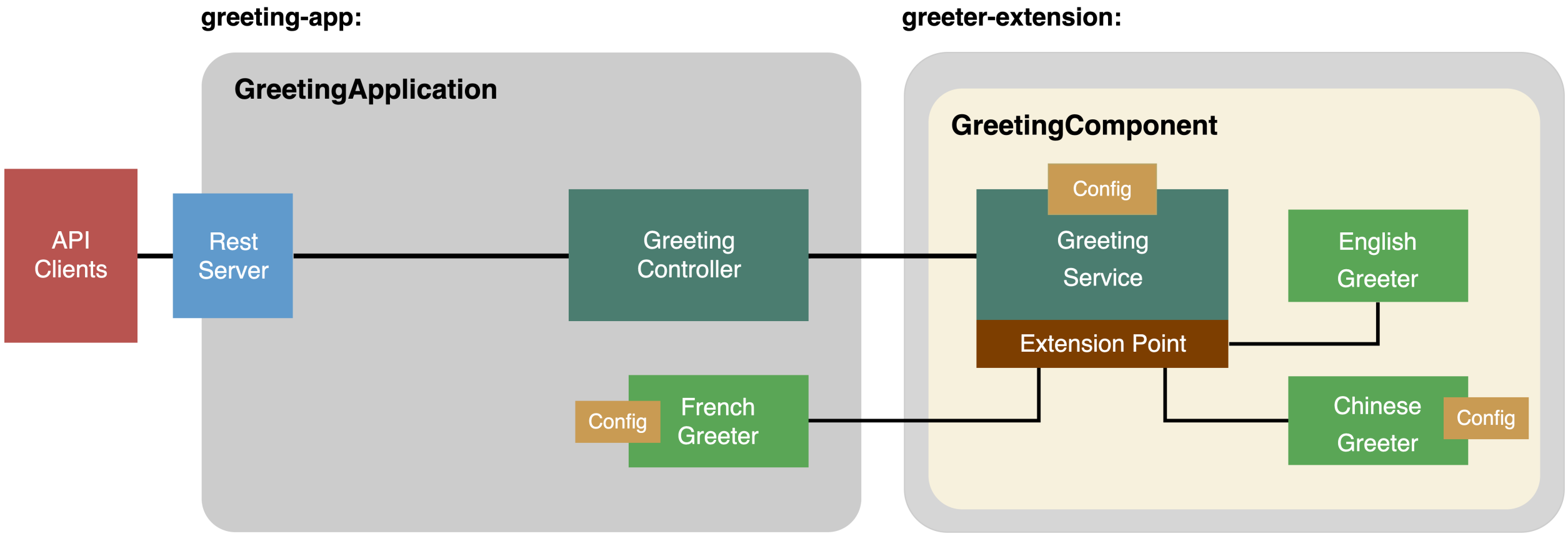
```
$ git stash
$ git checkout workshop-part2-completed
$ npm i
```

Part 3.1
Add a New Language Extension

Extension Point

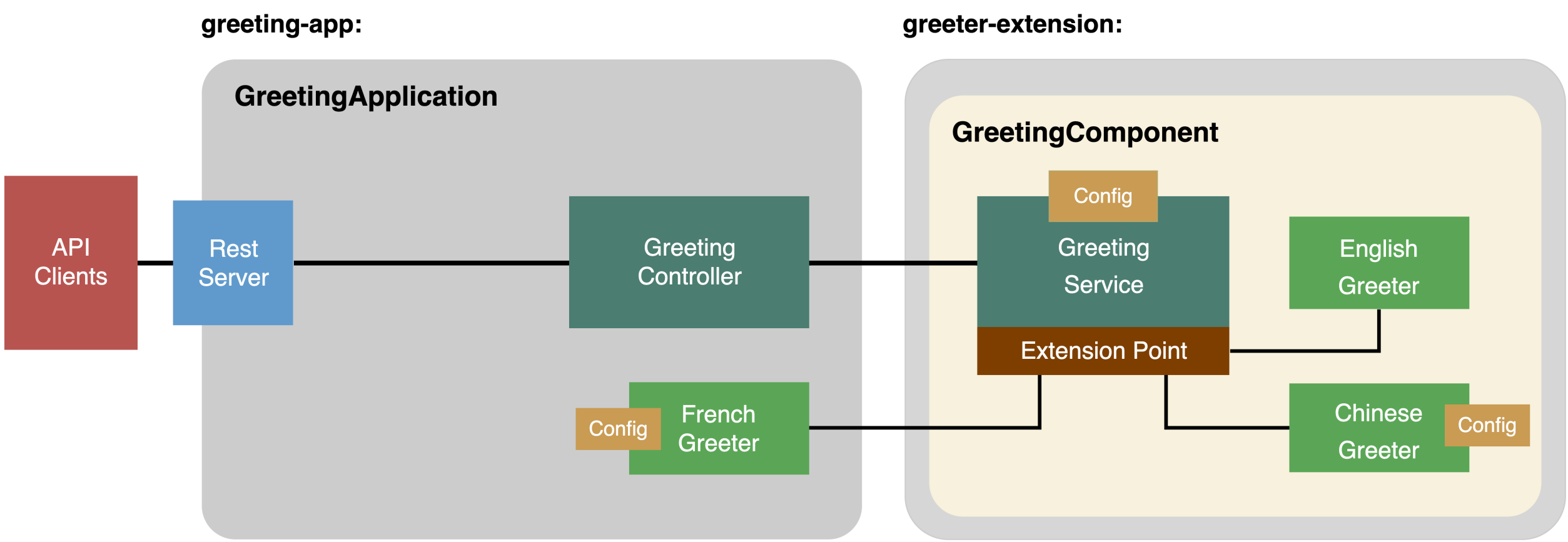
Extension


greeting-app:
greeter-extension:
GreetingApplication
API Clients
Rest Server
Greeting Controller
GreetingComponent
Config
Greeting Service
Extension Point
implements `Greeter` interface
English Greeter
Chinese Greeter
Config
`greeters`


greeting-app:
GreetingApplication
API Clients
Rest Server
Greeting Controller
Config
French Greeter
greeter-extension:
GreetingComponent
Config
Greeting Service
Extension Point
English Greeter
Chinese Greeter
Config

Part 3.2
Configuration

Configuration


greeting-app:
GreetingApplication
API Clients
Rest Server
Greeting Controller
Config
French Greeter
greeter-extension:
GreetingComponent
Config
Greeting Service
Extension Point
English Greeter
Chinese Greeter
Config

Part 4

If you haven't finished the previous step,
enter the following commands to start on next part

```
$ git stash
$ git checkout workshop-part3-completed
$ npm i
```

Part 4
Enable Caching

Interceptor

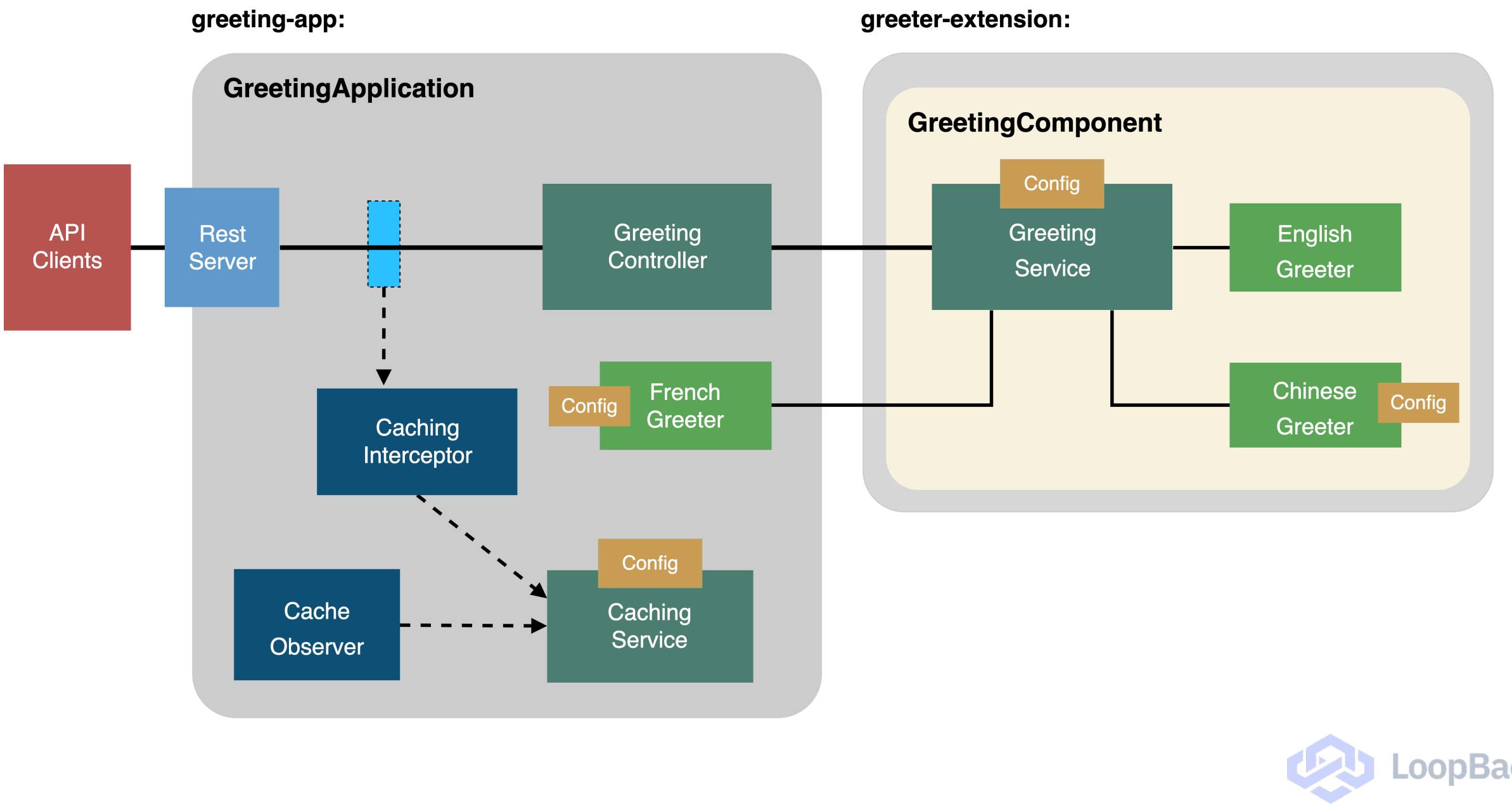
Observer

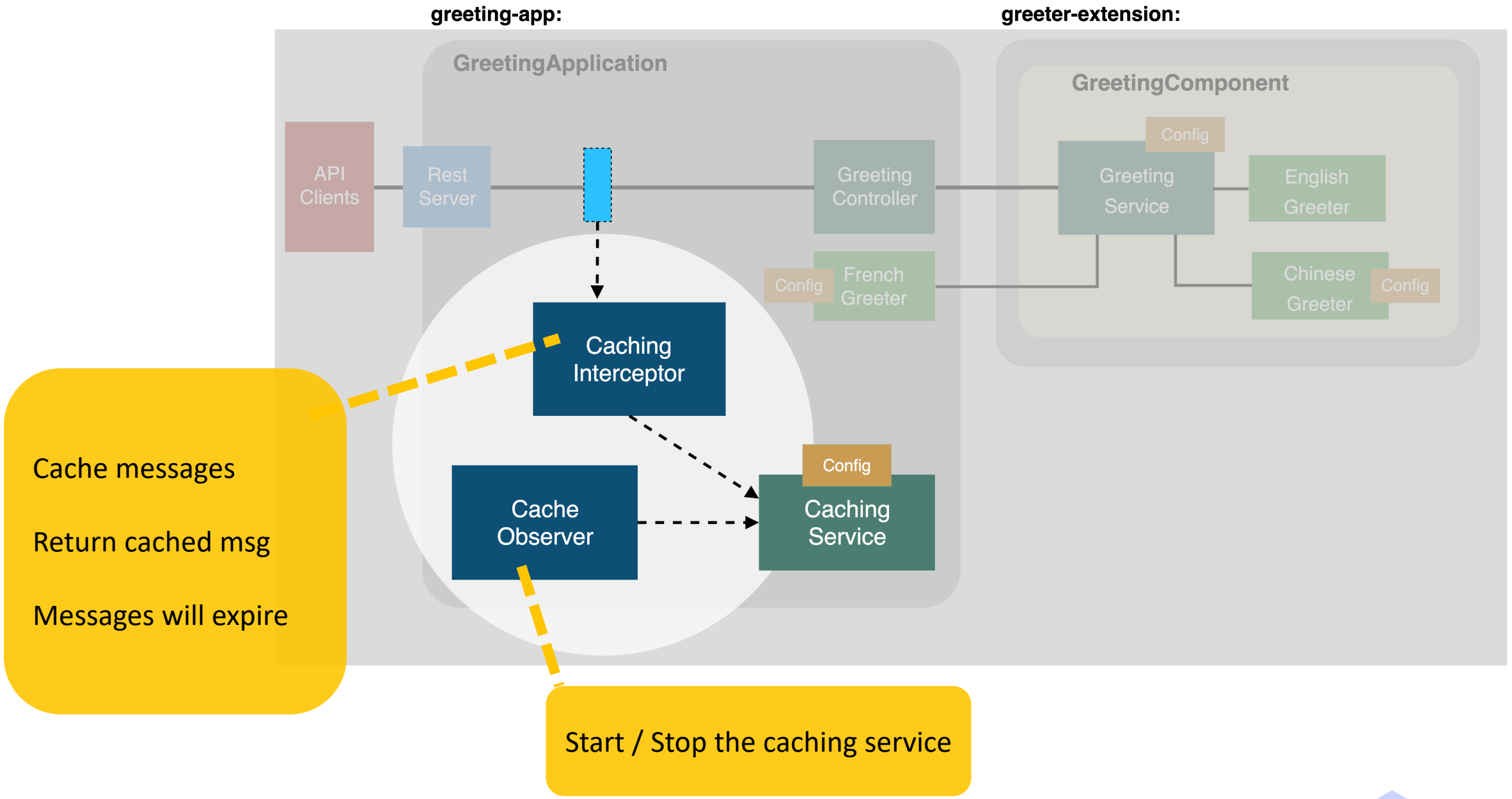
Let's Build a Caching System That Does:

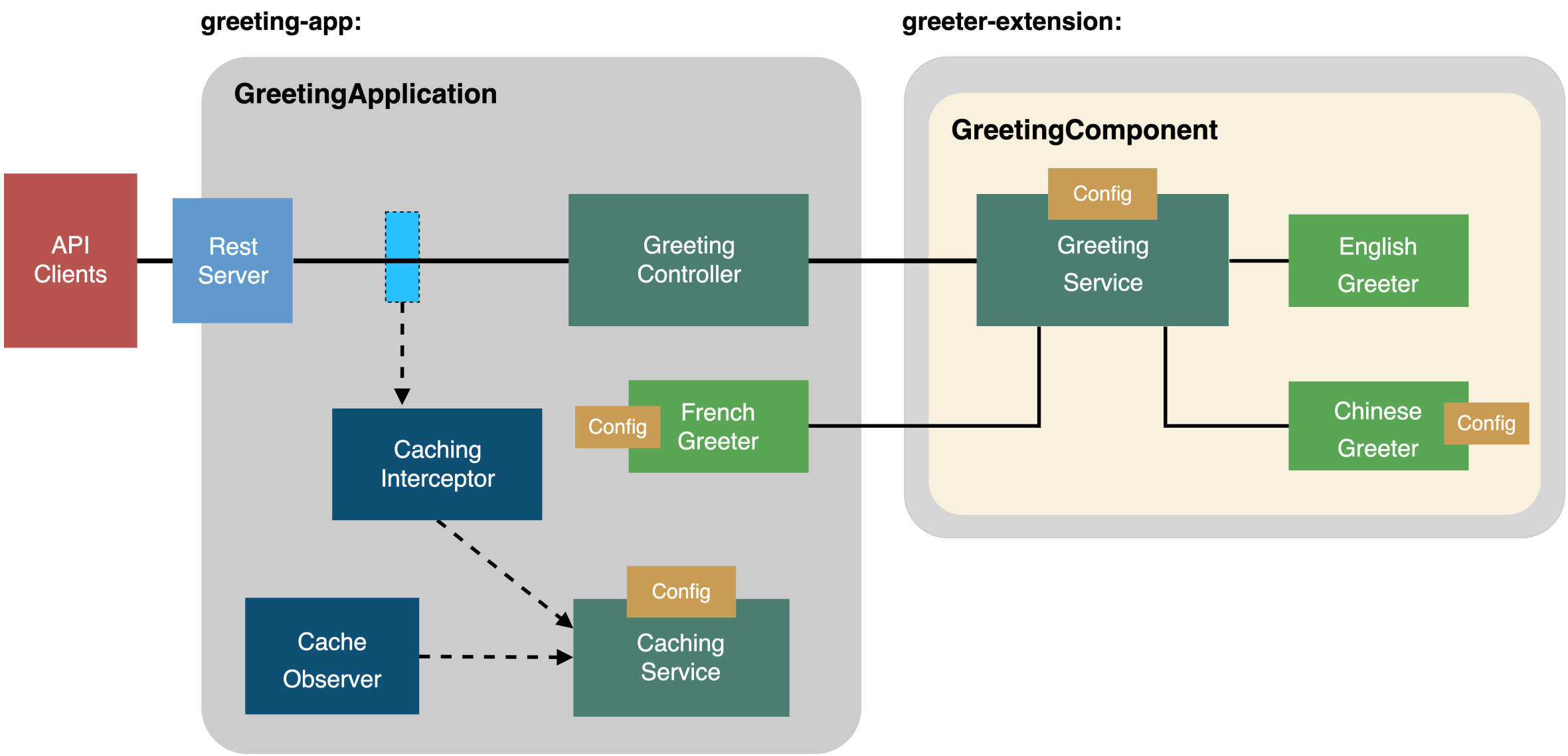
+ Cache the result using the request's URI and language as the key

+ The cached message has an expiration time

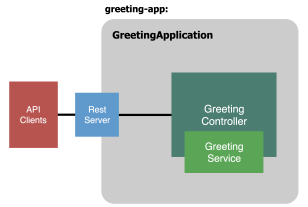
+ Expired messages will be swept out periodically


greeting-app:
greeter-extension:
GreetingApplication
GreetingComponent
API Clients
Rest Server
Greeting Controller
Caching Interceptor
Config
French Greeter
Cache Observer
Config
Caching Service
Config
Greeting Service
English Greeter
Chinese Greeter
Config


greeting-app:
greeter-extension:
GreetingApplication
GreetingComponent
API Clients
Rest Server
Greeting Controller
Greeting Service
Config
English Greeter
Chinese Greeter
Config
French Greeter
Config
Caching Interceptor
Cache Observer
Config
Caching Service
Cache messages
Return cached msg
Messages will expire
Start / Stop the caching service


greeting-app:
greeter-extension:
GreetingApplication
GreetingComponent
API Clients
Rest Server
Greeting Controller
Caching Interceptor
Config
French Greeter
Cache Observer
Config
Caching Service
Config
Greeting Service
English Greeter
Chinese Greeter
Config

Congratulations! You've completed the workshop

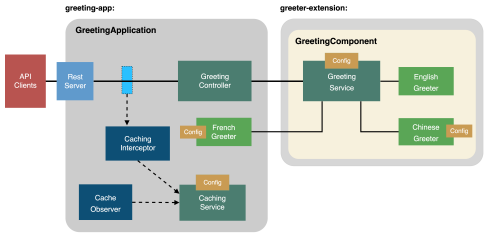


★ Built a LoopBack application and separated it into two modules.

★★ Enriched the functionality of the app by adding extensions (`FrenchGreeter`).

★★★ Created a caching service that is started and stopped by a lifecycle observer.

★★★★ Retrieved/stored request responses from/in the cache with an interceptor.



You can find the completed code of this workshop in the git branch:

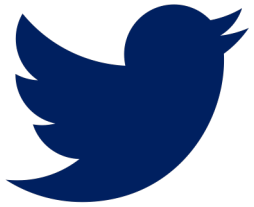
```
$ git checkout workshop-part4-completed
```

LoopBack

Thank You!

loopback.io

Strongloop/loopback-next



@StrongLoop

Acknowledgement

- The LoopBack team at IBM and our contributors from the community making good things happen in open source
- Taranveer Virk and Raymond Feng - allowing us to reuse some of their slides
- Iconography by FontAwesome is licensed under CC BY 4.0