

Micronaut vs Spring Boot, или «кто тут самый маленький?»



ЦИАН



@tolkv



@lavcraft



Кирилл



Максим

@gorelikoff



@gorelikov



Развлекательно – познавательный доклад

 @tolkv

 @lavcraft



Кирилл

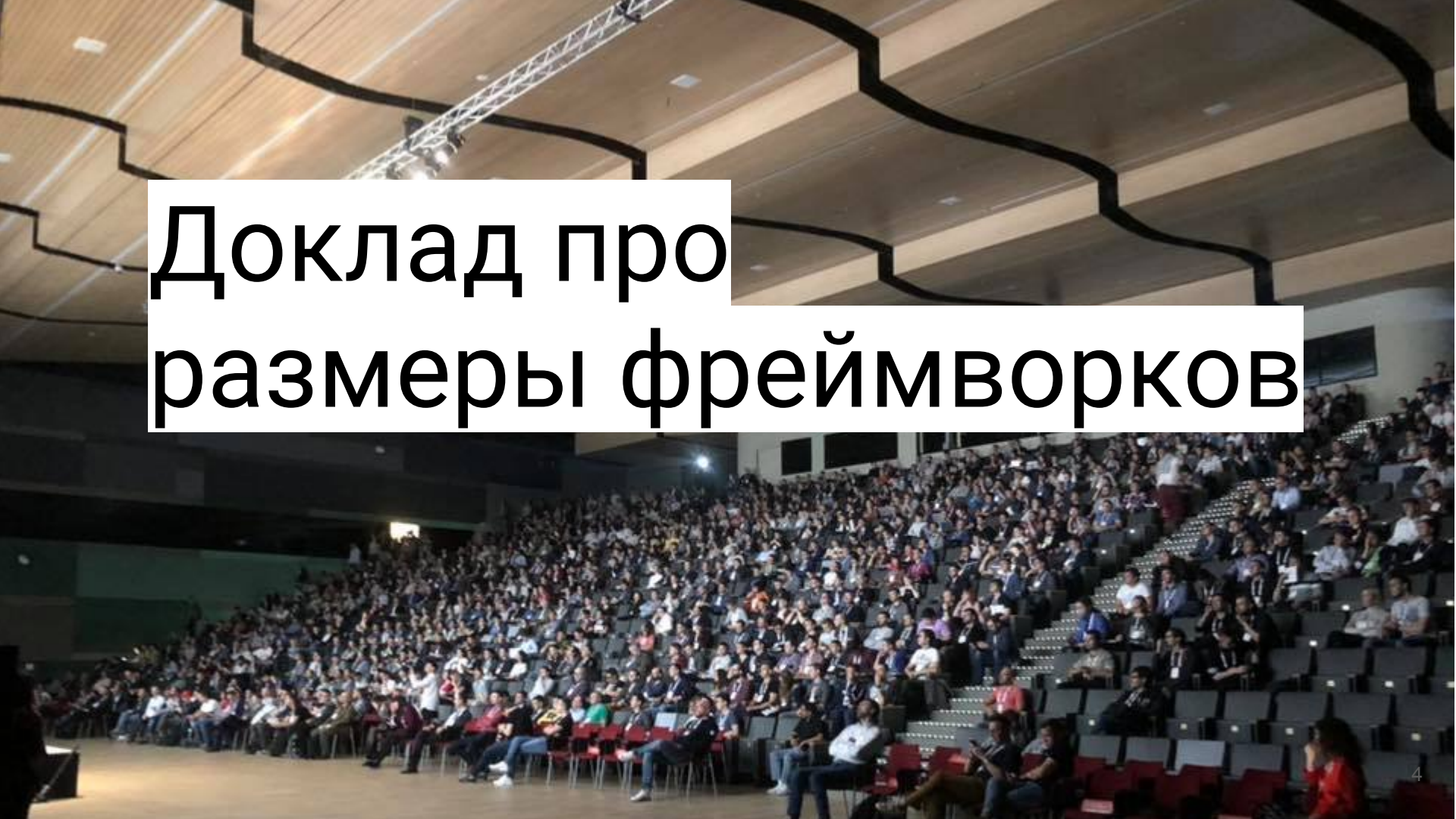


Максим

@gorelikoff 

@gorelikov 

Доклад про размеры фреймворков



A wide-angle photograph of a large, modern auditorium. The ceiling is a complex, multi-tiered structure with a light-colored wood-grain finish and dark, wavy, geometric patterns. A long, white truss structure with several spotlights hangs from the ceiling. The audience is seated in rows of dark grey chairs, filling the lower two-thirds of the frame. The stage area is visible in the foreground, with a few people seated on red chairs. The overall atmosphere is that of a large-scale event or performance.

Такой большой зал

A large audience is seated in a modern hall with a distinctive ceiling design. The audience is diverse in age and appearance. In the foreground, a cartoon character with a large, bushy yellow head and a long, thin yellow body is overlaid on the image. The character has a surprised expression with wide eyes and an open mouth. The hall has a wooden floor and a ceiling with a series of white, curved, overlapping panels. The lighting is bright, and the overall atmosphere is that of a large-scale event or conference.

Такой большой зал

И такой маленький фреймворк

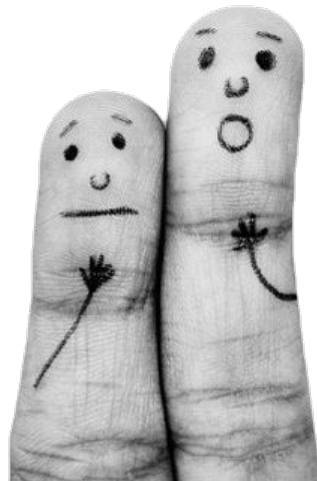
Развлекательно – познавательный доклад

- Сравнение «микро» фреймворков

Кто,
если
не мы?

Развлекательно – познавательный доклад

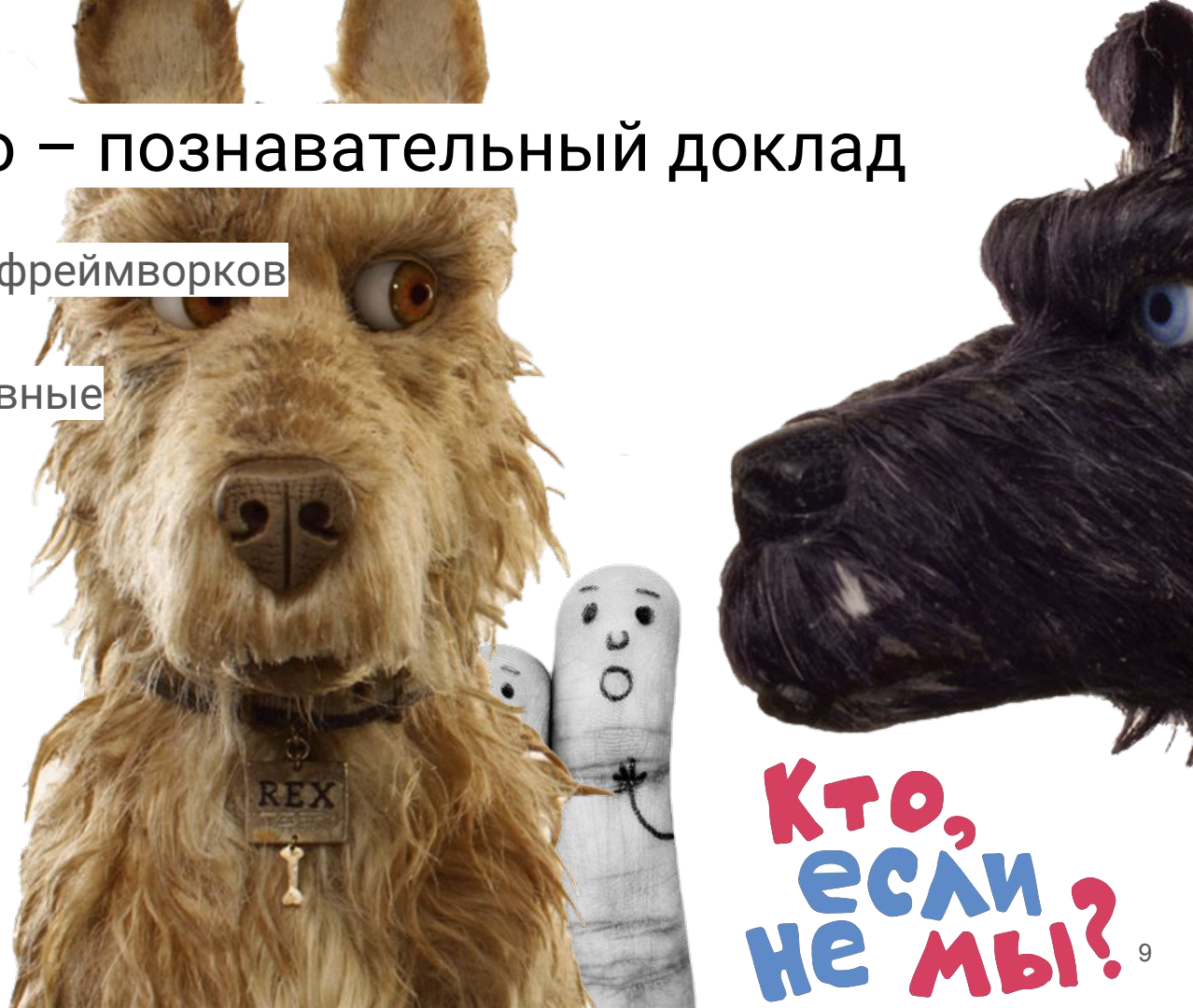
- Сравнение «микро» фреймворков
- Демонстрация – весёлая



**КТО,
ЕСЛИ
НЕ МЫ?**

Развлекательно – познавательный доклад

- Сравнение «микро» фреймворков
- Демо – весёлое
- Выводы – субъективные



КТО,
ЕСЛИ
НЕ МЫ? ⁹



Что мы делаем на работе?



Microservices



Что мы делаем на работе?



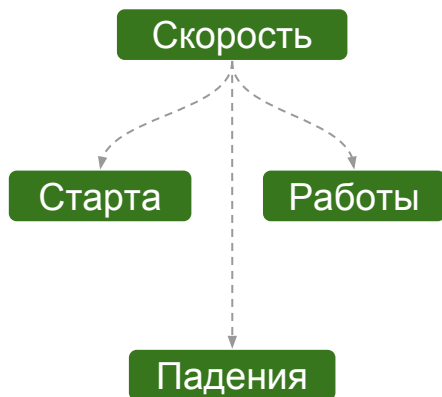
Что за микросервисы?



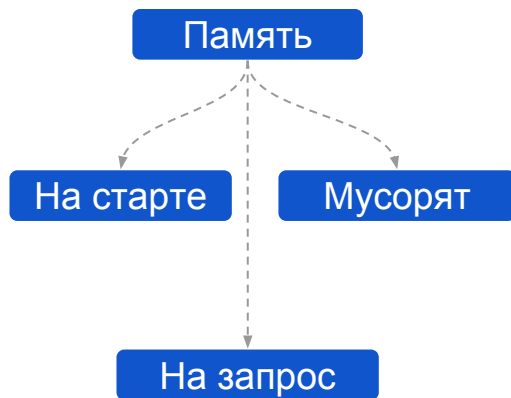
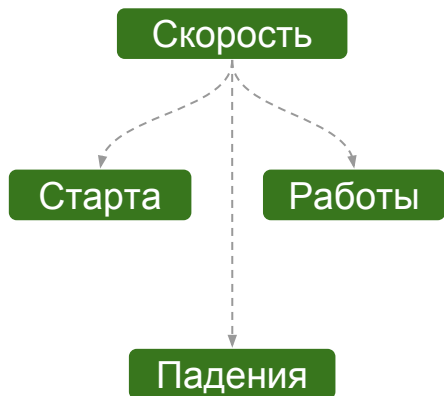


Что за микрофреймворки?

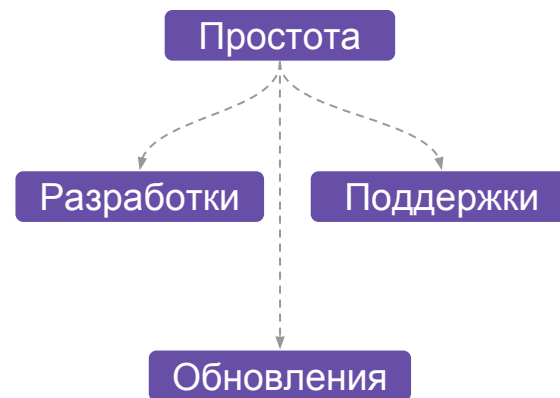
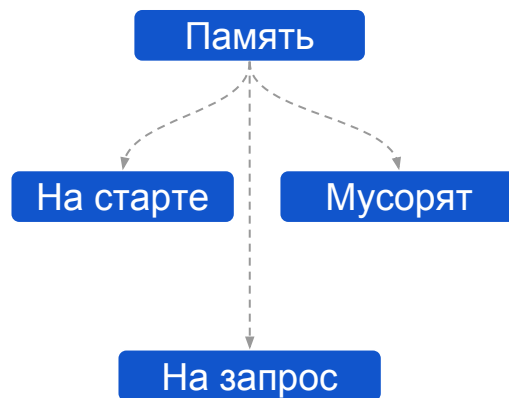
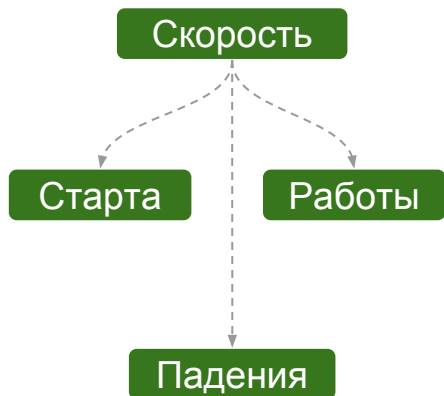
Как оценить «микроскопичность»? ?



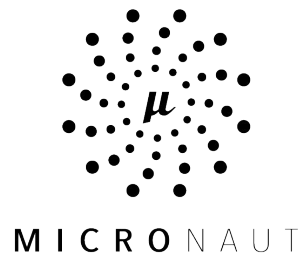
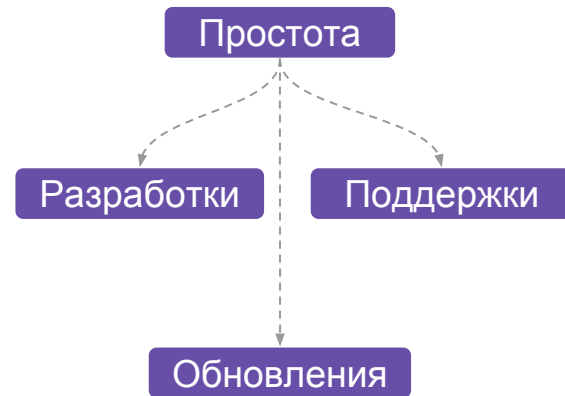
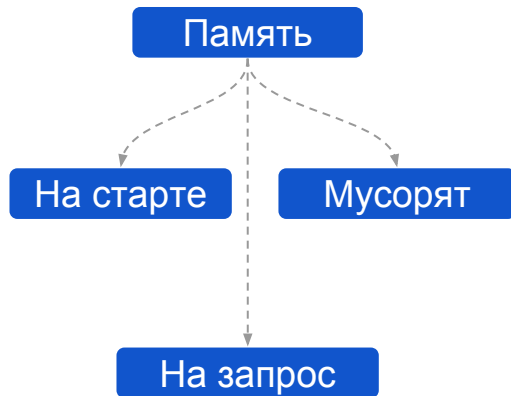
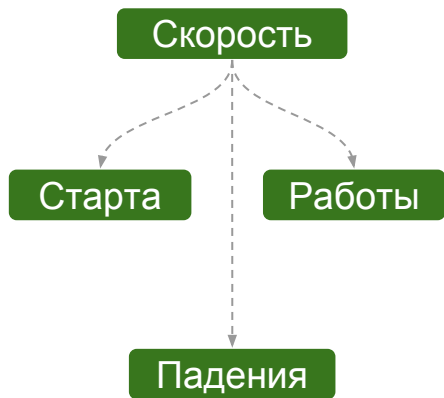
Как оценить «микроскопичность»?



Как оценить «микроскопичность»?



Как оценить «микроскопичность»? ?



Ktor

А в чем проблема-то?



A black and tan dog, possibly a German Shepherd, is lying on a wooden floor. The dog is facing left and has its head down. The floor is made of light-colored wooden planks. In the background, there is a green wall on the left and a white wall on the right.

Spring Boot потолстел со временем*

*По данным опросов и ВЦИОМ

Этот фреймворк
не в моде



Этот память жрёт
как не в себя



Этот слишком
замороженный
и сложный





А этот вообще,
запускается
больше 2 секунд

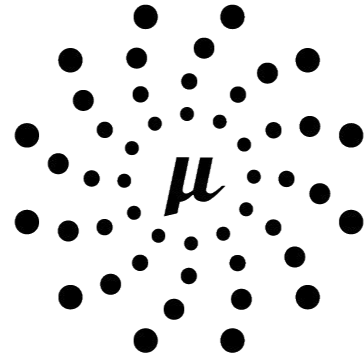
**Сравним
самое популярное
с самым хайповым**





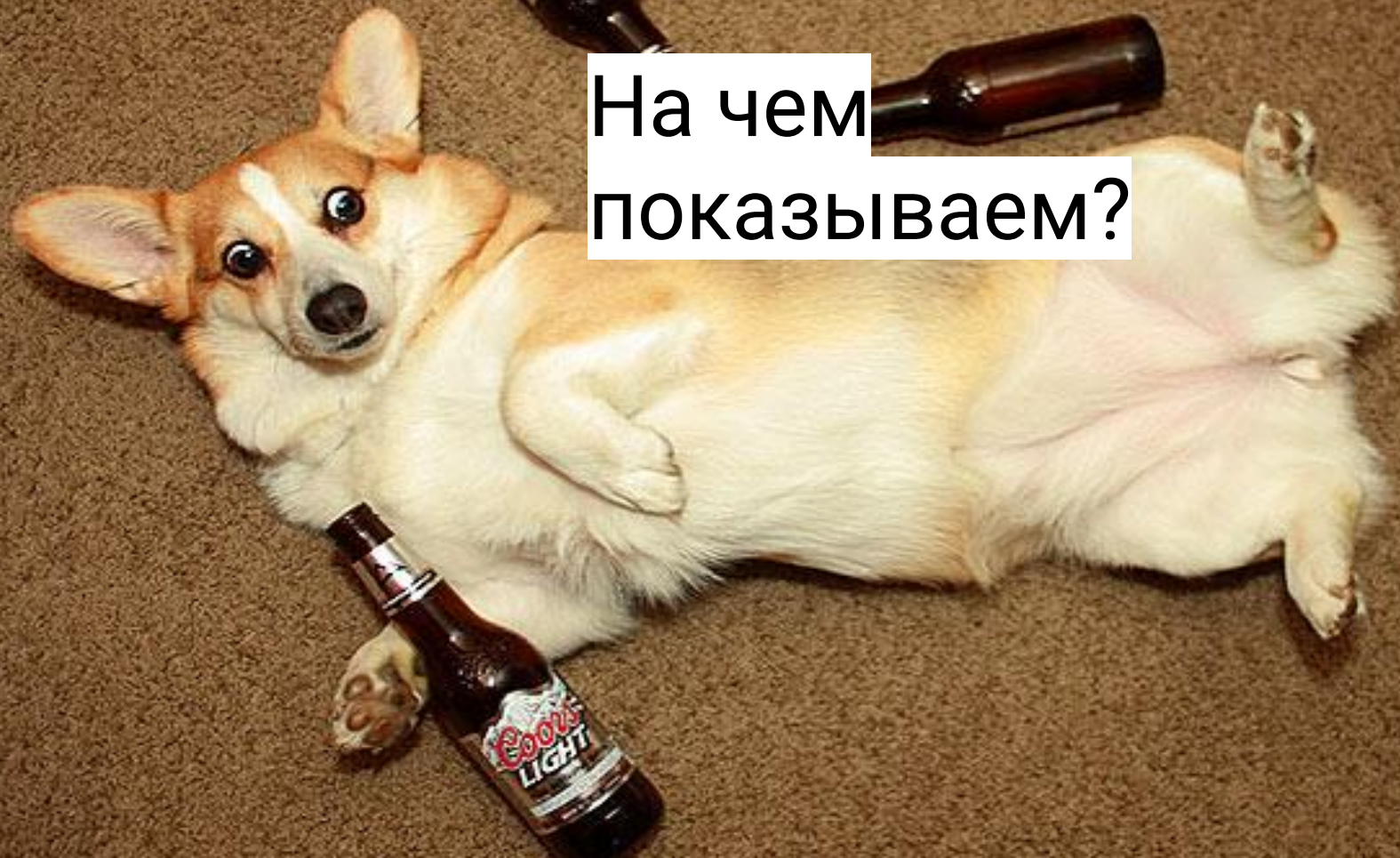
Spring Boot
vs
Micronaut

Spring Boot VS Micronaut

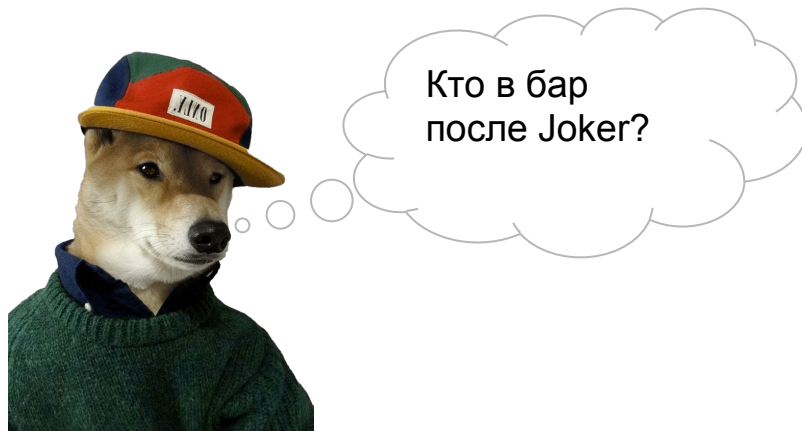


M I C R O N A U T

На чем
показываем?



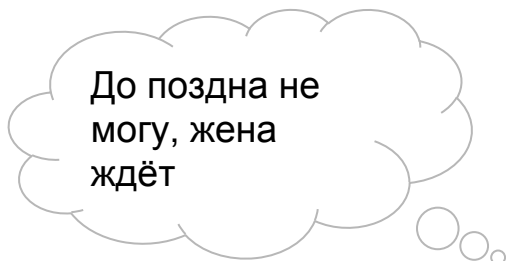
Стартап – Собери компанию в бар



Стартап – Собери компанию в бар



Стартап – Собери компанию в бар



Стартап – Собери команду в бар

По сколько скидываемся?



Стартап – Собери компанию в бар



Стартап – Собери компанию в бар



Бар Бульдог
каждое 3я
пинта
бесплатно!



Архитектура

Does it run? Just leave it alone.



Writing Code that Nobody Else Can Read

The Definitive Guide

@ThePracticalDev

O'REILLY?



4th Edition

АРХИТЕКТУРА

КАКАЯ ЕЩЕ АРХИТЕКТУРА?

о трех буквах на которые вас послал
проджект менеджер

Василий Пупкин

O'REILLY®



Росинский Е.Б. & Ахметов С.Ю.



Темный Столик

Essential

Googling the Error Message

The Practical Developer
@ThePracticalDev

O'REILLY?

Does it run? Just leave it alone.



Why Code that
Nobody Else Can Read

Не плачь,
херачь



4th Edition

КАК
ДЕЛАТЬ

АРХИТЕКТУРА

КАКАЯ ЕЩЕ АРХИТЕКТУРА?

три буквы на которые вас послал
проект менеджер

Василий Пупкин

O'REILLY®



Росинский Е.Б. & Ахметов С.Ю.



Темная
Корда

КАК
ДЕЛАТЬ

Essential

Googling the
Error Message

O'REILLY®

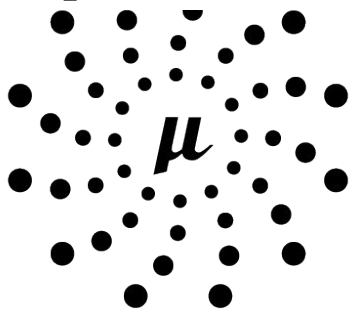
The Practical Developer
@ThePracticalDev



=



Да начнётся битва



=



M I C R O N A U T

Да начнётся битва



VS



Условия

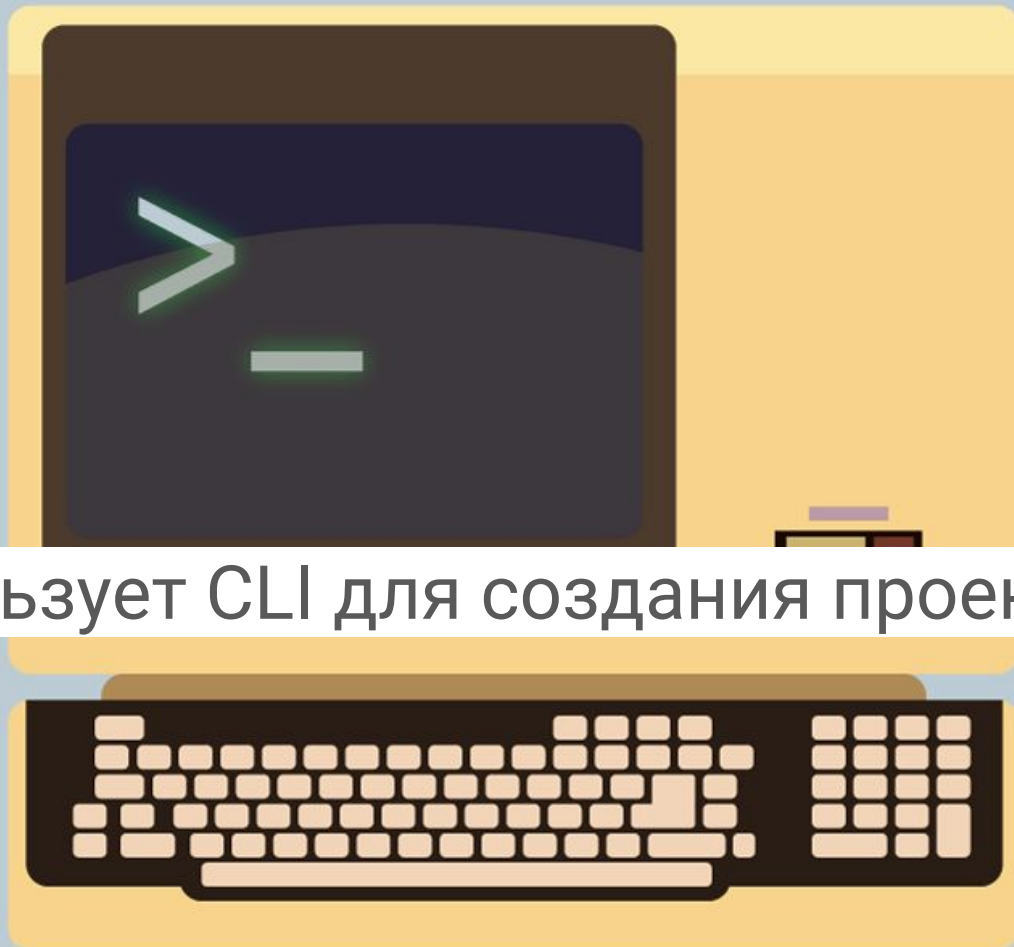
- Intel Core I5 3.3 GHz
- 16GB
- OSX 10.13.6
- Java 8.0.181-oracle(build 1.8.0_181-b13)



Шаг 1 - создание



CLI



Кто использует CLI для создания проекта?

Инструменты

- spring cli → `spring`
- micronaut cli → `mn`

Инструменты

- spring cli → spring → 6
- micronaut cli → mn → 2

Инструменты

- spring cli → spring → 6
- micronaut cli → mn → 2



Micronaut

Secure | https://docs.micronaut.io/latest/guide/index.html

Apps web eng documenting Обучение show processes Segmentation Fau... Other Bookmarks

Table of Contents

- 1 Introduction
 - 1.1 What's New?
- 2 Quick Start
 - 2.1.2 Install through Binary on Windows
 - 2.1.3 Building from Source
 - 2.2 Creating a Server Application
 - 2.3 Setting up an IDE
 - 2.4 Creating a Client
 - 2.5 Deploying the Application
- 3 Inversion of Control
 - 3.1 Defining Beans
 - 3.2 How Does it Work?
 - 3.3 The BeanContext

MICRONAUT™ Table of contents API Reference Configuration Ref

Updating Micronaut Gradle Build for 1.0.0 RC3

```
dependencyManagement {  
    bom: 'io.micronaut:bom:1.0.0.RC3' 1  
}  
  
dependencies {  
    annotationProcessor "io.micronaut:micronaut-inject-java" 2  
    compile "io.micronaut:micronaut-inject" 3  
    // repeat rename for remaining "io.micronaut" dependencies  
    ...  
}
```

- 1 Renamed bom → micronaut-bom
- 2 Renamed inject-java → micronaut-inject-java
- 3 Renamed inject → micronaut-inject

Обратимся к документации

MN CLI умеет создавать федерации

→ `mn create-federation buster --services application,categories,mess...`

<http://buster.igg.biz/iwantbeer/{nickname}/{money}>

A close-up photograph of a Corgi dog lying on a green lawn. The dog has its mouth open, showing its pink tongue and teeth. A white, textured object, resembling a beer can, is balanced on top of its head. The dog is wearing a purple collar with a metal buckle. The background is a bright, sunny grassy area.

Не все идет по плану

<http://buster.igg.biz/iwantbeer/{nickname}/{money}>

MN CLI federation RC1

```
| Generating Java project....  
| Error Error occurred running Micronaut CLI: null  
java.lang.NullPointerException  
  at io.mn.cli.pr.co.AbstractCreateCommand$_addDependentFeatures_closure17.doCall  
  at io.mn.cli.pr.co.AbstractCreateCommand.addDependentFeatures(AbstractCreateCommand.groovy:571)  
  at io.mn.cli.pr.co.AbstractCreateCommand.populateFeatures(AbstractCreateCommand.groovy:544)  
  at io.mn.cli.pr.co.AbstractCreateCommand.evaluateFeatures(AbstractCreateCommand.groovy:477)  
  at io.mn.cli.pr.co.AbstractCreateCommand.handle(AbstractCreateCommand.groovy:272)  
  at io.mn.cli.pr.co.CreateFederationCommand.handle(CreateFederationCommand.groovy:92)  
  at io.mn.cli.MicronautCli.executeCommand(MicronautCli.groovy:351)  
  at io.mn.cli.MicronautCli.execute(MicronautCli.groovy:234)  
  at io.mn.cli.MicronautCli.main(MicronautCli.groovy:175)  
| Error Error occurred running Micronaut CLI: null
```

<http://buster.igg.biz/iwantbeer/{nickname}/{money}>

Micronaut federation RC2

```
$ tree -L 1
```

```
├── bar-emulator  
├── places-api  
└── beer-api
```

Micronaut federation RC2

```
$ tree -L 1
```

```
.  
├── bar-emulator  
├── places-api  
└── beer-api
```

Никак не связанные
проекты

Micronaut federation RC3

```
$ tree -L 1
```

```
.  
├── bar-emulator  
├── places-api  
├── beer-api  
├── gradle  
├── gradlew  
├── gradlew.bat  
└── settings.gradle
```

Связанные на уровне сборки проекты

Бывает IDE странно реагирует на что-то новое

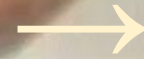
Information: 2018-09-25 09:49 - Compilation completed with 1 error and 0 warnings in 1 s 830 ms

Error: Groovyc: While compiling application-api_main: BUG! exception in phase 'canonicalization' in source unit
'/Users/m.gorelikov/Documents/workspace/conference/joker2018/buster/application-api/src/main/groovy/application/api/controller/ApplicationController.groovy' JVM class can't be loaded
application.api.repository.UserApplicationService
at org.codehaus.groovy.ast.decompiled.AsmReferenceResolver.resolveJvmClass(AsmReferenceResolver.java:88)
at org.codehaus.groovy.ast.decompiled.DecompiledClassNode.getTypeClass(DecompiledClassNode.java:176)
at org.codehaus.groovy.ast.ClassNode.getTypeClass(ClassNode.java:1359)
at io.micronaut.ast.groovy.utils.AstGenericUtils.resolveTypeReference(AstGenericUtils.groovy:192)
at io.micronaut.ast.groovy.utils.AstGenericUtils.resolveTypeReference(AstGenericUtils.groovy:153)
at io.micronaut.ast.groovy.InjectTransform\$InjectVisitor.visitProperty(InjectTransform.groovy:1015)
at org.codehaus.groovy.ast.ClassNode.visitContents(ClassNode.java:1091)
at org.codehaus.groovy.ast.ClassCodeVisitorSupport.visitClass(ClassCodeVisitorSupport.java:54)
at io.micronaut.ast.groovy.InjectTransform\$InjectVisitor.visitClass(InjectTransform.groovy:388)
at io.micronaut.ast.groovy.InjectTransform.visit(InjectTransform.groovy:161)
at org.codehaus.groovy.transform.ASTTransformationVisitor\$3.call(ASTTransformationVisitor.java:331)
at org.codehaus.groovy.control.CompilationUnit.applyToSourceUnits(CompilationUnit.java:968)
at org.codehaus.groovy.control.CompilationUnit.doPhaseOperation(CompilationUnit.java:633)
at org.codehaus.groovy.control.CompilationUnit.processPhaseOperations(CompilationUnit.java:609)
at org.codehaus.groovy.control.CompilationUnit.compile(CompilationUnit.java:586)
at org.jetbrains.groovy.compiler.rt.GroovyCompilerWrapper.compile(GroovyCompilerWrapper.java:62)
at org.jetbrains.groovy.compiler.rt.DependentGroovycRunner.runGroovyc(DependentGroovycRunner.java:119)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.jetbrains.groovy.compiler.rt.GroovycRunner.intMain2(GroovycRunner.java:90)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.jetbrains.jps.incremental.groovy.InProcessGroovyc.runGroovycInThisProcess(InProcessGroovyc.java:160)
at org.jetbrains.jps.incremental.groovy.InProcessGroovyc.lambda\$runGroovyc\$0(InProcessGroovyc.java:89)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:624)



**Сколько
нужно памяти**

для запуска



10m



100m



1g

Критерии успеха

Критерии успеха

1. Успешный запуск без исключений



Критерии успеха

1. Успешный запуск без исключений



2. Стабильно отвечает на простой запрос





→ Demo



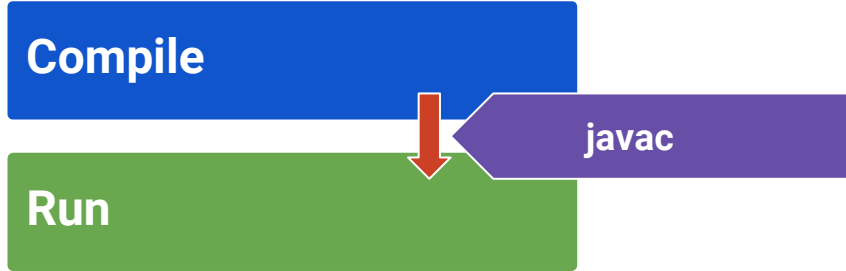
Добавим
немного Injection



Micronaut DI + AOP

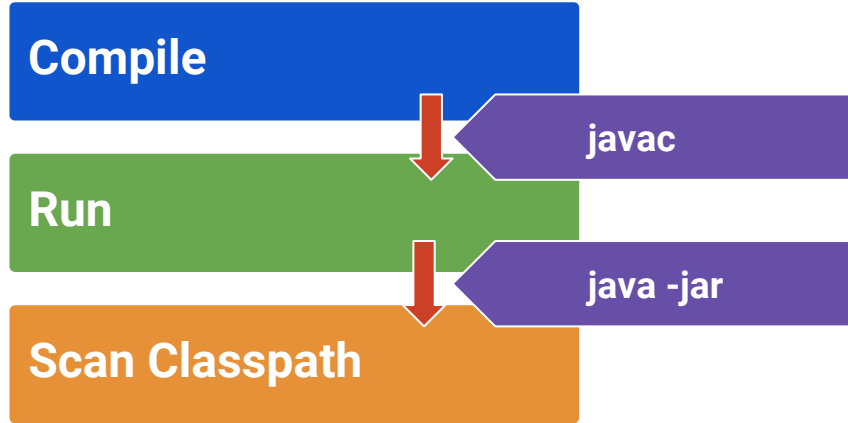
Compile

Spring DI + AOP



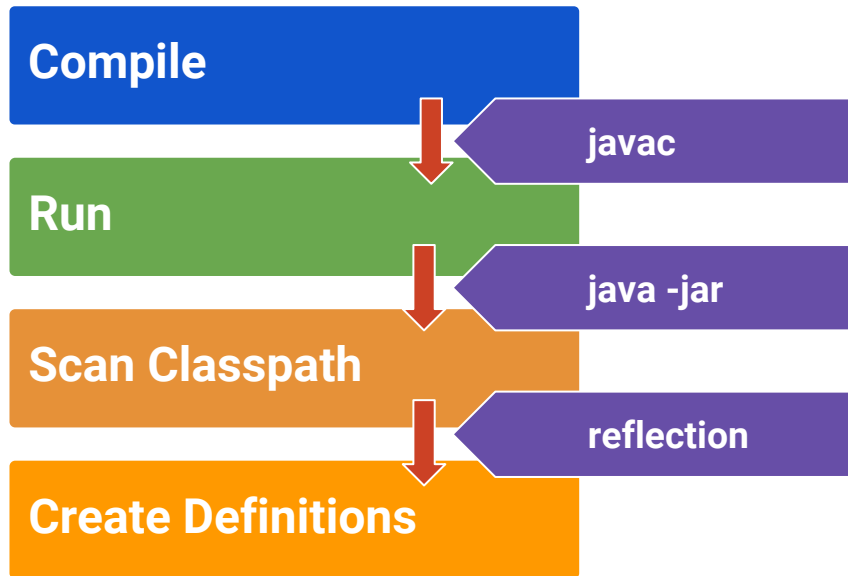
```
_$ javac -cp build/*.jar *.java
```

Spring DI + AOP



```
_ $ java -jar build/libs/app.jar \  
      --server.port=8081
```

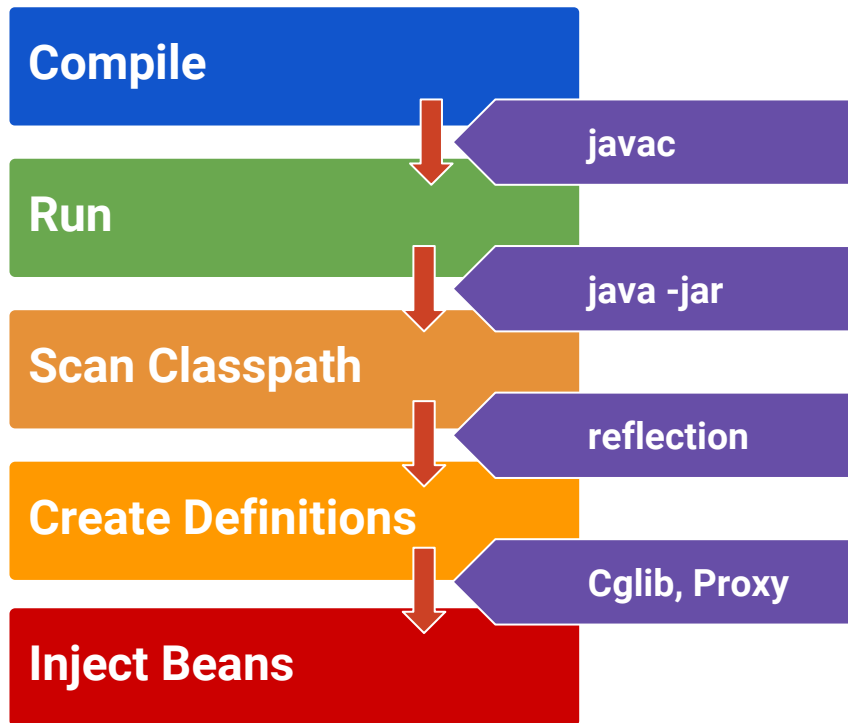
Spring DI + AOP



```
_$ java -jar build/libs/app.jar \
  --server.port=8081

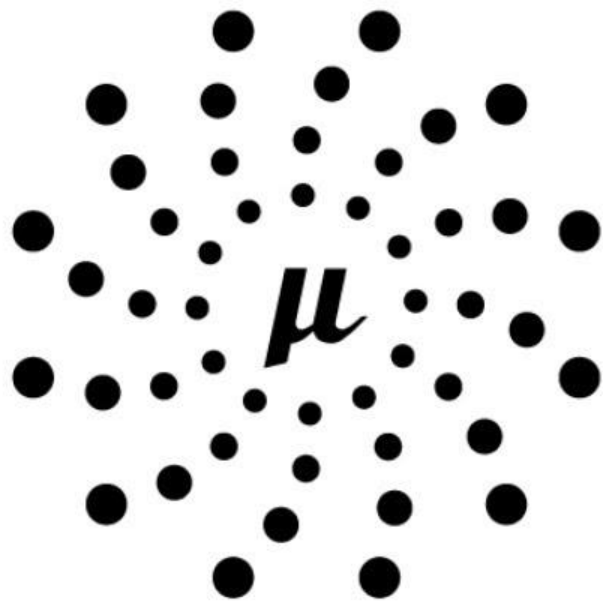
INFO 28687 main: Refreshing...
...
INFO 28687 nio-1: Started HttpServer on :8081
...
```

Spring DI + AOP



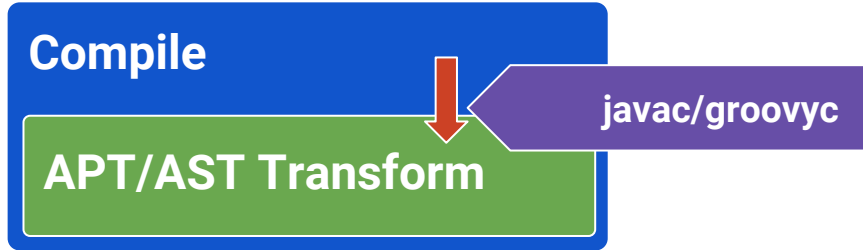
```
_ $ java -jar build/libs/app.jar \
  --server.port=8081

INFO 28687 main: Refreshing...
...
INFO 28687 main: BeanFactory id=...
INFO 28687 main:
f.a.AutowiredAnnotationBeanPostProcessor :
JSR-330 'javax.inject.Inject' annotation
found and supported for autowiring
...
INFO 28687 nio-1: Started HttpServer on :8081
```

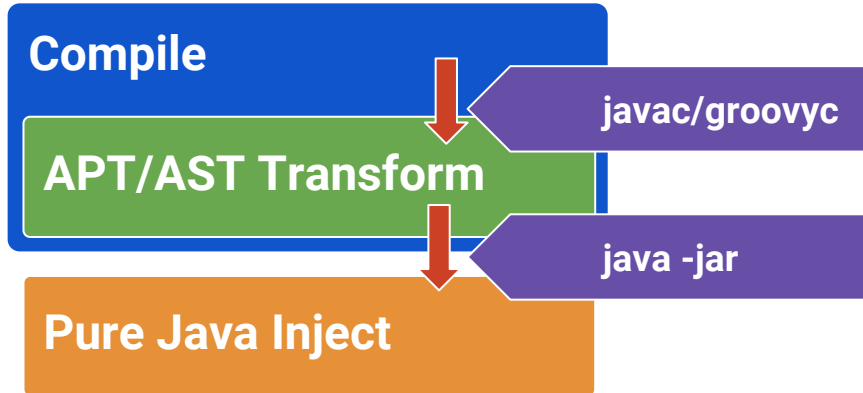



M I C R O N A U T

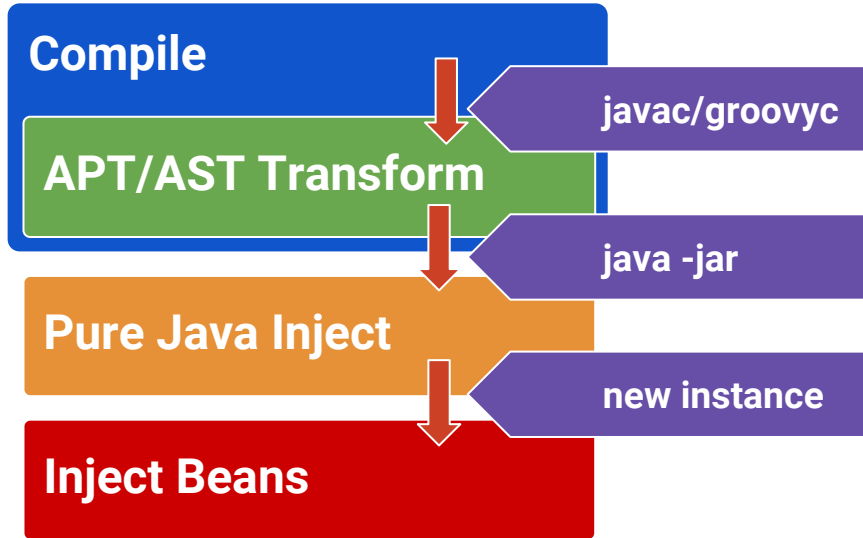
Micronaut DI + AOP



Micronaut DI + AOP



Micronaut DI + AOP



Что там генерит Micronaut?

```
m.b.a.controller.$BeerControllerDefinition$$exec1
m.b.a.controller.$BeerControllerDefinitionClass
m.b.a.service.$BarServiceDefinitionClass$$AnnotationMetadata
m.b.a.Application
m.b.a.service.$BarServiceDefinition
m.b.a.controller.BeerController
m.b.a.entity.Order
m.b.a.controller.$BeerControllerDefinition
m.b.a.controller.$BeerControllerDefinitionClass$$AnnotationMetadata
m.b.a.service.$BarServiceDefinitionClass
m.b.a.service.BarService
m.b.a.controller.$BeerControllerDefinition$$exec1$$AnnotationMetadata
```

Как работает статический Injection

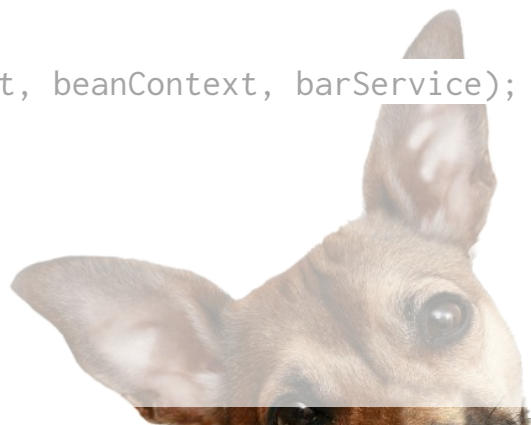


Как работает статический Injection

```
public class $BarServiceDefinition
    extends AbstractBeanDefinition<BarService>
    implements BeanFactory<BarService> {
    @Override
    public BarService build(BeanResolutionContext beanResolutionContext,
        BeanContext beanContext,
        BeanDefinition<BarService> beanDefinition) {

        BarService barService = new BarService();
        barService =
            (BarService)this.injectBean(beanResolutionContext, beanContext, barService);

        return barService;
    }
    ...
}
```



Как работает статический Injection

```
public class $BeerControllerDefinition
    extends AbstractBeanDefinition<BeerController>
    implements BeanFactory<BeerController> {

    @Override
    public BeerController build(BeanResolutionContext context,
                               BeanContext beanContext,
                               BeanDefinition<BeerController> beanDefinition) {
        BeerController beerController = new BeerController(
            (BarService)super.getBeanForConstructorArgument(context, beanContext, 0)
        );
        beerController = (BeerController)this.injectBean(beanResolutionContext,
                                                         beanContext,
                                                         beerController);

        return beerController;
    }
    ...
}
```



Spring Boot
следит за тенденциями



Spring Fu

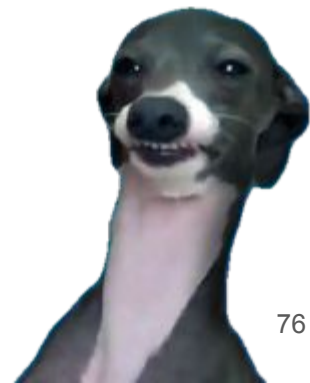
Kotlin Fu



→ KoFu

Кратко о работе с очередями

Пока удобнее
– **Spring Boot**



Java FU



→ JaFu

Spark





→ Demo

Шаг 1 - итоги



Табличка с резултатами

название →	-Xmx/mb	время старта/ms	время старта/ms
условие →		-Xmx=enough	-Xmx=min
mn	6	950	1113
mngroovy	7	950	1393
boot	11	2200	7093
spark	3	350	348
kofu	6	1500	2500
jafu	5	1041	5600

Победители в номинациях



Победители в номинациях



Основной победитель
в поединке – **Micronaut**



Победители в номинациях



Основной победитель
в поединке – **Micronaut**

Я не толстый
– **Spring Boot**



Победители в номинациях



Основной победитель
в поединке – **Micronaut**

Я не толстый
– **Spring Boot**



Шустрый мелкий
Старичок – **Spark**



Победители в номинациях



Основной победитель
в поединке – **Micronaut**

Я не толстый
– **Spring Boot**



Тёмная лошадка
– **Spring Fu**



Шустрый мелкий
Старичок – **Spark**

Это вообще
нормально
работает?





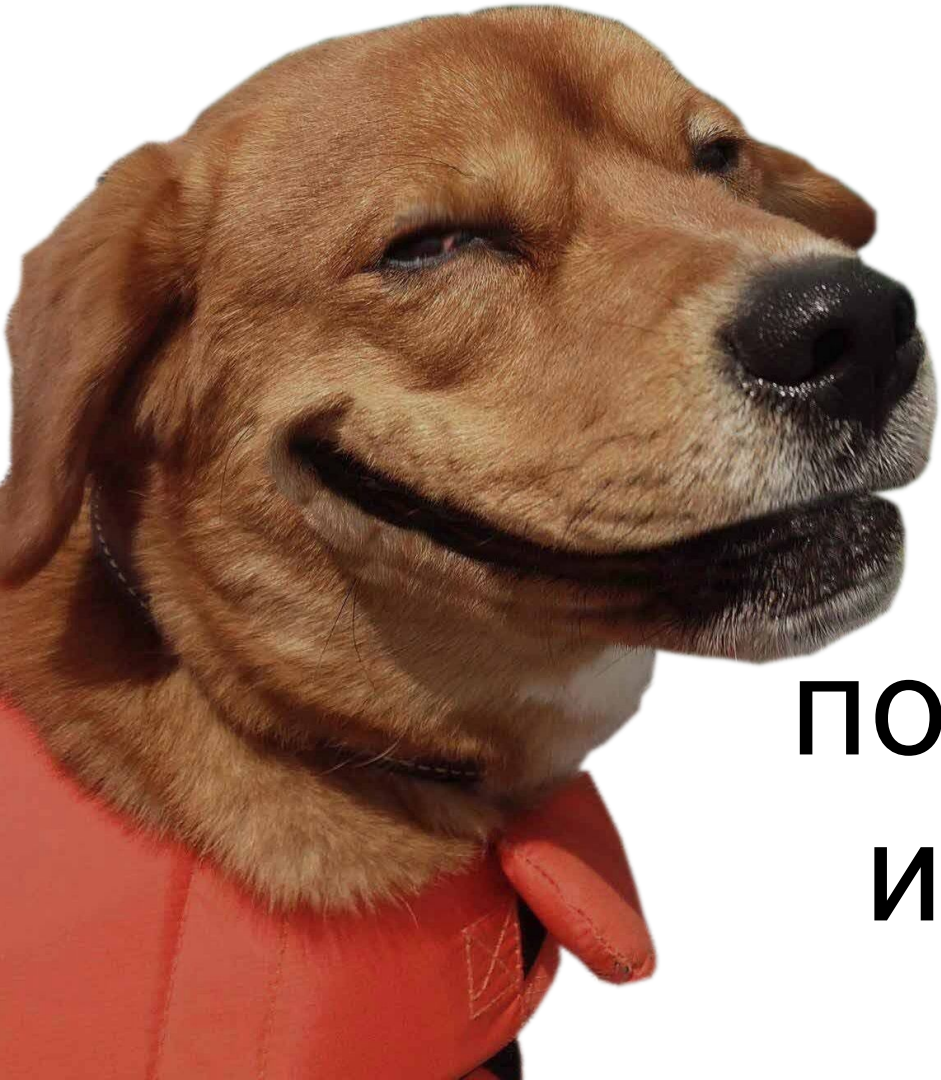
→ Demo

Табличка с резултатами

название →	-Xmx/mb	время старта/ms	время старта/ms
условие →		-Xmx=enough	-Xmx=min
mn	6	950	1113
mngroovy	7	950	1393
boot	11	2200	7093
spark	3	350	348
kofu	6	1500	2500
jafu	5	1041	5600

Погрешность
измерения?





Какая
погрешность
измерения?



GraalVM - в погоне за скоростью

- Собираем долго

GraalVM - в погоне за скоростью

- Собираем долго
- Запускаем быстро



GraalVM - в погоне за скоростью

- Собираем долго
- Запускаем быстро
- Работает – непонятно как



А в чем проблема-то?



В погоне за GraalVM - сборка

```
$ native-image --no-server \  
  -H:Name=target/app -Dio.netty.noUnsafe=true \  
  -H:+ReportUnsupportedElementsAtRuntime \  
  -cp "::$(echo target/BOOT-INF/lib/*.jar | tr ' ' ':')"\  
    :target/BOOT-INF/classes:target \  
  com.example.jafu.beer.api.Application
```



В погоне за GraalVM - сборка

maven/gradle dependencies

```
$ native-image --no-server \  
-H:Name=target/app -Dio.netty.noUnsafe=true \  
-H:+ReportUnsupportedElementsAtRuntime \  
-cp '.:$(echo target/BOOT-INF/lib/*.jar | tr ' ' ':')"\ \  
:target/BOOT-INF/classes:target \  
com.example.jafu.beer.api.Application
```



В погоне за GraalVM - сборка

```
$ native-image --no-server \  
-H:Name=target/app -Dio.netty.noUnsafe=true \  
-H:+ReportUnsupportedElementsAtRuntime \  
-cp "::$(echo target/BOOT-INF/lib/*.jar | tr ' ' ':')"\  
:target/BOOT-INF/classes:target \  
com.example.jafu.beer.api application
```

Скомпилированный src/



В погоне за GraalVM - сборка Reflection

```
java.lang.InstantiationException:
```

```
Type `o.a.l.l.m.ReusableMessageFactory`  
can not be instantiated reflectively  
as it does not have a no-parameter  
constructor or the no-parameter  
constructor has not been added  
explicitly to the native image
```



В погоне за GraalVM - сборка Reflection

```
java.lang.NoSuchMethodException:  
  c.e.j.b.a.c.BeerHandler.<init>():  
  ...
```



В погоне за GraalVM - сборка Reflection

```
1  { "name": "org.s.b.a...$.FactoryBean",
2    "allDeclaredConstructors": true },
3  { "name": "org.s.b.a.e.PropertiesPropertySourceLoader",
4    "allDeclaredConstructors": true },
5  { "name": "org.s.b.a.e.YamlPropertySourceLoader",
6    "allDeclaredConstructors": true },
7  { "name": "org.s.b.c.event.EventPublishingRun
8    "allDeclaredConstructors": true },
9
10 ...
11
12 { "name": "org.s.b.logging.java.JavaLoggingSys
13   "allDeclaredConstructors": true },
14 {
15   "name": "org.s.b.a.thymeleaf.ThymeleafTemp!
```



GraalVM Run - Micronaut Win



Сам генерит `reflect.json`

GraalVM Run - Micronaut Win



Сам генерит `reflect.json`

Пока все руками или из репозитория Dave Syer



GraalVM Run - Micronaut Win



Сам генерит `reflect.json`

Пока все руками или из репозитория Dave Syer





→ Demo

GraalVM Run - Micronaut Win!!!



1.m.r.Micronaut - Startup completed in 50 ms

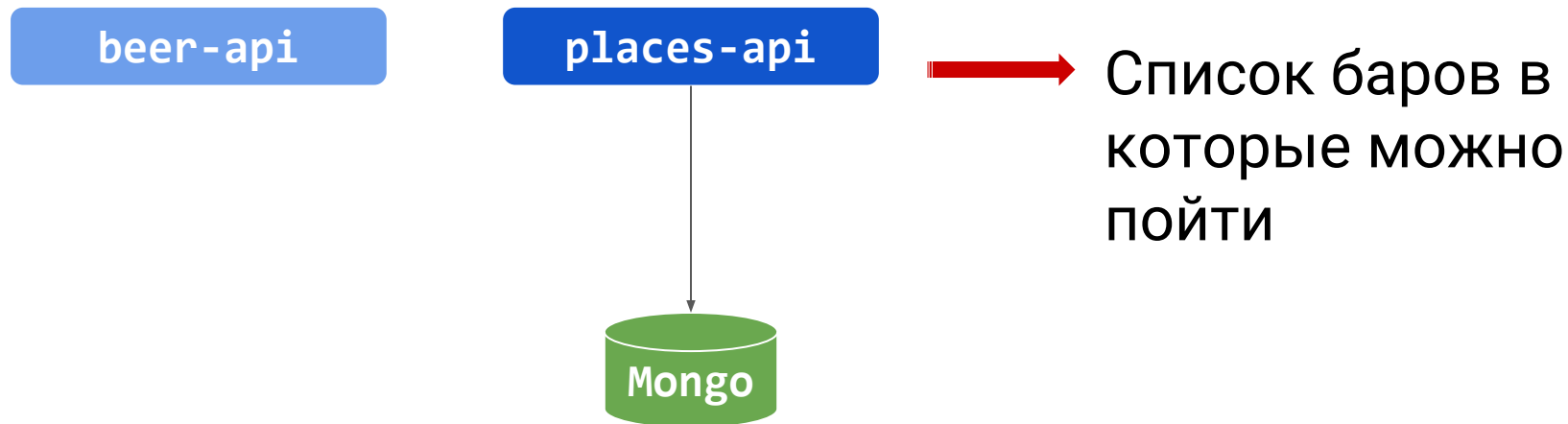
o.s.b.Spring - Started application in 120 ms





Шаг 2 - данные.

Развиваем стартап - зачатки архитектуры






→ Demo



Шаг 2 - итоги.

Табличка с резултатами

название →	<code>-Xmx</code> /mb	время старта/ms	время старта/ms
условие →	min	<code>-Xmx=enough</code>	<code>-Xmx=min</code>
mn	19	2400	6000
spring	15	3400	11000
spark	5	600	900

A white dog, possibly a Weimaraner, is sitting in the center of a cluttered room. The room is filled with various items, including cardboard boxes, a wooden frame, and a blue tarp. The dog is looking towards the camera with a happy expression. The text "Чем больше добавляем - тем меньше разницы" is overlaid on the image in a white box with black text.

Чем больше добавляем
– тем меньше разницы

Победители в номинациях



Уже не торт,
но еще вкусно
– **Micronaut**

Победители в номинациях



Уже не торт,
но еще вкусно
– **Micronaut**



Скрытая надежда
– **Spring**

Победители в номинациях



Уже не торт,
но еще вкусно
– **Micronaut**



Скрытая надежда
– **Spring**

построй счастье
своими руками
– **Spark**



A white dog, likely a Shiba Inu, is dressed as an astronaut in a grey space suit with a large, clear, spherical helmet. The dog is sitting at a control console in a spacecraft, with its paws resting on a keyboard. The console is filled with various controls, including joysticks, buttons, and screens. The background shows the interior of the spacecraft, with various panels and equipment. The dog's name tag reads "JSHIRATO".

Микросервисов в вакууме не бывает!

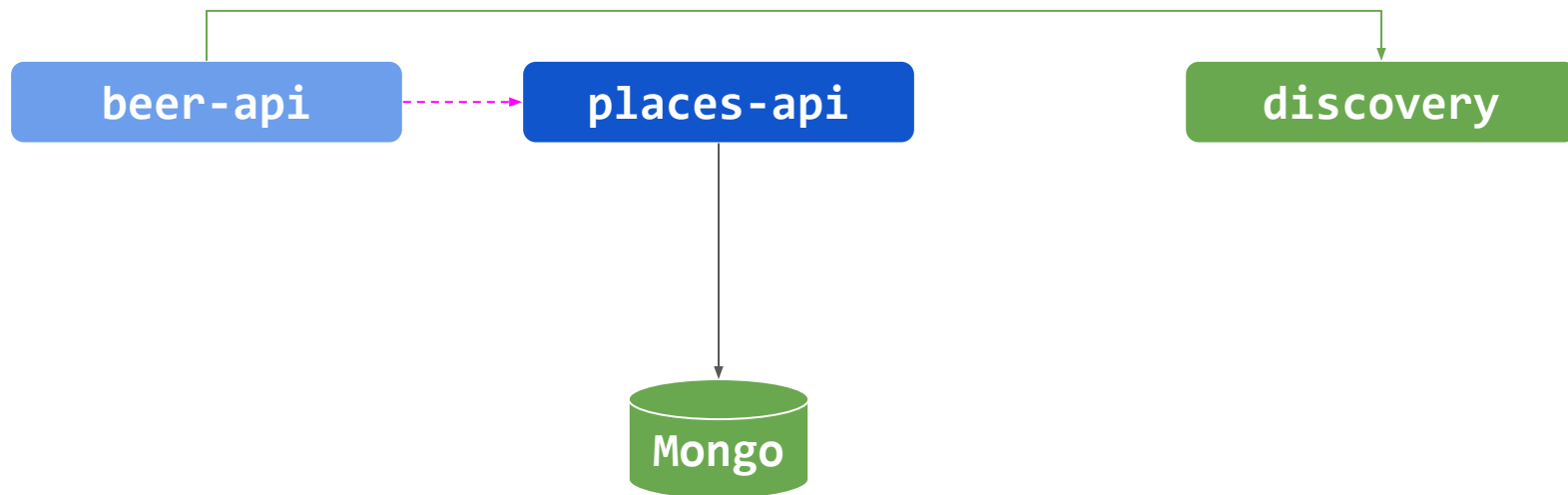


Шаг 3.

Идем в cloud

AGRAM

Картинка с архитектурой



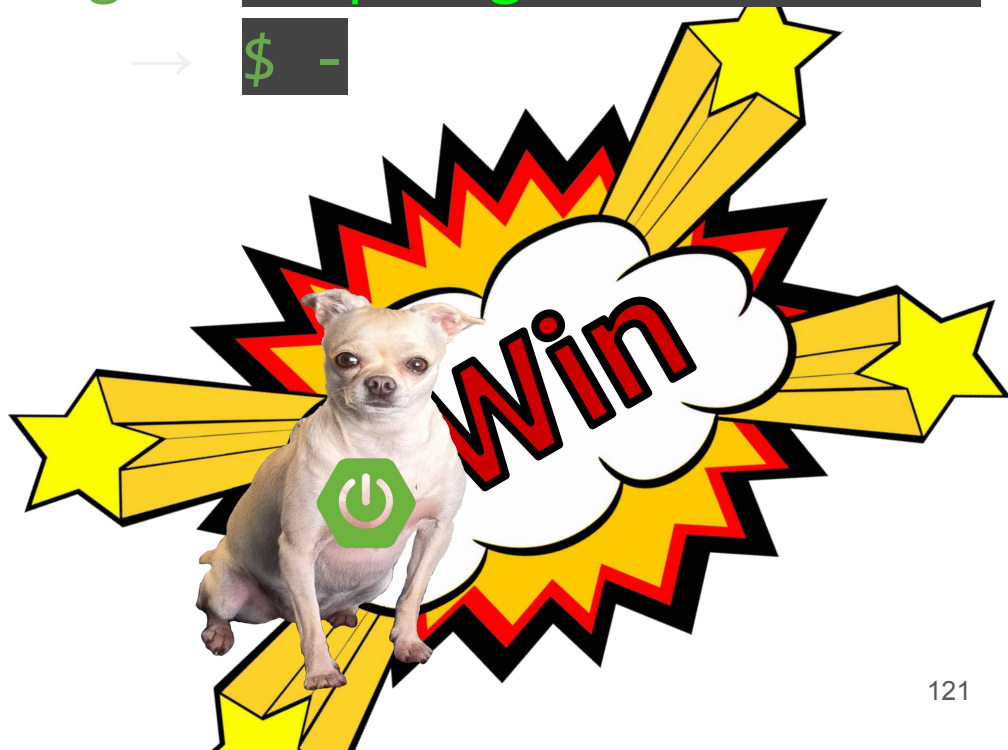


→ Demo

CLI – реванш

- spring cli
- micronaut cli

→ spring → `$ spring cloud eureka`
→ mn → `$ -`



Шаг 3 - итоги



Табличка с резултатами

название → условие →	-Xmx/mb	время старта -Xmx=enough/ms	время старта -Xmx=min/ms
mn	21	3000	7500
spring	22	5800	30000

Победители в номинациях



Уходит в отрыв
– **Micronaut**

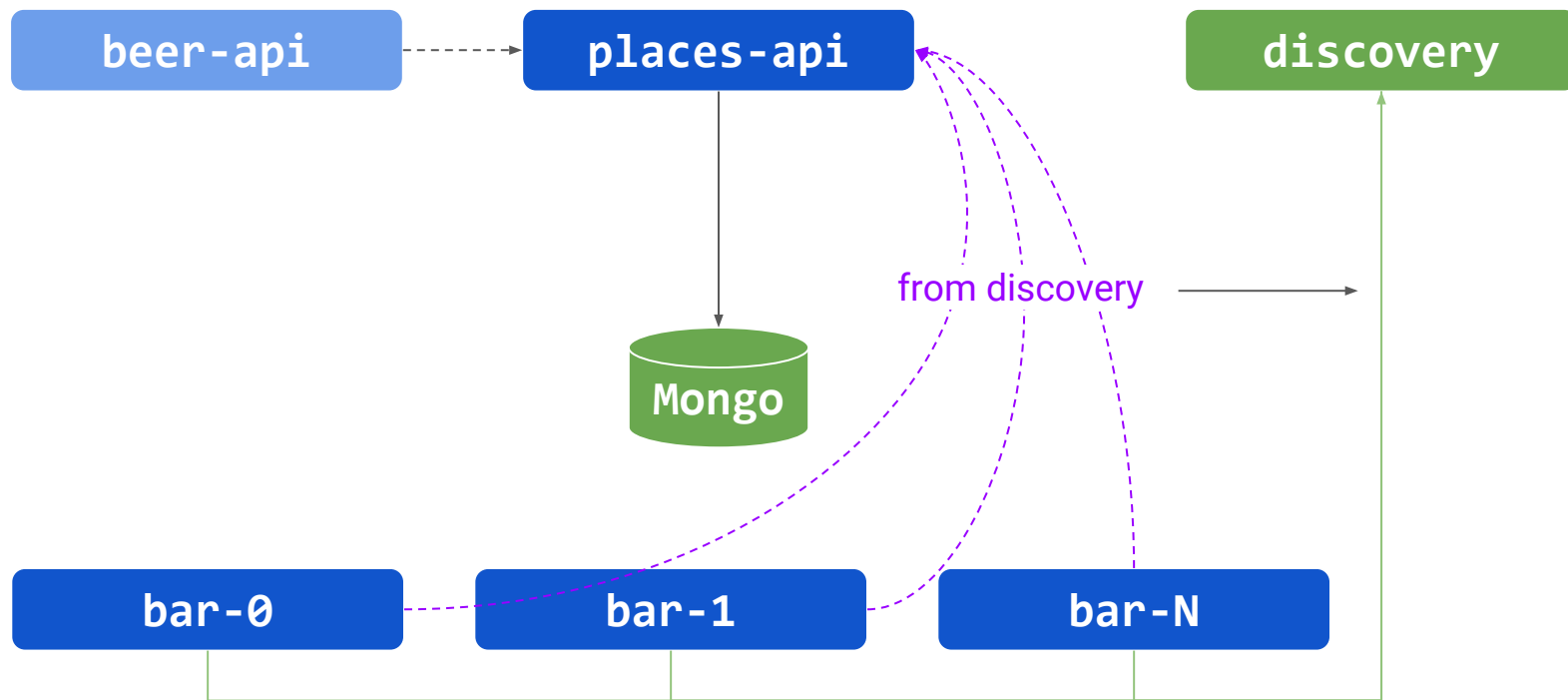
Разбитые надежды
– **Spring Boot**



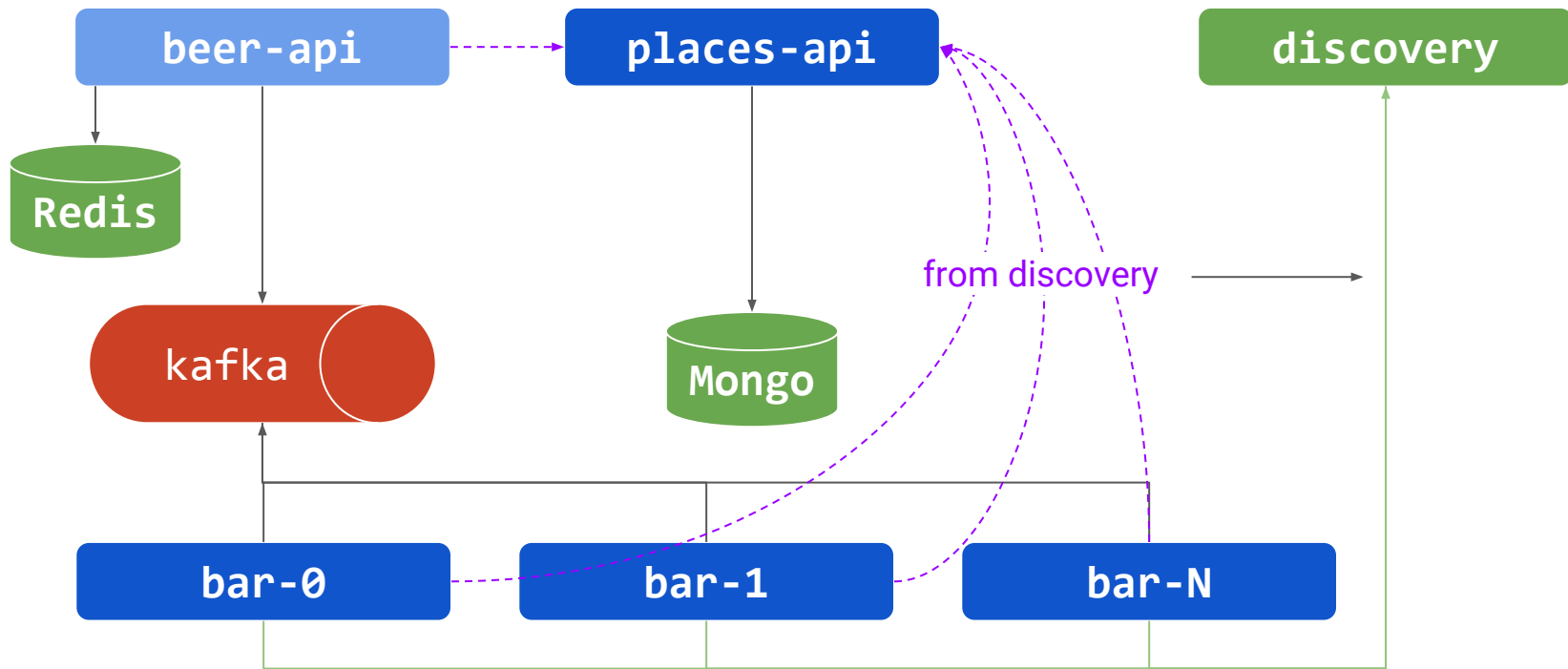


Шаг 4 - очереди/кэш

Картинка с архитектурой



В этой жизни надо попробовать все



В этой жизни надо попробовать все



Кратко о работе с Kafka и Redis

Кратко о работе с Kafka и Redis

**Только очень
кратко, пожалуйста.**

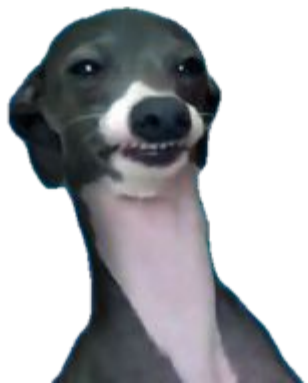


Кратко о работе с очередями

Пока удобнее
– **Spring Boot**



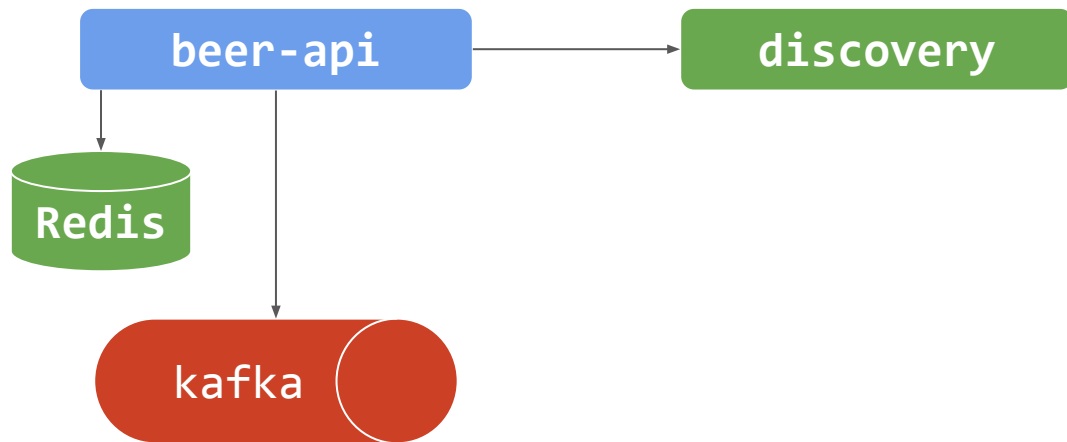
Погоди внедрять я доделаю
– **Micronaut**



Шаг 4 - итоги



Что запускаем для замера?



Табличка с результатами

название → **-Xmx**/mb

время старта

время старта

условие →

-Xmx=enough/ms

-Xmx=min/ms

mn_cloud_db

21

3000

7500

mn_cloud_mq_cache

15

3000

3000

spring_cloud_mq_cache

24

8000

64000

Табличка с результатами

название → условие →	-Xmx/mb	время старта -Xmx=enough/ms	время старта -Xmx=min/ms
mn	15	3000	3000
spring	24	8000	64000

**Надо было с
этого начинать**



Выводы

1. `MicroFramework` = 10^{-6} × `Framework`

Выводы

1. `MicroFramework` = 10^{-6} × `Framework`
2. **Область применимости узкая**



Выводы

1. `MicroFramework` = 10^{-6} × `Framework`
2. Область применимости узкая
3. **Разные внутри - одинаковые снаружи**



Выводы

1. `MicroFramework = 10-6 × Framework`
2. Область применимости узкая
3. **Разные внутри - одинаковые снаружи**
→ Лучшее взято из Spring



Интересные ресурсы

[Demo Source](#)

[Spring Blog] [The Evolution Of Spring Fu](#)

[Spring Fu Sample Sources](#)

[dsyer] [Spring Init Experiments source](#)

[GraalVM Blog] [Understanding Class Initialization in GraalVM Native Image Generation](#)

[GraalVM Blog] [Netty Native Image Generation](#)

[Micronaut docs](#)

Кто же пошел в бар и в какой

```
{  
  "type": "American Pale Ale",  
  "bottles": 1603,  
  "totalPrice": 223962,  
  "place": {  
    "name": "Relaxed-Beagle",  
    "averagePrice": 978,  
    "maxTableSize": 33  
  },  
  "companions": [  
    "ivan",  
    "pavel_ch",  
    "kolemik",  
    "sberkek",  
    "yegor256",  
    "vasya",  
    "NPE",  
    "baruh",  
    "alexanzer",  
    "ctap",  
    "alexander.kochin",  
    "andrew-q1",  
    "Vitoss",  
    "her_s_gory",  
    "andrewserdyuk",  
    "barnaulaltayka",  
    "=)",  
    "shurmeleva",  
    "mgorelikov",  
    "tolkv"  
  ]  
}
```



QA



@tolkv



@lavcraft



Кирилл



Максим

@gorelikoff



@gorelikov



QA



@tolkv



@lavcraft



Кирилл



Максим

@gorelikoff



@gorelikov

