

**JFOK**  
MAY THE SOURCE BE

# The ultimate microframework smackdown

---

Jaap Coomans



@JaapCoomans

Who is that guy?



# Microframeworks

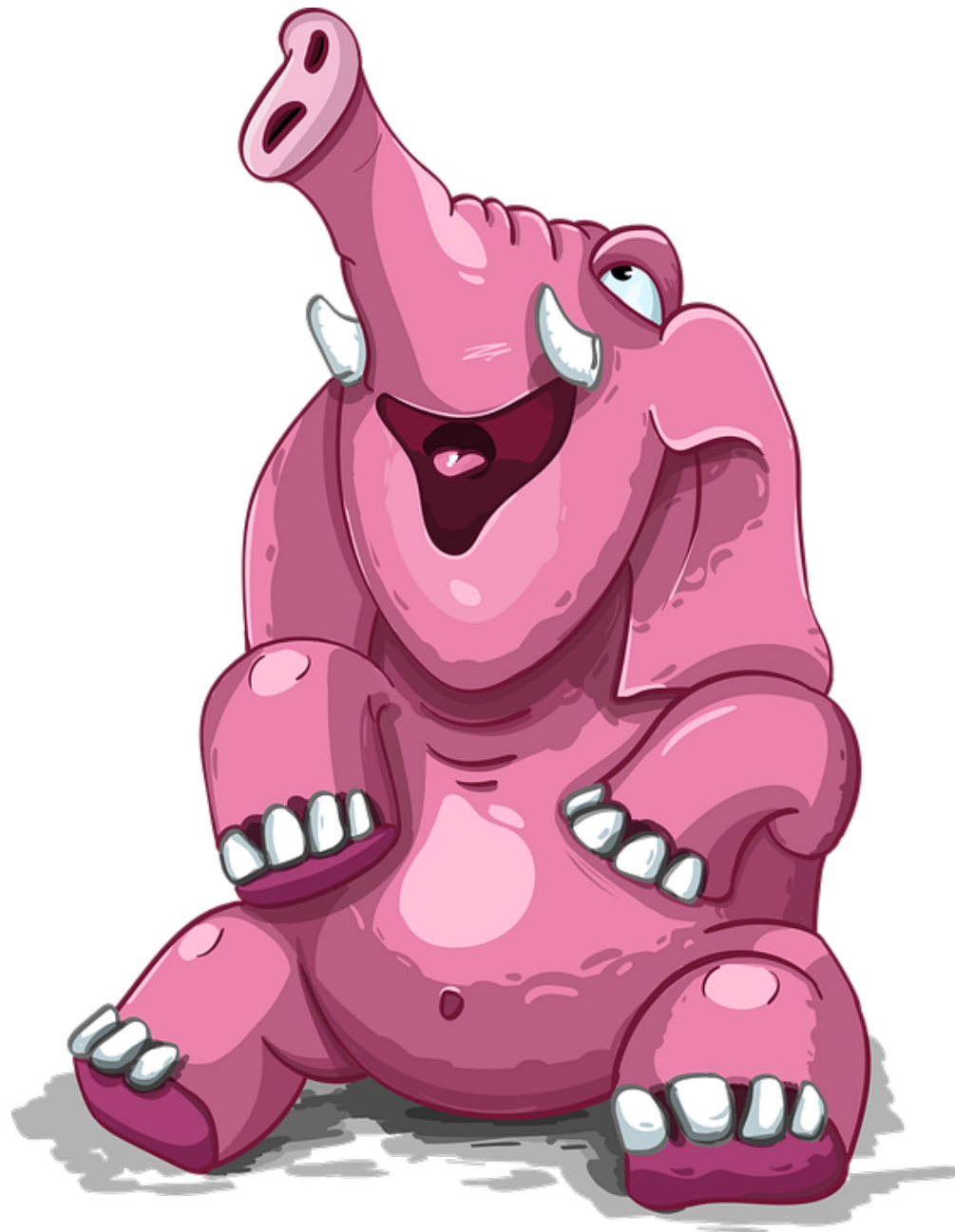
Microservices framework?



Microservices framework

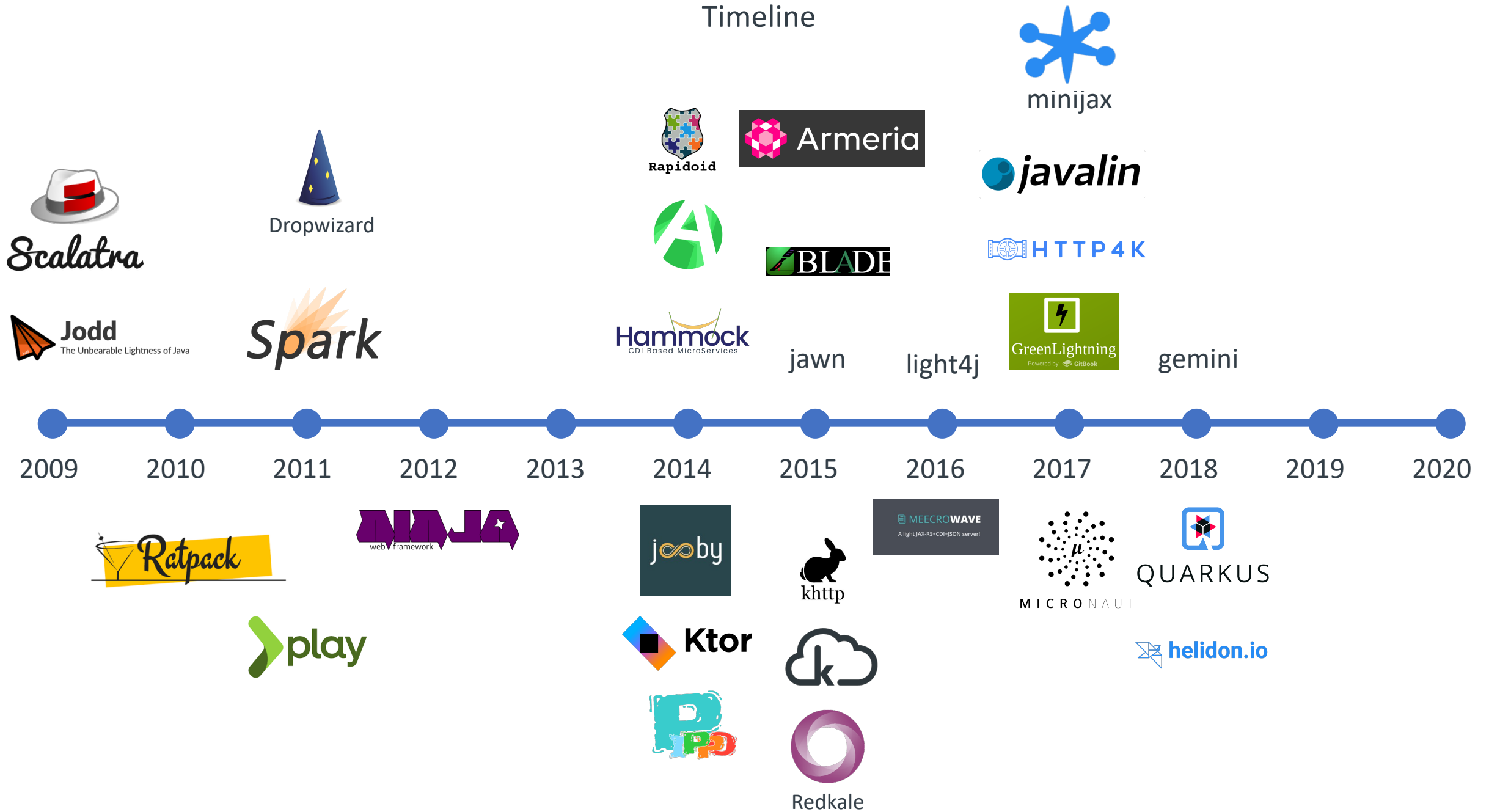
What is a  
microframework?

“A **microframework** is a term used to refer to minimalistic web application frameworks. It is contrasted with full-stack frameworks.”





# Timeline





## The long list

- |                   |                     |
|-------------------|---------------------|
| 1. Act Framework  | 17. Ktor            |
| 2. Armeria        | 18. KumuluzEE       |
| 3. Blade          | 19. Light-4j        |
| 4. Dropwizard     | 20. Meecrowave      |
| 5. Gemini         | 21. Micronaut       |
| 6. GreenLightning | 22. Minijax         |
| 7. Hammock        | 23. Ninja Framework |
| 8. Helidon SE     | 24. Pippo           |
| 9. Helidon MP     | 25. Play Framework  |
| 10. http4k        | 26. Proteus         |
| 11. Javalin       | 27. Quarkus         |
| 12. Jawn          | 28. Rapidoid        |
| 13. jlhttp        | 29. Ratpack         |
| 14. Jodd          | 30. Redkale         |
| 15. Jooby         | 31. Scalatra        |
| 16. khttp         | 32. Spark           |

## The long list

- |   |                     |
|---|---------------------|
| 1. Act Framework                            | 17. Ktor            |
| 2. Armeria                                  | 18. KumuluzEE       |
| 3. Blade                                    | 19. Light-4j        |
| 4. Dropwizard                               | 20. Meecrowave      |
| 5. Gemini                                   | 21. Micronaut       |
| 6. GreenLightning                           | 22. Minijax         |
| 7. Hammock                                  | 23. Ninja Framework |
| 8. Helidon SE                               | 24. Pippo           |
| 9. Helidon MP                               | 25. Play Framework  |
| 10. http4k                                  | 26. Proteus         |
| 11. Javalin                                 | 27. Quarkus         |
| 12. Jawn                                    | 28. Rapidoid        |
| 13. jhttp   GPL and custom hosting platform | 29. Ratpack         |
| 14. Jodd                                    | 30. Redkale         |
| 15. Jooby                                   | 31. Scalatra        |
| 16. khttp                                   | 32. Spark           |

## The long list

- |  |                     |
|--|---------------------|
| 1. Act Framework                                       | 17. Ktor            |
| 2. Armeria   | 18. KumuluzEE       |
| 3. Blade   | 19. Light-4j        |
| 4. Dropwizard  | 20. Meecrowave      |
| 5. Gemini  | 21. Micronaut       |
| 6. GreenLightning                                      | 22. Minijax         |
| 7. Hammock   | 23. Ninja Framework |
| 8. Helidon SE  | 24. Pippo           |
| 9. Helidon MP  | 25. Play Framework  |
| 10. http4k   | 26. Proteus         |
| 11. Javalin  | 27. Quarkus         |
| 12. Jawn   | 28. Rapidoid        |
| <del>13. jhttp</del>   GPL and custom hosting platform | 29. Ratpack         |
| <del>14. Jodd</del>   "Set of micro-frameworks"        | 30. Redkale         |
| 15. Jooby  | 31. Scalatra        |
| 16. khttp  | 32. Spark           |

## The long list

- |  |                     |
|--|---------------------|
| 1. Act Framework                                       | 17. Ktor            |
| 2. Armeria   | 18. KumuluzEE       |
| 3. Blade   | 19. Light-4j        |
| 4. Dropwizard  | 20. Meecrowave      |
| 5. Gemini  | 21. Micronaut       |
| 6. GreenLightning                                      | 22. Minijax         |
| 7. Hammock   | 23. Ninja Framework |
| 8. Helidon SE  | 24. Pippo           |
| 9. Helidon MP  | 25. Play Framework  |
| 10. http4k   | 26. Proteus         |
| 11. Javalin  | 27. Quarkus         |
| 12. Jawn   | 28. Rapidoid        |
| <del>13. jhttp</del>   GPL and custom hosting platform | 29. Ratpack         |
| <del>14. Jodd</del>   "Set of micro-frameworks"        | 30. Redkale         |
| 15. Jooby  | 31. Scalatra        |
| <del>16. khttp</del>   Dead project since 2017         | 32. Spark           |

## The long list

- |   |   |
|---|---|
| 1. Act Framework  | 17. Ktor  |
| 2. Armeria  | 18. KumuluzEE   |
| 3. Blade  | 19. Light-4j  |
| 4. Dropwizard   | 20. Meecrowave  |
| 5. Gemini   | 21. Micronaut   |
| 6. GreenLightning                                       | 22. Minijax   |
| 7. Hammock  | <del>23. Ninja Framework</del>   Full-stack (self-proclaimed) |
| 8. Helidon SE   | 24. Pippo   |
| 9. Helidon MP   | 25. Play Framework  |
| 10. http4k  | 26. Proteus   |
| 11. Javalin   | 27. Quarkus   |
| 12. Jawn  | 28. Rapidoid  |
| <del>13. jlhttp</del>   GPL and custom hosting platform | 29. Ratpack   |
| <del>14. Jodd</del>   "Set of micro-frameworks"         | 30. Redkale   |
| 15. Jooby   | 31. Scalatra  |
| <del>16. khttp</del>   Dead project since 2017          | 32. Spark   |

## The long list

- |   |   |
|---|---|
| 1. Act Framework  | 17. Ktor  |
| 2. Armeria  | 18. KumuluzEE   |
| 3. Blade  | 19. Light-4j  |
| 4. Dropwizard   | 20. Meecrowave  |
| 5. Gemini   | 21. Micronaut   |
| 6. GreenLightning                                       | 22. Minijax   |
| 7. Hammock  | <del>23. Ninja Framework</del>   Full-stack (self-proclaimed) |
| 8. Helidon SE   | 24. Pippo   |
| 9. Helidon MP   | <del>25. Play Framework</del>   Not very micro anymore        |
| 10. http4k  | 26. Proteus   |
| 11. Javalin   | 27. Quarkus   |
| 12. Jawn  | 28. Rapidoid  |
| <del>13. jlhttp</del>   GPL and custom hosting platform | 29. Ratpack   |
| <del>14. Jodd</del>   "Set of micro-frameworks"         | 30. Redkale   |
| 15. Jooby   | 31. Scalatra  |
| <del>16. khttp</del>   Dead project since 2017          | 32. Spark   |



## The long list

- |   |   |
|---|---|
| 1. Act Framework  | 17. Ktor  |
| 2. Armeria  | 18. KumuluzEE   |
| 3. Blade  | 19. Light-4j  |
| 4. Dropwizard   | 20. Meecrowave  |
| 5. Gemini   | 21. Micronaut   |
| 6. GreenLightning                                       | 22. Minijax   |
| 7. Hammock  | <del>23. Ninja Framework</del>   Full-stack (self-proclaimed) |
| 8. Helidon SE   | 24. Pippo   |
| 9. Helidon MP   | <del>25. Play Framework</del>   Not very micro anymore        |
| 10. http4k  | 26. Proteus   |
| 11. Javalin   | <del>27. Quarkus</del>   Only micro startup times...          |
| 12. Jawn  | 28. Rapidoid  |
| <del>13. jlhttp</del>   GPL and custom hosting platform | 29. Ratpack   |
| <del>14. Jodd</del>   "Set of micro-frameworks"         | 30. Redkale   |
| 15. Jooby   | 31. Scalatra  |
| <del>16. khttp</del>   Dead project since 2017          | 32. Spark   |

## The long list

1. Act Framework
2. Armeria
3. Blade
4. Dropwizard
5. Gemini
6. GreenLightning
7. Hammock
8. Helidon SE
9. Helidon MP
10. http4k
11. Javalin
12. Jawn
- ~~13. jhttp~~ | GPL and custom hosting platform
- ~~14. Jodd~~ | "Set of micro-frameworks"
15. Jooby
- ~~16. khttp~~ | Dead project since 2017
17. Ktor
18. KumuluzEE
19. Light-4j
20. Meecrowave
21. Micronaut
22. Minijax
- ~~23. Ninja Framework~~ | Full-stack (self-proclaimed)
24. Pippo
- ~~25. Play Framework~~ | Not very micro anymore
26. Proteus
- ~~27. Quarkus~~ | Only micro startup times...
- ~~28. Rapidoid~~ | Lack of docs for current version.
29. Ratpack
30. Redkale
31. Scalatra
32. Spark

## The long list

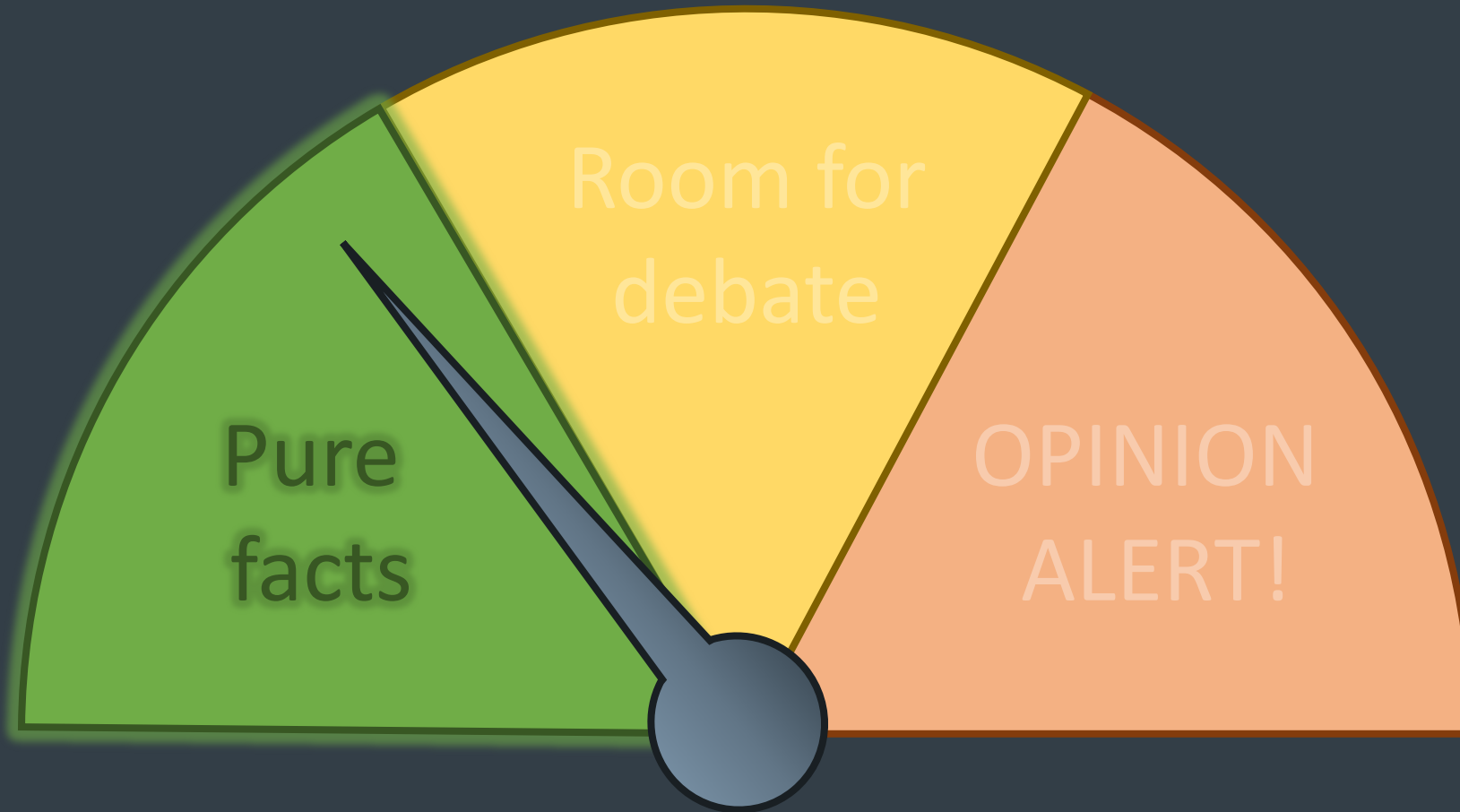
1. Act Framework
2. Armeria
3. Blade
4. Dropwizard
5. Gemini
6. GreenLightning
7. Hammock
8. Helidon SE
9. Helidon MP
10. http4k
11. Javalin
12. Jawn
- ~~13. jhttp~~ | GPL and custom hosting platform
- ~~14. Jodd~~ | “Set of micro-frameworks”
15. Jooby
- ~~16. khttp~~ | Dead project since 2017
17. Ktor
18. KumuluzEE
19. Light-4j
20. Meecrowave
21. Micronaut
22. Minijax
- ~~23. Ninja Framework~~ | Full-stack (self-proclaimed)
24. Pippo
- ~~25. Play Framework~~ | Not very micro anymore
26. Proteus
- ~~27. Quarkus~~ | Only micro startup times...
- ~~28. Rapidoid~~ | Lack of docs for current version.
29. Ratpack
- ~~30. Redkale~~ | Docs only in Chinese
31. Scalatra
32. Spark

Why is framework X  
not on your slides?



Let's sort it out

# Opinion-o-meter





## Licenses

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

Licenses

Apache 2.0

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

Licenses

Apache 2.0	BSD 2/3-clause				
Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

## Licenses

Apache 2.0

BSD 2/3-clause

MIT

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus

## Licenses

Apache 2.0

BSD 2/3-clause

MIT

LGPL 3.0

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus

## Programming Languages

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	



# Programming Languages

Java

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

# Programming Languages

Java

Kotlin

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus

# Programming Languages

Java

Kotlin

Groovy

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus

# Programming Languages

Java		Kotlin		Groovy		Scala	
Act Framework	Gemini	http4k		Ktor		Micronaut	Ratpack
Armeria	GreenLightning	Javalin		KumuluzEE	Node	Go	Minijax
Blade	Hammock	Jawn		Light-4j		Pippo	Spark Java
Dropwizard	Helidon	Jooby		Meecrowave		Proteus	

## Programming Model

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

# Programming Model

## Microprofile

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

# Programming Model

Microprofile

Programmatic

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus

# Programming Model

Microprofile

Programmatic

Annotations

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus



# Programming Model

Microprofile

Programmatic

Annotations

JAX-RS

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meerowave

Proteus

# Programming Model

Microprofile	Programmatic		Annotations		JAX-RS	Code generation	
Act Framework	Gemini		http4k		Ktor	Micronaut	Ratpack
Armeria	GreenLightning		Javalin		KumuluzEE	Minijax	Scalatra
Blade	Hammock		Jawn		Light-4j	Pippo	Spark Java
Dropwizard	Helidon		Jooby		Meerowave	Proteus	

## Organization

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

## Organization

Company

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

## Organization

Company

Non-profit org

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

## Organization

Company

Non-profit org

Community

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meecrowave

Proteus

## Web Engine

Act Framework	Gemini	http4k	Ktor	Micronaut	Ratpack
Armeria	GreenLightning	Javalin	KumuluzEE	Minijax	Scalatra
Blade	Hammock	Jawn	Light-4j	Pippo	Spark Java
Dropwizard	Helidon	Jooby	Meecrowave	Proteus	

## Web Engine

Jetty

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meecrowave

Proteus



# Web Engine

Jetty

Tomcat

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meecrowave

Proteus

# Web Engine

Jetty

Tomcat

Undertow

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

Helidon

Jooby

Meecrowave

Proteus

# Web Engine

Jetty

Tomcat

Undertow

Netty

Act Framework

Gemini

http4k

Ktor

Micronaut

Ratpack

Armeria

GreenLightning

Javalin

KumuluzEE

Minijax

Scalatra

Blade

Hammock

Jawn

Light-4j

Pippo

Spark Java

Dropwizard

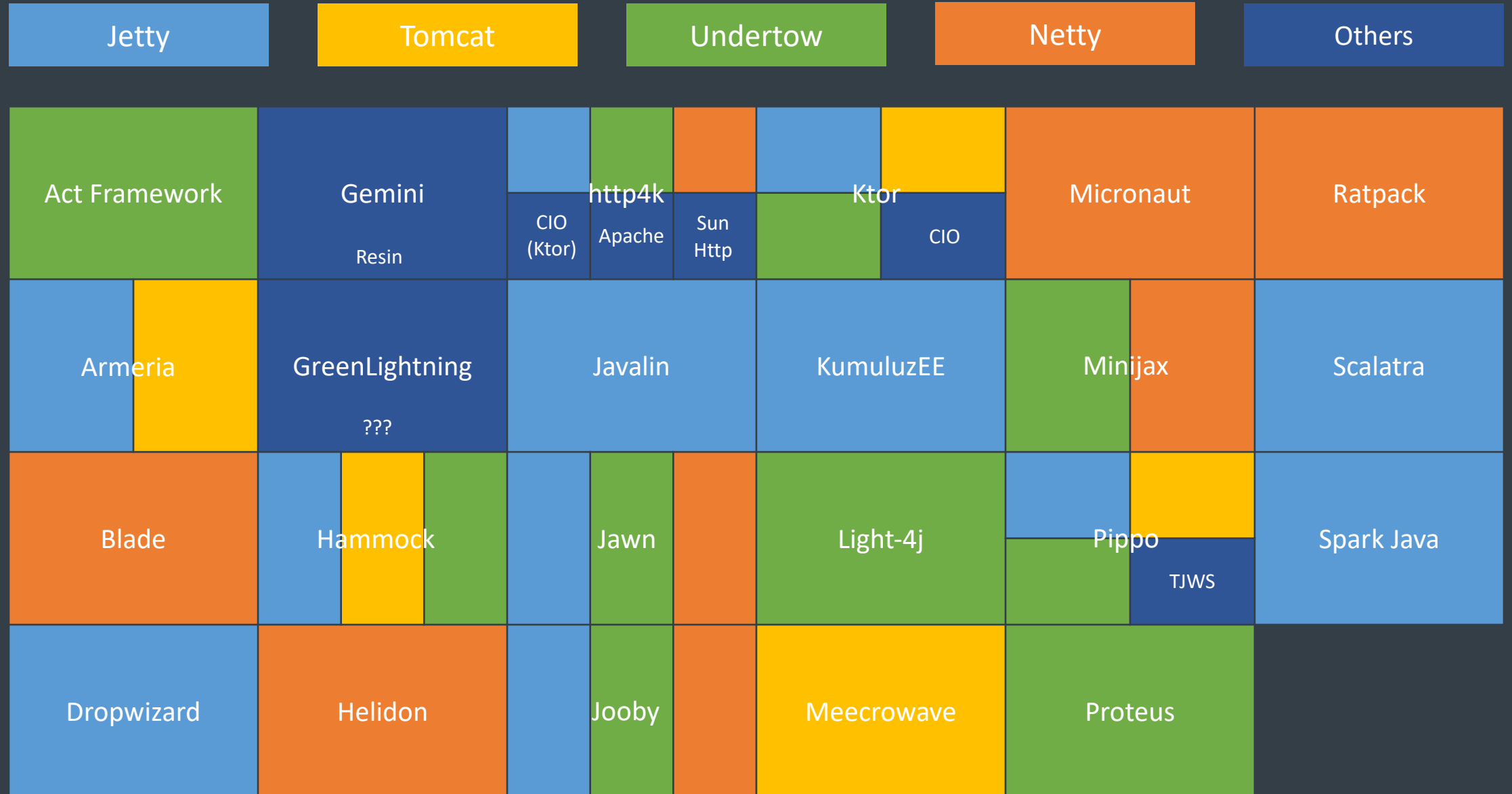
Helidon

Jooby

Meecrowave

Proteus

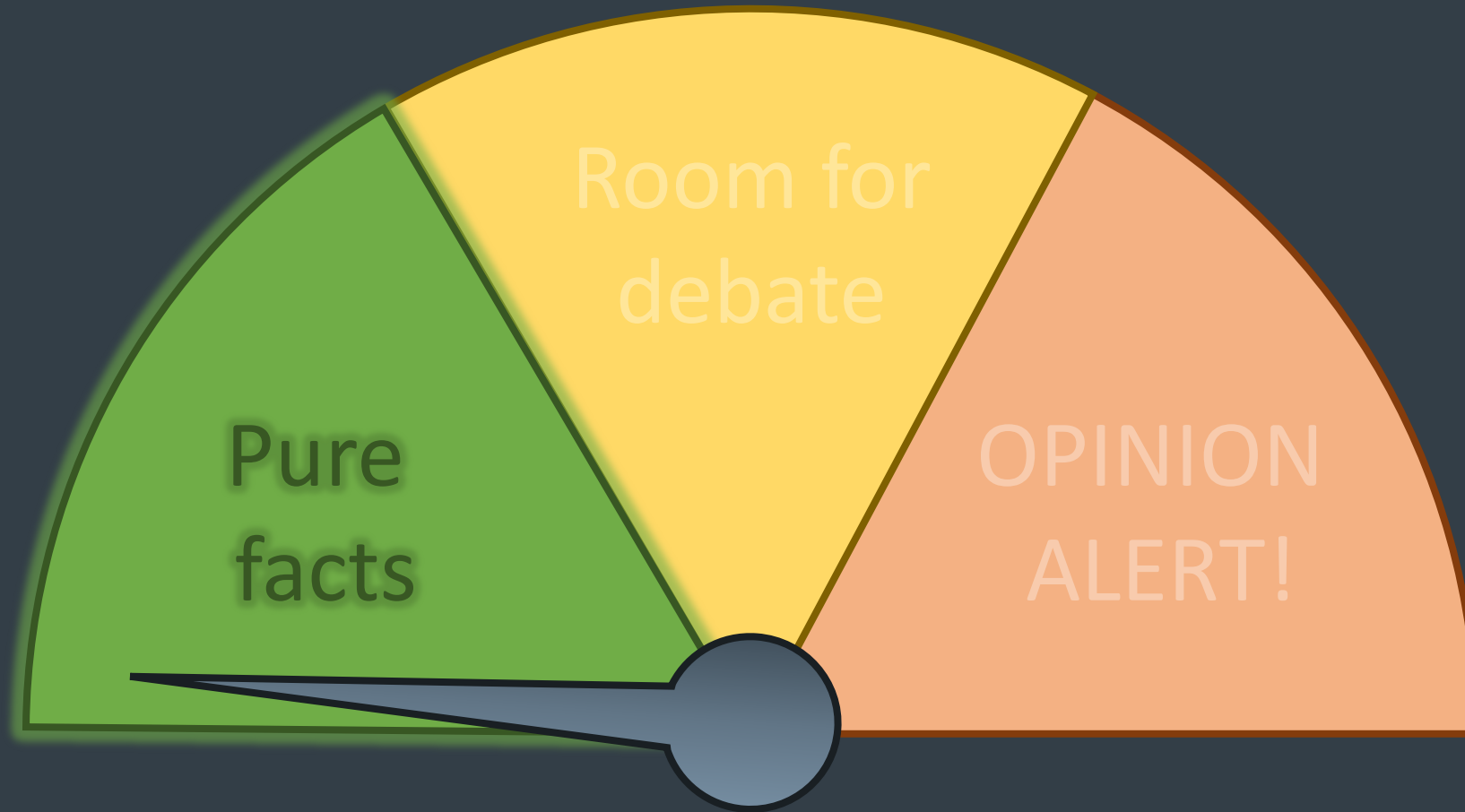
# Web Engine



# To measure is to know.



# Opinion-o-meter





Google Trends







# Number of active contributors\*

\* At least 20 commits and 1000 lines of code in the last 2 years

## Metrics

Number of active contributors\*

\* At least 20 commits and 1000 lines of code in the last 2 years

**Stars on github**

## Metrics

Number of active contributors\*

\* At least 20 commits and 1000 lines of code in the last 2 years

Stars on github

**Questions on Stack Overflow**

## Metrics

Number of active contributors\*

\* At least 20 commits and 1000 lines of code in the last 2 years

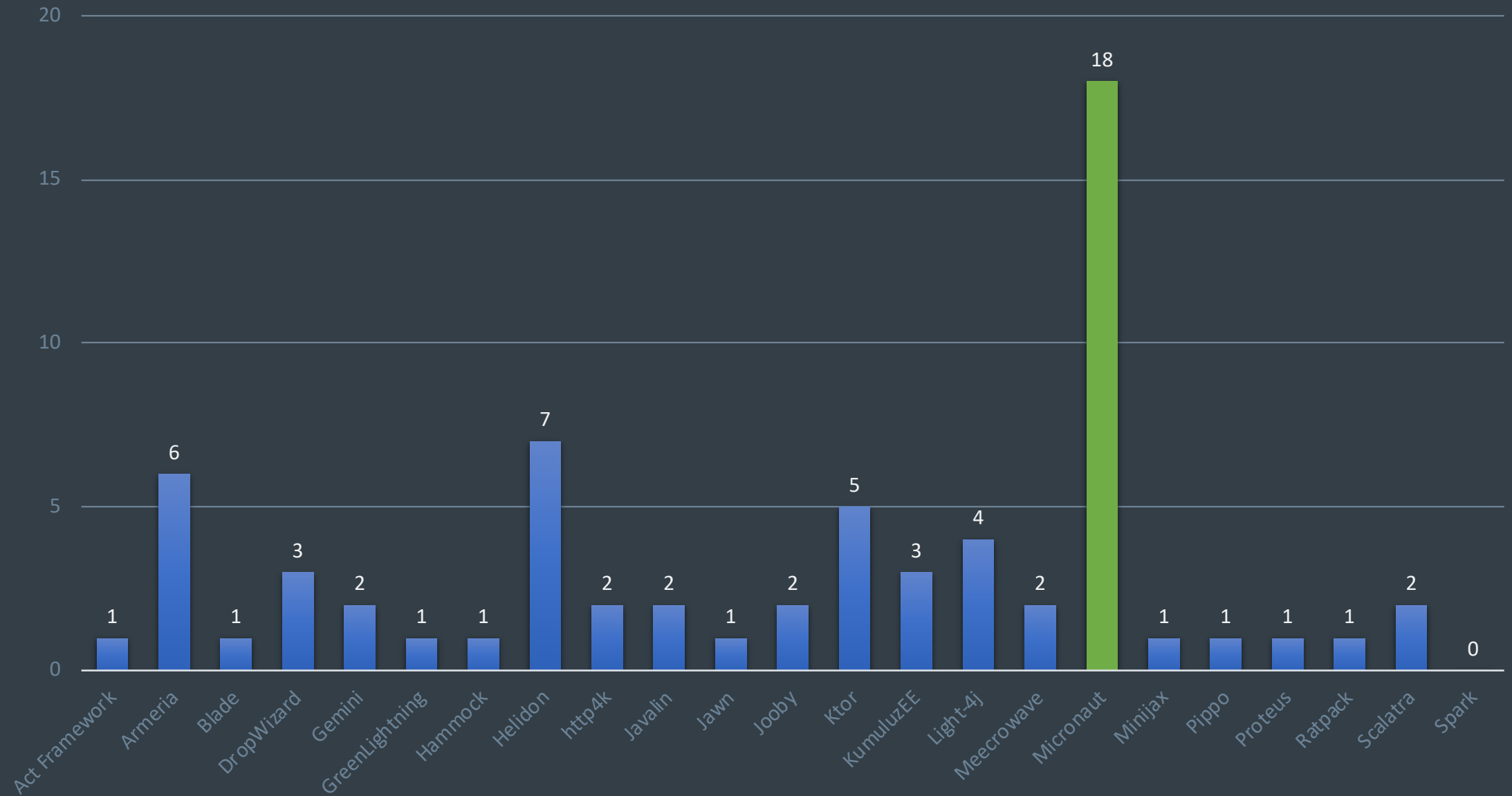
Stars on github

Questions on Stack Overflow

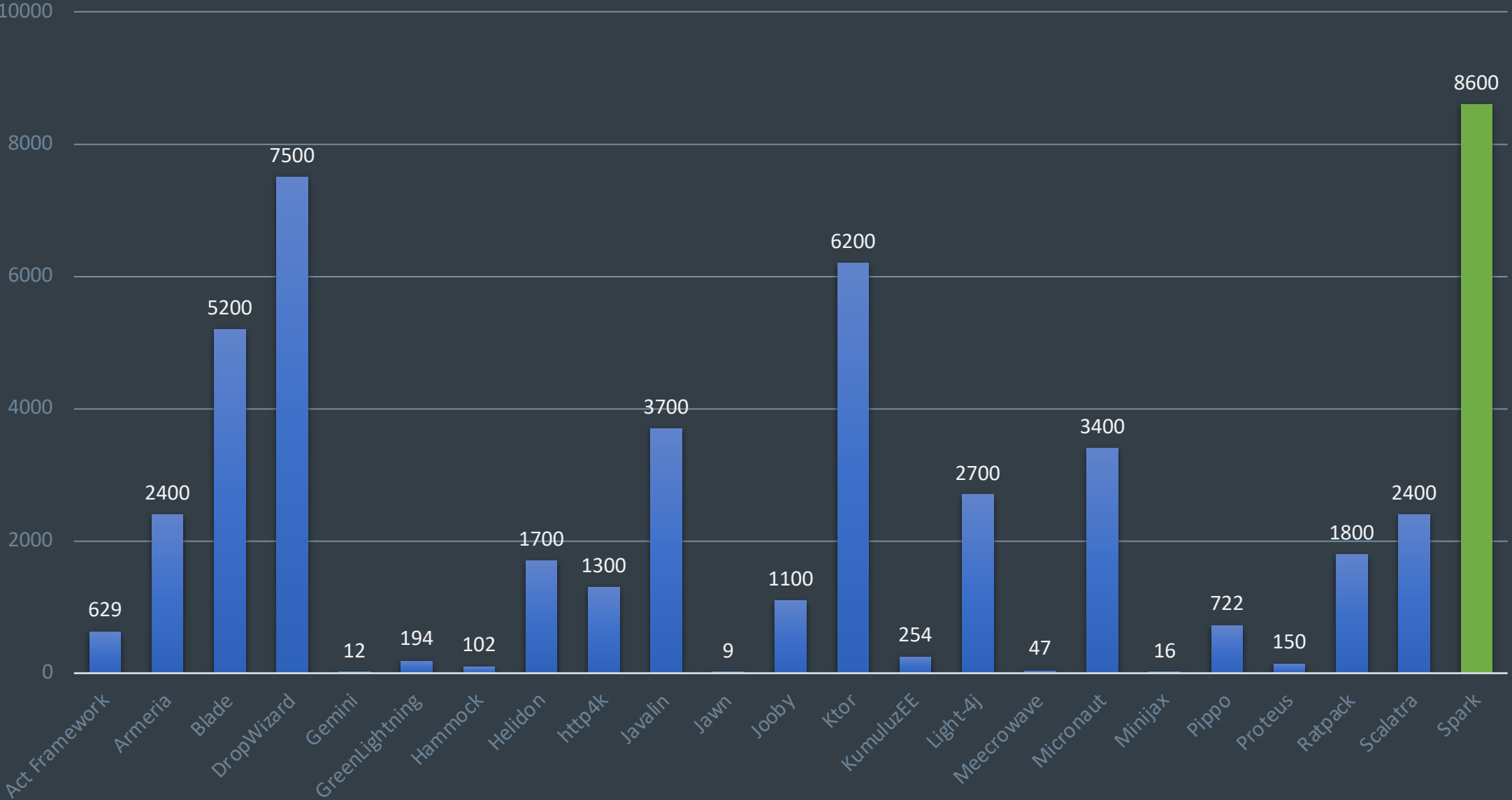
**Followers on Stack Overflow**

Framework	Active contributors		Github stars		Stack Overflow Questions		Stack Overflow Followers		Average Index
	Absolute	Relative	Absolute	Relative	Absolute	Relative	Absolute	Relative	
Act Framework	1	6	629	7	12	1	7	1	4
Armeria	6	33	2400	28	4	0	6	1	16
Blade	1	6	5200	60	0	0	0	0	17
Dropwizard	3	17	7500	87	1791	100	762	100	76
Gemini	2	11	12	0	0	0	0	0	3
GreenLightning	1	6	194	2	0	0	0	0	2
Hammock	1	6	102	1	48	3	1	0	2
Helidon	7	39	1700	20	11	1	14	2	15
http4k	2	11	1300	15	8	0	6	1	7
Javalin	2	11	3700	43	17	1	9	1	14
Jawn	1	6	9	0	0	0	0	0	1
Jooby	2	11	1100	13	13	1	6	1	6
Ktor	5	28	6200	72	205	11	93	12	31
KumuluzEE	3	17	254	3	4	0	0	0	5
Light-4j	4	22	2700	31	33	2	4	1	14
Meecrowave	2	11	47	1	2	0	0	0	3
Micronaut	18	100	3400	40	482	27	189	25	48
Minijax	1	6	16	0	0	0	0	0	1
Pippo	1	6	722	8	0	0	0	0	3
Proteus	1	6	150	2	0	0	0	0	2
Ratpack	1	6	1800	21	85	5	58	8	10
Scalatra	2	11	2400	28	303	17	70	9	16
Spark Java	0	0	8600	100	496	28	139	18	36

## Active contributors

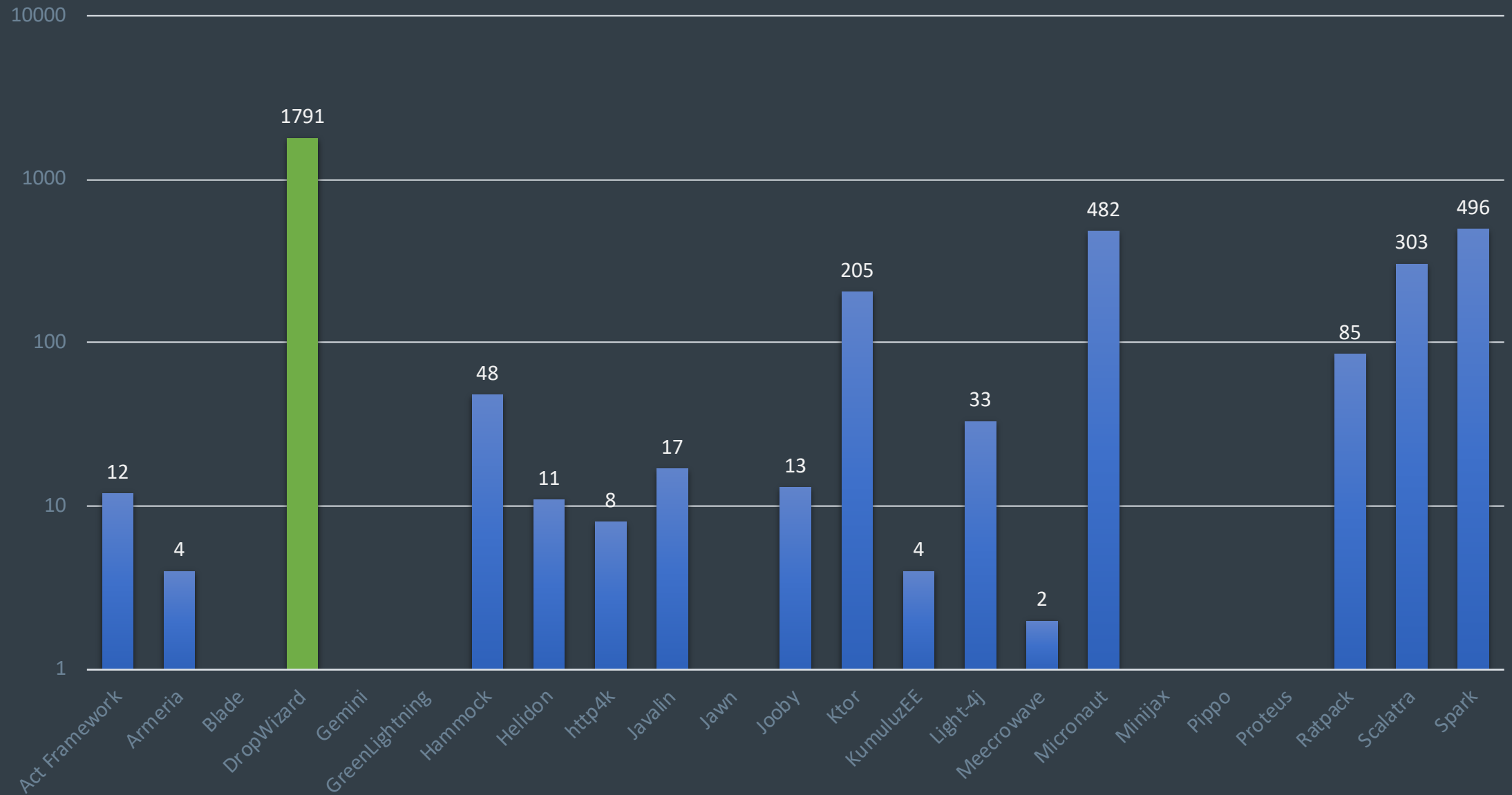


# Stars on Github

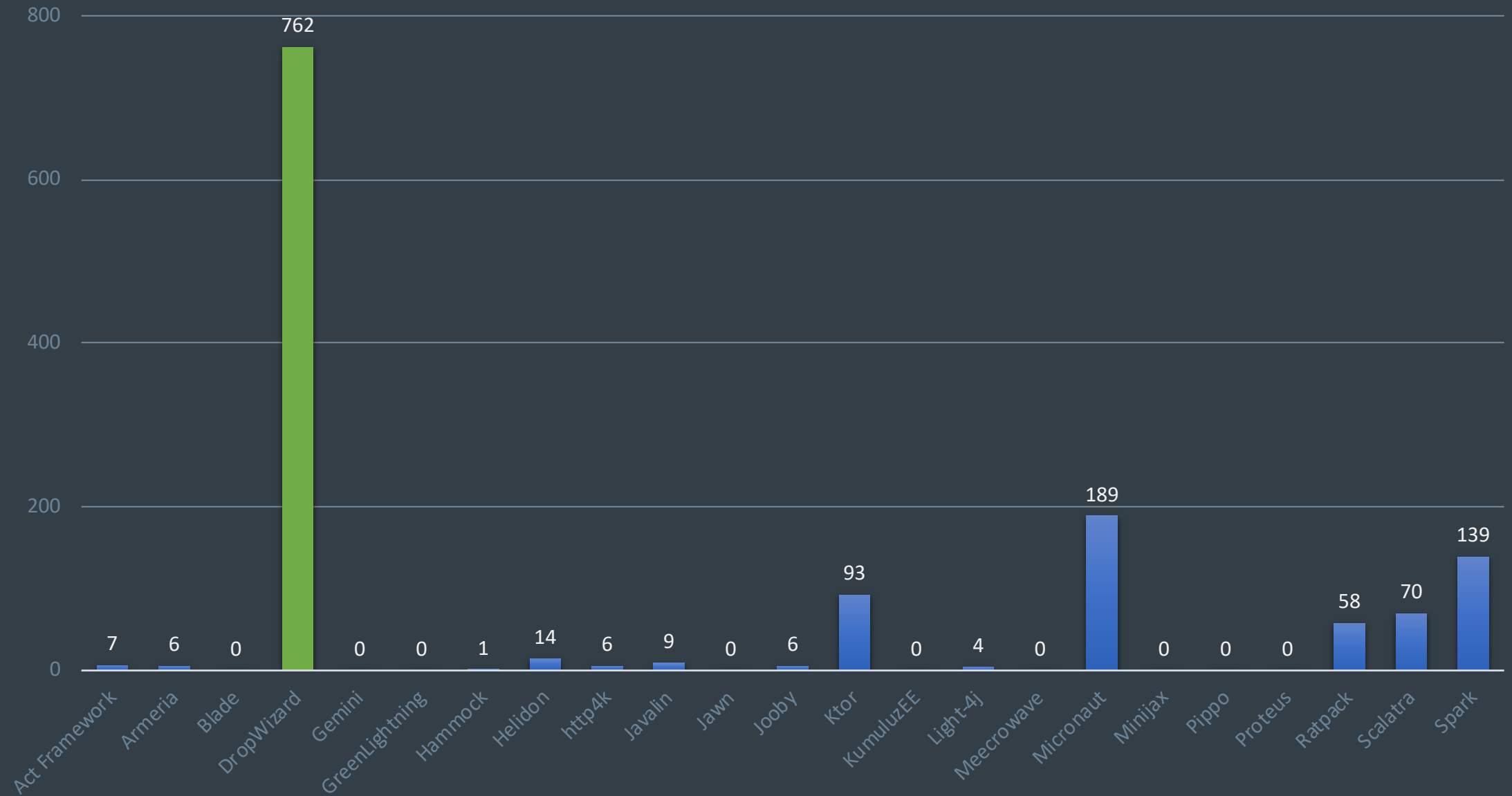




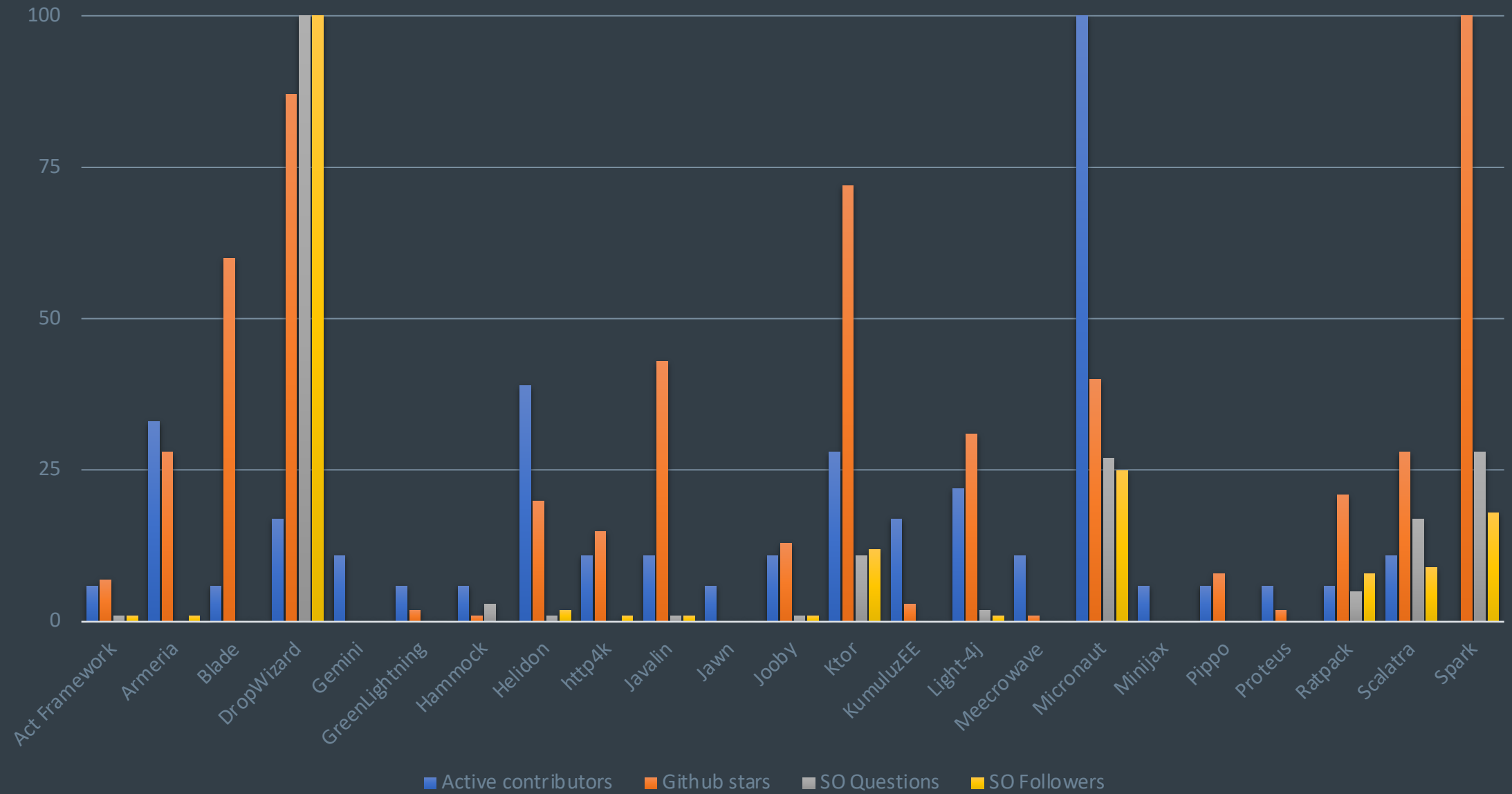
# Questions on Stack Overflow (logarithmic scale)



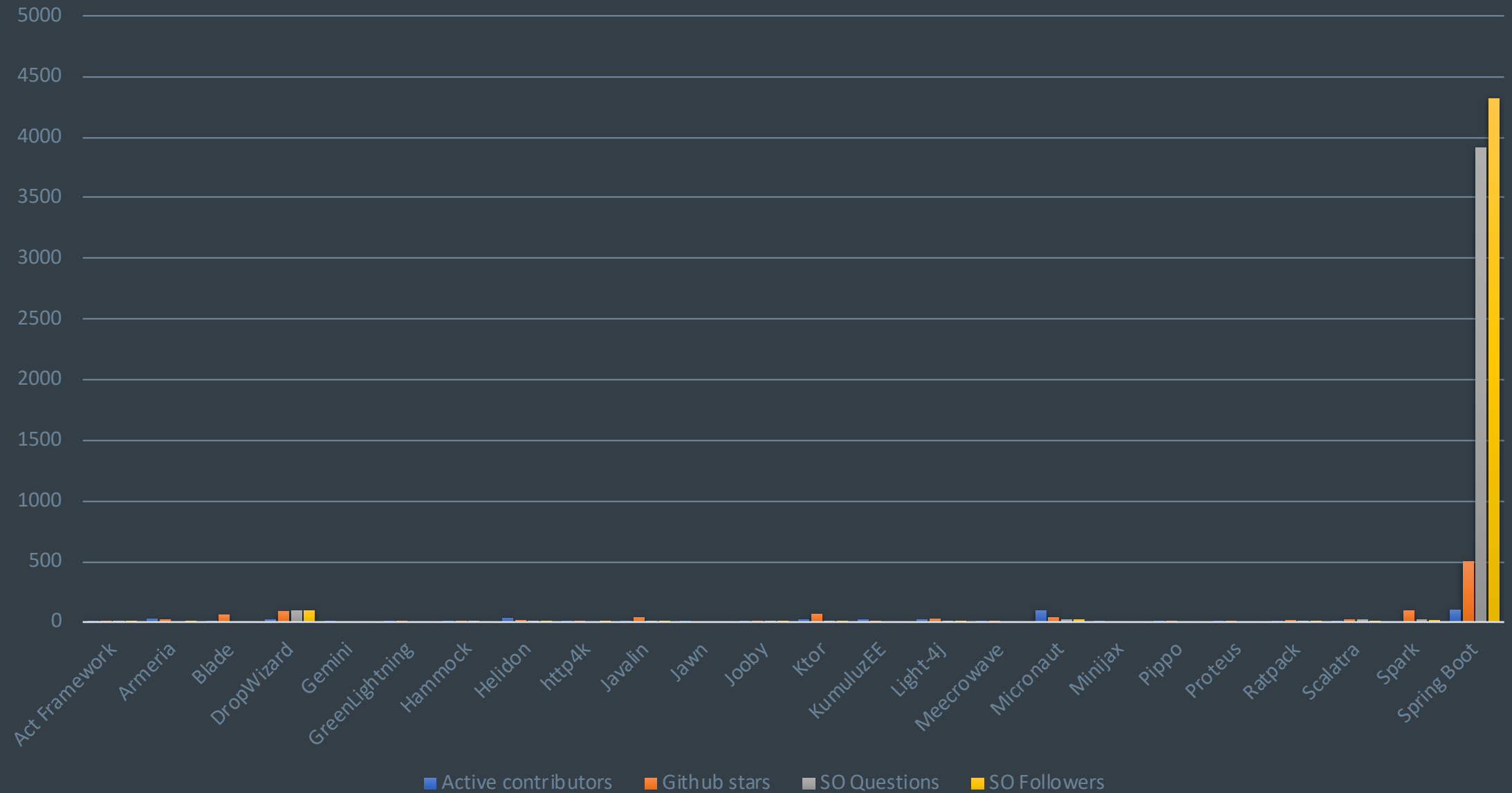
# Followers on Stack Overflow



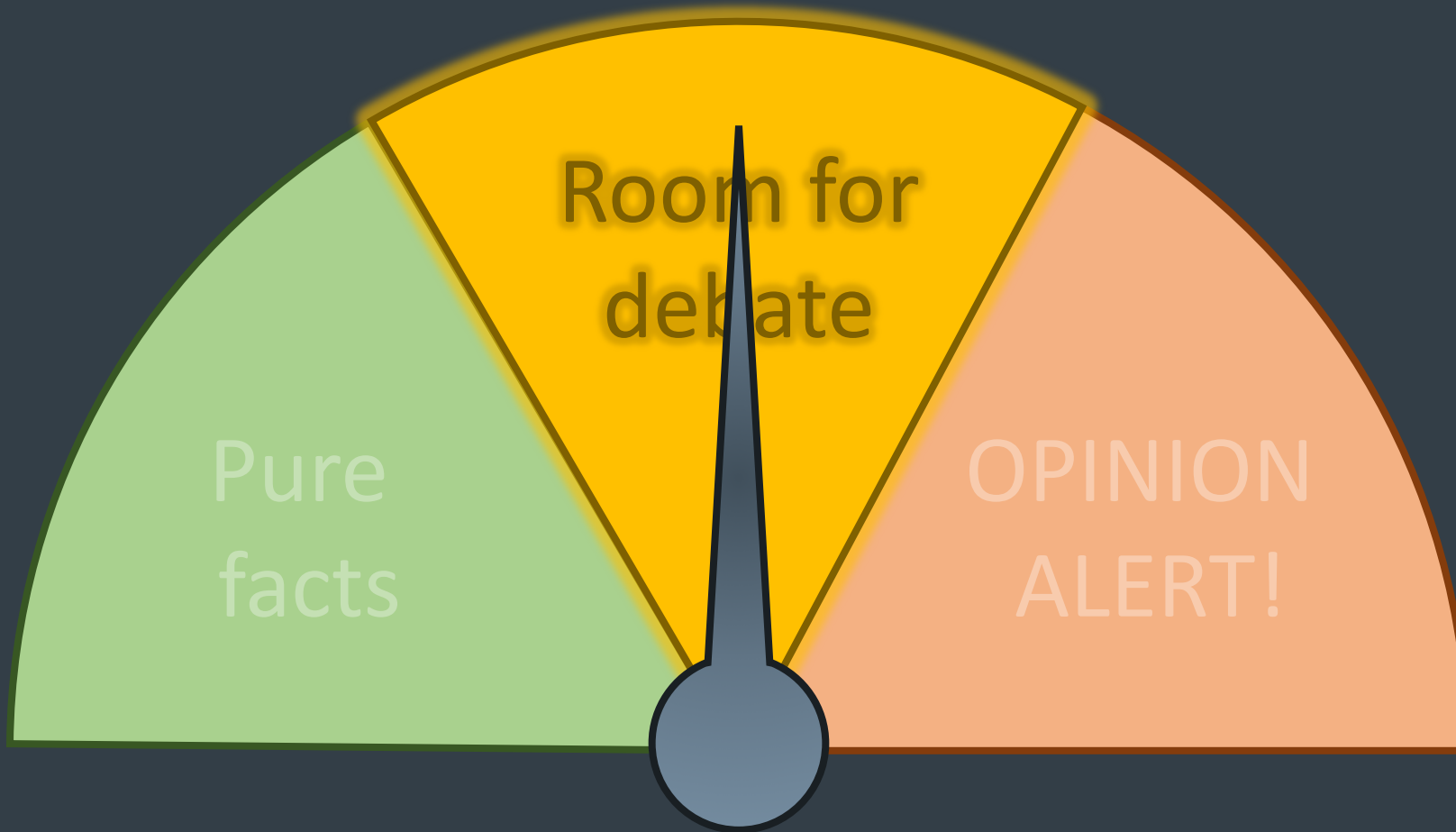
# All metrics, indexed



# All metrics, indexed, as compared to Spring Boot



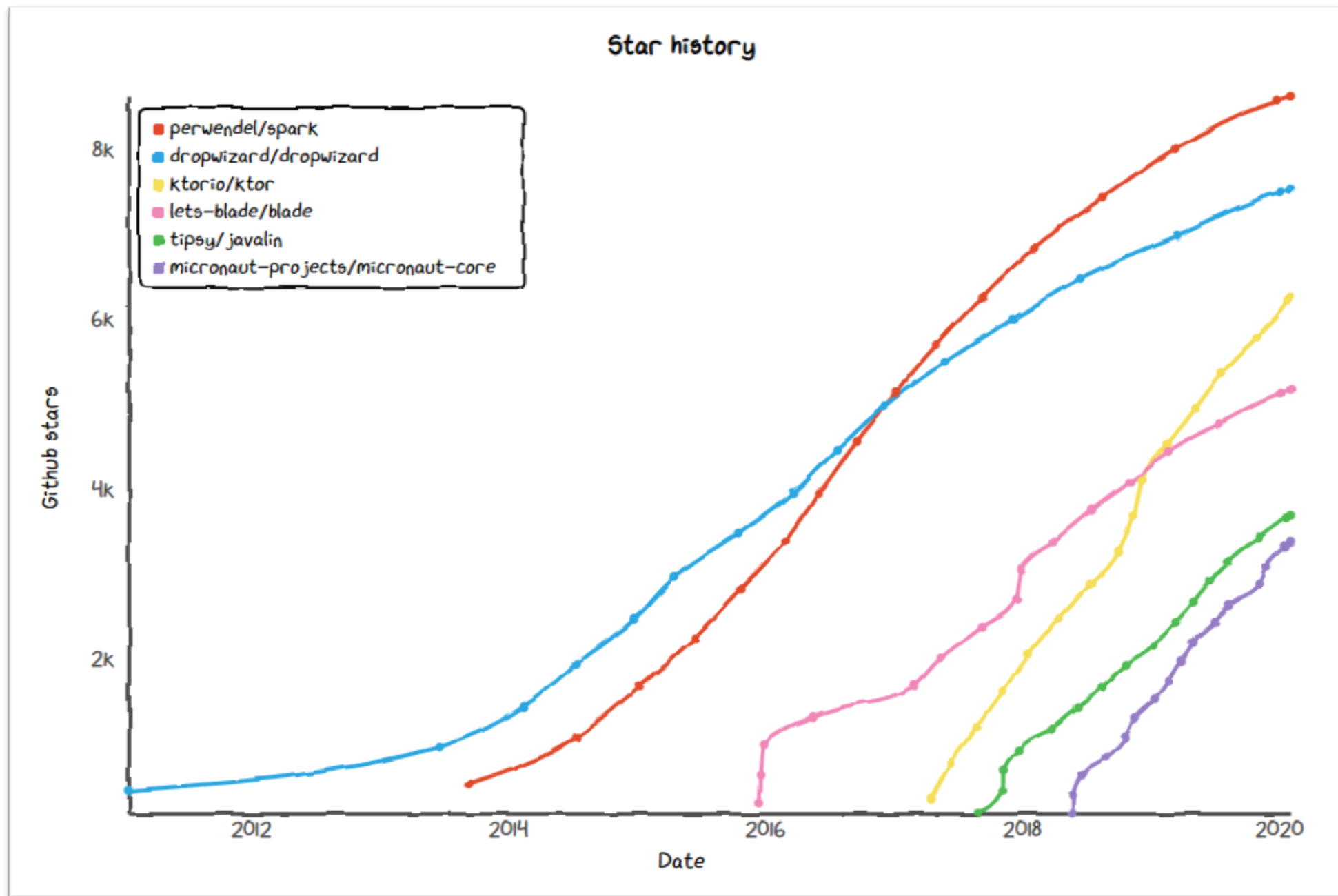
# Opinion-o-meter



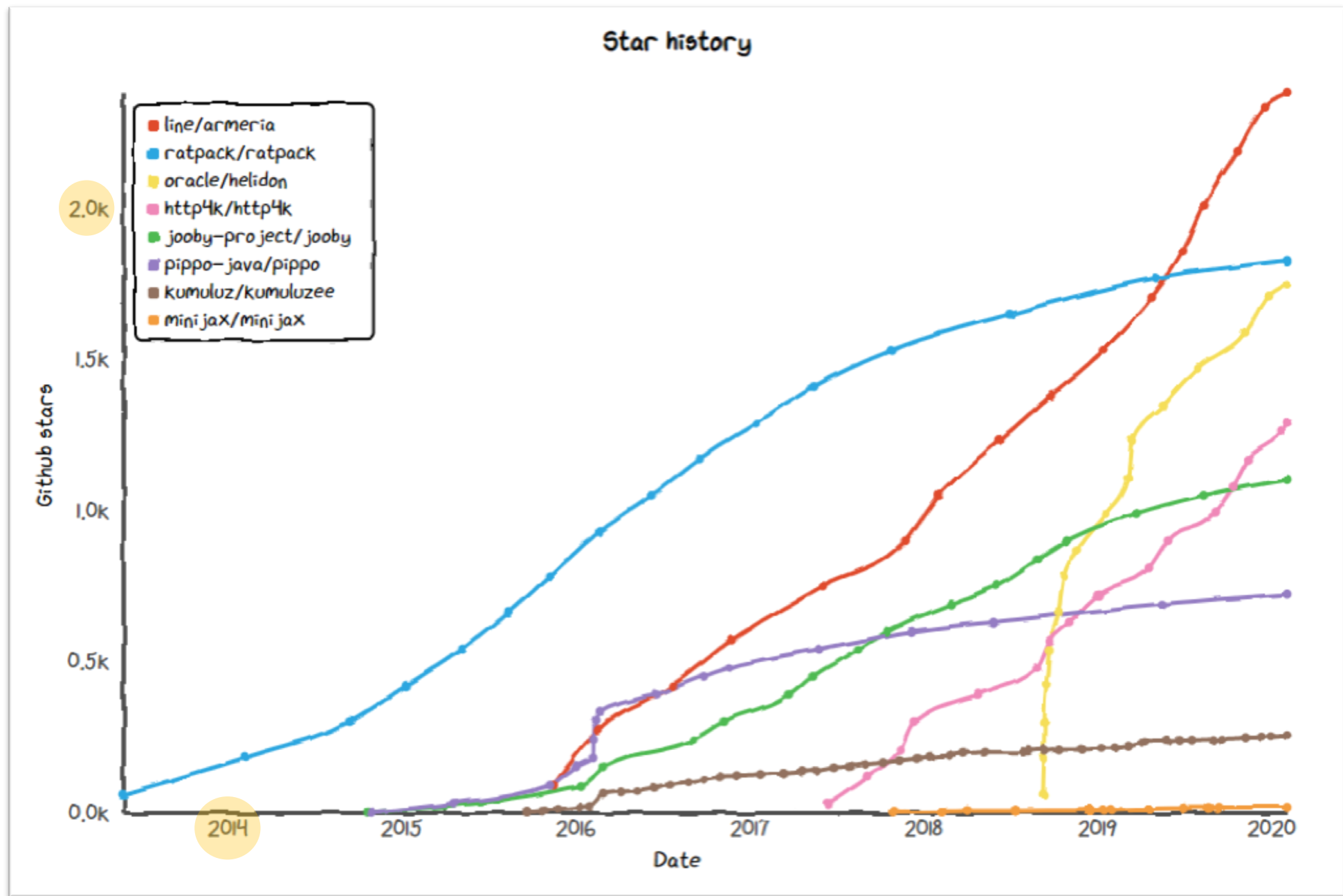
## The slightly-less-long list

- |               |               |
|---------------|---------------|
| 1. Armeria    | 9. Ktor       |
| 2. Blade      | 10. KumuluzEE |
| 3. Dropwizard | 11. Micronaut |
| 4. Helidon SE | 12. Minijax   |
| 5. Helidon MP | 13. Pippo     |
| 6. http4k     | 14. Ratpack   |
| 7. Javalin    | 15. Spark     |
| 8. Jooby      |               |









<https://star-history.t9t.io/>

// TODO:

<https://www.todobackend.com/>



# 6 REST endpoints

6 REST endpoints

JSON serialization (Jackson)

6 REST endpoints

JSON serialization (Jackson)

CORS support

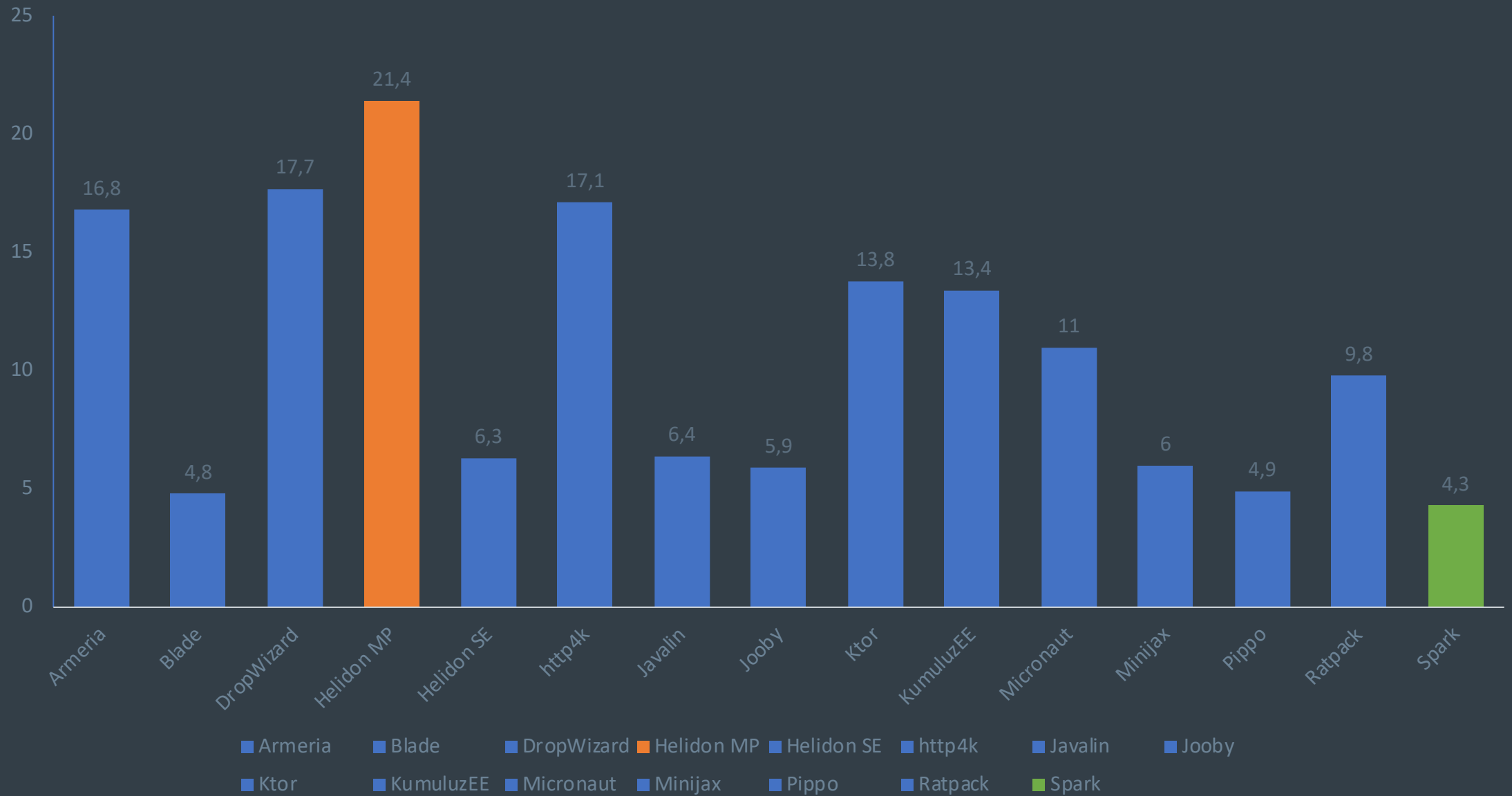
6 REST endpoints

JSON serialization (Jackson)

CORS support

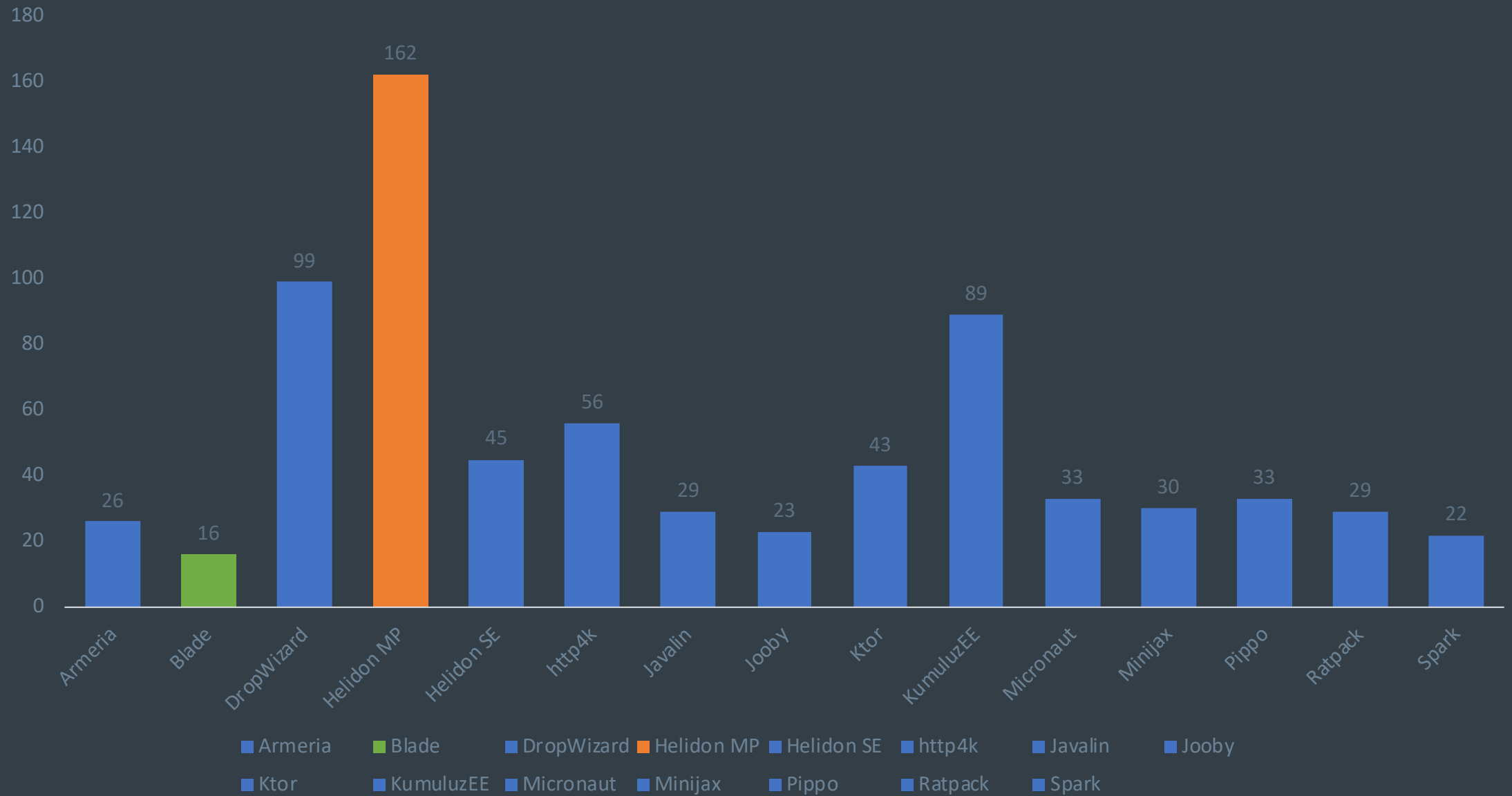
Minimalistic setup

Size of dependencies in MB (smaller is better)



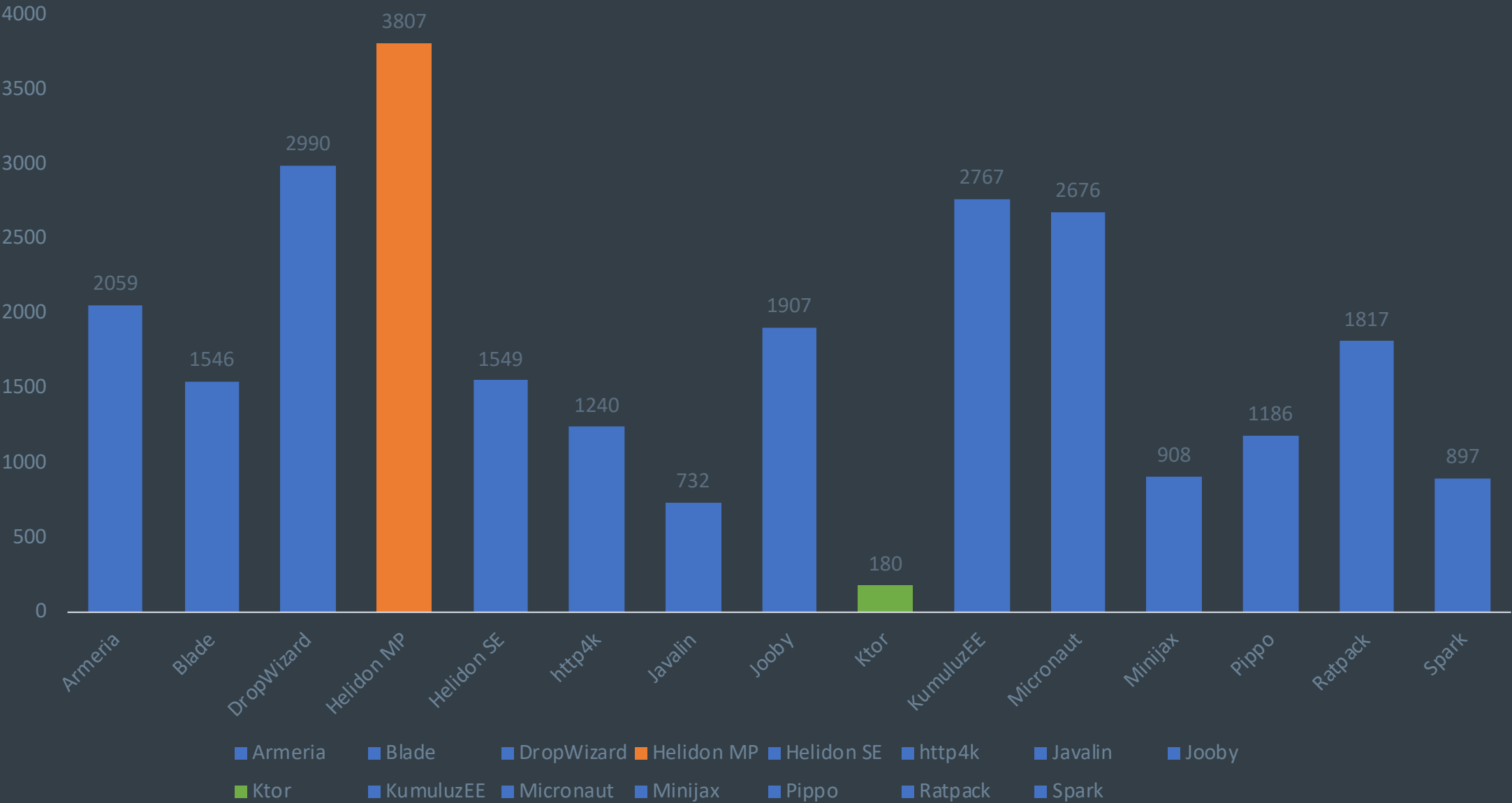


Number of dependencies (smaller is better)

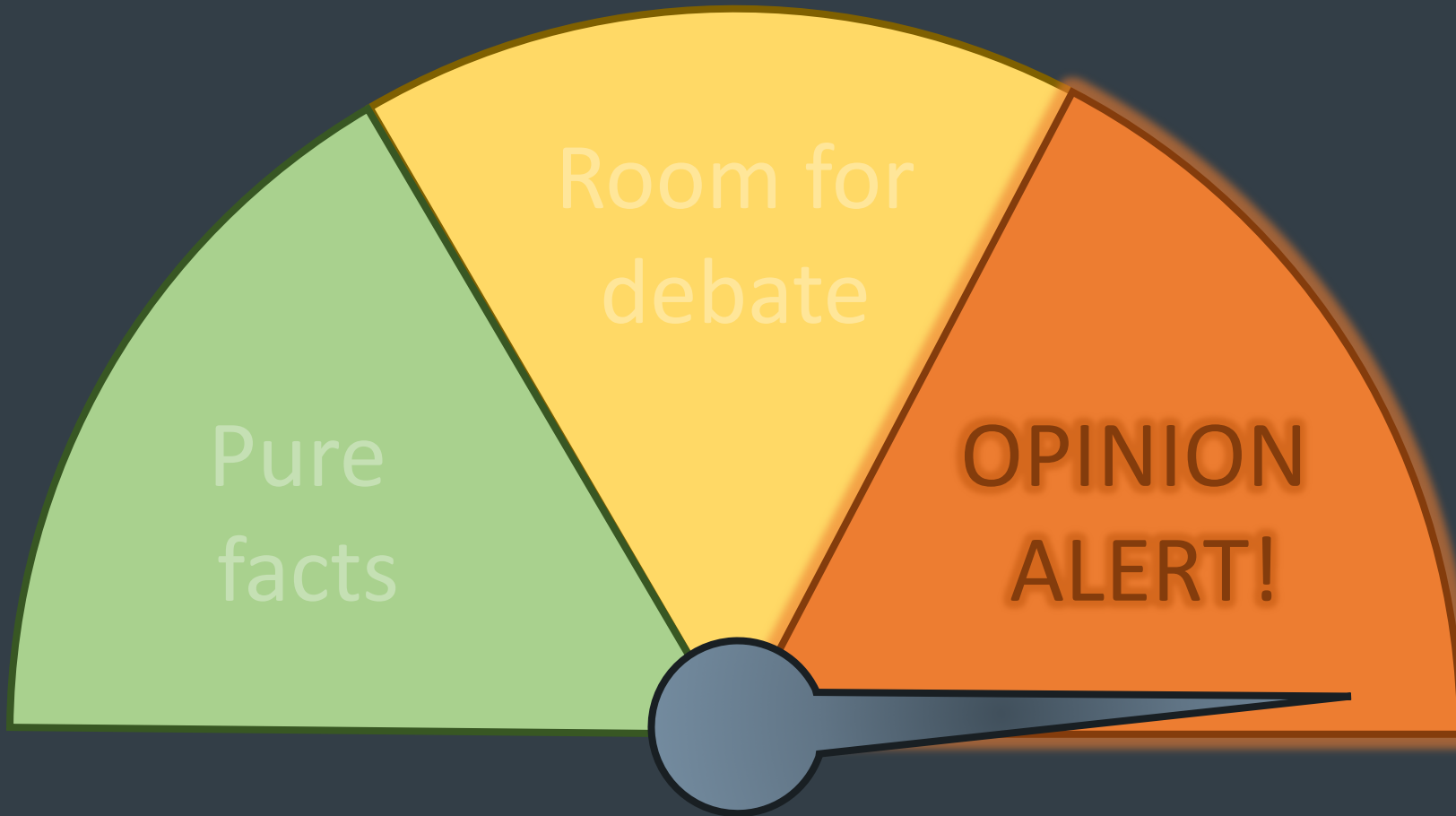


# Startup time in ms (smaller is better)

16 GB RAM | Intel i7 4 cores | JDK 13 / 1.8



# Opinion-o-meter





License **Apache 2.0**

Organization **LINE Corp.**

Language **Java**

Programming Model **Annotations**

Engines **Jetty, Tomcat**

Github stars **2400**

Active contributors **6**

Stack Overflow  
Questions **4**

Stack Overflow  
followers **6**

Armeria

Async and reactive

Anonymous inner classes

Decent documentation

Object or 404 is hard



Armeria

```
public static void main(String[] args) {  
    ServerBuilder sb = Server.builder();  
    sb.http(PORT);  
  
    sb.service("/hello", ArmeriaApplication::helloWorld);  
    sb.service("/hello/{name}", ArmeriaApplication::hello);  
  
    TodoRestController todoBackendApi = createTodoBackend();  
    sb.annotatedService(todoBackendApi);  
  
    Server server = sb.build();  
    CompletableFuture<Void> future = server.start();  
    future.join();  
  
    LOGGER.info("Server Started");  
}
```



Armeria

```
@Post("/todos")
@ProducesJson
@ConsumesJson
public TodoDTO createTodo(CreateTodoCommand command) {
    Todo todo = this.todoService.createNewTodo(command.getTitle(),
        command.getOrder());
    return this.wrap(todo);
}

@Get("/todos/{id}")
@ProducesJson
public HttpResponse getTodo(@Param("id") String id) throws JsonProcessingException {
    return this.todoService.findById(UUID.fromString(id))
        .map(this::wrap)
        .map(todo -> HttpResponse.of(HttpStatus.OK, MediaType.JSON,
            this.convertToJson(todo)))
        .orElse(HttpResponse.of(404));
}
```

Little documentation

Most documentation in Chinese

Not very lively

Spelling errors in public API



License **Apache 2.0**

Organization **Community**

Language **Java**

Programming Model **Programmatic**

Engine **Netty**

Github stars **5200**

Active contributors **1**

Stack Overflow  
Questions **0**

Stack Overflow  
followers **0**



```
public static void main(String[] args) {  
    JsonKit.jsonSupprt(new JacksonJsonSupport());  
  
    Blade application = Blade.of()  
        .enableCors(true, corsConfigurer())  
        .get("/hello", BladeApplication::helloWorld)  
        .get("/hello/:name", BladeApplication::hello);  
  
    createTodoBackend(application);  
  
    application.start(BladeApplication.class, args);  
}
```

Blade





```
void initializeRoutes(Blade application) {
    application.contextPath("/todos")
        .get("/", this::getAll)
        .post("/", this::createTodo)
        .delete("/", this::deleteAll)
        .get("/:id", this::getTodo)
        .delete("/:id", this::deleteTodo)
        .routeMatcher().addRoute("/:id", this::patchTodo,
                                HttpMethod.PATCH);
}

private void getAll(RouteContext context) {
    List<TodoDTO> todos = this.todoService.findAll().stream()
        .map(this::wrap)
        .collect(Collectors.toList());

    context.json(todos);
}
```



License **Apache 2.0**

Organization **Community**

Language **Java**

Programming Model **JAX-RS**

Engine **Jetty**

Github stars **7500**

Active contributors **3**

Stack Overflow  
Questions **1790**

Stack Overflow  
followers **759**

Dropwizard

Built on de-facto standards

Targeted at microservices

Many additional modules

Highest adoption of all

Metrics used by other frameworks



# Dropwizard

```
public class DropWizardApplication extends Application<Configuration> {  
    public static void main(String[] args) throws Exception {  
        var application = new DropWizardApplication();  
        application.run(args);  
    }  
  
    @Override  
    public void run(Configuration configuration, Environment environment) {  
        this.enableCors(environment);  
  
        var helloWorldController = new HelloWorldController();  
        environment.jersey().register(helloWorldController);  
  
        var todoRestController = createTodoBackend();  
        environment.jersey().register(todoRestController);  
    }  
}
```



# Dropwizard

```
@GET
@Path("/{id}")
@Produces(MediaType.APPLICATION_JSON)
public Response getTodo(@PathParam("id") String id) {
    return this.todoService.findById(UUID.fromString(id))
        .map(this::wrap)
        .map(body -> Response.ok(body).build())
        .orElse(Response.status(Response.Status.NOT_FOUND).build());
}
```

No CORS Support

Quite heavy to be micro

Less mature than expected

Good documentation



License **Apache 2.0**

Organization **Oracle**

Language **Java**

Programming Model **Microprofile**

Engine **Netty**

Github stars **1700**

Active contributors **7**

Stack Overflow  
Questions **11**

Stack Overflow  
followers **14**



```
public static void main(String[] args) {  
    Server.builder()  
        .addApplication(new HelloWorldApplication())  
        .addApplication(new TodoApplication())  
        .build()  
        .start();  
}
```



License **Apache 2.0**

Organization **Oracle**

Language **Java**

Programming Model **Programmatic**

Engine **Netty**

Github stars **1700**

Active contributors **7**

Stack Overflow  
Questions **11**

Stack Overflow  
followers **14**

Helidon SE

Far more light-weight than MP

Elegant programming model

No native support for PATCH

Good documentation



```
var serverConfig =  
    ServerConfiguration.builder().port(8080).build();  
var server = WebServer.create(serverConfig, createRouting());  
  
server.start()  
    .thenAccept(ws -> {  
        System.out.println(  
            "WEB server is up! http://localhost:" + ws.port());  
        ws.whenShutdown().thenRun(()  
            -> System.out.println("WEB server is DOWN."));  
    })  
    .exceptionally(t -> {  
        System.err.println("Startup failed: " + t.getMessage());  
        t.printStackTrace(System.err);  
        return null;  
    });
```





```
@Override
public void update(Routing.Rules rules) {
    rules.anyOf(List.of(GET, DELETE, POST), this::corsFilter)
        .options(this::corsPreflight)
        .get("/", this::getAll)
        .post("/",
            Handler.create(CreateTodoCommand.class, this::createTodo))
        .delete("/", this::deleteAll)
        .get("/{id}", this::getTodo)
        .delete("/{id}", this::deleteTodo)
        .anyOf(List.of(PATCH), "/{id}",
            Handler.create(PartialTodo.class, this::patchTodo));
}
```



# Helidon SE

```
private void getTodo(ServerRequest request, ServerResponse response) {  
    UUID id = UUID.fromString(request.path().param(ID_PATH_PARAM));  
  
    this.todoService.findById(id)  
        .map(this::wrap)  
        .ifPresentOrElse(response::send,  
            () -> response.status(NOT_FOUND_404).send());  
}
```

Application as a Function

Path mapping cumbersome

Request attribute, return response



License **Apache 2.0**

Organization **Community**

Language **Kotlin**

Programming Model **Programmatic**

Engines **Jetty, Undertow, Netty, CIO, Apache, Sun Http**

Github stars **1300**

Active contributors **2**

Stack Overflow Questions **8**

Stack Overflow followers **6**



http4k

```
val app = routes(  
    "/hello" bind GET to { Response(OK).body("Hello World!") },  
    "/hello/{name}" bind GET to helloController,  
    "/todos" bind GET to todoRestController::getAll,  
    "/todos" bind POST to todoRestController::createTodo,  
    "/todos" bind DELETE to todoRestController::deleteAll,  
    "/todos/{id}" bind GET to todoRestController::getTodo,  
    "/todos/{id}" bind PATCH to todoRestController::patchTodo,  
    "/todos/{id}" bind DELETE to todoRestController::deleteTodo  
)  
.withFilter(ServerFilters.Cors(CorsPolicy(  
    origins = listOf("*"),  
    headers = listOf("Content-Type"),  
    methods = listOf(GET, POST, DELETE, PATCH)  
)))  
  
val server = app.asServer(Jetty(8080))  
  
server.start()
```



```
fun createTodo(req: Request): Response {  
    val command = json.asA(req.bodyString(), CreateTodoCommand::class)  
  
    val todo = todoService.createNewTodo(command.title, command.order)  
  
    return Response(Status.OK)  
        .header("Content-Type", ContentType.APPLICATION_JSON.value)  
        .body(todo.wrap().asJsonObject().toPrettyString())  
}
```

http4k



License **Apache 2.0**

Organization **Community**

Language **Kotlin, Java**

Programming Model **Programmatic**

Engine **Jetty**

Github stars **3700**

Active contributors **2**

Stack Overflow  
Questions **17**

Stack Overflow  
followers **9**

Javalin

Forked from Spark

Small, but extensible

Well documented

API + static files



# Javalin

```
public static void main(String[] args) {  
    Javalin.create(JavalinApplication::configure)  
        .routes(() -> {  
            path("/hello", helloWorldApi());  
            path("/todos", todoApi());  
        })  
        .start(PORT);  
}  
  
private static void configure(JavalinConfig config) {  
    config.enableCorsForAllOrigins();  
}
```



Javalin

```
EndpointGroup defineEndpoints() {  
    return () -> {  
        get(this::getAll);  
        post(this::createTodo);  
        delete(this::deleteAll);  
        get(ID_PATH, this::getTodo);  
        patch(ID_PATH, this::patchTodo);  
        delete(ID_PATH, this::deleteTodo);  
    };  
}  
  
void getAll(Context context) {  
    List<TodoDTO> todos = this.todoService.findAll().stream()  
        .map(this::wrap)  
        .collect(Collectors.toList());  
  
    context.json(todos);  
}
```



Supports non-blocking

Copied JAX-RS annotations?

Extensible via extensions

Very well documented



License **Apache 2.0**

Organization **Community**

Language **Kotlin, Java**

Programming Model **Programmatic OR  
Annotations**

Engines **Jetty, Undertow, Netty**

Github stars **1100**

Active contributors **2**

Stack Overflow  
Questions **13**

Stack Overflow  
followers **6**

```
runApp(args, app -> {  
    app.setServerOptions(new ServerOptions().setPort(PORT));  
    app.encoder(MediaType.JSON, jacksonModule);  
    app.decoder(MediaType.JSON, jacksonModule);  
    app.decorator(new CorsHandler());  
  
    app.get("/hello", JoobyApplication::helloWorld);  
    app.get("/hello/{name}", JoobyApplication::hello);  
  
    todoController.initializeRoutes(app);  
  
    app.onStarted(() -> {  
        long bootTime = System.currentTimeMillis() - startTime;  
        System.out.println("Started in " + bootTime + "ms.");  
    });  
});
```

```
private TodoDTO patchTodo(Context context) {  
    UUID id = UUID.fromString(context.path(ID_PATH_PARAM).value());  
  
    var command = context.body(PartialTodo.class);  
  
    return this.todoService.updateTodo(id, command)  
        .map(this::wrap)  
        .orElseThrow(() -> new StatusCodeException(NOT_FOUND));  
}
```



License **Apache 2.0**

Organization **JetBrains**

Language **Kotlin**

Programming Model **Programmatic**

Engine **Jetty, Tomcat,  
Undertow, CIO**

Github stars **6200**

Active contributors **5**

Stack Overflow  
Questions **205**

Stack Overflow  
followers **93**

Ktor

Aimed at asynchronosity

Kotlin DSL adds learning curve

From the makers of Kotlin

Very rich, yet lightweight



Ktor

```
fun Application.module() {  
    install(ContentNegotiation) {  
        jackson {}  
    }  
    install(CORS) {  
        /* snip */  
    }  
  
    routing {  
        get("/hello") {  
            call.respondText("Hello World!", ContentType.Text.Plain)  
        }  
        get("/hello/{name}") {  
            call.respondText("Hello, ${call.parameters["name"]}!")  
        }  
  
        todoRoutes()  
    }  
}
```



Ktor

```
get("/{id}") {  
    val id = call.parameters["id"]  
    val todo = todoService.findById(UUID.fromString(id))  
        .map { it.wrap() }  
        .orElse(null)  
  
    when(todo) {  
        null -> call.respond(HttpStatusCode.NotFound)  
        else -> call.respond(todo)  
    }  
}
```

Early Microprofile adopter

Duke's Choice Award 2015

Part of a larger product suite



License **MIT**

Organization **Sunesis**

Language **Java, Node, Go**

Programming Model **Microprofile**

Engine **Jetty**

Github stars **254**

Active contributors **3**

Stack Overflow  
Questions **4**

Stack Overflow  
followers **0**



KumuluzEE

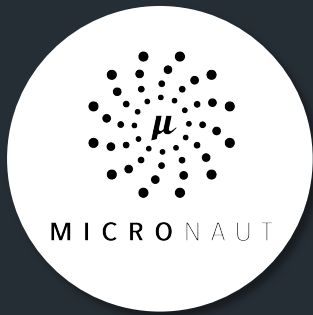
```
@ApplicationPath("")
public class KumuluzEEApplication extends Application {
    @Override
    public Set<Class<?>> getClasses() {
        Set<Class<?>> classes = new HashSet<>();
        classes.add(HelloWorldController.class);
        classes.add(TodoController.class);
        return classes;
    }
}
```





```
@ApplicationScoped
public class TodoBackendFactory {
    @Produces
    public TodoRepository todoRepository() {
        return new InMemoryTodoRepository();
    }

    @Produces
    public TodoService todoService() {
        return new TodoService(todoRepository());
    }
}
```



License **Apache 2.0**

Organization **Object Computing Inc.**

Language **Java, Kotlin, Groovy**

Programming Model **Annotations**

Engine **Netty**

Github stars **3400**

Active contributors **18 (!)**

Stack Overflow  
Questions **482**

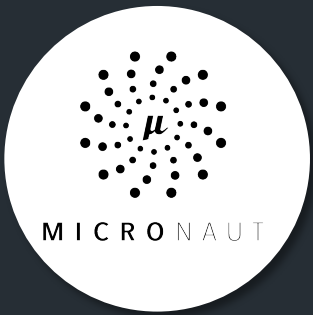
Stack Overflow  
followers **189**

Micronaut

# Spring-inspired API

## AOT compilation

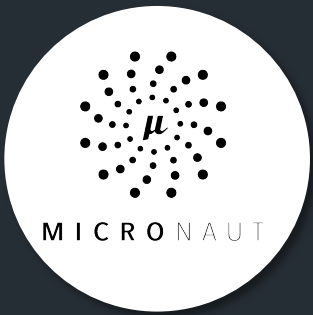
## Gaining terrain fast



# Micronaut

```
public class MicronautApplication {  
    public static void main(String[] args) {  
        Micronaut.run(MicronautApplication.class);  
    }  
}
```

```
@Factory  
public class TodoFactory {  
    @Bean()  
    public TodoService todoService() {  
        return new TodoService(new InMemoryTodoRepository());  
    }  
}
```



Micronaut

```
@Patch(uri =("/{id}"),  
    produces = MediaType.APPLICATION_JSON,  
    consumes = MediaType.APPLICATION_JSON)  
public Optional<TodoDTO> patchTodo(@PathVariable String id,  
    PartialTodo command) {  
  
    return this.todoService.updateTodo(UUID.fromString(id), command)  
        .map(this::wrap);  
}
```

## Basically JAX-RS

Several useful extensions available

Sparse documentation

Basic dependency injection

Simple server configuration



License **Apache 2.0**

Organization **Community**

Language **Java**

Programming Model **JAX-RS**

Engine **Undertow, Netty**

Github stars **16**

Active contributors **1**

Stack Overflow  
Questions **0**

Stack Overflow  
followers **0**



Minijax

```
new Minijax()  
  .port(8080)  
  .register(JsonFeature.class)  
  .register(CorsFilter.class)  
  .register>HelloWorldController.class)  
  .register(new TodoService(todoRepository))  
  .register(TodoController.class)  
  .start();
```



License **Apache 2.0**

Organization **Community**

Language **Java**

Programming Model **Programmatic**

Engine **Jetty, Tomcat,  
Undertow, TWJS**

Github stars **722**

Active contributors **1**

Stack Overflow  
Questions **0**

Stack Overflow  
followers **0**

Pippo

Integrates with Spring, Guice, Weld

Extensive documentation

“Easy to use and hack”



Pippo

```
public static void main(String[] args) {  
    Pippo pippo = new Pippo();  
    pippo.getServer().setPort(PORT);  
  
    pippo.ANY("/*", createCorsHandler());  
    pippo.GET("/hello", PippoApplication::helloWorld);  
    pippo.GET("/hello/{name}", PippoApplication::hello);  
  
    var todoRestController = createTodoBackend();  
    todoRestController.initializeRoutes(pippo);  
  
    pippo.start();  
}
```





PipPO

```
private void patchTodo(RouteContext context) {  
    var id = UUID.fromString(context.getParameter(ID_PATH_PARAM)  
        .toString());  
  
    var command = context.createEntityFromBody(PartialTodo.class);  
  
    this.todoService.updateTodo(id, command)  
        .map(this::wrap)  
        .ifPresentOrElse(context.json()::send,  
            () -> context.status(404));  
}
```

API / DSL targeted at Groovy

Paths for RESTful CRUD hard

Asynchronous, non-blocking



License **Apache 2.0**

Organization **Community**

Language **Groovy, Java**

Programming Model **Programmatic**

Engine **Netty**

Github stars **1800**

Active contributors **1**

Stack Overflow  
Questions **85**

Stack Overflow  
followers **58**



```
public static void main(String[] args) throws Exception {
    RatpackServer.start(RatpackApplication::init);
}

private static void init(RatpackServerSpec server) {
    var todoRestController = createTodoBackend();

    server.serverConfig(ServerConfig.embedded().port(PORT))
        .handlers(chain -> {
            chain.all(RatpackApplication::cors)
                .get("hello", RatpackApplication::helloWorld)
                .get("hello/:name", RatpackApplication::hello);
            todoRestController.initializeEndpoints(chain);
        });
}
```

Ratpack



```
void initializeEndpoints(Chain chain) throws Exception {
    chain
        .path("todos", byMethod(chain.getRegistry(), spec ->
            spec.get(this::getAll)
                .post(this::createTodo)
                .delete(this::deleteAll)
            )
        )
        .path("todos/:id", byMethod(chain.getRegistry(), spec ->
            spec.get(this::getTodo)
                .delete(this::deleteTodo)
                .patch(this::patchTodo)
            )
        );
}
```



License **Apache 2.0**

Organization **Community**

Language **Java, Kotlin**

Programming Model **Programmatic**

Engine **Jetty**

Github stars **8600**

Active contributors **0 (!)**

Stack Overflow  
Questions **496**

Stack Overflow  
followers **139**

Spark

The original Java microframework

Unfortunately low activity

Forked by Javalin

Very basic

Fairly well documented



Spark

```
public static void main(String[] args) {  
    port(PORT);  
    addCORS();  
  
    get("/hello", SparkApplication::helloWorld);  
    get("/hello/:name", SparkApplication::hello);  
  
    setupTodoBackend();  
}
```



Spark

```
var jsonTransformer = new JacksonResponseTransformer();

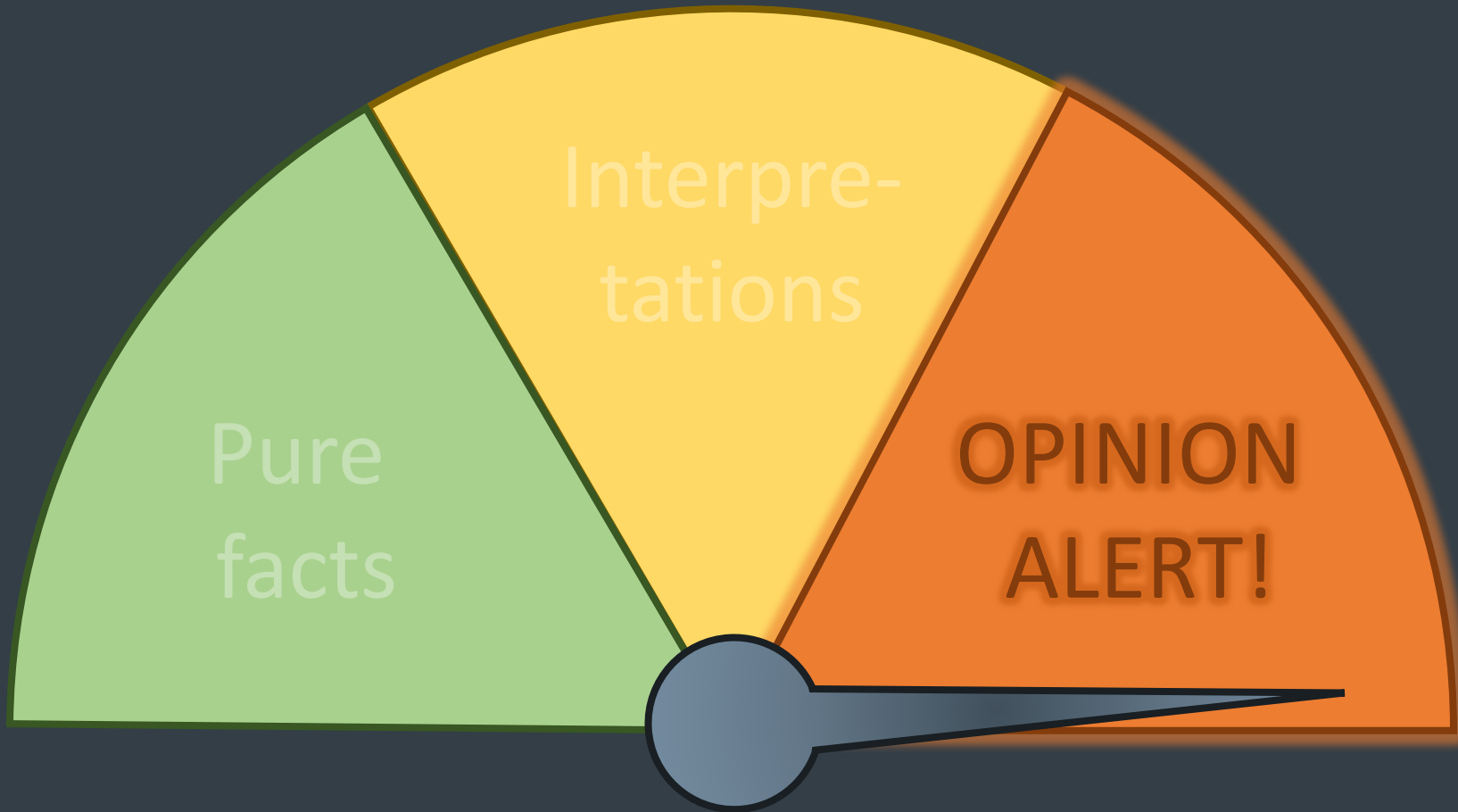
path("/todos", () -> {
  get("", todoRestController::getAll, jsonTransformer);
  post("", todoRestController::createTodo, jsonTransformer);
  delete("", todoRestController::deleteAll);
  path("/:id", () -> {
    get("", todoRestController::getTodo, jsonTransformer);
    delete("", todoRestController::deleteTodo);
    patch("", todoRestController::patchTodo, jsonTransformer);
  });
});
```

# The Winners





# The opinion-o-meter





The winners

# Ktor

The winners

Ktor

Javalin

The winners

Ktor

Javalin

Dropwizard

The winners

Ktor

Javalin

Dropwizard

**Micronaut**

The winners

Ktor

Javalin

Dropwizard

Micronaut

Helidon SE





# Maven archetypes are not dead

Maven archetypes are not dead

Micro as in micro-adoption

Maven archetypes are not dead

Micro as in micro-adoption

Think before you choose

# Thank you!

Please help me improve:

 @JaapCoomans

 In person after the session

 Rate my talk in the app

Get the code:

 <https://github.com/jaapcoomans/microframeworks>

**GROUP9**  
ELITE.JAVA.DEVELOPMENT.