



Micronaut in Action



About us



Silvio D'Alessandro
Senior Software Engineer



Carlos Barragan
Distinguished Software Engineer

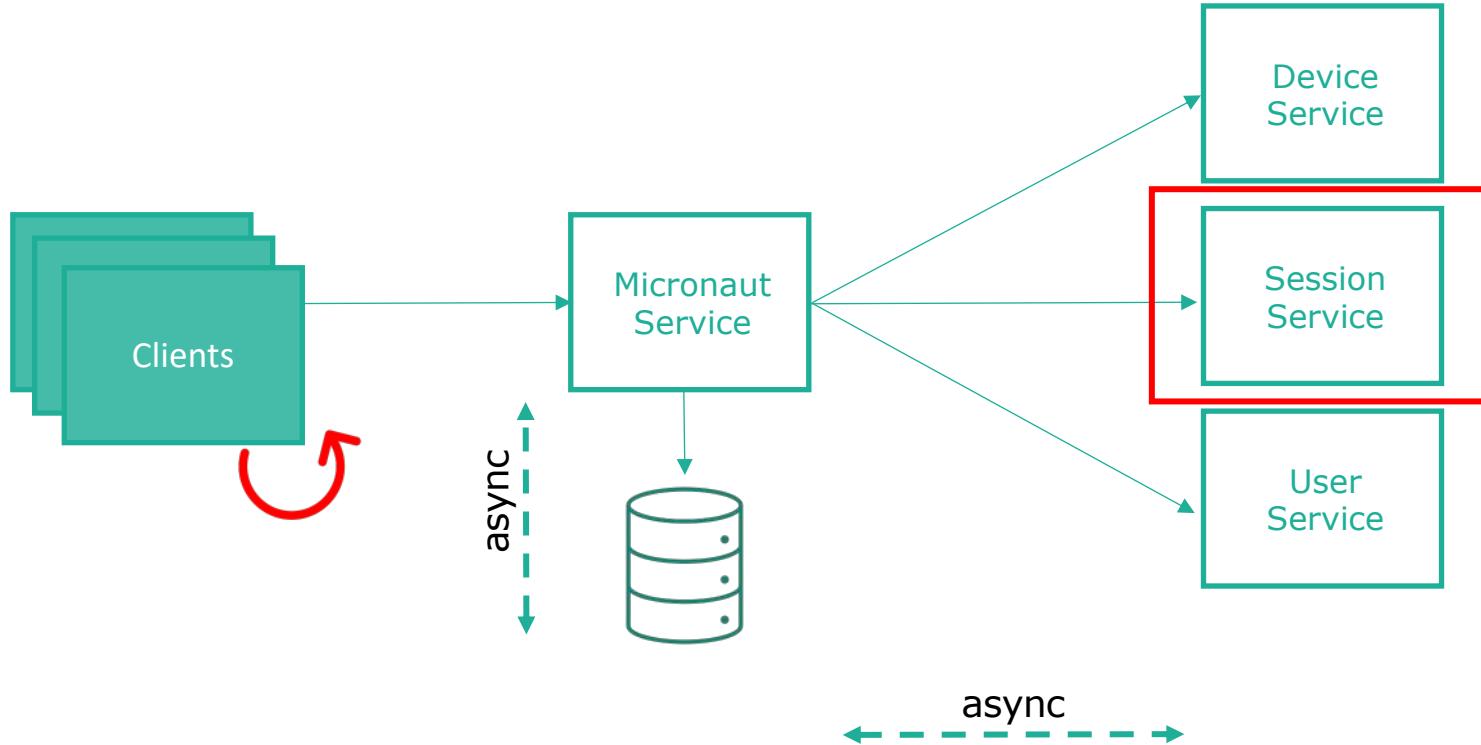
Agenda

1. What is Micronaut?
2. Business Case
3. Technical Stack
4. Why Micronaut?
5. Production Figures
6. Micronaut vs. Spring Boot
7. Conclusion

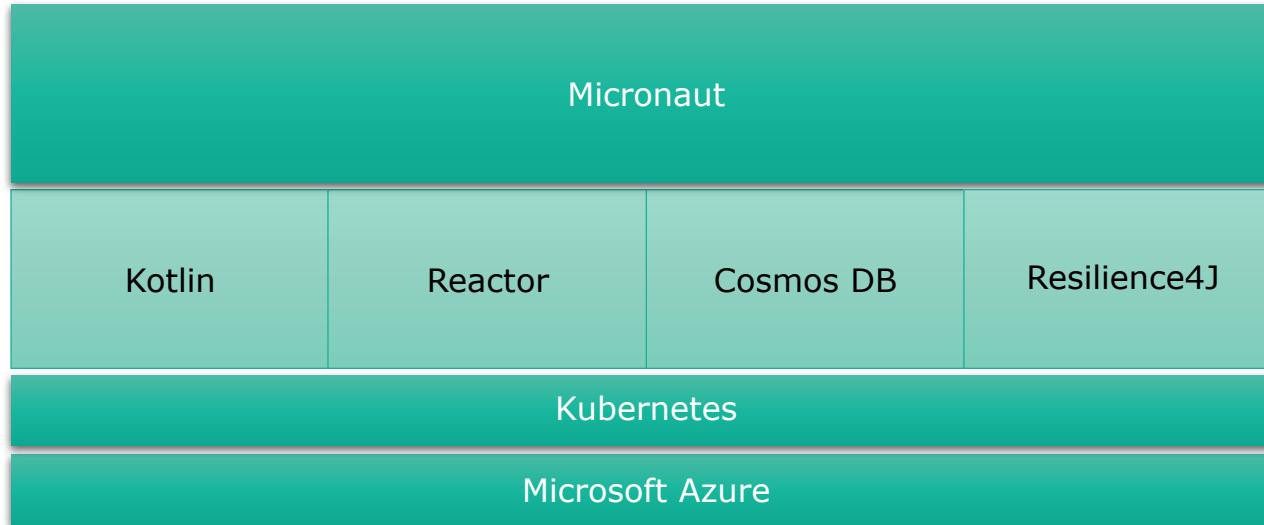
What is Micronaut?

- Micronaut is a **Micro-Framework**
- Micro-Frameworks are not reflection-based, which makes them more likely
 - ... to be **lightweight**
 - ... to have less memory consumption
 - ... to have faster start-up times
- Micronaut focuses on **simplicity** and **speed** of development

Business Case

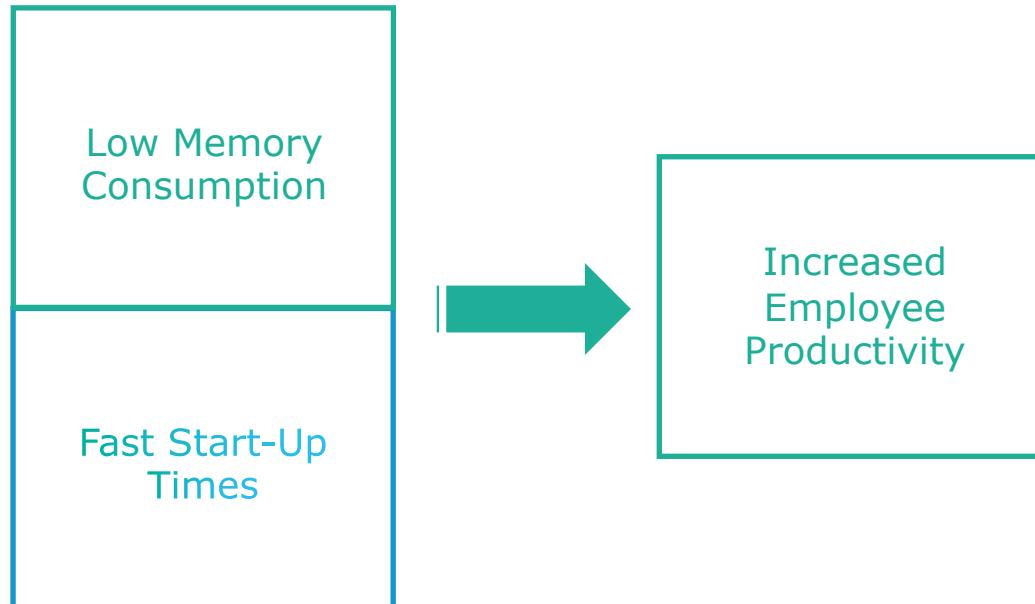


Technical Stack



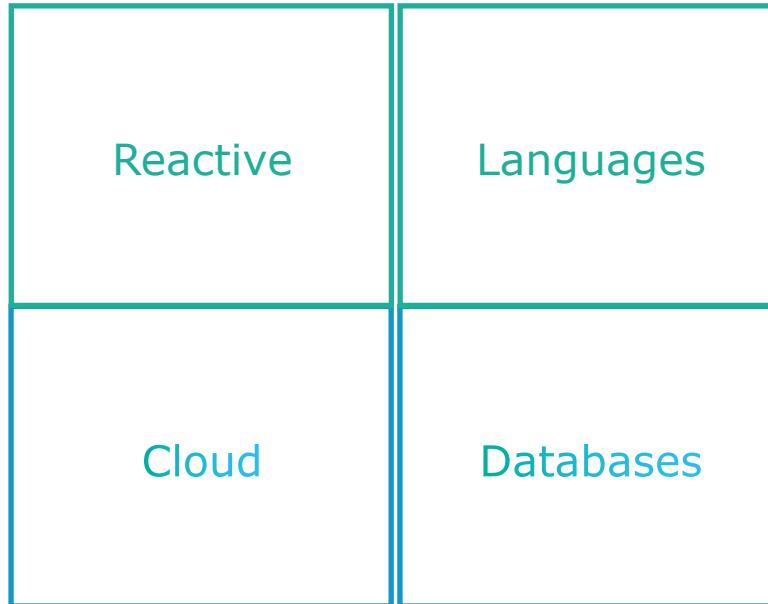
Why Micronaut?

Performance



Why Micronaut?

Diversity of Features



Why Micronaut?

Low Entry Barrier

```
import io.micronaut.http.HttpRequest
import io.micronaut.http.HttpResponse
import io.micronaut.http.annotation.Controller
import io.micronaut.http.annotation.Get
import io.micronaut.http.annotation.PathVariable
import javax.inject.Singleton

@Controller(value = "/sample")
class GreetController(private val service: MyImportantService) {

    @Get(value = "/greet/{name}")
    fun greet(request: HttpRequest<*>, @PathVariable(value = "name") name: String): HttpResponse<String> {
        val greet: String = service.constructGreet(name)
        return HttpResponse.ok(greet)
    }
}

@Singleton
class MyImportantService {

    fun constructGreet(name: String): String {
        return "Hello $name"
    }
}
```

Why Micronaut?

Responsive Community

<https://github.com/micronaut-projects/micronaut-core/issues/2665>

Memory leak when request body is anything but JSON #2665

Closed

carlosbarragan opened this issue on Jan 20 · 26 comments

<https://github.com/micronaut-projects/micronaut-core/issues/4350>

Mutating an instance of NettyMutableHttpRequest throws
UnsupportedOperationException #4350

Closed

22dasi1bwi opened this issue 7 days ago · 6 comments

<https://github.com/micronaut-projects/micronaut-core/issues/3964>

ProxyHttpClient hangs when the response contains no body #3964

Closed

carlosbarragan opened this issue on Aug 25 · 0 comments

Normal Production Figures

7500

Request per Minute

6

Nodes

220

MB heap in average

Peak production figures

>200K

Request per Minute

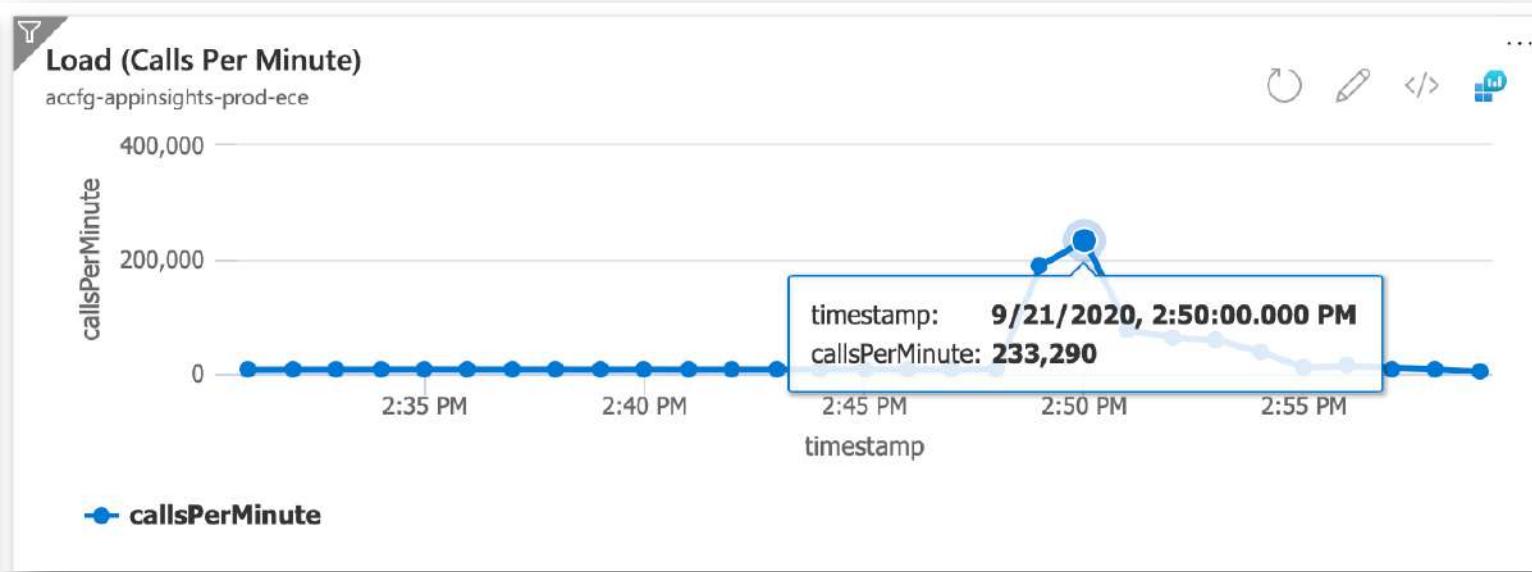
6

Nodes

400

MB heap in average

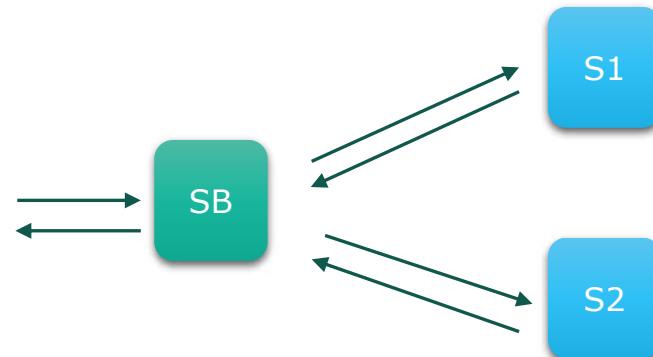
Peak production figures





Spring Boot Comparison

- Spring Boot app is very similar in functionality.
- Fully reactive.
- It does not have a Database, it only routes requests.



Micronaut vs Spring Boot App in Production

	Micronaut (2.0.2)	Spring Boot (2.3.5)
Memory Consumption (heap)	220 MB	250 MB
Request / Minute	~ 7500	~ 60
Startup time	9s	18s
Build Time *	57s (gradle)	27s (maven)
Jar Size	40 MB	36 MB

* Micronaut app executes 170 tests. Many of them start a docker container (using testcontainers).
Spring boot app executes 89 tests. None of them uses docker.

Micronaut Camunda BPM Integration

The screenshot shows the GitHub repository page for 'NovatecConsulting / micronaut-camunda-bpm'. The repository has 14 issues, 5 pull requests, and 30 stars. It contains 4 branches and 19 tags. The repository is described as integrating Micronaut and Camunda BPM Process Engine. It includes links to a blog post and various tags like micronaut, camunda-bpm, bpmn, etc. The repository has 180 commits from tobiaschaefer.

About

Integration between Micronaut and Camunda BPM Process Engine. We configure Camunda BPM with sensible defaults, so that you can get started with minimum configuration: simply add a dependency in your Micronaut project to embed the process engine!

www.novatec-gmbh.de/en/blog/mic...

micronaut camunda-bpm bpmn hacktoberfest business-process java kotlin dmn

Readme Apache-2.0 License

Releases 19

v0.10.1 (Latest) 6 days ago + 18 releases

<https://github.com/NovatecConsulting/micronaut-camunda-bpm>

One more thing...

One more thing...

We have yet another
Micronaut Service in
Production!

“New Service” Production Figures

25K

Request per Minute

8

Nodes

100

ms avg latency

140

MB heap in average

Conclusion

- Micronaut is **production ready**.
- It **delivers** on its promise of **low memory** consumption and **fast** start-up times.
- **Quicker feedback** times → You get misconfiguration errors at compile time.
- **Response** time and quality from Micronaut team are great.
- **Easy to use** and understand as it uses a similar programming model as Spring Boot.

Thanks!