The Death of Microservices

Diego Pacheco

About me...

- ❑ Cat's Father
- ❑ Head of Software Architecture
- ❑ Agile Coach
- ❑ SOA/Microservices Expert
- ❑ DevOps Practitioner
- ❑ Author
- ❑ Speaker

diegopacheco

@diego_pacheco

http://diego-pacheco.blogspot.com.br/

tinyurl.com/diegopacheco

Diego Pacheco

Building Applications with Scala

Write modern, scalable, and reactive applications with the power of Scala

Packt>

Diego Pacheco

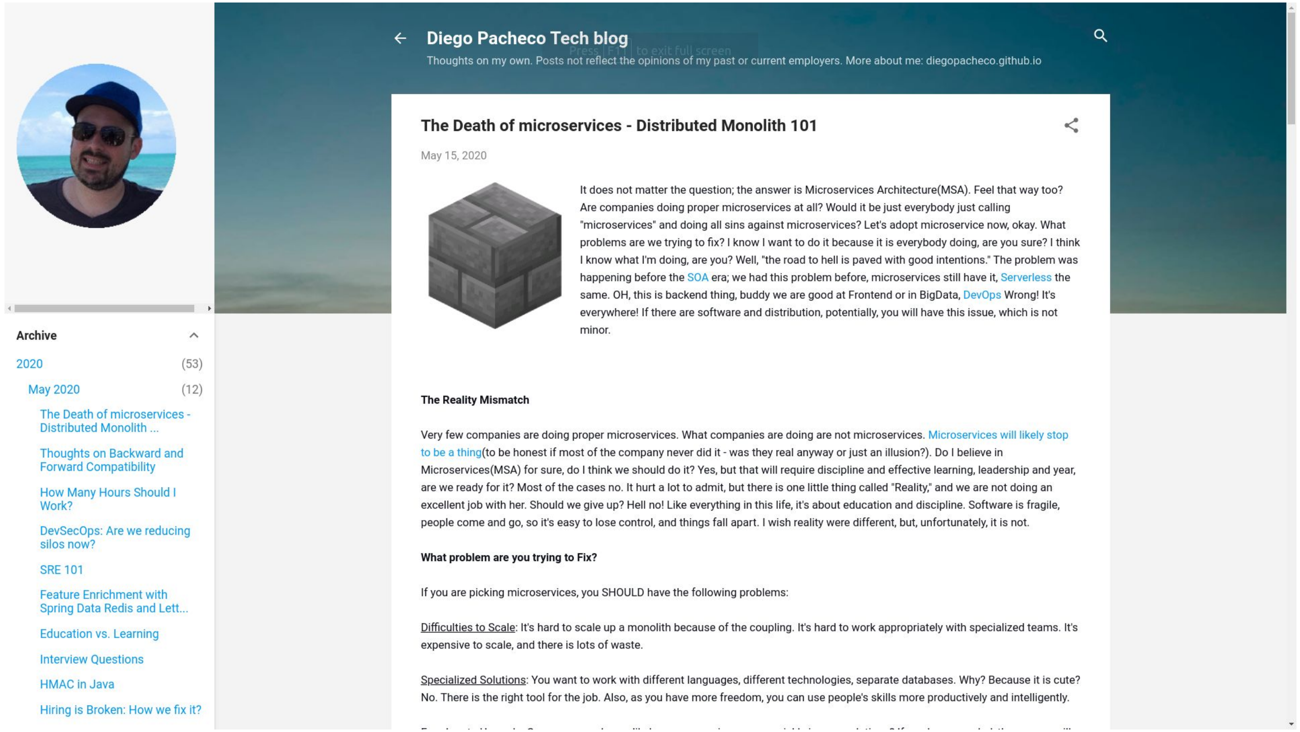
Building Effective Microservices

Explore microservices and their implementation hands-on

Packt>

https://diegopacheco.github.io/

Companion Blog Post...



Diego Pacheco Tech blog

Thoughts on my own. Posts not reflect the opinions of my past or current employers. More about me: diegopacheco.github.io

Archive

2020 (53)

May 2020 (12)

- The Death of microservices - Distributed Monolith ...
- Thoughts on Backward and Forward Compatibility
- How Many Hours Should I Work?
- DevSecOps: Are we reducing silos now?
- SRE 101
- Feature Enrichment with Spring Data Redis and Lett...
- Education vs. Learning
- Interview Questions
- HMAC in Java
- Hiring is Broken: How we fix it?

The Death of microservices - Distributed Monolith 101

May 15, 2020

It does not matter the question; the answer is Microservices Architecture(MSA). Feel that way too? Are companies doing proper microservices at all? Would it be just everybody just calling "microservices" and doing all sins against microservices? Let's adopt microservice now, okay. What problems are we trying to fix? I know I want to do it because it is everybody doing, are you sure? I think I know what I'm doing, are you? Well, "the road to hell is paved with good intentions." The problem was happening before the SOA era; we had this problem before, microservices still have it, Serverless the same. OH, this is backend thing, buddy we are good at Frontend or in BigData, DevOps Wrong! It's everywhere! If there are software and distribution, potentially, you will have this issue, which is not minor.

The Reality Mismatch

Very few companies are doing proper microservices. What companies are doing are not microservices. Microservices will likely stop to be a thing(to be honest if most of the company never did it - was they real anyway or just an illusion?). Do I believe in Microservices(MSA) for sure, do I think we should do it? Yes, but that will require discipline and effective learning, leadership and year, are we ready for it? Most of the cases no. It hurt a lot to admit, but there is one little thing called "Reality," and we are not doing an excellent job with her. Should we give up? Hell no! Like everything in this life, it's about education and discipline. Software is fragile, people come and go, so it's easy to lose control, and things fall apart. I wish reality were different, but, unfortunately, it is not.

What problem are you trying to Fix?

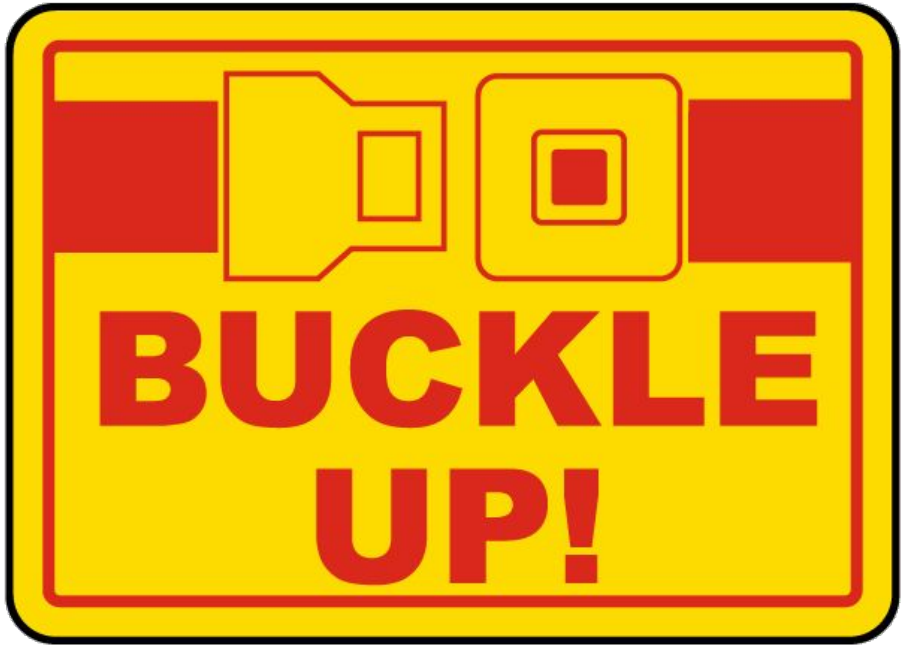
If you are picking microservices, you SHOULD have the following problems:

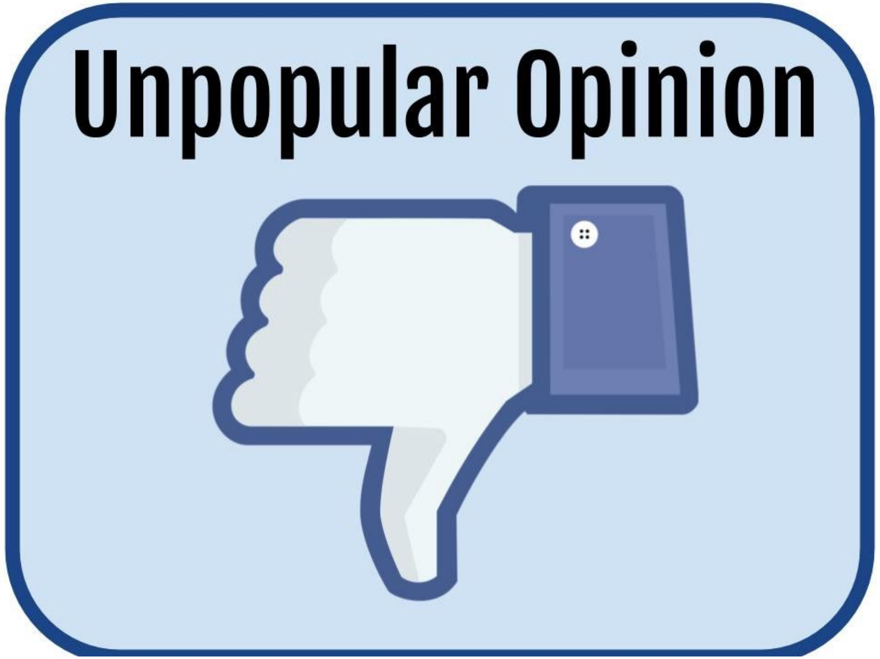
Difficulties to Scale: It's hard to scale up a monolith because of the coupling. It's hard to work appropriately with specialized teams. It's expensive to scale, and there is lots of waste.

Specialized Solutions: You want to work with different languages, different technologies, separate databases. Why? Because it is cute? No. There is the right tool for the job. Also, as you have more freedom, you can use people's skills more productively and intelligently.

http://diego-pacheco.blogspot.com/2020/05/the-death-of-microservices-distributed.html

Disclaimer



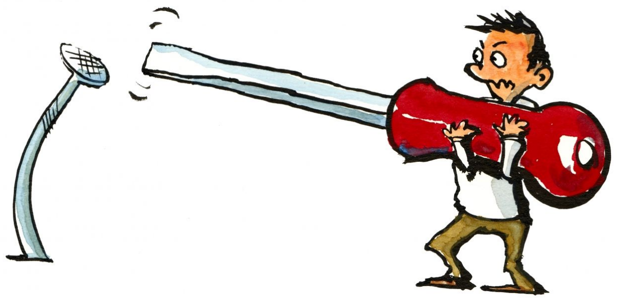


Reality Mismatch

- ❑ Do we have Proper Microservices?
- ❑ There is Database Isolation?
- ❑ Calling "Microservices" vs being?
- ❑ Can we easily upgrade libs?
- ❑ Is the Future about microservices?

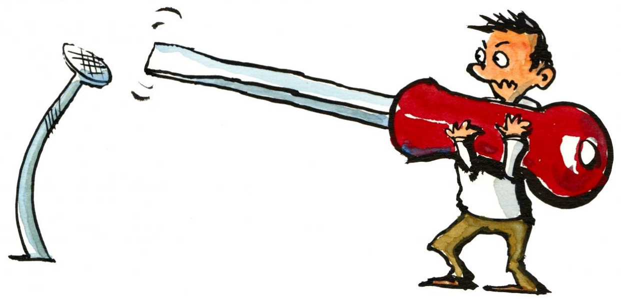
What Problem are we trying to fix?

- ❑ Difficulties to Scale?
- ❑ Specialized Solutions?
- ❑ Freedom to Upgrade?
- ❑ Better team Organization?



Often is...

- ❏ Social Pressure / "Envy"
- ❏ Sounds "easy" / looks "cool"
- ❏ Lack of knowing other options
- ❏ Agile lack of "Architecture" practices



Microservices Benefits

- ❑ Deploy services independently
- ❑ Scale services independently
- ❑ Technology Diversity
- ❑ Freedom to Upgrade services in isolation
- ❑ Better team organizations

Microservices drawbacks

- ❏ Infrastructure Complexity
- ❏ General Complexity (distributed systems)
- ❏ Duplication (not necessarily bad)

Microservices: Trade Offs

- ❑ Isolation has a price == duplication
- ❑ Eat the cake or have the cake.
- ❑ Distributed Monolith
  - ❑ By denial of the drawbacks
  - ❑ Lack of isolation

Code Duplication

- ❑ Bugs, Repetition -> Other approaches
- ❑ Is better than coupling
- ❑ When you need you upgrade you can't.
- ❑ It's fine to duplicate code!

Traps: How to build a distributed monolith 101

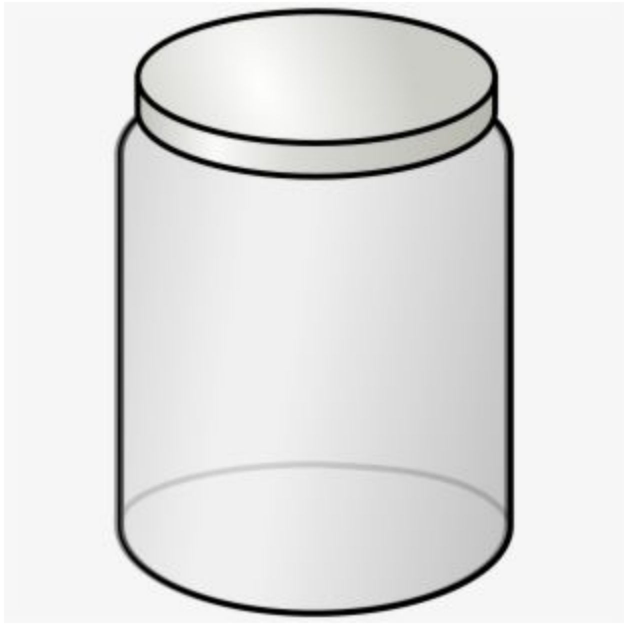
- ❑ Monorepo
- ❑ Shared-Libs
- ❑ Drivers
- ❑ Corporate Frameworks
- ❑ Tables

Traps - Monorepo

- ❑ Problem or Solution?
- ❑ It's the same thing or not? (angular)
- ❑ Do we have isolation or Not?
- ❑ Make easy to couple things.
- ❑ You are not Google (2B LoC)

Traps - Shared-libs

- ❏ Easiest form of reuse.
- ❏ Are evil! Why? 3rd party libs.
- ❏ 10s/100s exact libs to work
- ❏ Product vs Service company
- ❏ Distributed Monolith
- ❏ There is something worse!

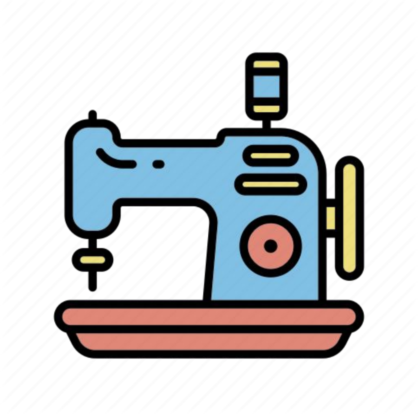


Traps - Drivers

- ❑ Clients / Network Drivers.
- ❑ Pure Coupling - Cant call the service.
- ❑ Bad deal: Optimization for Freedom.
- ❑ There is a even worse problem.

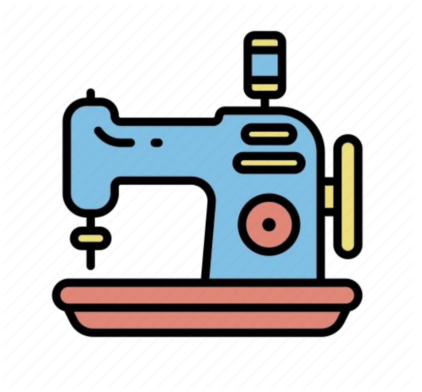
Traps - Corporate Frameworks

- ☐ Expensive and Complex to use
- ☐ Lack of documentation / samples
- ☐ Lack of good Abstractions (level 1/2)
- ☐ People hate it
  - ☐ Can't search in Stackoverflow
  - ☐ Can't use in other companies



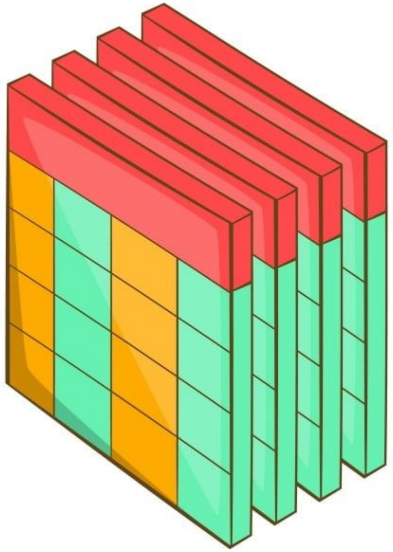
Traps - Frameworks as Libs

- ❑ Looks like a lib but it's a framework.
- ❑ Impose one specific model.
- ❑ Coupled to other libs.
- ❑ Hides to many things from you.



Traps - Tables

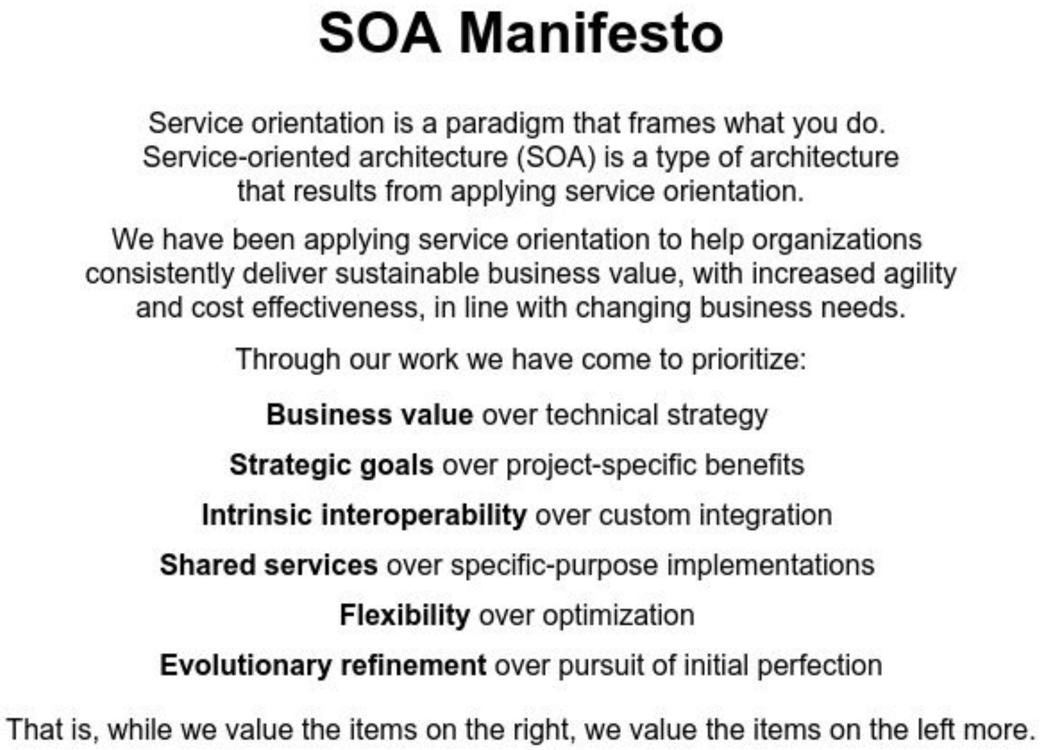
- ❑ Contract by accident
- ❑ All Datastores
- ❑ All Caches(i.g IMOC like Redis)
- ❑ Configurations (xml,json,yaml)



How we fix it?

- ☐ Back to the basics
- ☐ Duplication
- ☐ Packages
- ☐ Isolation
- ☐ Platforms
- ☐ Education

Back to the basics == SOA



SOA Manifesto

Service orientation is a paradigm that frames what you do.
Service-oriented architecture (SOA) is a type of architecture
that results from applying service orientation.

We have been applying service orientation to help organizations
consistently deliver sustainable business value, with increased agility
and cost effectiveness, in line with changing business needs.

Through our work we have come to prioritize:

Business value over technical strategy

Strategic goals over project-specific benefits

Intrinsic interoperability over custom integration

Shared services over specific-purpose implementations

Flexibility over optimization

Evolutionary refinement over pursuit of initial perfection

That is, while we value the items on the right, we value the items on the left more.

http://www.soa-manifesto.org/

Don't accept deps from strangers

- ❑ Protect your classpath.
- ❑ Stop using shared-libs.
- ❑ NetflixOss / Spring Boot (In vs Out)
- ❑ Same capability: Let's reuse!
  - ❑ Bad idea.
  - ❑ Famous last words!

Packages are your friends

- ❑ Provide Isolation
- ❑ FAT Jars
- ❑ Package Explosion Tools
- ❑ Memory increase
- ❑ Sometimes: just copy the code.

Isolate everything

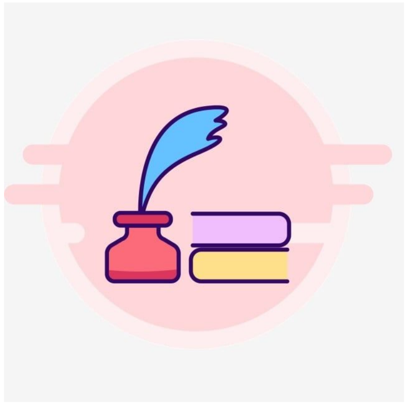
- ❑ Datastores
- ❑ Caches
- ❑ Configs
- ❑ Implementation
- ❑ Backward/Forward at Contract level

Platforms

- ❑ Tooling
- ❑ Old days of Spring framework
- ❑ Self-service automated Service
- ❑ Sidecars. .i.g: Lyft Envoy.
- ❑ Runtime Platforms: Kubernetes

Education

- ❑ SOA
- ❑ Training
- ❑ Accepting trade-offs (duplication)
- ❑ Continuous Effort.



Thank you!

The Death of Microservices

Diego Pacheco