



Реальная альтернатива микросервисам

15 марта
10:30 MSK

Курсы по ИТ-архитектуре

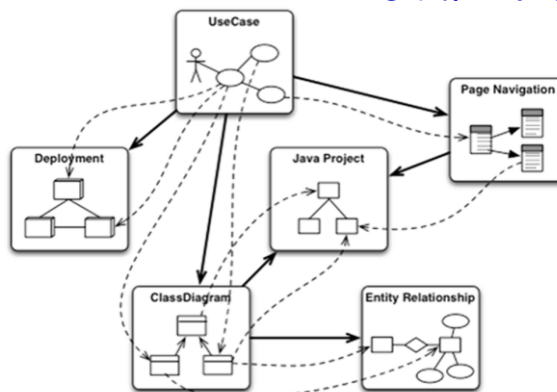
Микросервисная Архитектура онлайн



8-11 апреля

Software architecture

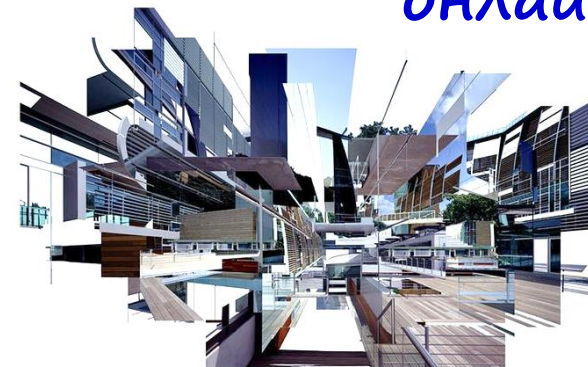
Мастерская Проектирования ИТ-решений онлайн



25-28 марта

Solution architecture

Модели корпоративной архитектуры онлайн



22-25 апреля

Enterprise architecture

Microservices

a definition of this new architectural term



James Lewis



Martin Fowler

In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

<https://martinfowler.com/articles/microservices.html>

- 1. Когда же наконец это закончится?**
- 2. Что будет после микросервисной архитектуры?**

О чём пойдет речь

1. Давайте посмотрим на микросервисы с другой стороны (альтернативное определение)
2. Замечание об архитектурных стилях
3. Перечислим некоторые альтернативы микросервисной архитектуре
4. Почему актуально обсуждение альтернатив
5. Реальная альтернатива микросервисам
6. Ответы на вопросы



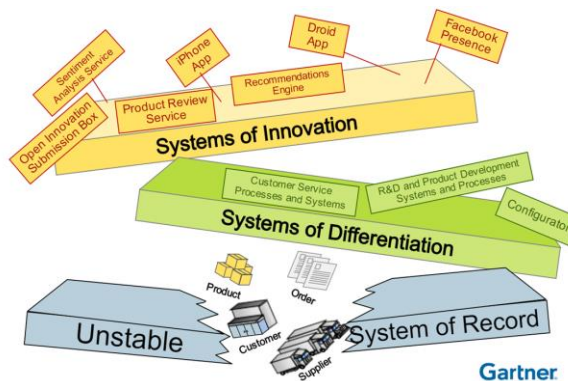
Три шага к микросервисам

1. Потребность в масштабировании



Стартапам и корпорациям потребовались распределенные вычисления (вслед за продуктовыми интернет-компаниями)

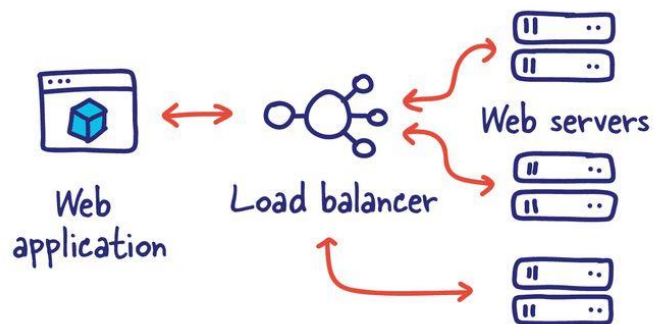
2. Межпроцессные взаимодействия



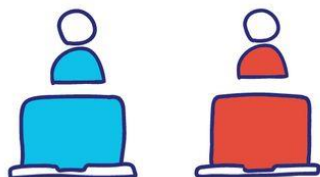
Оказалось, что не все сетевые взаимодействия одинаково полезны (8 заблуждений распределенных вычислений)

8 FALLACIES & DISTRIBUTED SYSTEMS

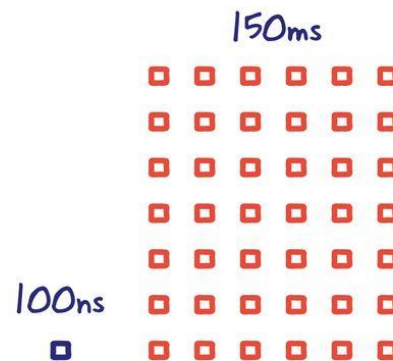
1 The network is reliable.



6 There is one administrator.



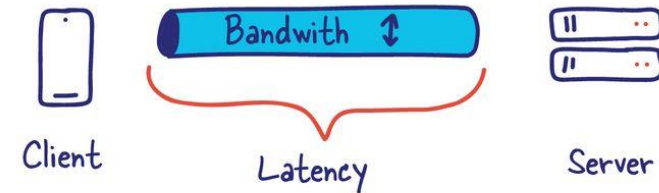
2 Latency is zero.



Transport cost is zero.



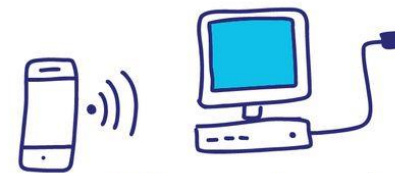
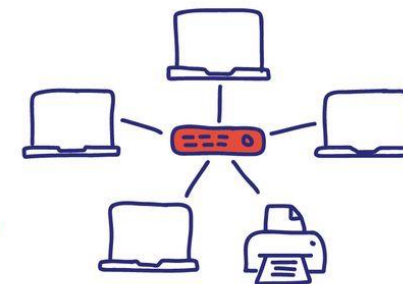
3 Bandwidth is infinite.



4 The network is secure.



5 Topology doesn't change.



8 The network is homogeneous.



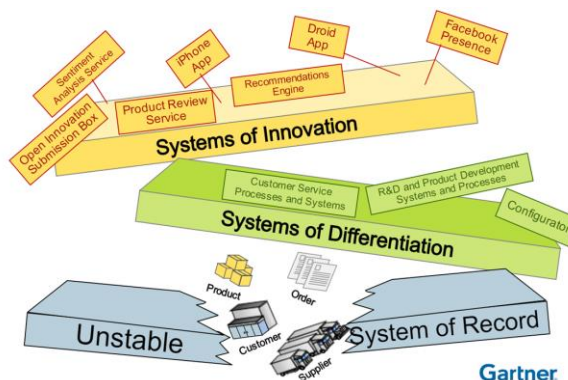
Три шага к микросервисам

1. Потребность в масштабировании



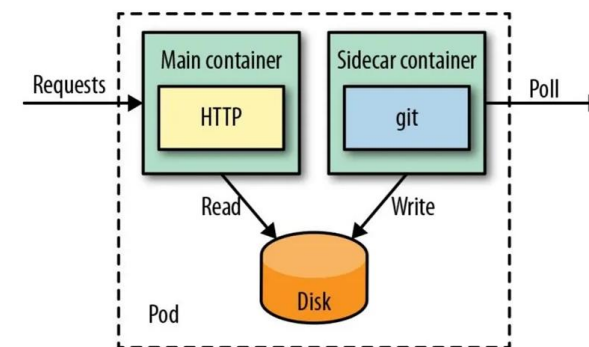
Стартапам и корпорациям потребовались распределенные вычисления (вслед за продуктовыми интернет-компаниями)

2. Межпроцессные взаимодействия



Оказалось, что не все сетевые взаимодействия одинаково полезны (8 заблуждений распределенных вычислений)

3. Ограничения взаимодействия



Оставить в сети взаим., которые устойчивы к сбоям (stateless запросы, асинхронная обработка команд и событий)

								
	layered	modular monolith	microkernel	microservices	service-based	service-oriented	event-driven	space-based
agility	★	★★	★★★	★★★★★	★★★★★	★	★★★★	★★
abstraction	★	★	★★★	★	★	★★★★★	★★★★	★
configurability	★	★	★★★★	★★★	★★	★	★★	★★
cost	★★★★★	★★★★★	★★★★★	★	★★★★	★	★★★	★★
deployability	★	★★	★★★	★★★★★	★★★★★	★	★★★	★★★★
domain part.	★	★★★★★	★★★★★	★★★★★	★★★★★	★	★	★★★★★
elasticity	★	★	★	★★★★★	★★	★★★	★★★★	★★★★★
evolvability	★	★	★★★	★★★★★	★★★	★	★★★★★	★★★★
fault-tolerance	★	★	★	★★★★★	★★★★	★★★	★★★★★	★★★★
integration	★	★	★★★	★★★	★★	★★★★★	★★★★	★★
interoperability	★	★	★★★	★★★	★★	★★★★★	★★★★	★★
performance	★★★	★★★	★★★	★★	★★★	★★	★★★★★	★★★★★
scalability	★	★	★	★★★★★	★★★	★★★	★★★★★	★★★★★
simplicity	★★★★★	★★★★★	★★★★	★	★★★	★	★	★
testability	★★	★★	★★★	★★★★★	★★★★	★	★★	★
workflow	★	★	★★	★	★	★★★★★	★★★★★	★

Как понимать термин architecture styles





UNIVERSITY OF CALIFORNIA, IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

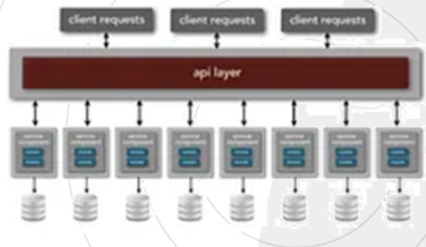
in Information Systems

An **architectural style** is a coordinated set of architectural **constraints** that restricts the roles/features of architectural elements and the allowed relationships among those elements...

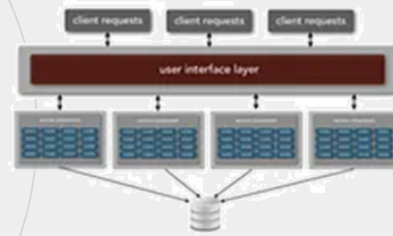
Chapter 1.5

<https://www.ics.uci.edu/~tielking/publications>

Software Architecture Patterns 2017



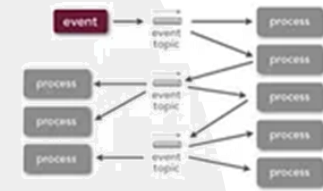
microservices
architecture



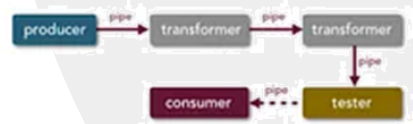
service-based
architecture



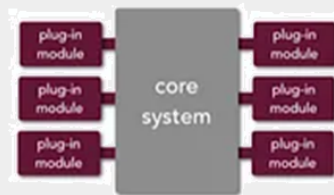
layered
architecture



event-driven
architecture



pipeline
architecture



microkernel
architecture



space-based
architecture

APPSIGNAL

The Citadel Architecture at AppSignal



Thijs Cadier on Apr 8, 2020

<https://blog.appsignal.com/2020/04/08/the-citadel-architecture-at-appsignal.html>



DHH  
@dhh · [Follow](#)



Replying to @dhh

In addition to the Majestic Monolith, someone should write up the pattern of The Citadel: A single Majestic Monolith captures the majority mass of the app, with a few auxiliary outpost apps for highly specialized and divergent needs.

4:51 PM · Apr 7, 2020



211



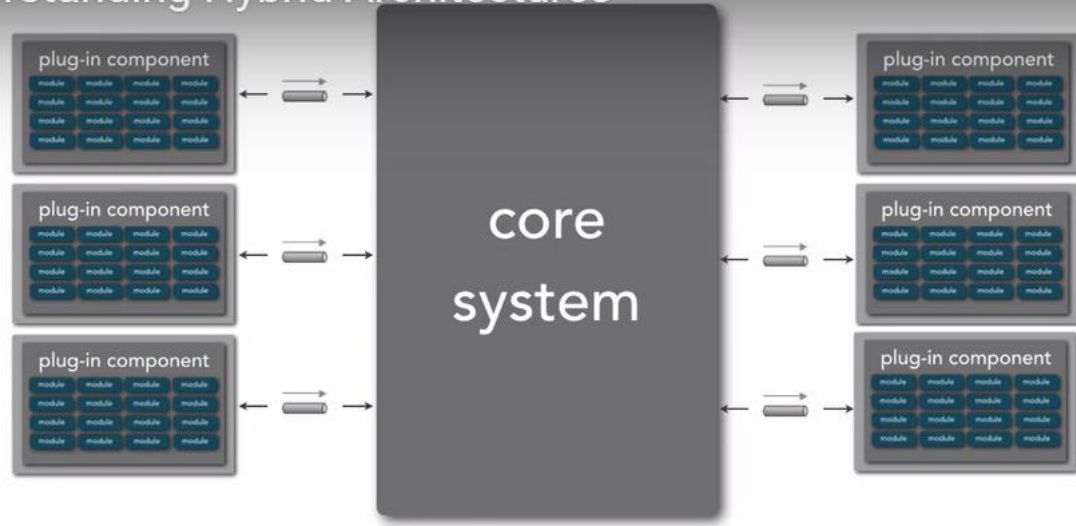
Reply



Copy link

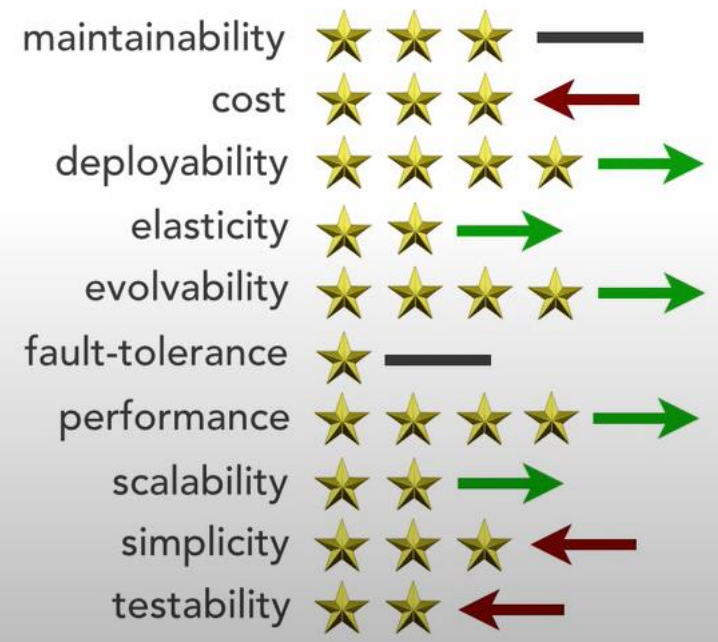
[Read 4 replies](#)

Lesson 92 - Understanding Hybrid Architectures



event-driven microkernel

- + deployability
- + elasticity
- + evolvability
- + performance
- + scalability
- cost
- simplicity
- testability



Какие есть альтернативы

МОНОЛИТ 29

МОДУЛЬНЫЙ 12

SOA 9

ACTORS 3 ГИБРИД 3

SBA 3

EDA 2

ОТВЕТИЛИ 64 из 221

Modular Monolith: Is This the Trend in Software Architecture?

Ruoyu Su
M3S, University of Oulu
Oulu, Finland
ruoyu.su@oulu.fi

Xiaozhou Li
M3S, University of Oulu
Oulu, Finland
xiaozhou.li@oulu.fi



*Bozlanmıca
& monar mı?*



Modular Monolith is a **software architecture pattern** that strategically combines the simplicity of a monolithic structure with the advantages of microservices...

This **architectural style** seeks to strike a middle ground between traditional monolithic and microservices architectures. It emphasizes the interchangeability and potential reusability of modules, promoting a clear programming interface between them

<https://arxiv.org/pdf/2401.11867.pdf>

Google Service Weaver

Introducing Service Weaver: A Framework for Writing Distributed Applications

March 1, 2023

Towards Modern Development
of Cloud Applications

Sanjay Ghemawat, Robert

Grandl, Srdjan Petrovic,

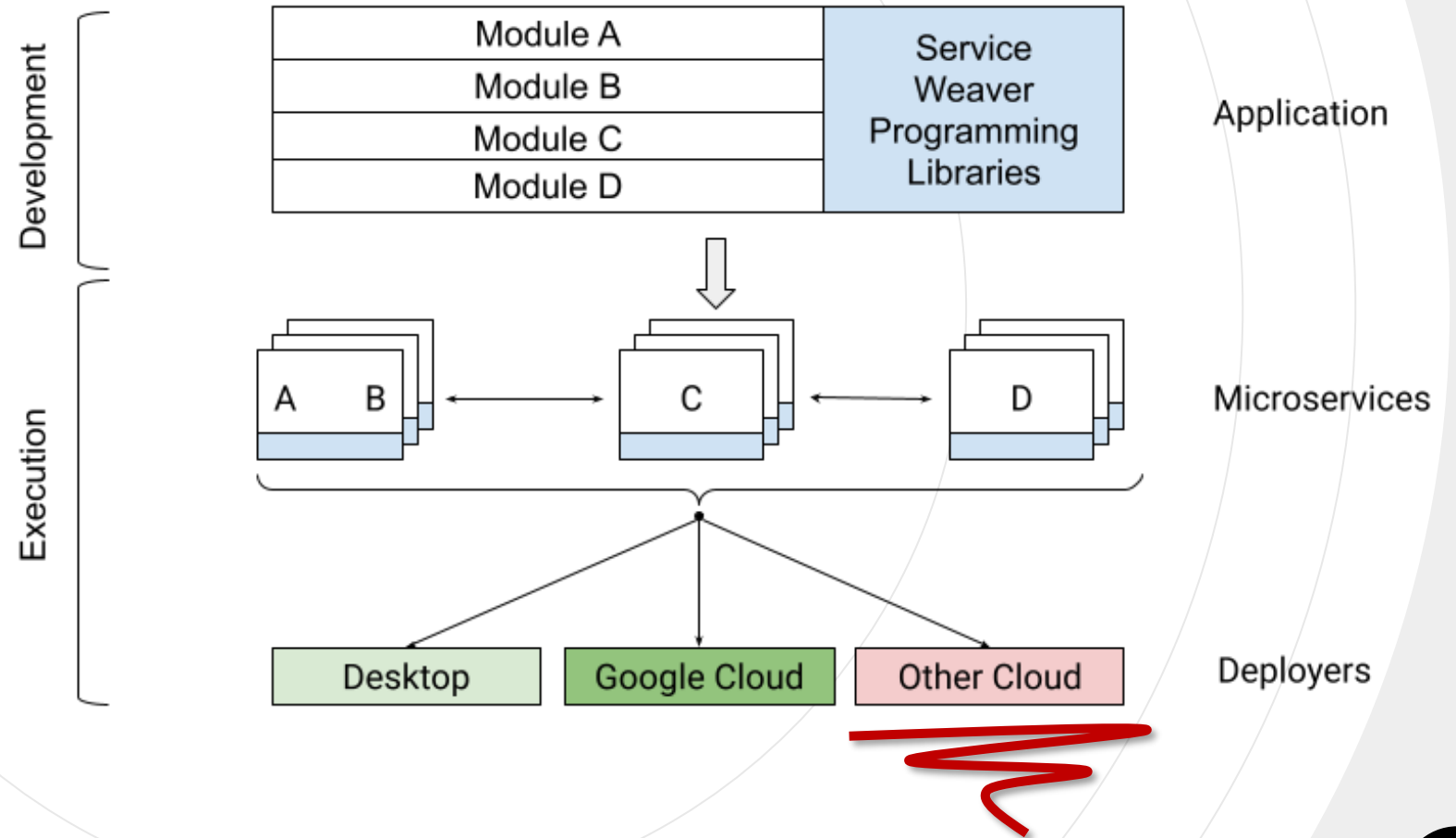
Michael Whittaker Parveen

Patel, Ivan Posva, Amin Vahdat

Google

<https://doi.org/10.1145/3593>

[856.3595909](https://doi.org/10.1145/3593)



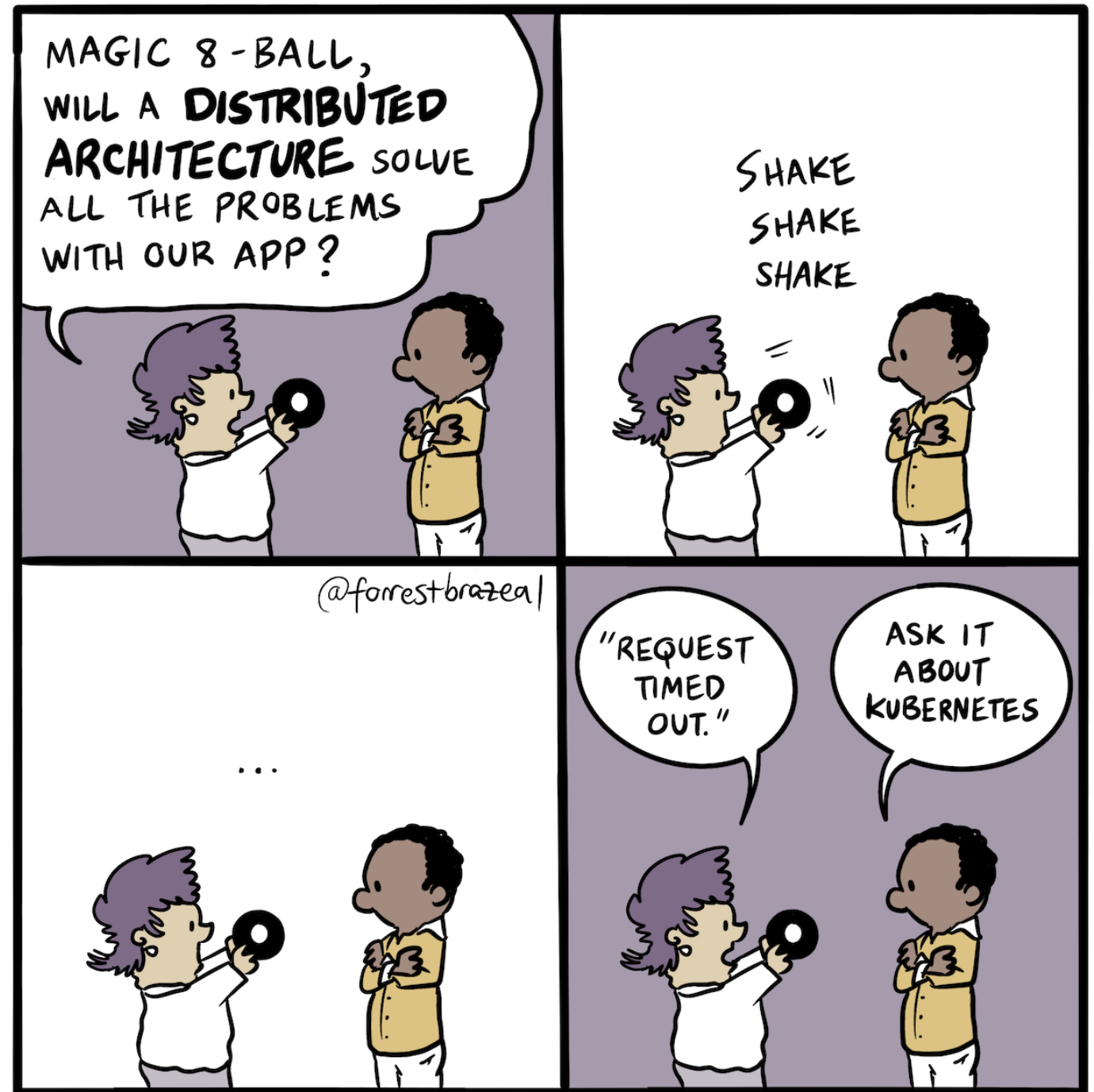
Calm down about Service Weaver

It isn't a silver bullet, and it's not claiming to be

FORREST BRAZEAL
3 MAR. 2023 Г.

[https://newsletter.goodtechthings.com/
p/calm-down-about-service-weaver](https://newsletter.goodtechthings.com/p/calm-down-about-service-weaver)

Good Tech Things



Calm down about Service Weaver

It isn't a silver bullet, and it's not claiming to be

FORREST BRAZEAL
3 MAR. 2023 Γ.

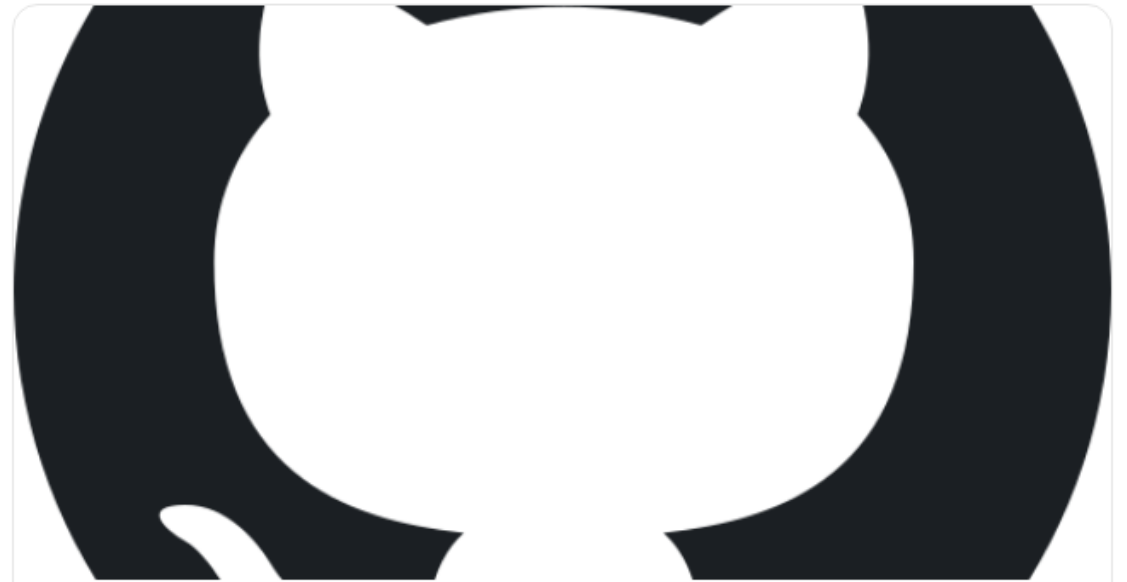
[https://newsletter.goodtechthings.com/
p/calm-down-about-service-weaver](https://newsletter.goodtechthings.com/p/calm-down-about-service-weaver)

Good Tech Things



Martin Kleppmann @martin@nondeterministic.computer
@martinkl

Google's Service Weaver claims that it can hide the difference between RPC and local method calls. Just like CORBA claimed in 1991. Is it different this time? The docs say barely a word about handling failures, such as RPCs timing out



serviceweaver.dev
Service Weaver

10:29 AM · Mar 3, 2023

266 Likes 34 Retweets

<https://twitter.com/martinkl/status/1631602767524495363>

There are two reasons why these microservice-based solutions fall short of solving challenges C1-C5.

The first reason is that they all assume that the developer manually splits their application into multiple binaries. This means that the network layout of the application is predetermined by the application developer.

Moreover, once made, the network layout becomes hardened by the addition of networking code into the application (e.g., network endpoints, client/server stubs, network-optimized data structures like [30]). This means that it becomes harder to undo or modify the splits, even when it makes sense to do so

The second reason is the assumption that application binaries are individually (and in some cases continually) released into production. This makes it more difficult to make changes to the cross-binary protocol. Additionally, it introduces versioning issues and forces the use of more inefficient data formats like [23, 30]

<https://doi.org/10.1145/3593856.3595909>

Infrastructure ~~as~~ From Code



Infrastructure-from-Code (IfC) is a process that analyzes your application code to infer the cloud resources you need, and then creates and maintains them without you having to manually define them

Programming Languages

- Wing
- DarkLang

Облачные языки программирования с новыми конструкциями (очереди, хранилища, функц.)

Annotations + Frameworks

- Encore Различные способы
- Shuttle объявления ресурс.

Pure Annotations

- Klotho Только аннотации

SDK-Based

- Ampt
- Nitric

Для создания инфраструктуры анализируются вызовы конкретных API

<https://klo.dev/state-of-infrastructure-from-code-2023/>

	A	B	C	D	E	G	I	J	K	L	M
1		Language Support	Technologies Used	Compute Backing Tech	Hosting Model	Developer	Low Level Config	Out-of-the-box Cloud Support	IaC Support	Project Age	Github Star Count (If Applicable)
2	Service Weaver (Google)	Go	Pre-selected	Kubernetes	Self-Hosted	SDK		Google Cloud	✓	4 months	3000
3	Caio	Python	Pre-selected	Serverless Kubernetes	Self-hosted	Standard/OSS interfaces	High-level	AWS	✓	3 years	N/A
4	Modal	Python	Proprietary	Proprietary	PaaS	Annotations		None	✗	2 Years	100
5	Klotho	JS TS Go Python C#	Adaptive	Serverless Kubernetes	Self-Hosted	Standard/OSS Frameworks + Annotations	High Level + Low Level using IaC	AWS	✓	3 years	610
6	Encore	Go	Pre-selected	Serverless	PaaS (Full Featured) Self-Hosted (Limited)	SDK + Annotations		AWS, GCP, Azu	✗	2 years	3600
7	Nitric	JS TS Go Python Java C#	Adaptive	Serverless	PaaS Self-Hosted	SDK		AWS, GCP, Azu	✓	2.5 Years	498
8	Chalice	Python	Explicit	Serverless	Self-Hosted	Annotations		AWS	✗	7 Years	9600
9	Wing	Wing IPM/JS/TS intero	Adaptive	Serverless	Self-Hosted	Language	High-level + Config plugins	WS, Azure, GC	✓	1 year	282
10	Shuttle	Rust	Pre-selected	Serverless	PaaS	SDK + Annotations	N/A	AWS	✗	1 Year	2000