



Microservices

Kari Systä, 12.10.2021

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**.

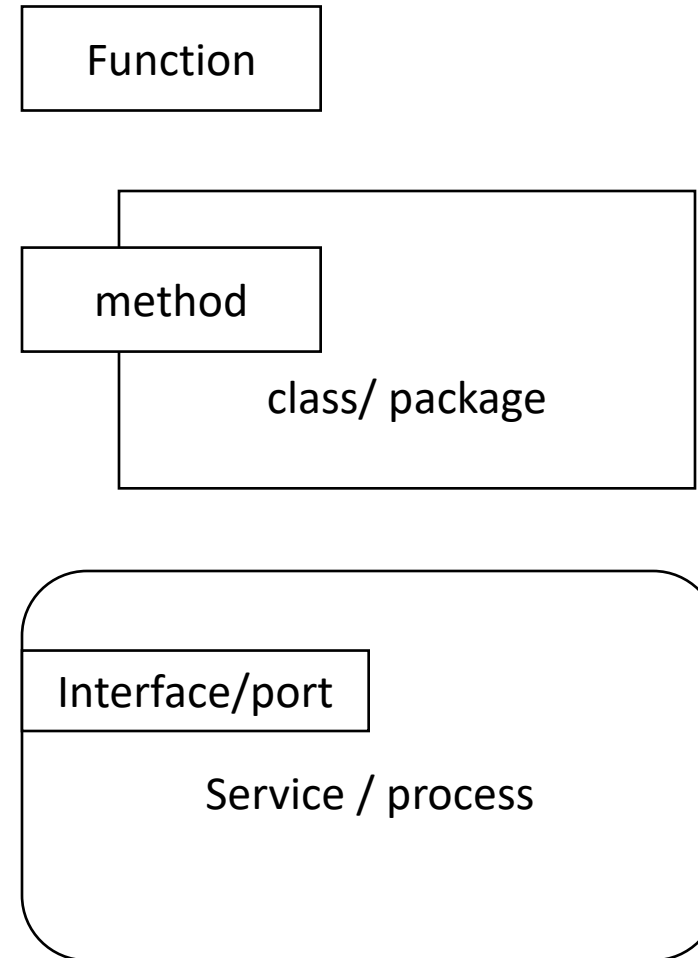
There is a **bare minimum of centralized management** of these services, which may be written in different programming languages and use different data storage technologies.

I. Nadareishvili et al., *Microservice Architecture: Aligning Principles, Practices, and Culture*, O'Reilly, 2016.

- small
- messaging enabled,
- bounded by contexts,
- **autonomously developed**,
- independently deployable,
- decentralized, and
- built and released with automated processes.

Microservices

- Modular and decomposed ?
- Service-oriented ?
- Distributed ?
- Message-oriented ?
- Independently developed ?
- Independently deployed ?



What do you remember from the “Software Engineering Methodologies” course?

- Properties of an Agile (e.g. Scrum) team?
- Self-organizing ?
- Cross-functional ?
- Co-located ?
- 7 +/- 2/4 ?

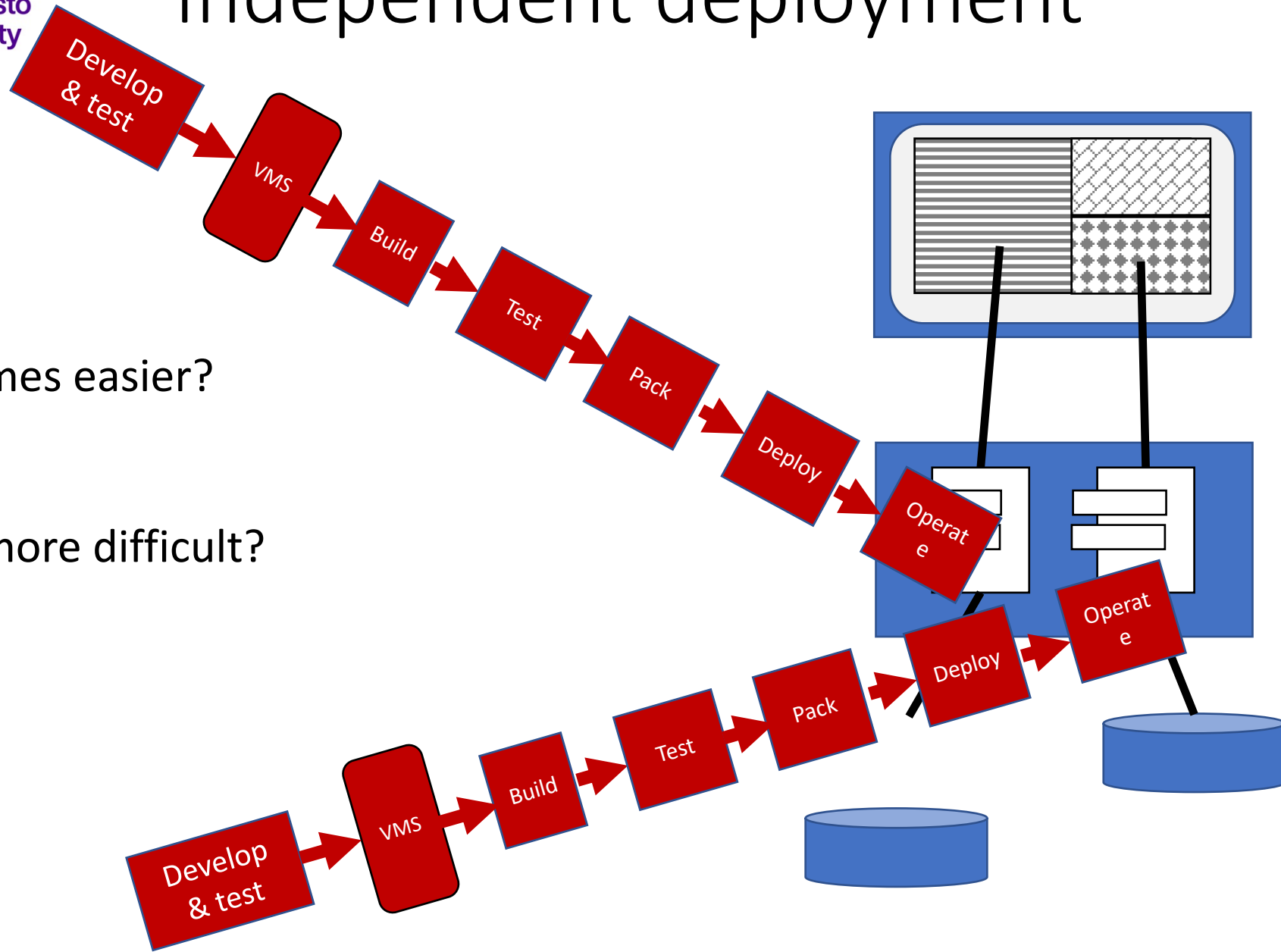
Independent development

- Separate team
 - Two pizza rule
 - Scrum propose 7 ± 2
 - Small team is more efficient
- Independently selected
 - Run-time
 - Libraries
 - Programming language
- Or, arrogant developers want to use their own favorite?

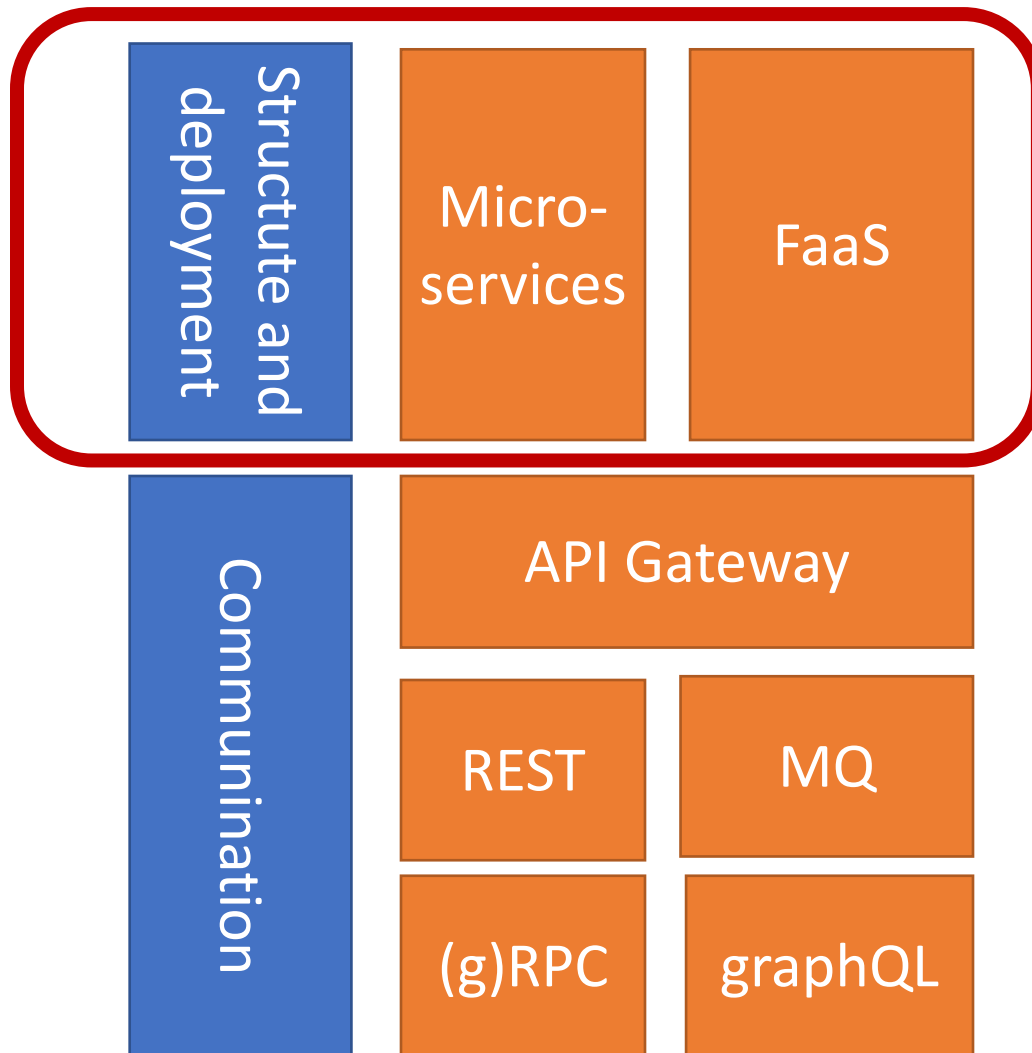
Independent deployment

What becomes easier?

What gets more difficult?



More about cloud-native architectures



Build around business capabilities?

- A way to split monolith to micro services
- Traditional SOA way is to look at static or dynamic dependencies
 - And minimize inter-service calls
- Should support independent development:
 - *Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.*
 - -- Melvyn Conway, 1967
- Should support independent deployment
 - Testing
 - Timing

Faculty of Information Technology and Communication Sciences > ... > TIE-23536 > plusa-syky2019 > Pipelines

All **91** Pending **0** Running **0** Finished **91** Branches Tags

Run Pipeline

Status Pipeline Triggerer Commit Stages

	#10909 latest		release 4a643309 Saved modified emacs b...		00:01:02 3 days ago	
	#10908 latest		master 4a643309 Saved modified emacs b...		00:01:02 3 days ago	
	#10907		master 4e1301f7 fixed folder name in root ...		00:01:05 3 days ago	
	#8363		release a5954f38 Push deadline		00:00:57 2 weeks ago	
	#8362		release bd544248		00:00:56	

Kari Systä
@systa

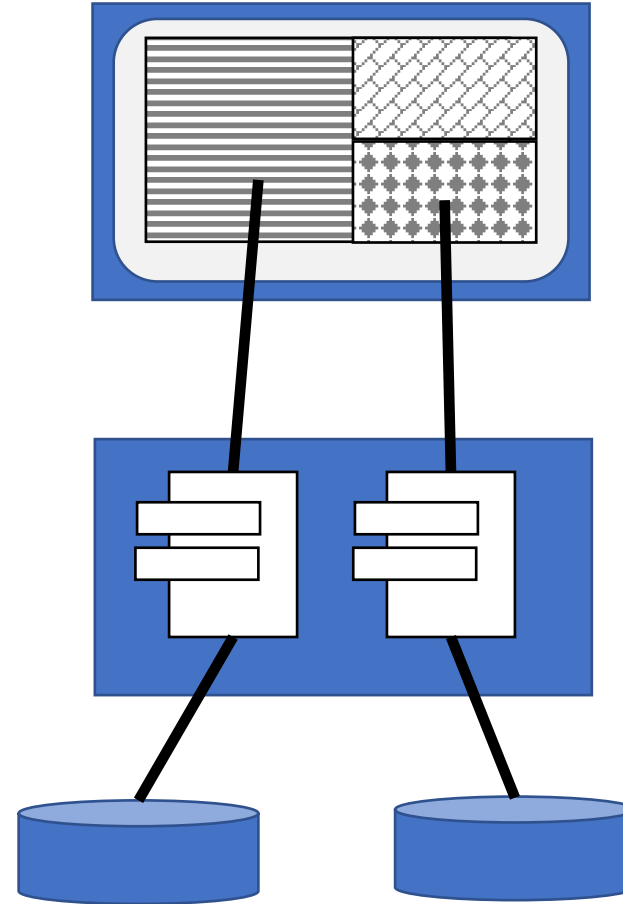
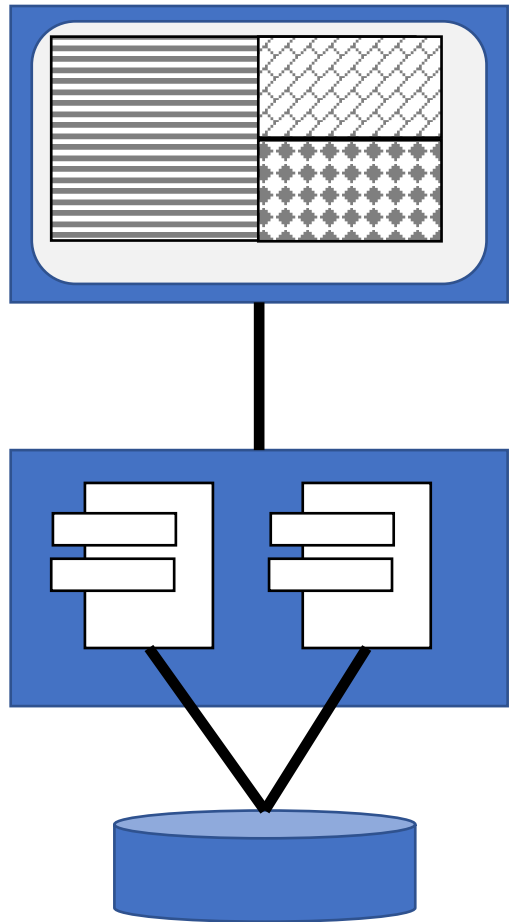
Set status

Profile

Settings

Sign out

Build around business capabilities?



Faculty of Information Technology and Communication Sciences > ... > TIE-23536 > plusa-syky2019 > Pipelines

All **91** Pending **0** Running **0** Finished **91** Branches Tags

Run Pipeline

Status Pipeline Triggerer Commit Stages

	#10909 latest		release 4a643309 Saved modified emacs b...		00:01:02 3 days ago	
	#10908 latest		master 4a643309 Saved modified emacs b...		00:01:02 3 days ago	
	#10907		master 4e1301f7 fixed folder name in root ...		00:01:05 3 days ago	
	#8363		release a5954f38 Push deadline		00:00:57 2 weeks ago	
	#8362		release bd544248		00:00:56	

Kari Systä
@systa

Set status

Profile

Settings

Sign out

Homework

- Read article “The Monolith Strikes Back: Why Istio Migrated From Microservices to a Monolithic Architecture”
<https://ieeexplore.ieee.org/document/9520758>
(access requires VPN connection to TUNI)
- And prepare a list of microservice drawbacks you find from the article

Table 1. Sam Newman's recommendations on when not to use microservices and whether they apply to Istio.

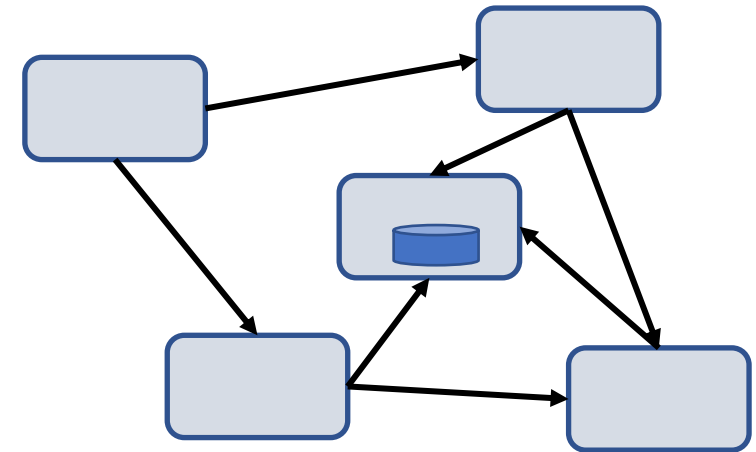
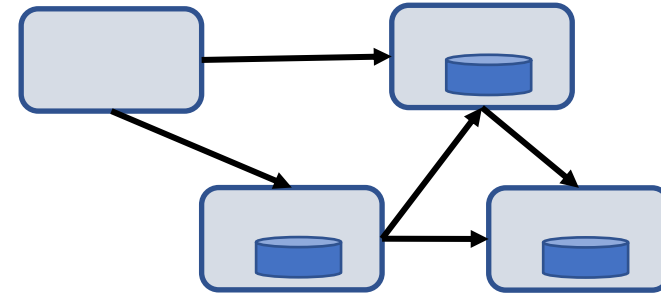
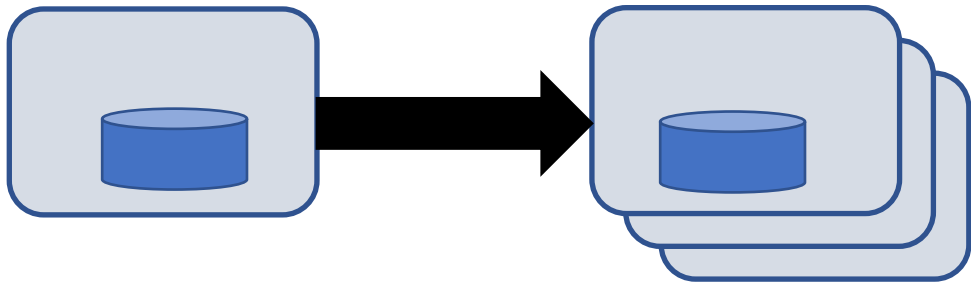
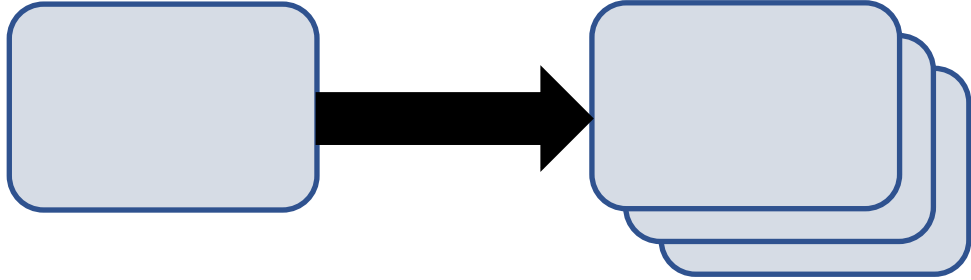
Situation	Why Microservices Are Bad	Does It Apply to Istio?
Unclear domain	Getting service boundaries wrong can be expensive.	In part. From the perspective of fault and security isolation, it can be argued that splitting the Istio control plane into multiple independent services was an unnecessary decision.
Start-ups	A start-up needs to focus all its attention on finding the right fit for its product. Microservices primarily solve the sorts of problems start-ups have once they've found that fit with their customer base.	In part. While Istio was initially designed by mature organizations, it was run like a start-up and did indeed need to focus on finding the right fit. As it turned out, microservices were solving a problem Istio didn't actually have.
Customer-installed and managed software	Microservices push a lot of complexity into the operational domain. Coping with this complexity isn't something you can typically expect of your end customers.	Yes. Negative user feedback on the complexity of deploying and managing Istio was the main reason for the team's decision to consolidate the control plane microservices into a single binary.
Not having a good reason!	Do not adopt microservices if you don't have a clear idea of what exactly it is that you're trying to achieve.	In part. Although the Istio team had a clear view of the benefits and cost of microservices, they didn't realize right from the start that, in their case, the costs would outweigh the benefits.

Issues with microservices

- Decoupling from the monolithic system
- Database migration and data splitting
- Communication among services
- Service orchestration complexity

- <https://microservices.io/patterns/apigateway.html>
- <https://whatis.techtarget.com/definition/API-gateway-application-programming-interface-gateway>

Stateful vs stateless



From Twitter



Jack Kleeman @JackKleeman · Nov 1

1500 microservices at @monzo; every line is an enforced network rule allowing traffic

[Show this thread](#)

