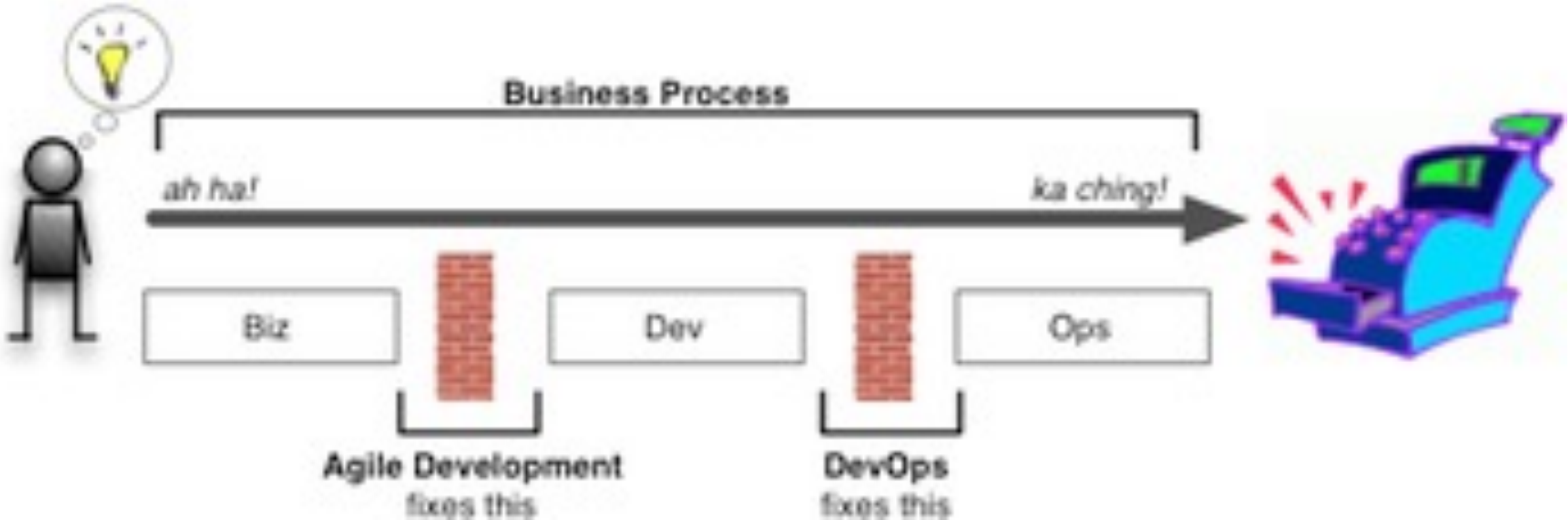
Tampere University

COMP.SE.140 Recap 05.12.2023

Kari Systä

What is DevOps

The lifecycle



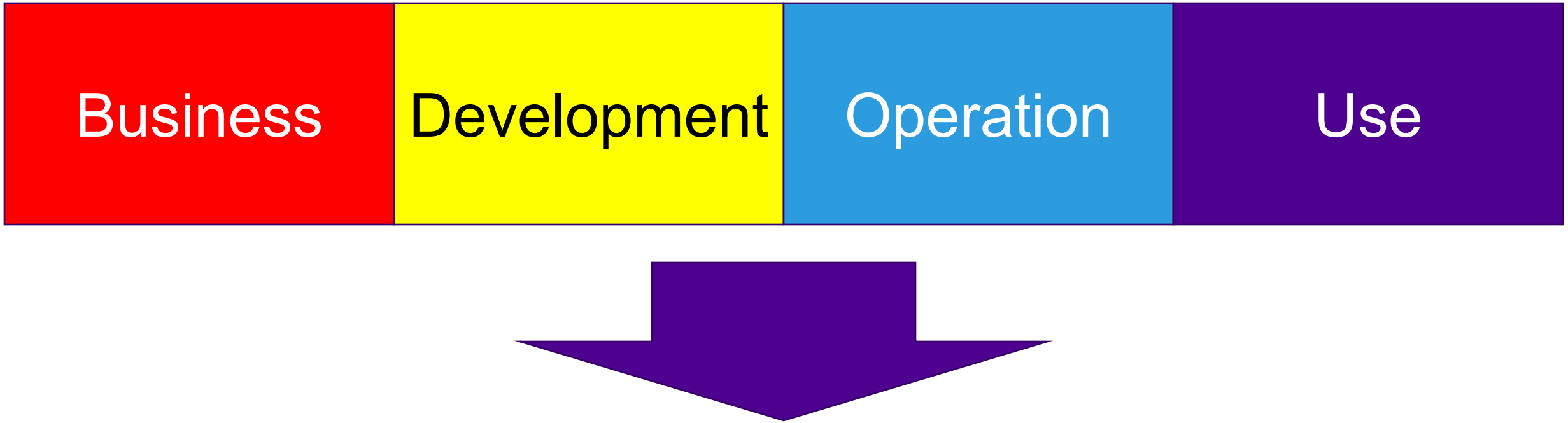
DevOps practices

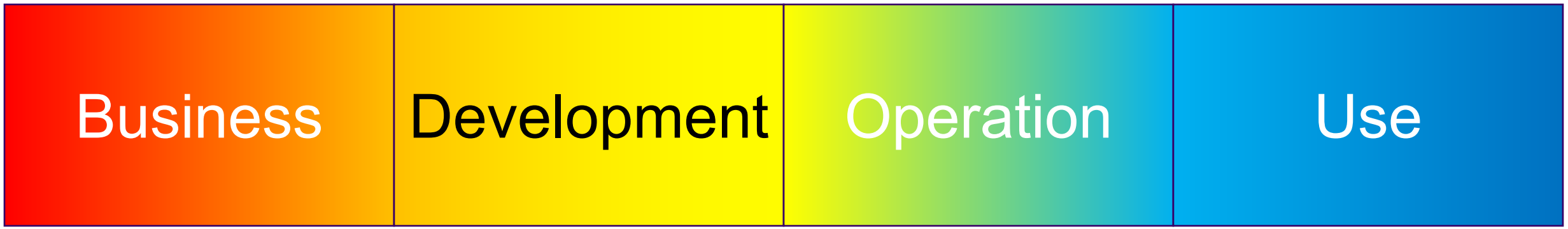
- Organizational
  - increased scope of responsibilities for developers;
  - intensified cooperation between development and operations.

- Technical
  - automation,
  - monitoring
  - measurement

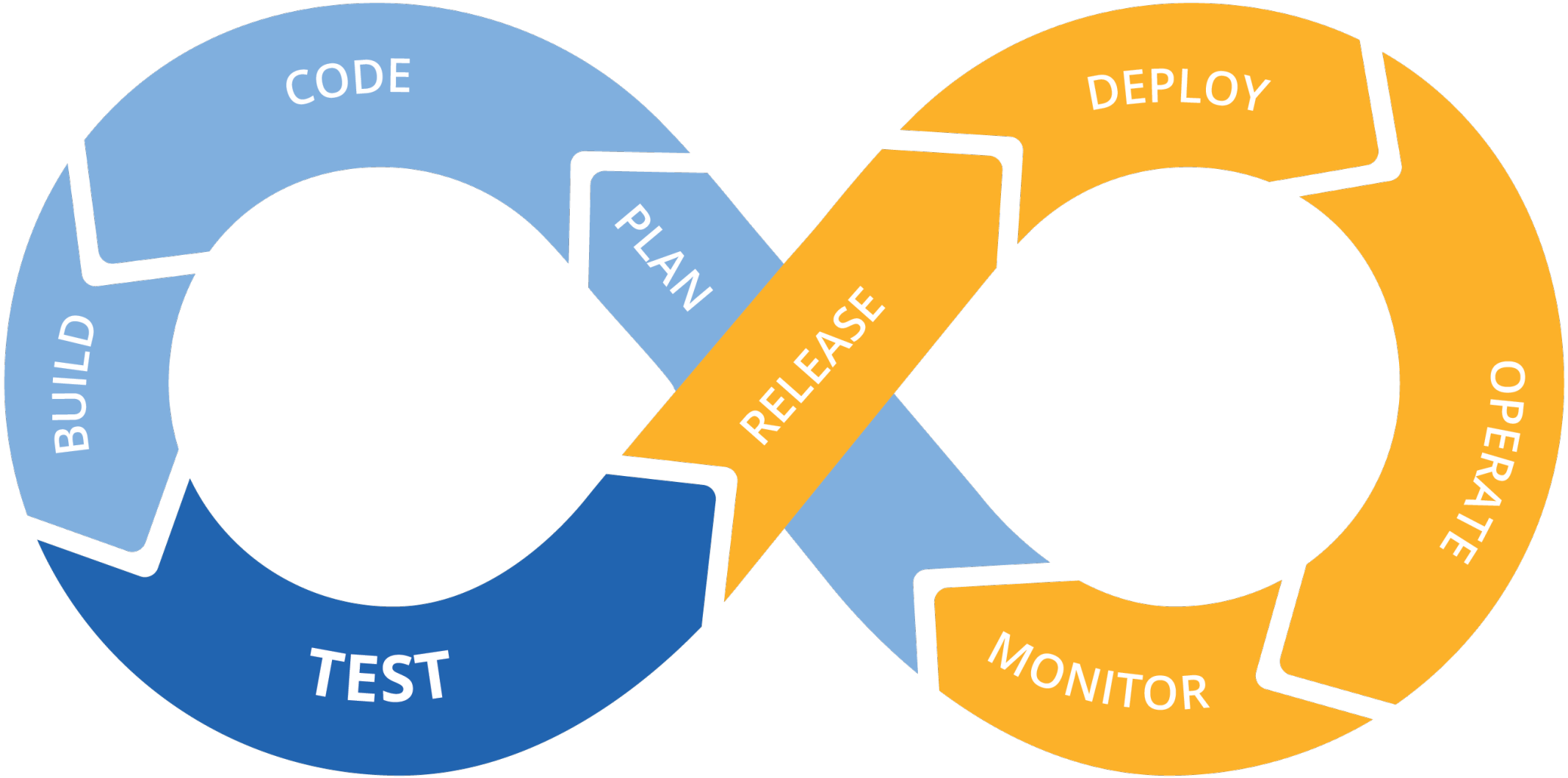
Tampere University

Where was the beef?





DevOps

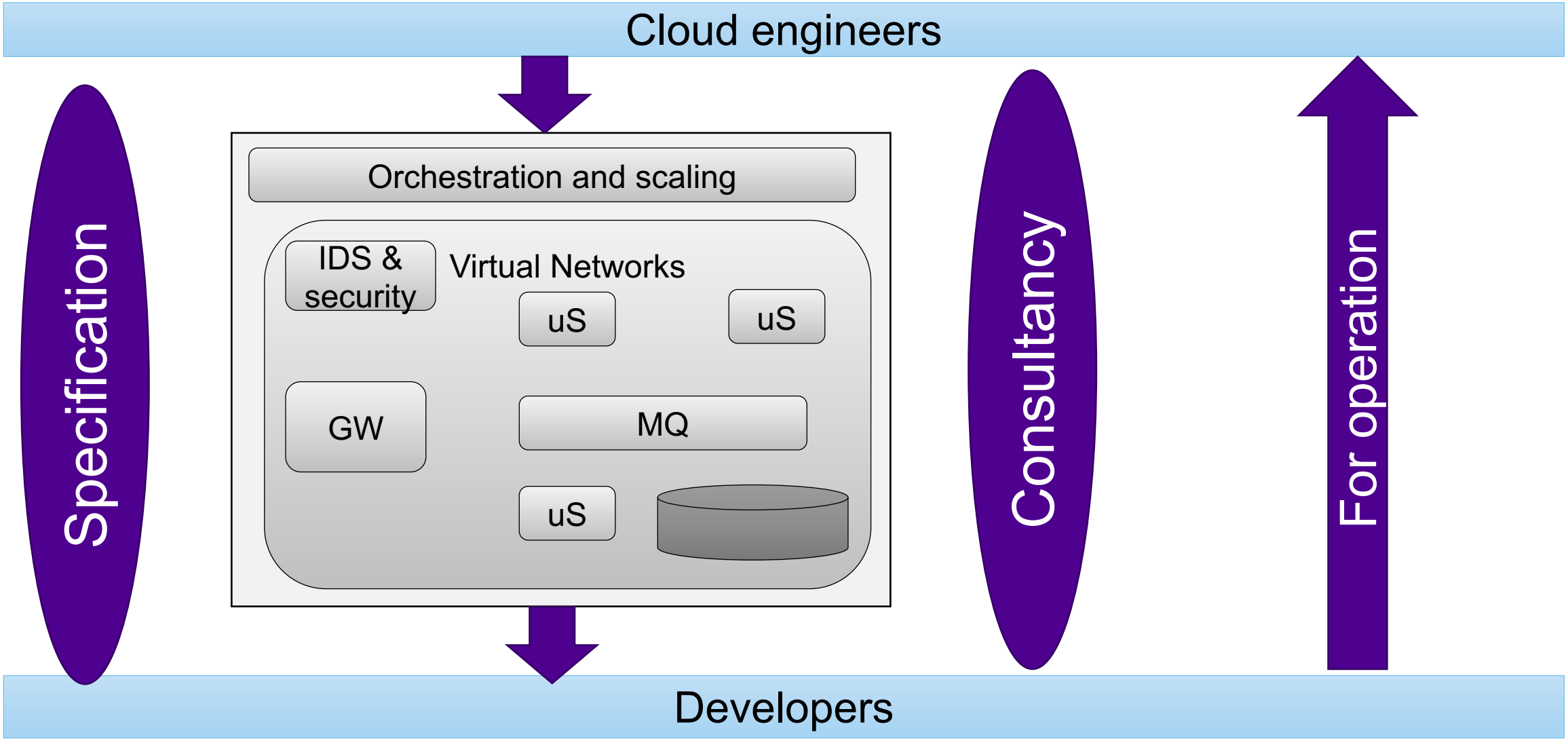


Misconceptions

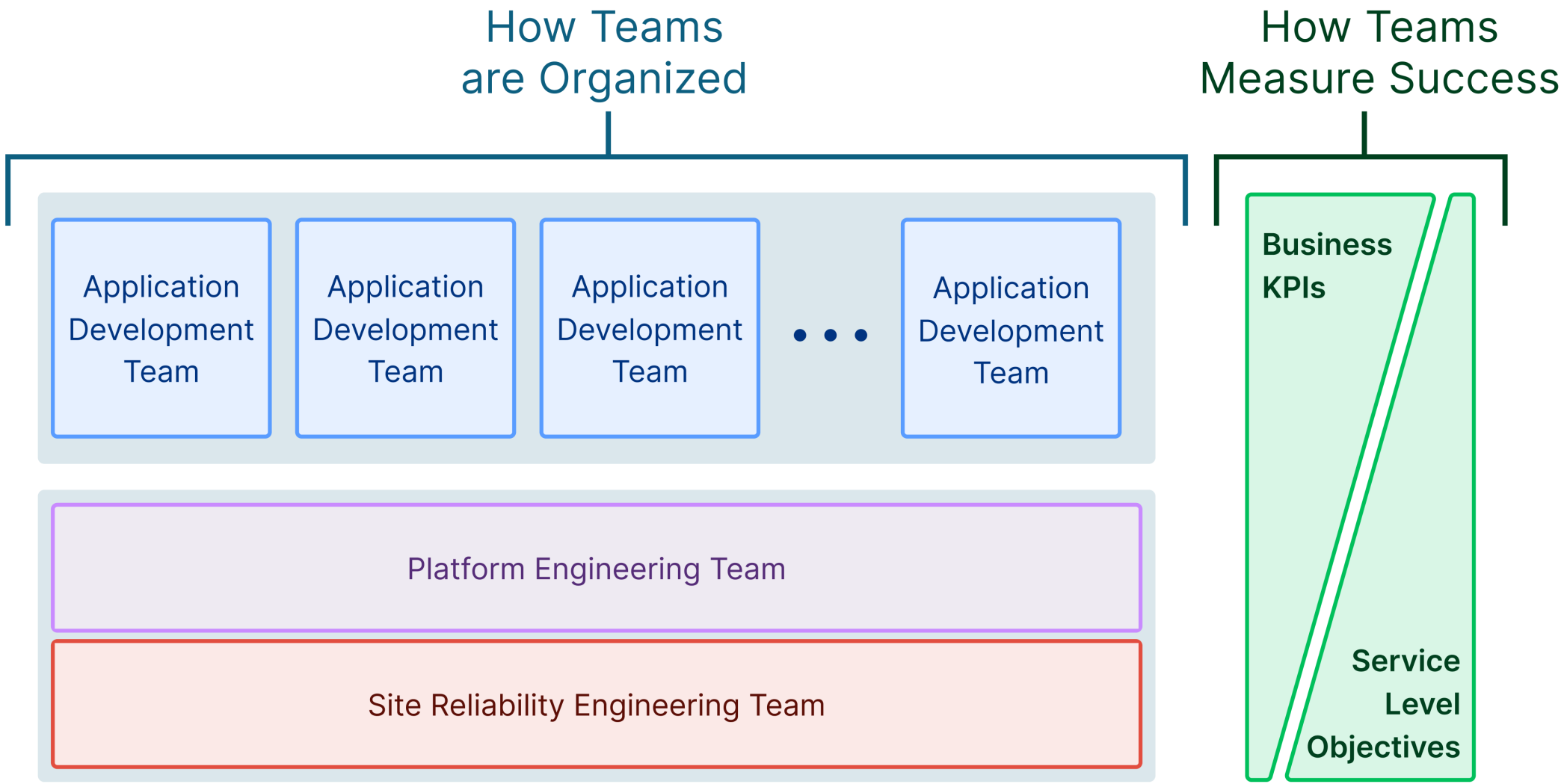
- “When the code is ready, let’s send it to the DevOps team”
- In DevOps developers need to be both Dev and Ops experts.

Tampere University

Split of work



https://www.getambassador.io/resources/rise-of-cloud-native-engineering-organizations



Site reliability engineer

https://aws.amazon.com/what-is/sre/

- Application monitoring
  - service-level agreements (SLAs), service-level indicators (SLIs), and service-level objectives (SLOs)
- Gradual change implementation
  - SRE practices encourage the release of frequent but small changes to maintain system reliability
- Automation for reliability improvement
  - policies and processes that embed reliability principles in every step of the delivery pipeline
- *SRE is the practical implementation of DevOps.*

Principles

- Clear mission and role
- Treat your platform as a product
- Focus on common problems
- Glue is valuable
- Don’t reinvent the wheel

DORA metric for DevOps

Metric	Explanation
Deployment Frequency	Refers to the frequency of successful software releases to production.
Lead Time for Changes	Captures the time between a code change commit and its deployable state.
Mean Time to Recovery	Measures the time between an interruption due to deployment or system failure and full recovery.
Change Failure Rate	Indicates how often a team's changes or hotfixes lead to failures after the code has been deployed.

xhttps://www.leanix.net/en/wiki/vsm/dora-metrics

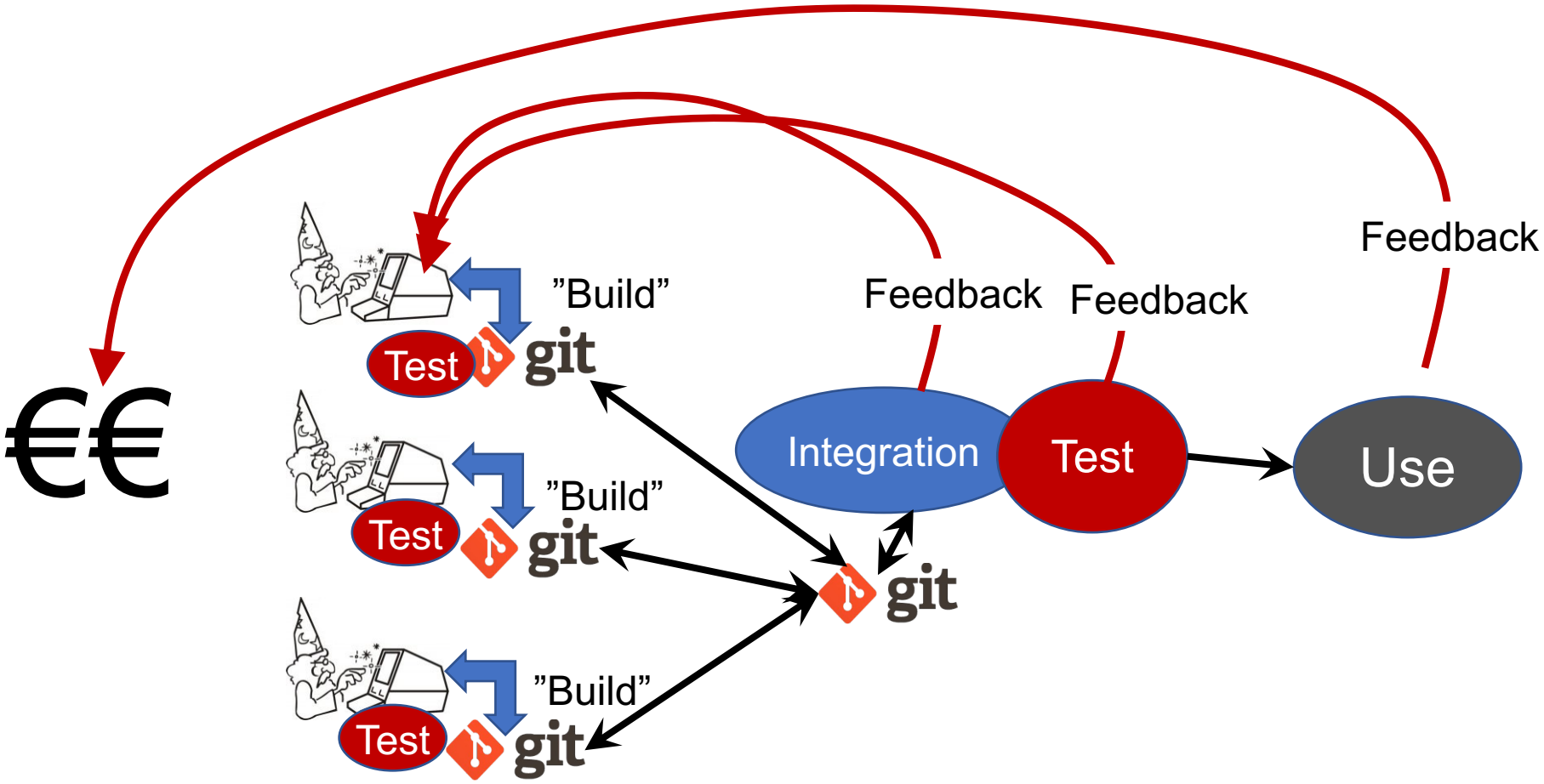
Tampere University

Reading material for the exam

- Lwakatare, Lucy Ellen, Doctoral Dissertation, University of Oulu, 2017, DevOps adoption and implementation in software development practice : concept, practices, benefits and challenges, http://jultika.oulu.fi/files/isbn9789526217116.pdf
  - Pages 25-30
- Taivalsaari, A., Mikkonen, T., Pautasso, C., & Systä, K. (2021). Full Stack Is Not What It Used to Be. In M. Brambilla, R. Chbeir, F. Frasincar, & I. Manolescu (Eds.), Web Engineering - 21st International Conference, ICWE 2021, Proceedings (pp. 363-371). (Lecture Notes in Computer Science; Vol. 12706 LNCS). Springer. https://helda.helsinki.fi/items/47b6de91-ad0b-4bd4-9e40-ac2b5c61a040
- https://www.getambassador.io/resources/rise-of-cloud-native-engineering-organizations

Continuous Deployment

Continuous deployment



- *“Continuous Deployment (CD) advocates for quick and frequent deployments of software to production. The goal is to bring new functionality as early as possible to users while learning from their usage.”*

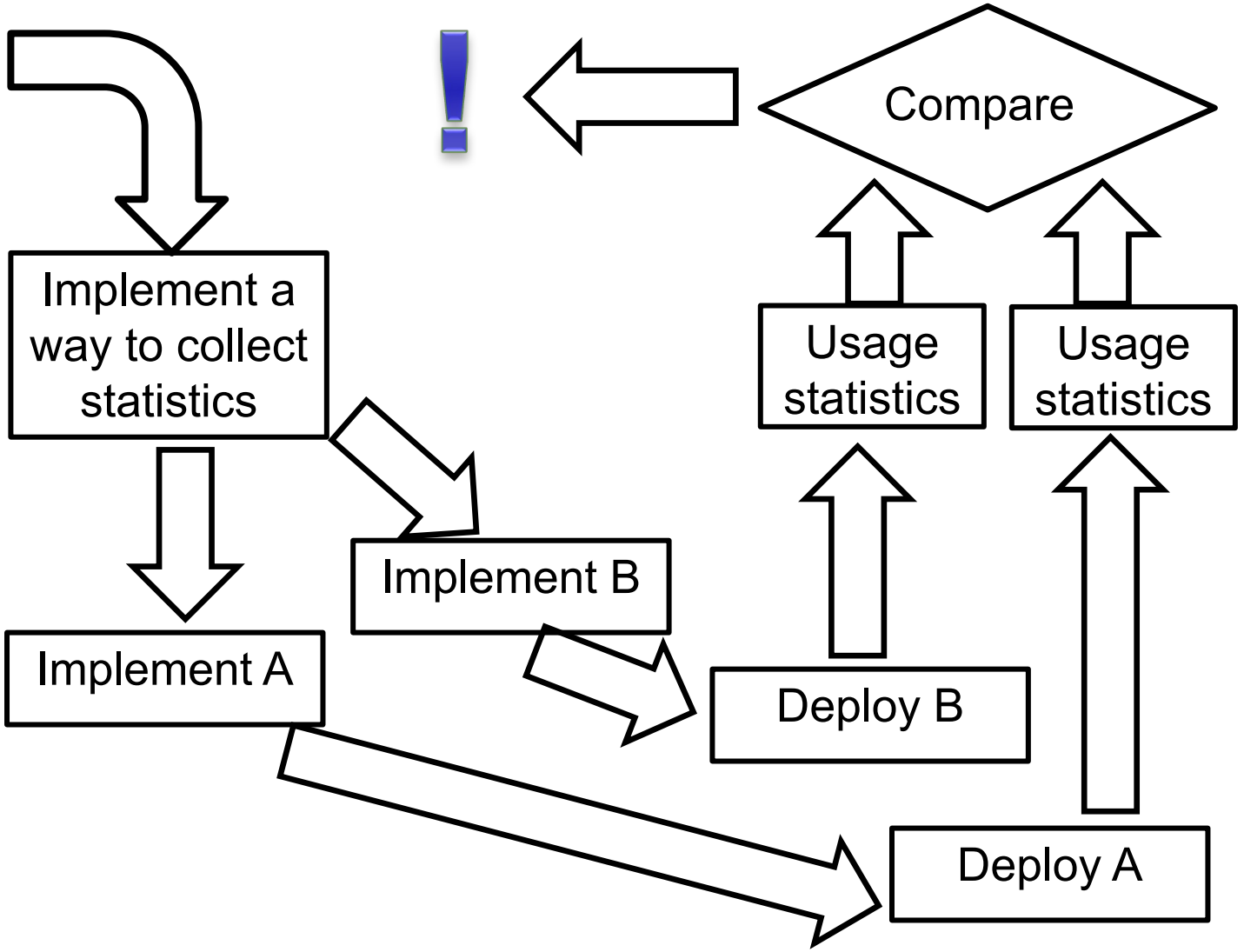
Tampereen yliopisto
Tampere University

Perceived benefits

- **Improved delivery speed** of software changes Improved speed in the development and deployment of software changes to production environment.
- **Improved productivity in operations** work. Decreased communication problems, bureaucracy, waiting overhead due to removal of manual deployment hand-offs and organisational boundaries; Lowered human error in deployment due to automation and making explicit knowledge of operation-related tasks to software development
- **Improvements in quality**. Increased confidence in deployments and reduction of deployment risk and stress; Improved code quality; Improved product value to customer resulting from production feedback about users and usage.
- **Improvements in organisational-wide culture** and mind-set. Enrichment and wider dissemination of DevOps in the company through discussions and dedicated training groups 'communities of practice'

A/B Testing

Should our project have A or B?



Reading for the exam

- https://continuousdelivery.com

- M. Leppänen et al., "The highways and country roads to continuous deployment," in IEEE Software, vol. 32, no. 2, pp. 64-72, Mar.-Apr. 2015. doi: 10.1109/MS.2015.50, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7057604&isnumber=7057030

Cloud

https://www.computerworld.com/article/3427741/a-brief-history-of-salesforce-com.html

“The way the story goes is that Marc Benioff was floating in the sea just off Big Island in his beloved Hawaii during a sabbatical when he thought:

why can't buying software be as a simple as Amazon is for consumer goods?

This line of thinking eventually led to Benioff and a team of developers pioneering the software-as-a-service (SaaS) model by delivering its customer relationship management (CRM) software

over the internet on a per seat, per month payment plan, instead of deployed on-premise servers under a hefty licensing agreement.”

Essential characteristics

- *On-demand self-service.*
- *Broad network access.*
- *Rapid elasticity.*
- *Resource pooling.*
- *Measured service.*

Tampere University

Use case example

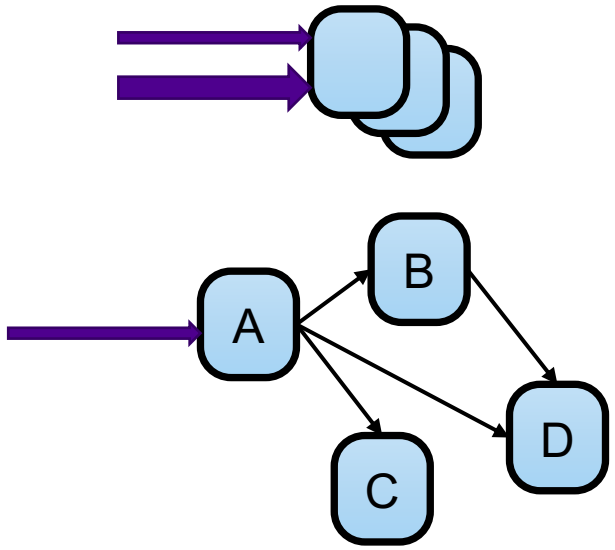
- Your application needs
  - Certain version of nodejs
  - Set of libraries (certain versions)
  - Mongo database

- Your system has
  - Wrong version of nodejs
  - Mongo serving another application

- Solution
  - Create a docker image (container)
  - Install the image
  - Run the image

What are typical cloud applications

- Networks of containers!



Logically like:

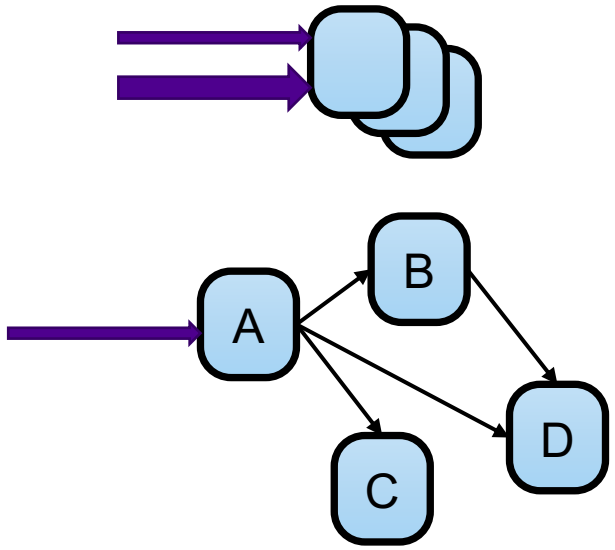
```
A() {
    B();
    C();
    D();
}
```

But implemented as inter-process communication.

```
A() {
    http.get(B:80);
    http.get(C:80);
    http.get(D:80);
}
```

What are typical cloud applications

- Networks of containers!



Logically like:

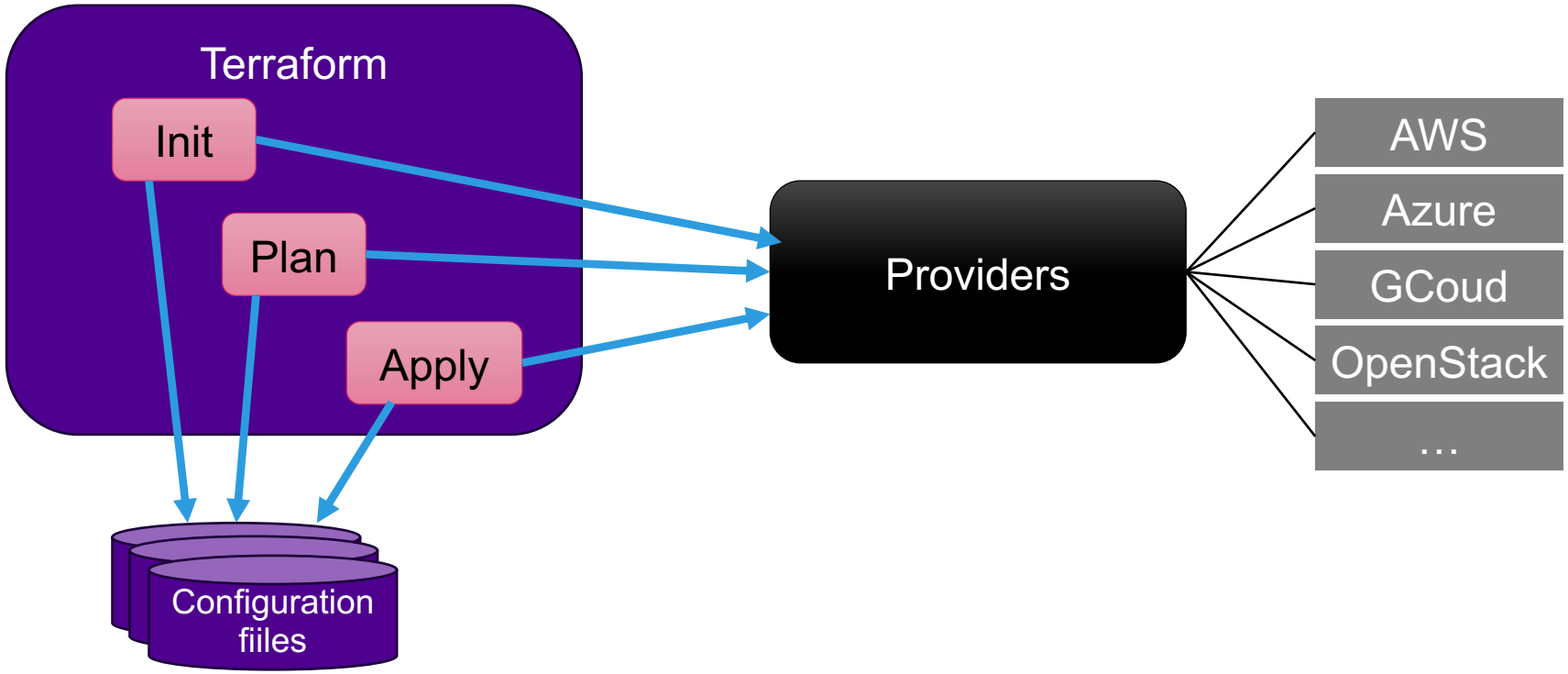
```
A() {
    B();
    C();
    D();
}
```

But implemented as inter-process communication.

```
A() {
    http.get(B:80);
    http.get(C:80);
    http.get(D:80);
}
```

Tampere University

Terraform



Tampere University

Ansible vs Terraform?

- Ansible: update configuration of a ”machine”
- Terraform: create virtual (cloud) infrastructure from IaC

Automation, Automation

Tampereen yliopisto
Tampere University

7R's of cloud Micration

Replace
with similar or improved but SaaS

Reuse
in the new SaaS version

Refactor
towards cloud-native architecture

Replatform
by using cloud services

Rehost
to a VM

Retain

Retire

Reading for the exam

- Peter Mell; Timothy Grance (September 2011). The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
- D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," in IEEE Cloud Computing, vol. 4, no. 5, pp. 12-14, September/October 2017, doi: 10.1109/MCC.2017.4250932. https://ieeexplore.ieee.org/document/8125545

Cloud native

Cloud-native – what/how?

1. Packaged as lightweight containers
2. Developed with best-of-breed languages and frameworks
3. Designed as loosely coupled microservices
4. Centered around APIs for interaction and collaboration
5. Architected with a clean separation of stateless and stateful services
6. Isolated from server and operating system dependencies
7. Deployed on self-service, elastic, cloud infrastructure
8. Managed through agile DevOps processes
9. Automated capabilities
10. Defined, policy-driven resource allocation

Some definitions

- If an app is "cloud-native," it's specifically designed to provide a consistent development and automated management experience across private, public, and hybrid clouds.
- A native cloud application (NCA) is a program that is designed specifically for a cloud computing architecture.
  - NCAs are designed to take advantage of cloud computing frameworks,
  - which are **composed of loosely-coupled cloud services**.
  - That means that developers must break down tasks into separate **services that can run on several servers in different locations**.
  - Because the infrastructure that supports a native cloud app does not run locally, NCAs must be **planned with redundancy** in mind so the application can withstand equipment failure and be able to re-map IP addresses automatically should hardware fail.

Cloud-native – what/how?

1. Packaged as lightweight containers
2. Developed with best-of-breed languages and frameworks
3. Designed as loosely coupled microservices
4. Centered around APIs for interaction and collaboration
5. Architected with a clean separation of stateless and stateful services
6. Isolated from server and operating system dependencies
7. Deployed on self-service, elastic, cloud infrastructure
8. Managed through agile DevOps processes
9. Automated capabilities
10. Defined, policy-driven resource allocation

Some definitions

- If an app is "cloud-native," it's specifically designed to provide a consistent development and automated management experience across private, public, and hybrid clouds.
- A native cloud application (NCA) is a program that is designed specifically for a cloud computing architecture.
  - NCAs are designed to take advantage of cloud computing frameworks,
  - which are **composed of loosely-coupled cloud services**.
  - That means that developers must break down tasks into separate **services that can run on several servers in different locations**.
  - Because the infrastructure that supports a native cloud app does not run locally, NCAs must be **planned with redundancy** in mind so the application can withstand equipment failure and be able to re-map IP addresses automatically should hardware fail.

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

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.

Tampereen yliopisto
Tampere University

Serverless computing

Baldini et all: Serverless Computing: Current Trends and Open Problems, Research Advantages in Cloud Computing, Springer, 2017.

A cloud-native platform

for

- short-running, stateless computation
- event driven applications

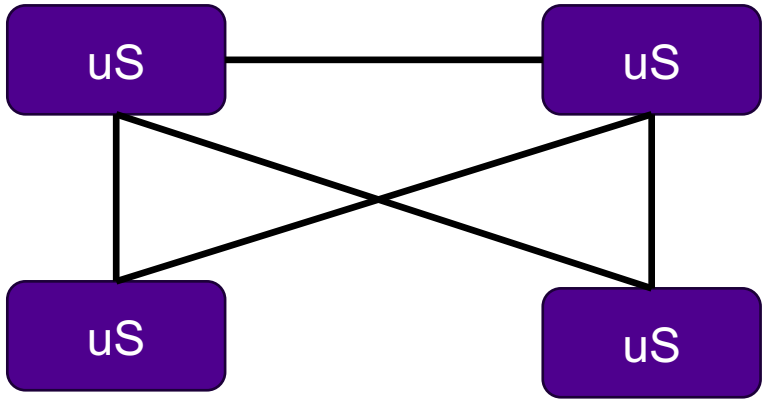
which

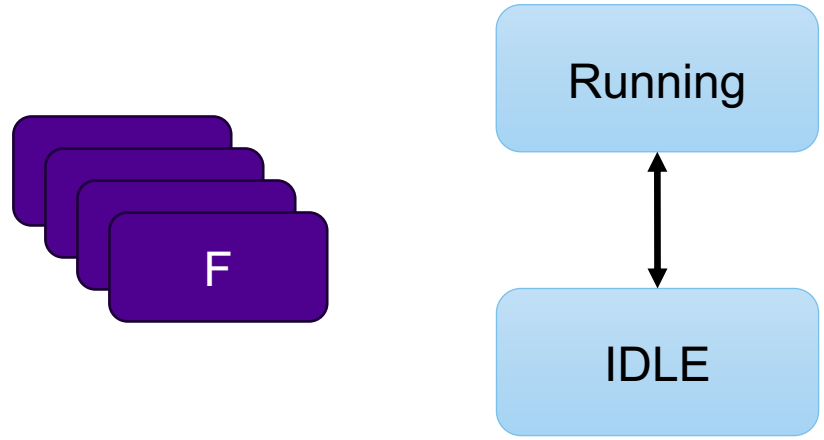
- scale up and down instantly and automatically

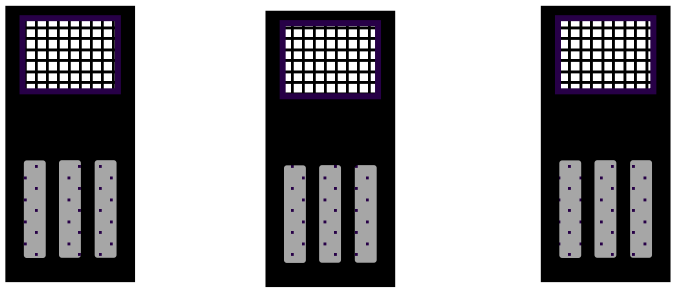
and

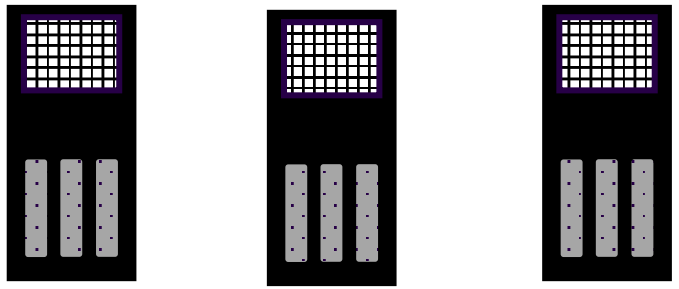
- charge for actual usage and high granulatity

The difference









Claimed FaaS advantages

- Smaller for developer since infrastructure is handled by somebody else => more time for writing application code
- Inherently scalable
- No need to pay for idle resources (temptation to miss-use)
- Available and fault tolerant
- No explicit multi-tenancy
- Forces modular business logic

Tampereen yliopisto
Tampere University

Claimed FaaS disadvantages

- Decreased transparency
- Maybe challenging to debug
- Autoscaling of functions may lead to autoscaling of cost
- Keeping track of huge numbers of functions is tough
- Chaching of requests?

Tampereen yliopisto
Tampere University

Microservices vs. Serverless/FaaS
(They are different – do not call serveless microservices)

- Microservice
  - Small services running in their own process and communicating with lightweight services
  - Can be stateful

- Serverless / FaaS
  - Short term execution triggered by a request, then closes down
  - For stateless computing

Exam reading

- Baldini et all: Serverless Computing: Current Trends and Open Problems, Research Advanges in Cloud Computing, Springer, 2017. https://arxiv.org/abs/1706.03178

Tampere University

Exam

- Exam window 12.-19.12.
- Enrollment in Sisu open already
- Enrollment in Exam opens today evening