Tampereen yliopisto
Tampere University

Lecture 12 - Hosting

17.11.2020

Kari Systä

Schedule for coming weeks

Week	Lecture	Plussa exercises (deadlines)
10/45	03.11 Testing and testing automation	
11/46	10.11 Guest Lecture, CD pipeline at cargotec	
12/47	**17.11 Deployment, hosting and monitor**ing	
13/48	24.11 Introduction of some popular tools	
14/40	01.12 Recap	

Tampereen yliopisto
Tampere University

Some general stuff

Again some jobs / master thesis places

- PricewaterhouseCoopers Oy
  - Integration of new data source to a BI system
  - Knowledge about Microsoft frameworks is a “plus”
  - Contact: viivi.litmanen@pwc.com

- Fingrid
  - Two posts in digitalization of smart grids
  - Finnish (language) is required
  - https://rekry.fingrid.fi/MepRekry/Application/Index/338/
  - https://rekry.fingrid.fi/MepRekry/Application/Index/336/

- MyLab
  - Medical SW development
  - Contact: timo.pellinen@mylab.fi

Some feedback from message queue exercise

About Docker:

- As you run software in containers, you should store all code files, libraries, etc. in the containers themselves. Don't store these in the host.
- The image "node:latest" seems to be a bad idea and may cause Docker Compose build to fail.
- Before you return your exercise, clone the repo once into another location or even another host. This helps you notice if some files are missing from the repo.

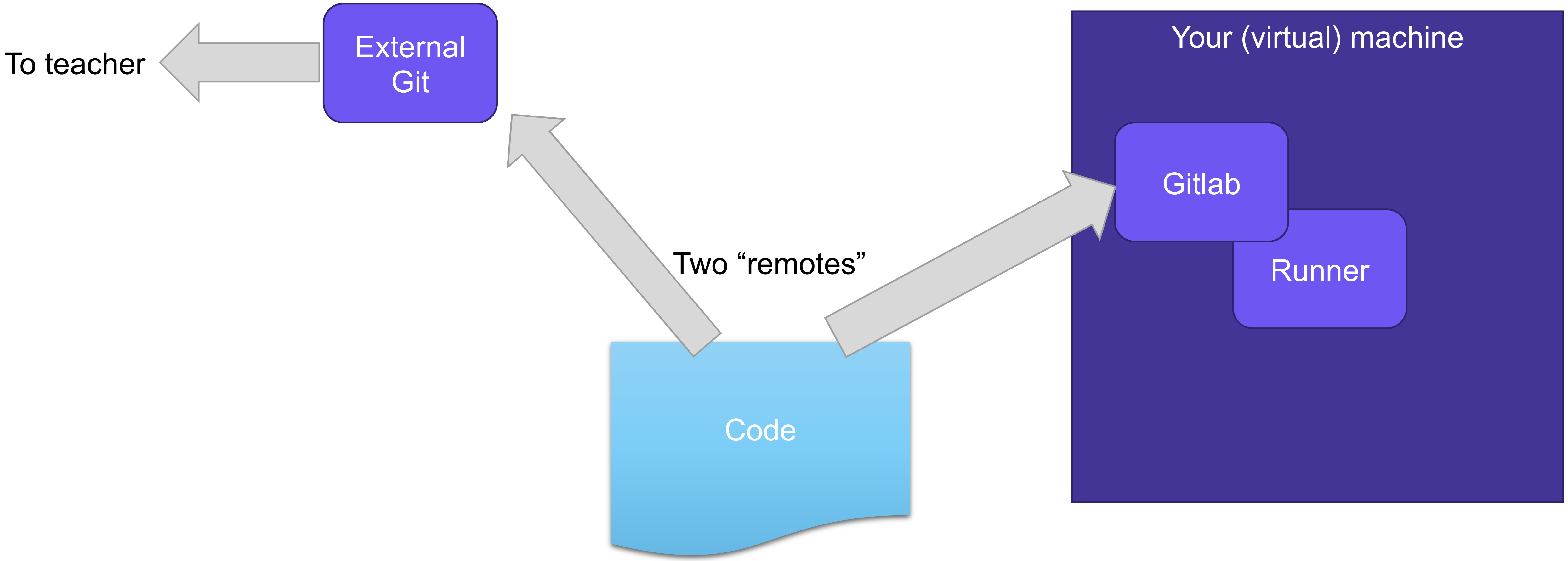
Read the requirements.

- Use a topic exchange instead of fanout or sending directly to queues.
- The requirements said "send three messages" instead of "send forever".
- Please re-check the formatting requirement of log entries.
- The messages should be appended to the log (this was not explicit in the instructions, but only one student seems to have misunderstood).

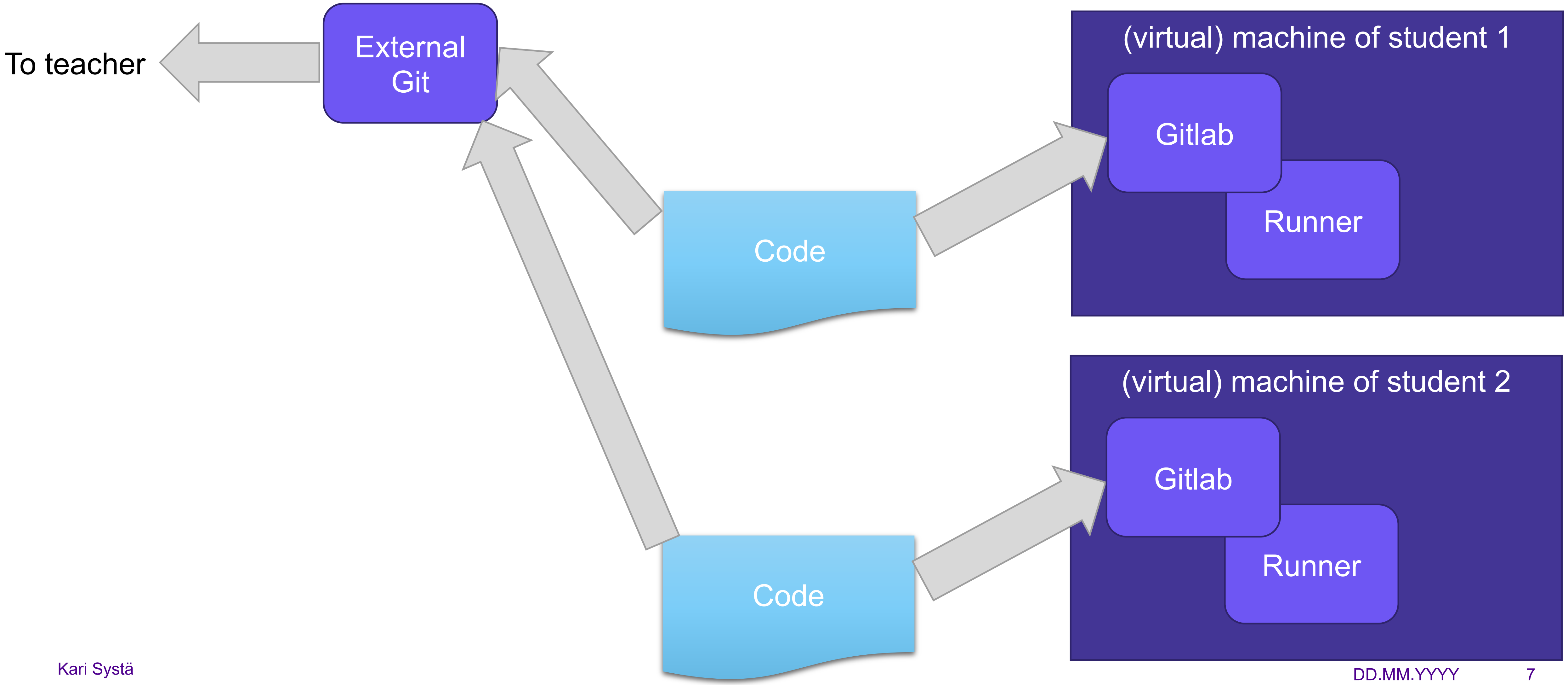
Back to basics:

- Use comments in source code.
- Use descriptive variable names.
- Use error handling. Don't let the HTTP server die if there is (yet) nothing to serve.

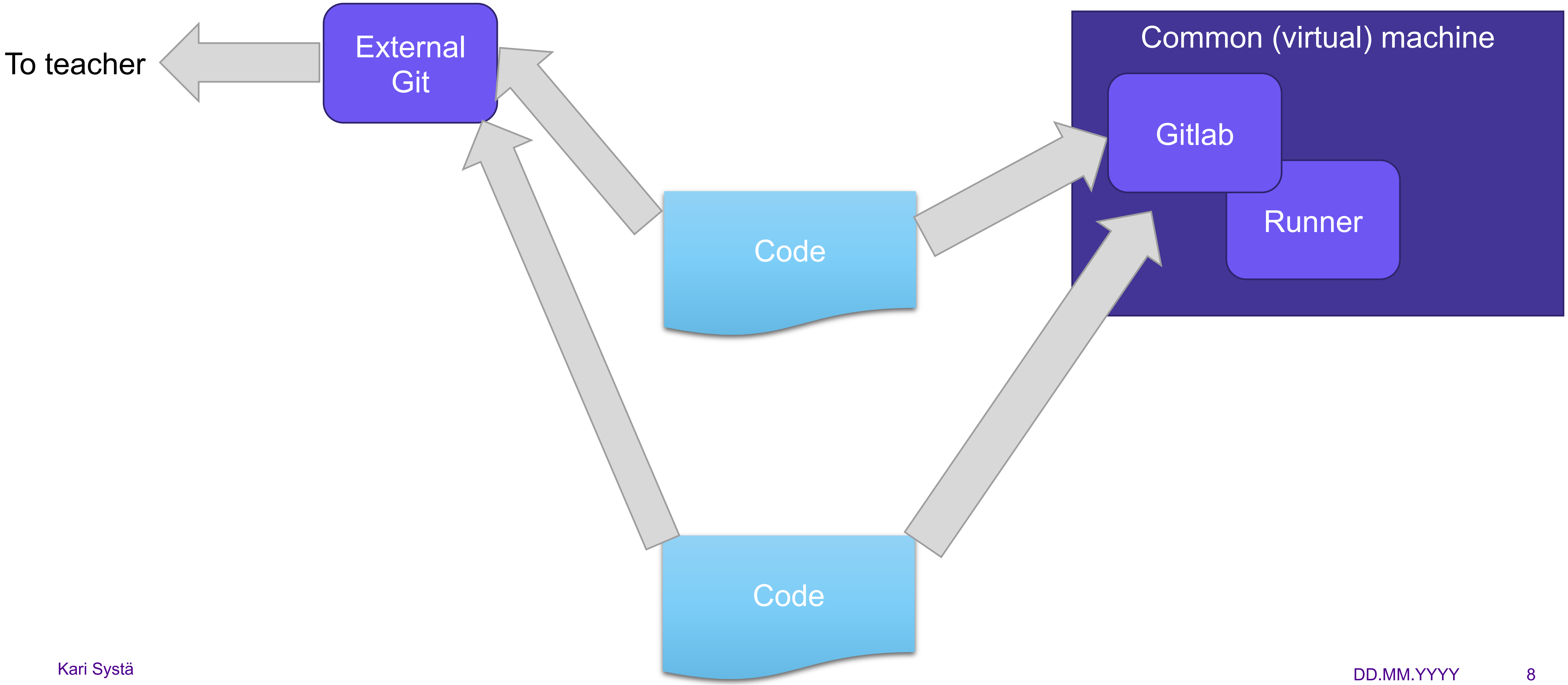
Clarifications for the project



Possible option for a pair


To teacher
External Git
Code
Code
(virtual) machine of student 1
Gitlab
Runner
(virtual) machine of student 2
Gitlab
Runner

Another option for a pair



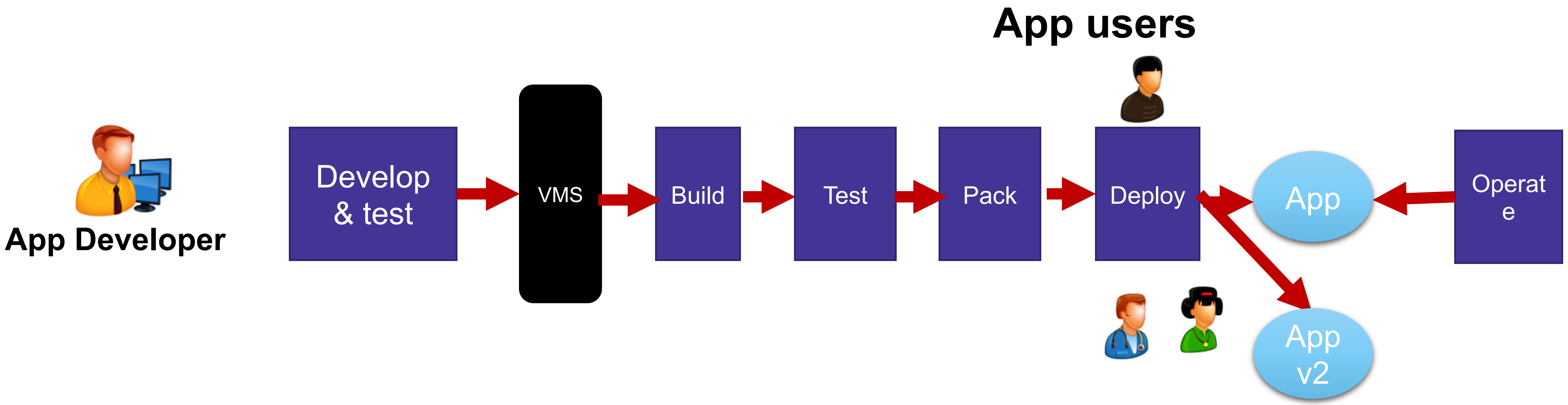
About the exam

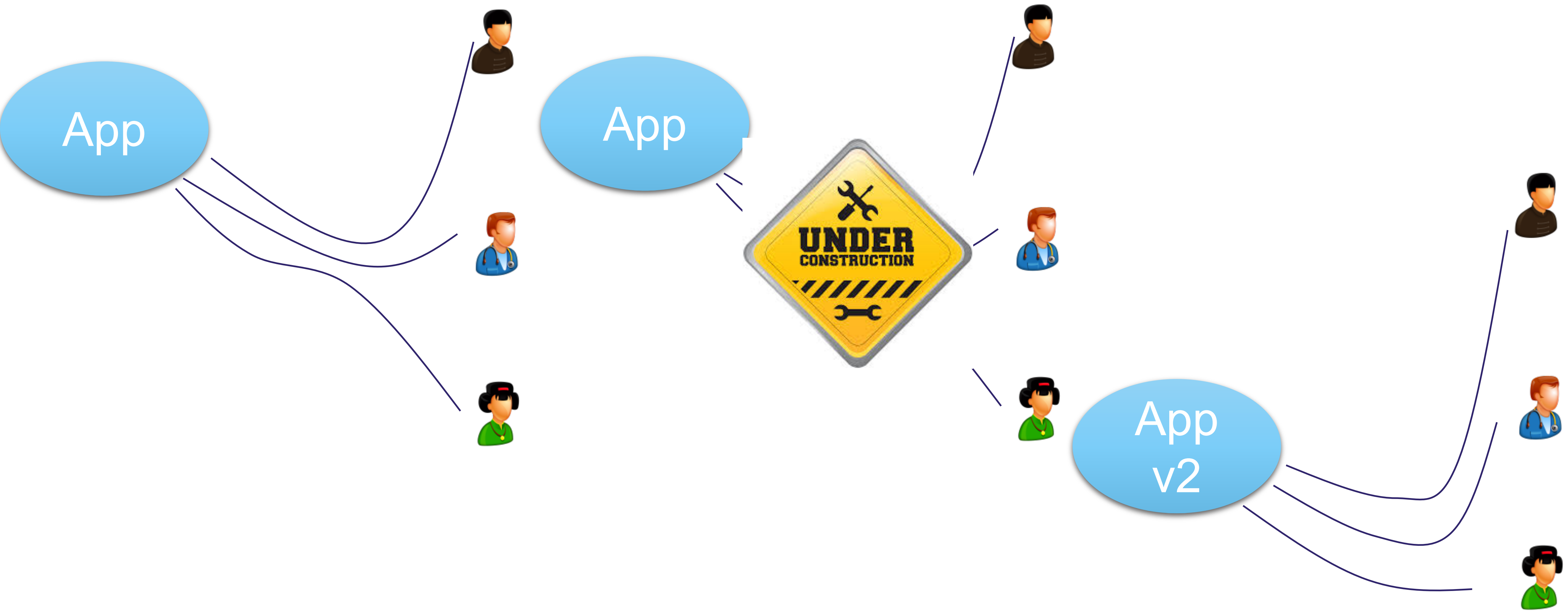
- The exam was initially scheduled for December (just broad schedule 07.12 – 31.12 was made).
- Before narrowing the timeframe I want to know student preferences. For example,
  - 7.-11.12 and then I could try check the exams before Christmas.
  - 14.-18.12 but then results will be postponed to January.

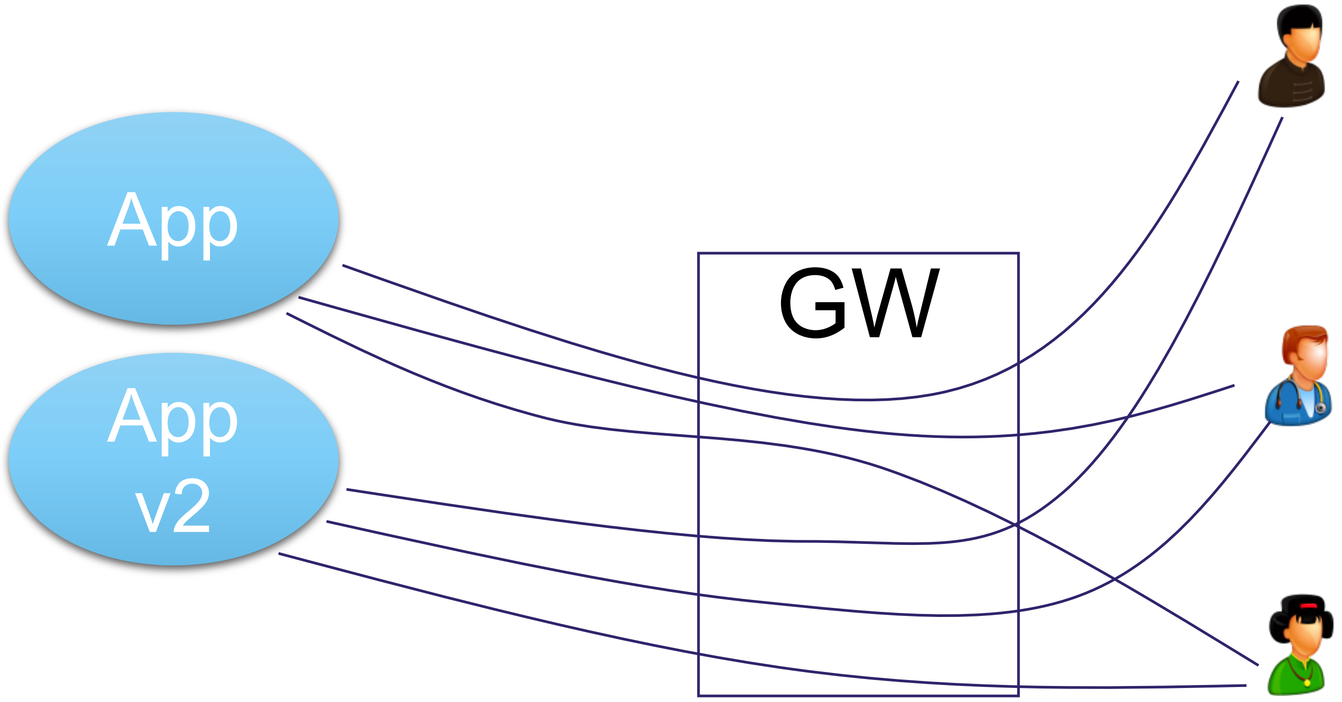
- The assumption has been that we use the exam rooms (with the security precautions), but before deciding on anything I need to ask how many of you is not in Finland at the moment?

- Exam represents 40% of the total grading.

Deployment and hosting


App users
App Developer
Develop & test
VMS
Build
Test
Pack
Deploy
App
App v2
Operate


App
App
UNDER CONSTRUCTION
App v2



Problems & issues?

Deployment strategies

Basic Deployment (aka Suicide)

(https://harness.io/2018/02/deployment-strategies-continuous-delivery/)

all nodes are updated at the same time

Tampereen yliopisto
Tampere University

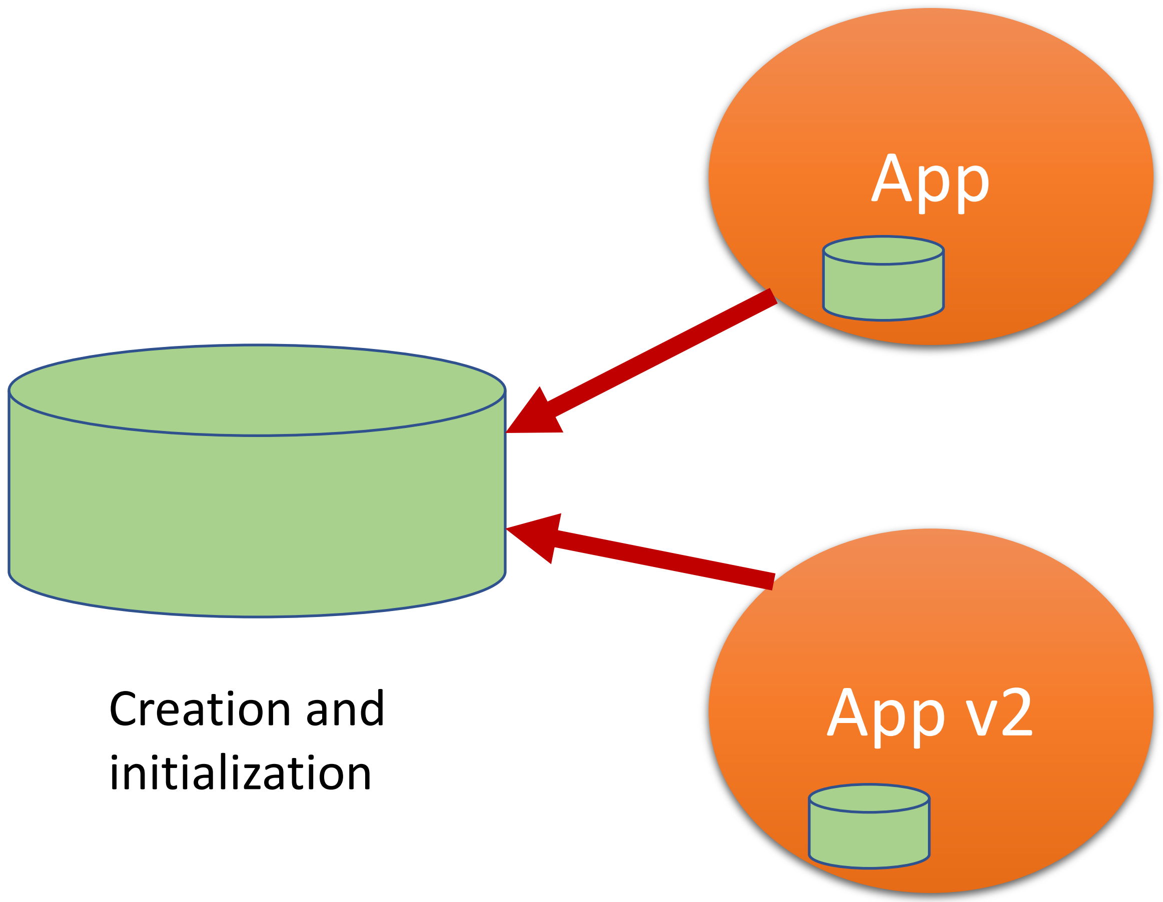
Rolling Deployment
(https://harness.io/2018/02/deployment-strategies-continuous-delivery/)

nodes are updated incrementally,

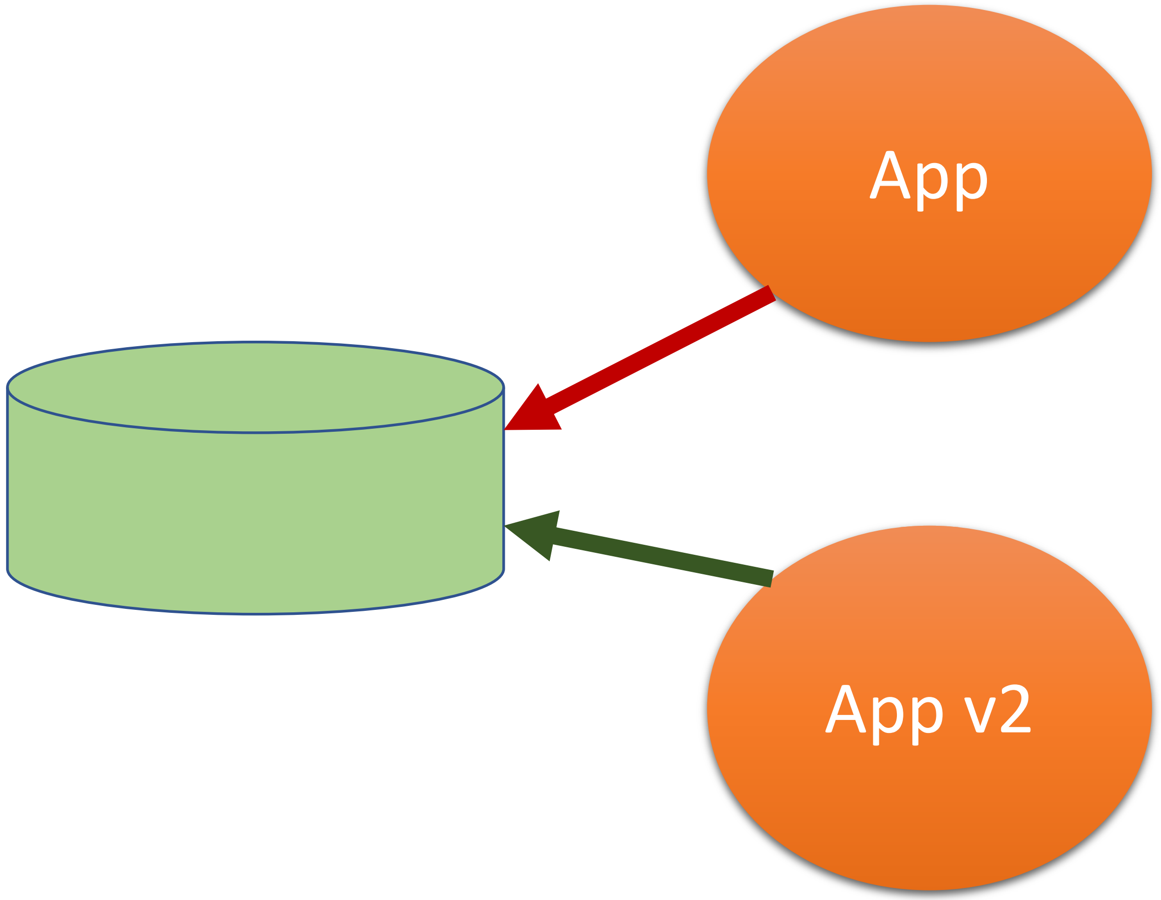
(http://martinfowler.com/bliki/BlueGreenDeployment.html)

the new version (called green) is set up in parallel with the current (blue). When new (green) is ready, the router is switched to new (green) and blue is left as a backup. If something goes wrong with new, the router can be switched back to old - that means easy “rollback”.

How about the data?



Creation and initialization

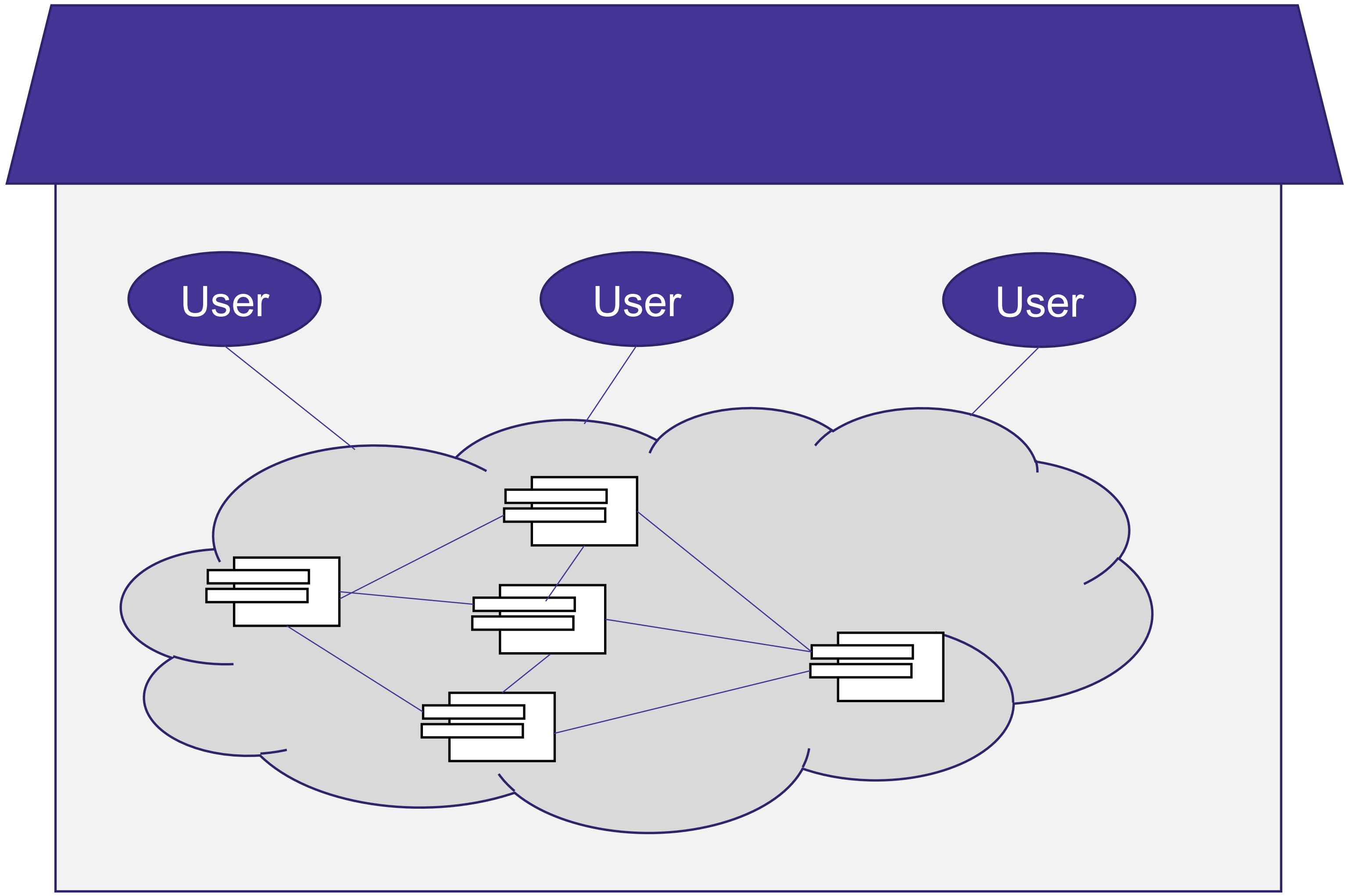


Tampereen yliopisto
Tampere University

Data migration

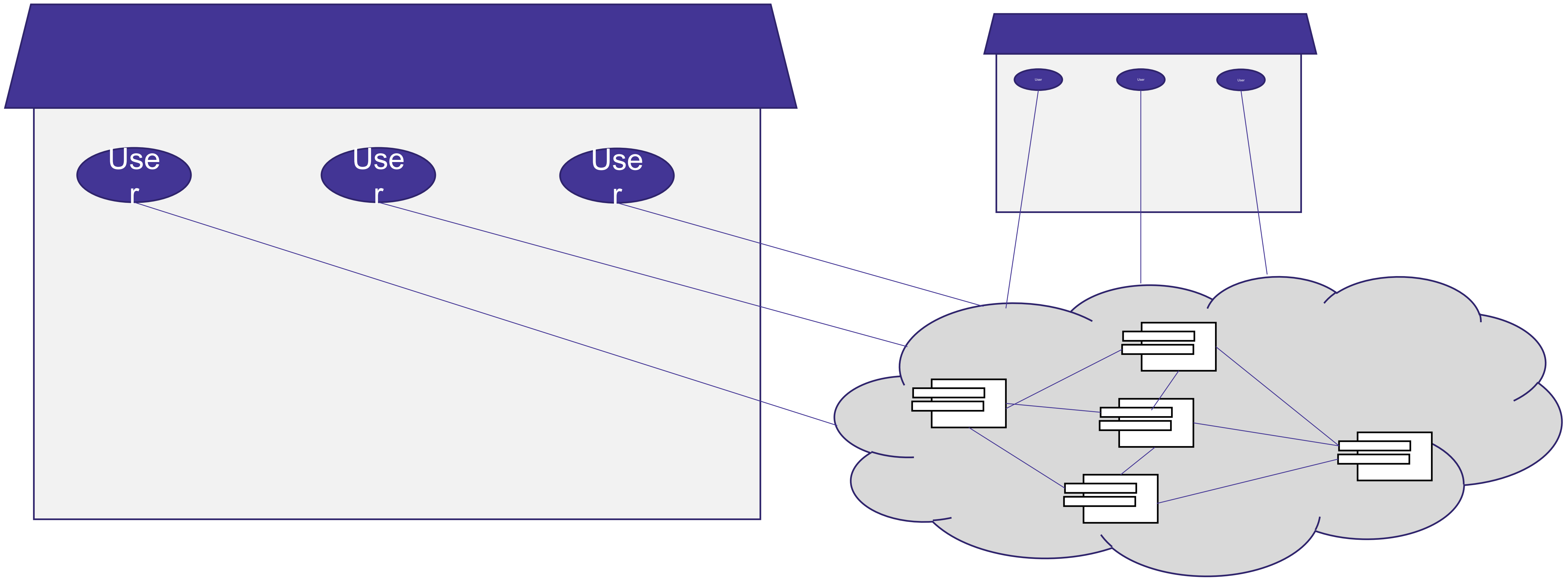
- Versions of the data bases
- Data migration scripts are needed.
- Rollback need to be possible

Private cloud

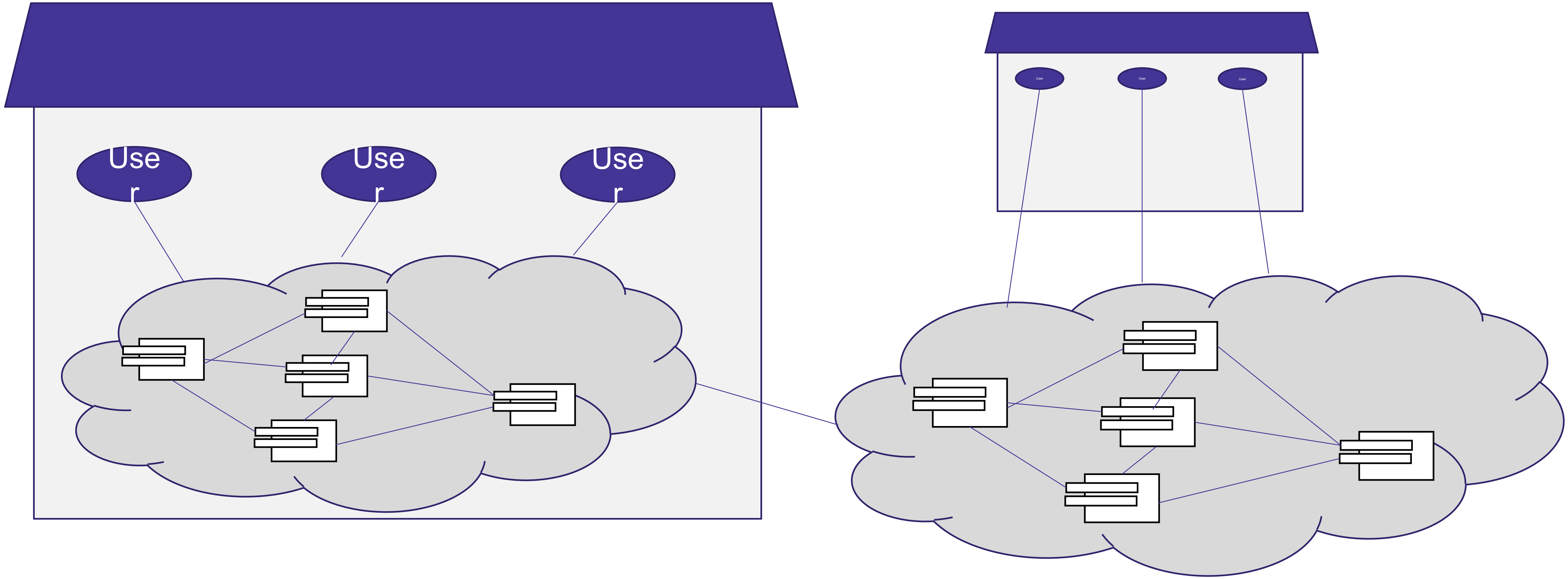


Tampereen yliopisto
Tampere University

Public cloud



Hybrid cloud



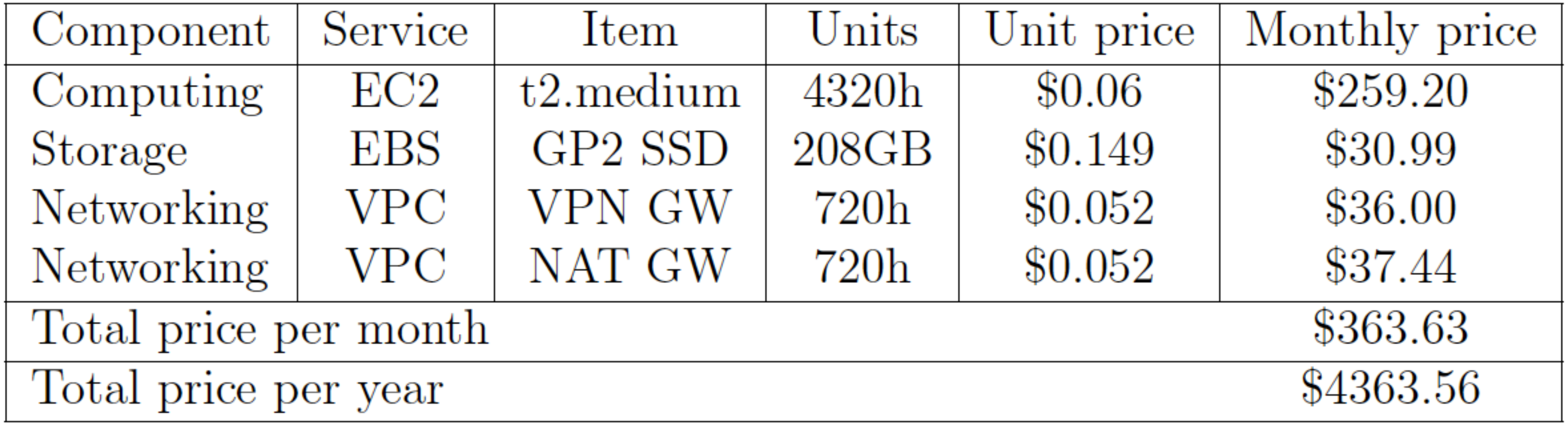
Tampereen yliopisto
Tampere University

Comparison

	Investment cost	Operation cost	Security	Flexibility
Private cloud	High, buying the HW	Low, although increases labour costs	Can be high but depends on skills	Low, there is a limit; change is difficult.
Public cloud	Low	High	Is matter of trusts, sometimes regulation limits	High
Hybrid cloud	High/Medium	Low/Medium	Who do you trust?	Medium

Tampereen yliopisto
Tampere University

One master thesis estimated for CI/CD pipeline



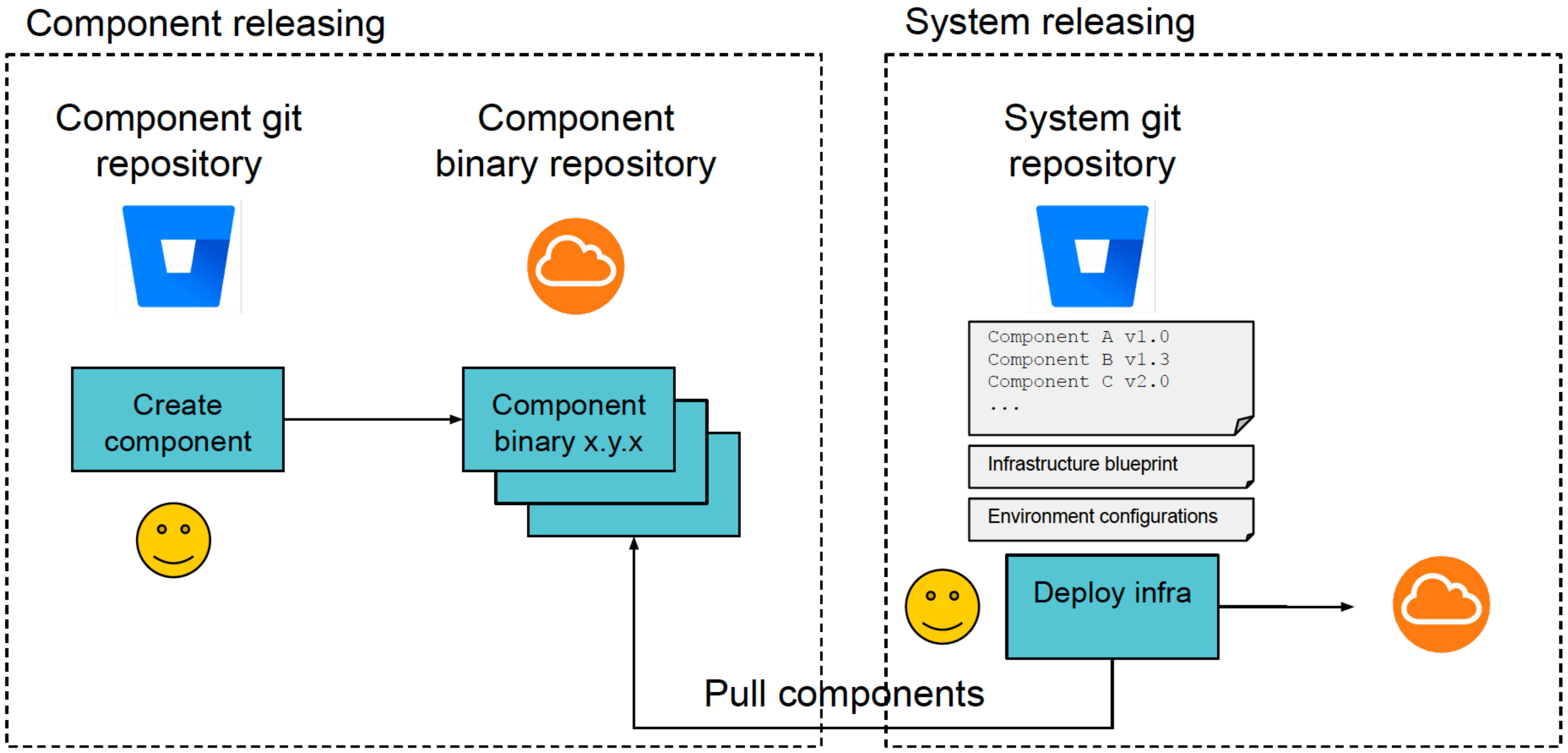
Component	Service	Item	Units	Unit price	Monthly price
Computing	EC2	t2.medium	4320h	$0.06	$259.20
Storage	EBS	GP2 SSD	208GB	$0.149	$30.99
Networking	VPC	VPN GW	720h	$0.052	$36.00
Networking	VPC	NAT GW	720h	$0.052	$37.44
Total price per month					$363.63
Total price per year					$4363.56

How about running costs?

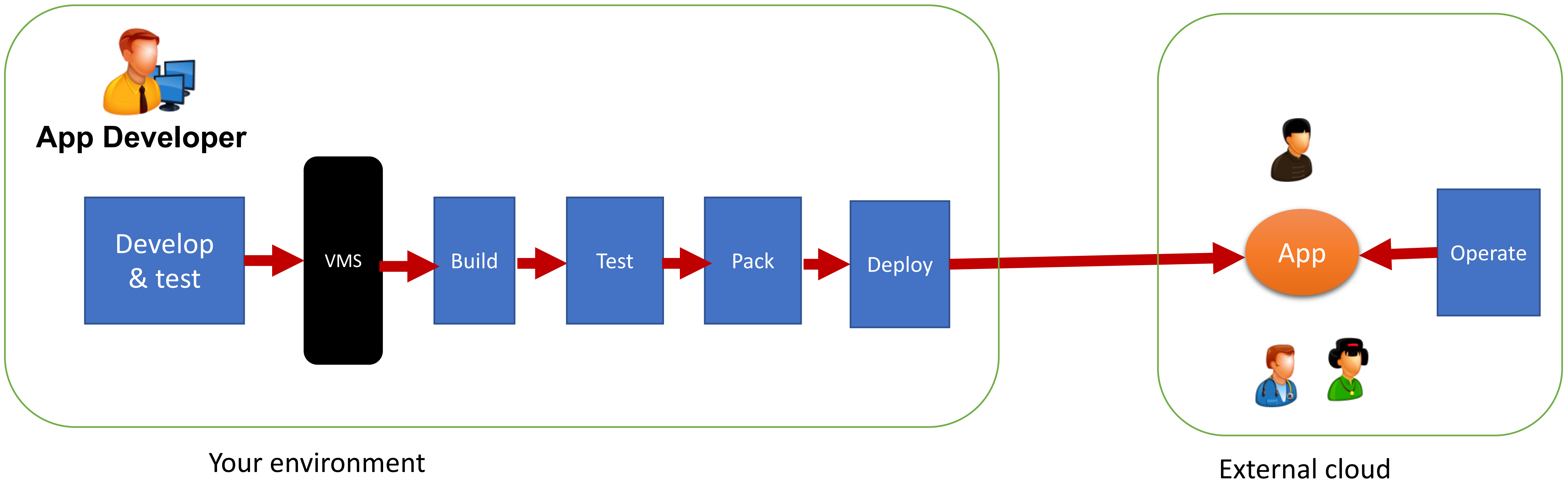
CAPEX, OPEX?

Tampereen yliopisto
Tampere University

From the Cargotec presentation



```
Component A v1.0
Component B v1.3
Component C v2.0
...
```


App Developer
Develop & test
VMS
Build
Test
Pack
Deploy
App
Operate
Your environment
External cloud

Tampereen yliopisto
Tampere University

Steps

- Provision – set up the infrastructure
- Configure – the infratructure
- Deploy – the application
- Reserve/scale resources
- Monitor

- Nice AWS-specific overview:
  https://d1.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf

Tool examples in AWS

(Provision, configure, deploy scale, monitor)

- **AWS CloudForma**tion is a service that enables customers to provision and manage almost any AWS resource using a custom template language expressed in YAML or JSON.
- **AWS Elastic Beanstalk** is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, or Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
- **AWS CodeDeploy** is a fully managed deployment service that automates application deployments to compute services such as Amazon EC2, Amazon Elastic Container Service (Amazon ECS), AWS Lambda, or on-premises servers.

Tampereen yliopisto
Tampere University

AWS for containers

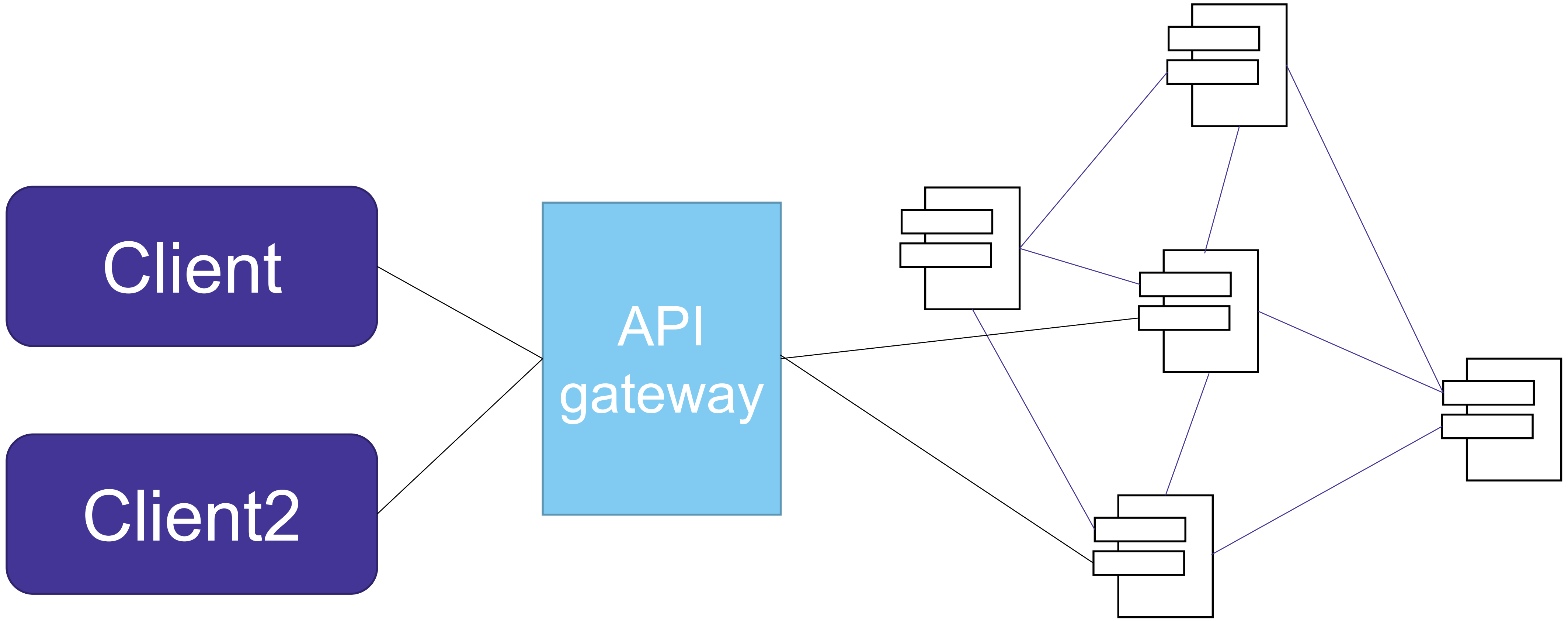
(According to AWS)

- **Amazon Elastic Container Service (Amazon ECS)** is a fully managed container orchestration service that supports Docker containers and allows you to easily run applications on a managed cluster. Amazon ECS eliminates the need to install, operate, and scale container management infrastructure, and simplifies the creation of environments with familiar AWS core features like Security Groups, Elastic Load Balancing, and AWS Identity and Access Management (IAM).
- **Amazon Elastic Kubernetes Service (Amazon EKS)** is a fully-managed, certified Kubernetes conformant service that simplifies the process of building, securing, operating, and maintaining Kubernetes clusters on AWS.

How to monitor

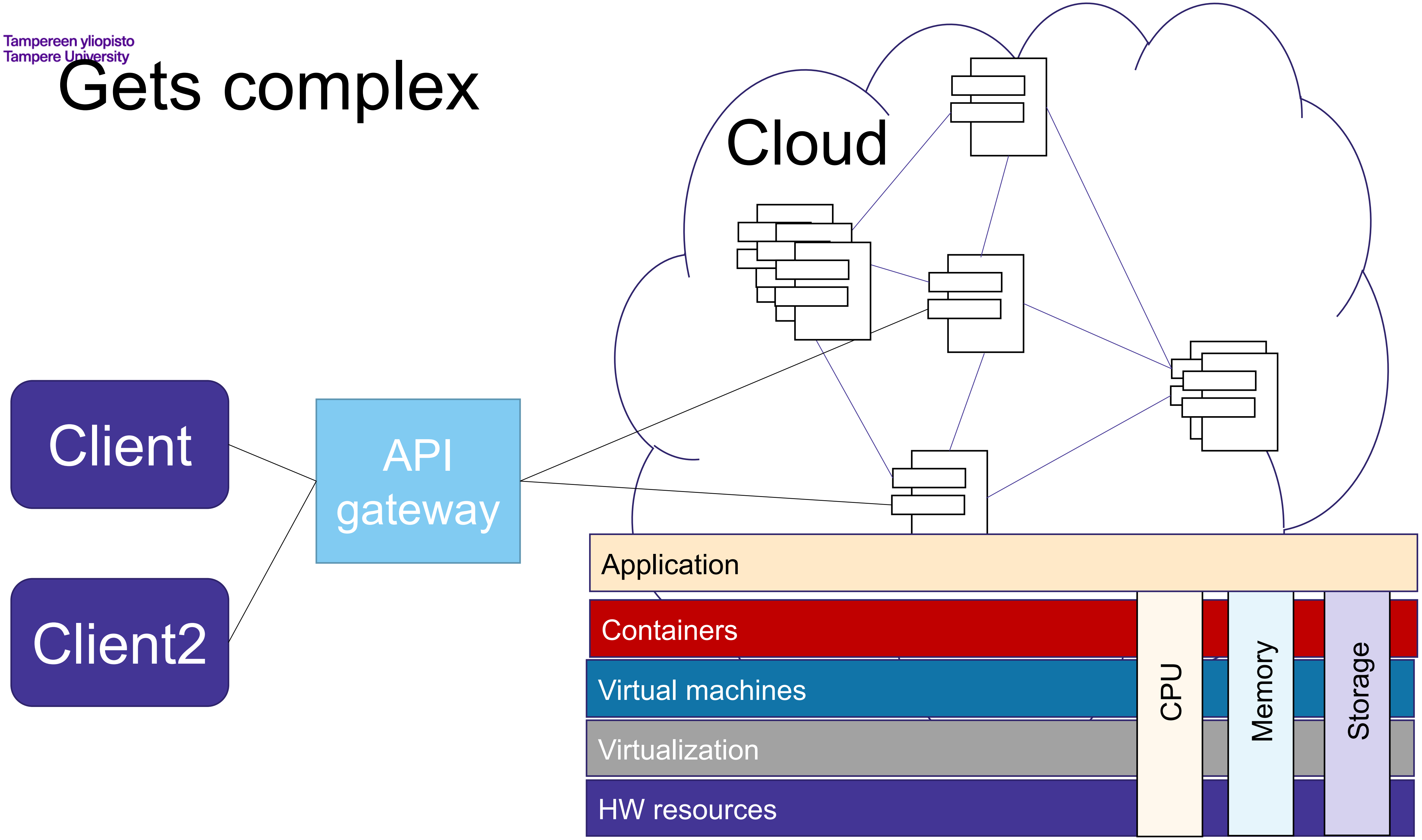
Tampereen yliopisto
Tampere University

Recall a possible microservice architecture



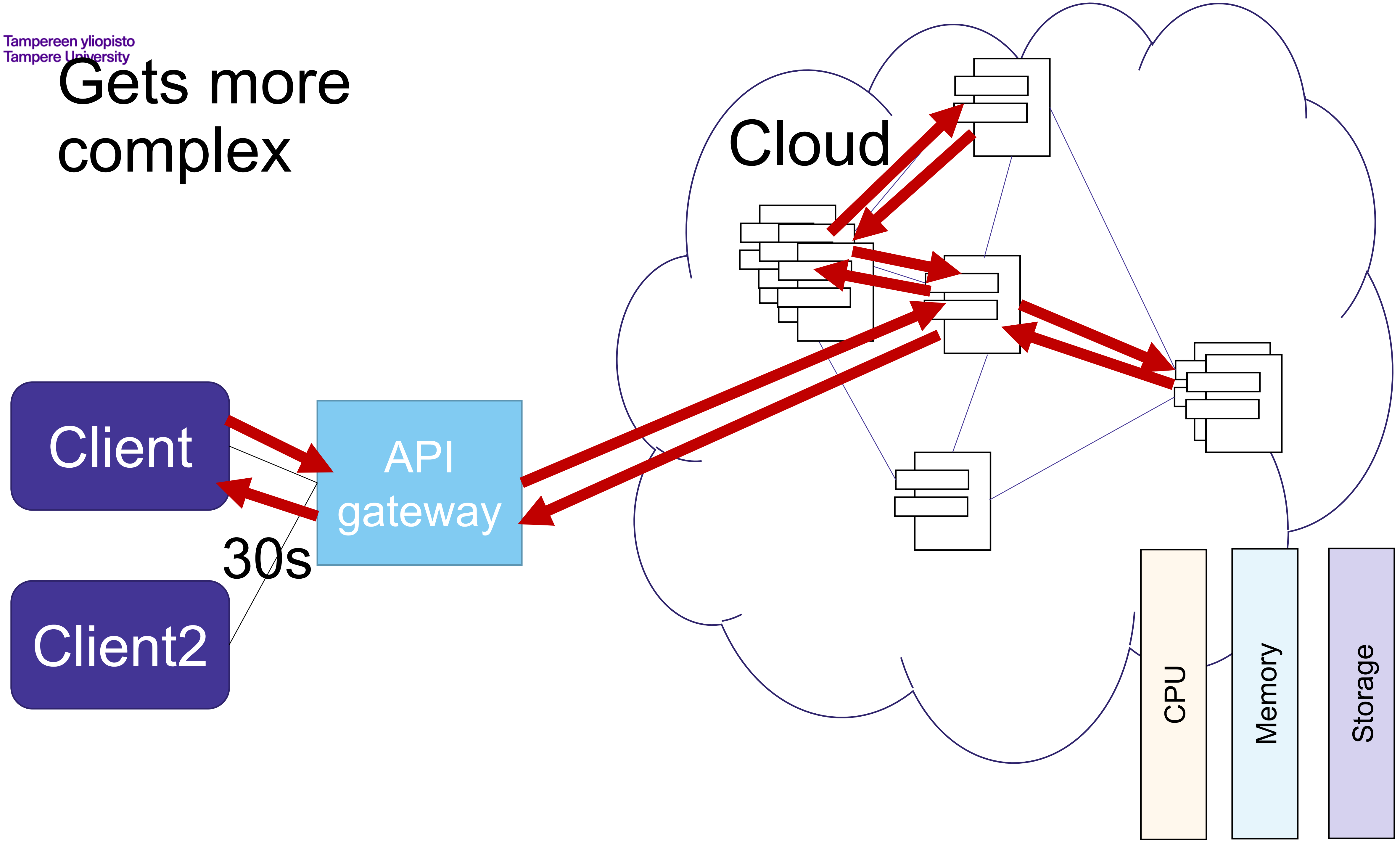
Tampereen yliopisto
Tampere University

Gets complex



Tampereen yliopisto
Tampere University

Gets more complex

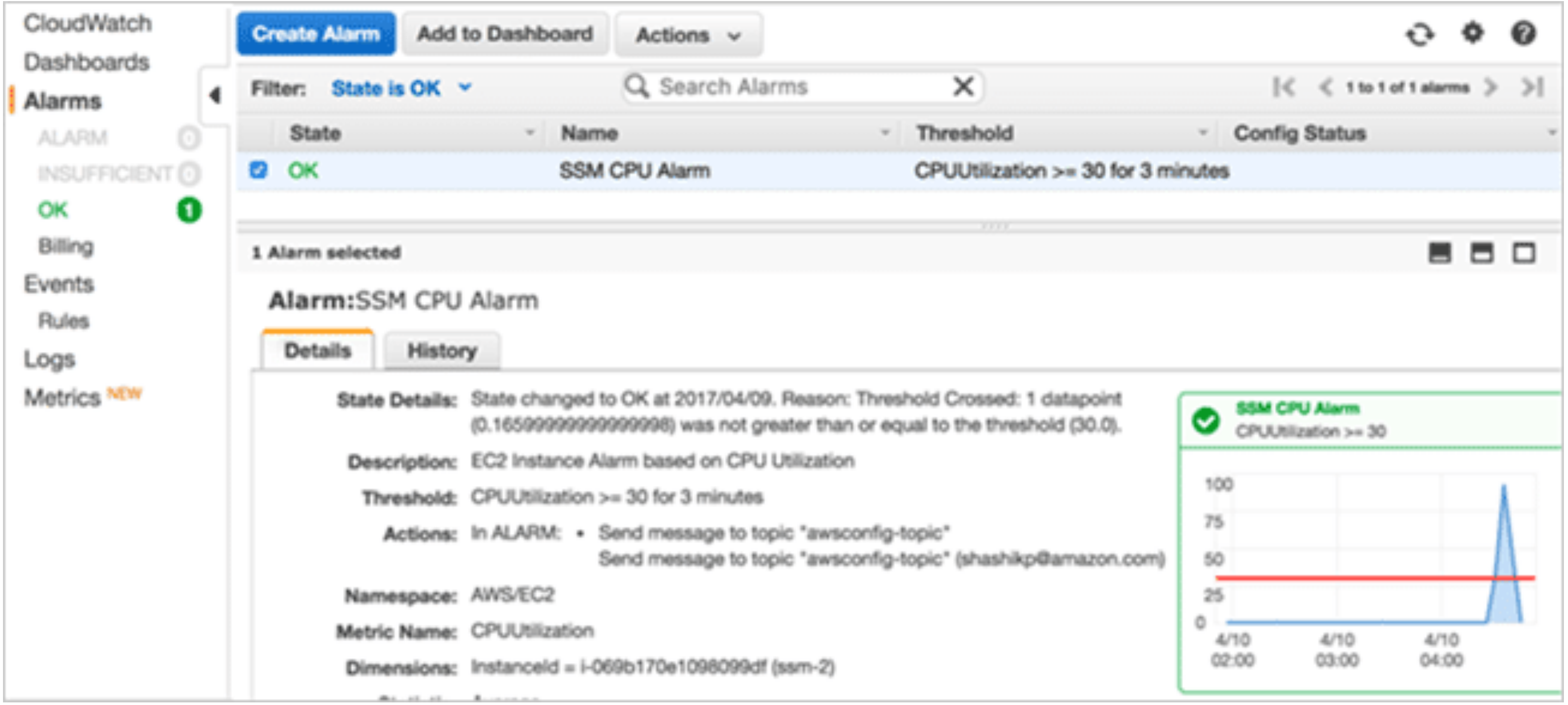


Is the problem

- In your application code?
- In your intrastructure code?

Tampereen yliopisto
Tampere University

Example: Amazon CloudWatch



State	Name	Threshold	Config Status
OK	SSM CPU Alarm	CPUUtilization >= 30 for 3 minutes	

Alarm:SSM CPU Alarm

State Details: State changed to OK at 2017/04/09. Reason: Threshold Crossed: 1 datapoint (0.16599999999999998) was not greater than or equal to the threshold (30.0).

Description: EC2 Instance Alarm based on CPU Utilization

Threshold: CPUUtilization >= 30 for 3 minutes

Actions: In ALARM: • Send message to topic "awsconfig-topic"
Send message to topic "awsconfig-topic" (shashikp@amazon.com)

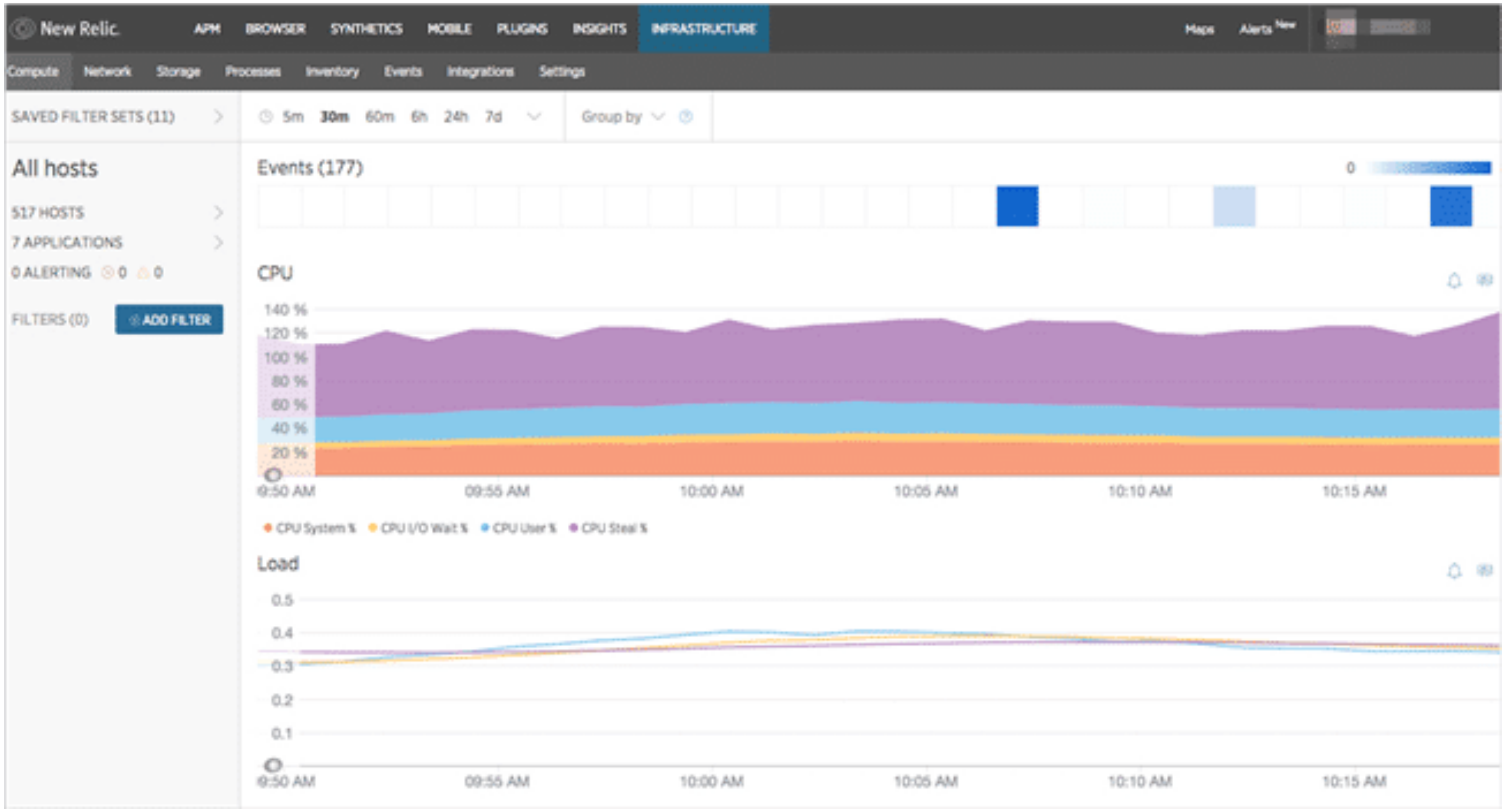
Namespace: AWS/EC2

Metric Name: CPUUtilization

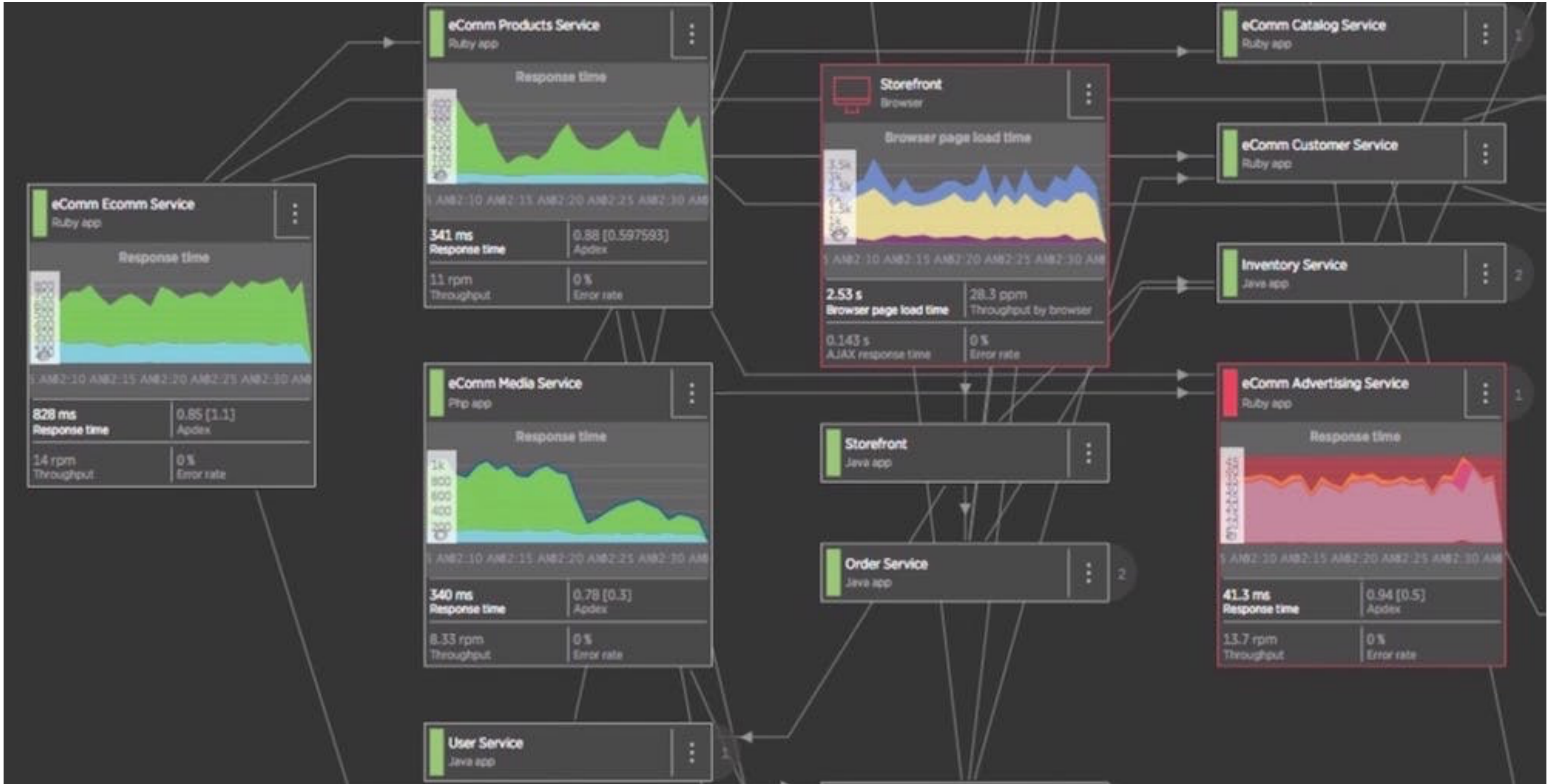
Dimensions: InstanceId = i-069b170e1098099df (ssm-2)

Tampereen yliopisto
Tampere University

Example: New Relic

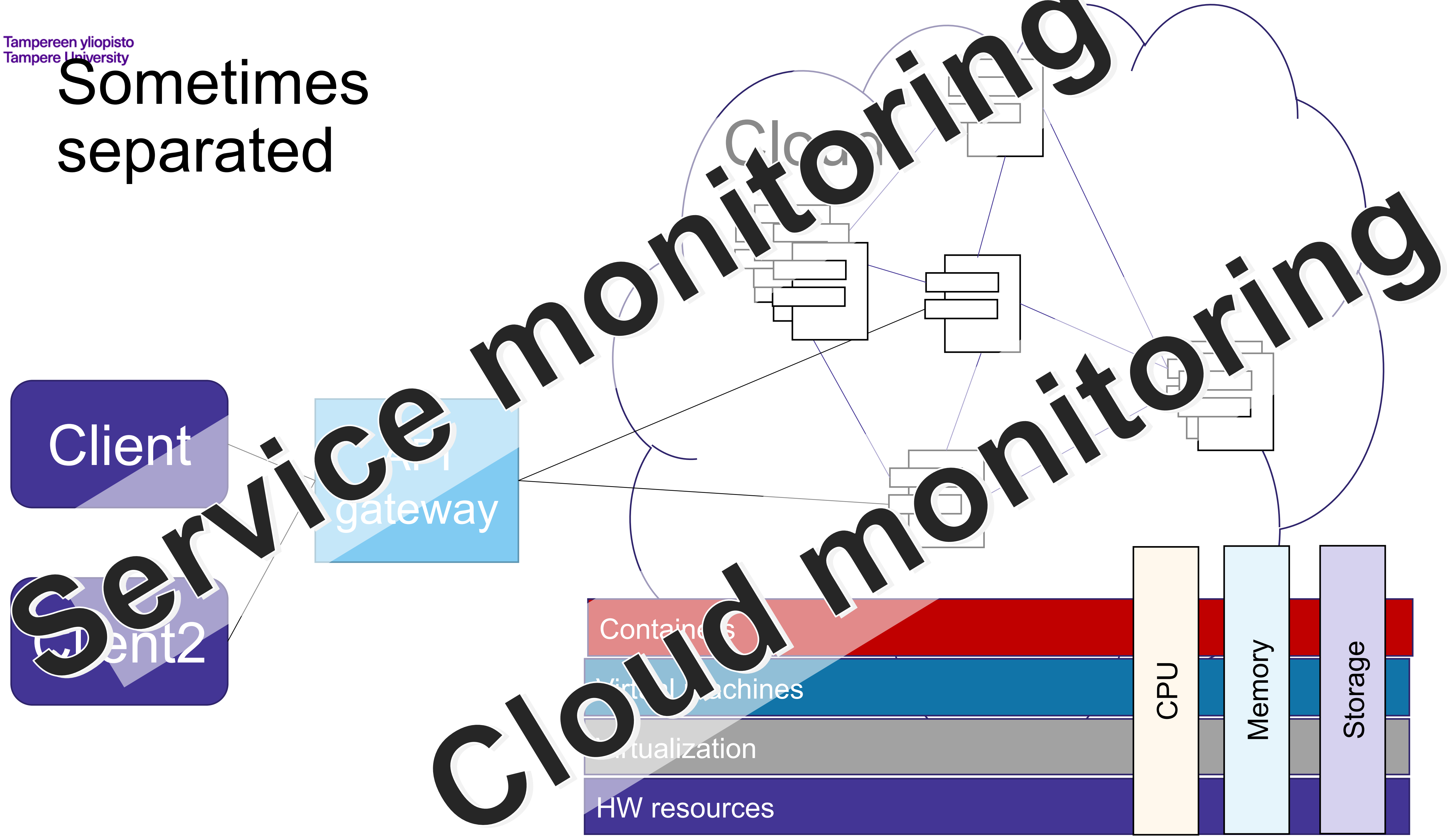


New Relic



eComm Products Service
Ruby app
Response time
341 ms
Response time
0.88 [0.597593]
Apdex
11 rpm
Throughput
0 %
Error rate
eComm Ecomm Service
Ruby app
Response time
828 ms
Response time
0.85 [1.1]
Apdex
14 rpm
Throughput
0 %
Error rate
Storefront
Browser
Browser page load time
2.53 s
Browser page load time
28.3 ppm
Throughput by browser
0.143 s
AJAX response time
0 %
Error rate
eComm Catalog Service
Ruby app
eComm Customer Service
Ruby app
Inventory Service
Java app
eComm Media Service
Php app
Response time
340 ms
Response time
0.78 [0.3]
Apdex
8.33 rpm
Throughput
0 %
Error rate
Storefront
Java app
Order Service
Java app
eComm Advertising Service
Ruby app
Response time
41.3 ms
Response time
0.94 [0.5]
Apdex
13.7 rpm
Throughput
0 %
Error rate
User Service
Java app

Sometimes separated



Tampereen yliopisto
Tampere University

What might be monitored

- Availability
- MTBF (mean time between failures)
- Throughput
- Response time
- Latency
- Security threats
- Scalability
- Cost per customer
- Usage (recall A/B testing)
- Application specific measures

In your project

- *(Optional)* implement monitoring and logging for troubleshooting. This should be a separate service that the user can use through browser. It should show at least start time of the service, number of requests it has received after start.

- Waiting for creative solutions !

Tampereen yliopisto
Tampere University

Couple of cloud quality ”terms”

- QoS (Quality of Service): measure of capacity, performance etc.
- SLA (Service Level Agreement): an agreement between provider client about capacity, performance etc.
  - Or at least promise

https://www.wired.com/insights/2011/12/service-level-agreements-in-the-cloud-who-cares/
(Thomas J. Trappler; “If It’s in the Cloud, Get it on Paper: Cloud Computing Contract Issues”
https://er.educause.edu/articles/2010/6/if-its-in-the-cloud-get-it-on-paper-cloud-computing-contract-issues.)

- Codifies the specific parameters and minimum levels required for each element of the service, as well as remedies for failure to meet those requirements.
- Affirms your institution’s ownership of its data stored on the service provider’s system, and specifies your rights to get it back.
- Details the system infrastructure and security standards to be maintained by the service provider, along with your rights to audit their compliance.
- Specifies your rights and cost to continue and discontinue using the service.

Tampereen yliopisto
Tampere University

Possible content

- Availability (e.g. 99.99% during work days, 99.9% for nights/weekends)
- Performance (e.g. maximum response times)
- Security / privacy of the data (e.g. encrypting all stored and transmitted data)
- Disaster Recovery expectations (e.g. worse case recovery commitment)
- Location of the data (e.g. consistent with local legislation)
- Access to the data (e.g. data retrievable from provider in readable format)
- Portability of the data (e.g. ability to move data to a different provider)
- Process to identify problems and resolution expectations (e.g. call center)
- Change Management process (e.g. changes – updates or new services)
- Dispute mediation process (e.g. escalation process, consequences)
- Exit Strategy with expectations on the provider to ensure smooth transition