



Advanced Topics in Computer Science (CS3001) Seminars

Topic 4: Microservices

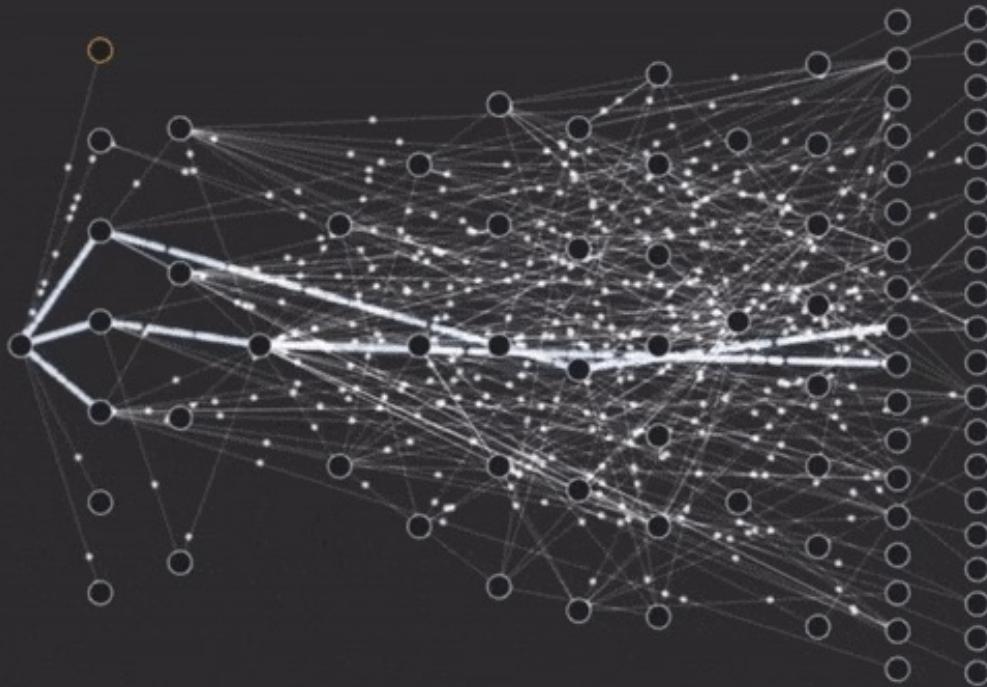
Dr Rumyana Neykova

Seminar outline

- Recap
- A Poll Everywhere test
- Case Study Cool Socks company
 - Summary
 - Main questions
- Case Study from a Research paper
 - Summary
 - Main questions



Part 1: Recap



Monolith



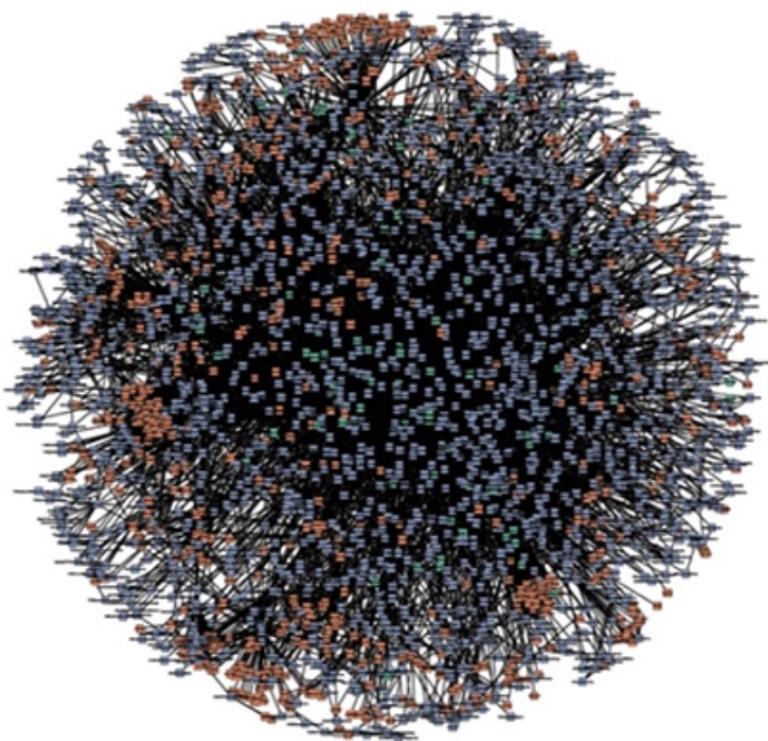
How to improve?

Microservice architecture to the rescue

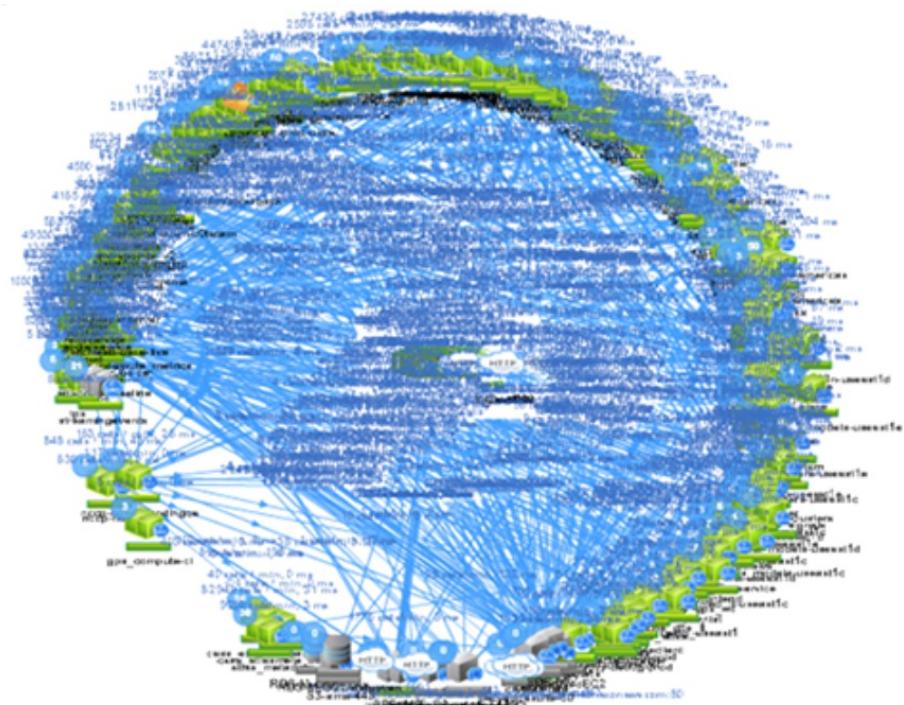


Microservices are NOT a silver bullet

The Death Star architecture



amazon.com®



NETFLIX

Part 1: Poll Everywhere

Topic 3CS Quiz

When survey is active, respond at PollEv.com/rumyananeyko590

0 surveys done

↻ 0 surveys underway

Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app

Topic 4CS Quiz

When survey is active, respond at pollev.com/rumyananeyko590

0 done

 0 underway

Powered by  **Poll Everywhere**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

Monolithic architecture is based on what paradigm?

Object-oriented programming.

The integration of different parts (view, model, data) into a single unit.

The separation of service logic from controller logic.

Non-modular paradigm.

The division into tiers – presentation, logic, and data.

Which of the following are of the main principles of a microservice architecture? 1. Independent deployment; 2) Small life cycle; 3) Composed application; 4) Functional module; 5) Well-defined interface.

- 1 and 2
- 2 and 4
- 3, 4 and 5
- 1 and 5
- All of them

How can the scope of changes made in a microservice be reduced?

Do not change the interface of the service.

Modify parameter types, but not method or function names.

Only make changes to the external interface.

Coordinate all code changes among teams.

Implement the microservice in a few lines of code.

Which of the following statements best describes microservice design?

Design is dictated by business domain.

Design is dictated by technology.

Design is governed by the expertise of the different teams.

Design depends on the Enterprise Service Bus, or ESB.

Design is based on one of the tiers: view, model, controller, data.

Which pattern takes a broken component out of service so time-outs do not bog down the consuming services?

Service discovery

Circuit breaker

Bulkhead

API Gateway

Chaos Monkey

Which pattern\tool isolates a failure so it does not cause the whole system to fail?

Circuit breaker

Service registry

API Gateway

The Simian Army

Bulkhead

A blue-green deployment is a strategy to achieve which goal?

Zero downtime deployment.

Continuous deployment.

Continuous integration.

Fault tolerance.

Small team size.

What is a container in the context of a microservice architecture?

A container encapsulates all the software dependencies associated with running a program.

A container encapsulates all microservices for a given application.

A virtual machine that runs different operating system and is used for testing of microservice.

A netflix invention designed to keep their system alive even when a failure occurs.

All of the above.

What is 'continuous integration'?

A way of verifying that failure will not cause disruption in production.

The routine of continuously testing new code.

Deploying new code changes to the production environment immediately.

A way of automating and integrating different services into a new service.

The practice of merging code changes to a mainline as soon as they are ready.

Which of the following is considered as advantage of microservices (in comparison with monolithic architecture)?

Services must find instances of cooperating services in the operational environment.

Frequent deployment of new changes.

Microservices must be written in the same languages.

Latency can lead to improved performance.

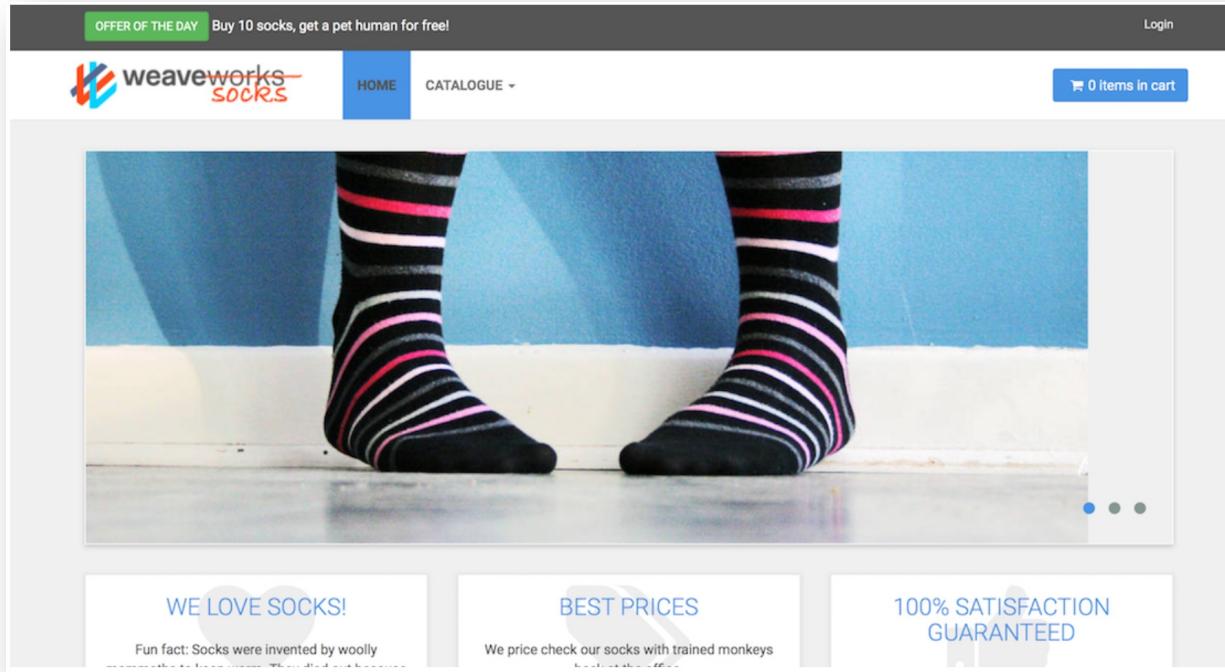
Identifying the business boundaries is easier.



Part 2: Use case

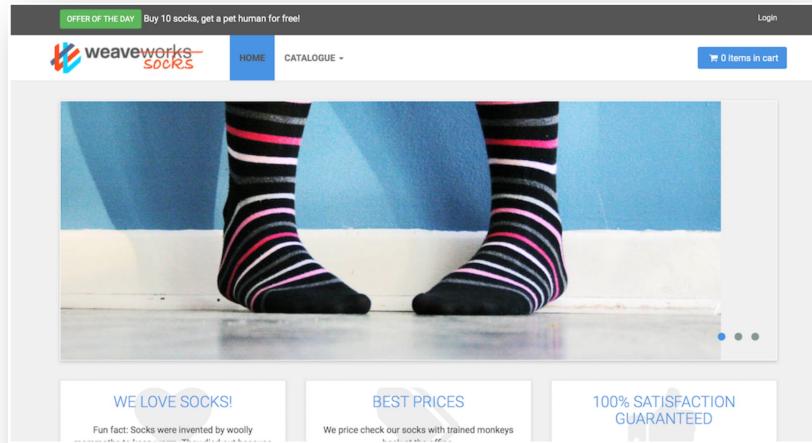
CoolSocks features

CoolSocks is an online store for selling socks with funny patterns. In addition to selling and delivering goods, the store offers recommendation system based on what other users have bought, a wish list (users can save a product in their profile without buying it), users also write reviews and are incentivized by earning points for their reviews



Problems with the current architecture

CoolSocks are a growing business and are expanding to several new counties, increasing their user base. At the same time, their team size is growing and has reached the 100 employees mark. CoolSocks is built using monolithic architecture, their whole application is written in Java with MySql database. The codebase is thousands of lines of code and it takes approximately two minutes to compile the project. CoolSocks releases a new feature every quarter, while their CEO wants to release a feature every week. CoolSocks team has also observed that more users buy socks in the evening, while in the morning users use the website to write reviews.



Use Case: Questions

- What are the technical challenges that the company is currently facing?
- Which of CoolSocks's problems can be addressed by shifting to microservice architecture?
- How a microservice architecture can help with 24/7 availability of the CoolSocks website?
- Draw an architecture diagram outlining the new components in the system and how they are connected. Explain each component with few words.

Use Case: Questions

- The recommendation system uses a third-party API that often returns an error. How will you ensure that an error in the recommendation system will not bring down the whole website?
- What design patterns can be used to make the system easy to deploy?
- What are the risks of moving to microservice architecture? Be specific and give examples.
- How will you split the teams in the company?
- What are the challenges in implementing the solution?



Part 3: FX Core case study

Microservices: Migration of a Mission Critical System

Manuel Mazzara^{id}, Nicola Dragoni^{id}, Antonio Bucchiarone^{id}, Alberto Giaretta^{id}, Stephan T. Larsen, and Schahram Dustdar^{id}, *Fellow, IEEE*

Abstract—An increasing interest is growing around the idea of microservices and the promise of improving scalability when compared to monolithic systems. Several companies are evaluating pros and cons of a complex migration. In particular, financial institutions are positioned in a difficult situation due to the economic climate and the appearance of agile competitors that can navigate in a more flexible legal framework and started their business since day one with more agile architectures and without being bounded to outdated technological standard. In this paper, we present a real world case study in order to demonstrate how scalability is positively affected by re-implementing a monolithic architecture (MA) into a microservices architecture (MSA). The case study is based on the *FX Core* system, a mission critical system of Danske Bank, the largest bank in Denmark and one of the leading financial institutions in Northern Europe. The technical problem that has been addressed and solved in this paper is the identification of a repeatable migration process that can be used to convert a real world Monolithic architecture into a Microservices architecture in the specific setting of financial domain, typically characterized by legacy systems and batch-based processing on heterogeneous data sources.

Index Terms—Service computing, software architecture, scalability, microservices

Use Case: Questions

1. Why was scalability important for Danske Bank?
2. What were the existing **scalability solutions** in Danske Bank? Why were they not enough? Create a list with existing solutions that worked and solutions that did not work.
3. What **problems** was Danske Bank facing. Pick three and explain.
4. Choose three **changes** that were introduced in the system **during the migration?**
5. What is **orchestration** and why was it introduced in the system?
6. Did **containerization** bring any benefits. Why was it introduced?
7. What **problems** were **solved by migrating** to MSA. How were they solved?
8. What **aspects of the architecture were not improved** by the migration?
9. Did migration to MSA improved the **scalability** of the system. How is that **evidenced** in the paper?
10. Did the migration to MSA affected the **fault-tolerance** of the system? How?