

The Maersk Mc-Kinney Møller Institute

# Advanced Topics in Software Architecture (E23)

## Introduction

**SDU** Sune Chung Jepsen and Torben Worm

September 2023

sdudk

#sdudk

1

1

The Maersk Mc-Kinney Møller Institute

# Agenda

- Motivation
- Practical details
- Course structure
  - Learning objectives
  - Schedule
  - Exam
- Expectations
  - Itslearning
  - Preparation and exercises
- What is software architecture
- Why is software architecture important

**SDU** Sune Chung Jepsen and Torben Worm

September 2023

sdudk

#sdudk

2

2


The Maersk Mc-Kinney Møller Institute

## What is your understanding of software architecture?

→ Think (1 minute)

→ Pair (3 minutes)

→ Share

SDU  Sune Chung Jepsen and Torben Worm

September 2023

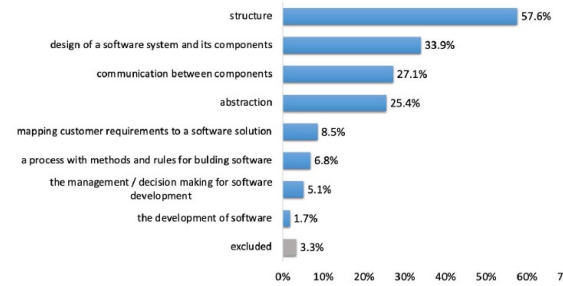
3

#sdumk


3

The Maersk Mc-Kinney Møller Institute

## What is understood by software architecture in practice?



Definition	Percentage
structure	57.6%
design of a software system and its components	33.9%
communication between components	27.1%
abstraction	25.4%
mapping customer requirements to a software solution	8.5%
a process with methods and rules for building software	6.8%
the management / decision making for software development	5.1%
the development of software	1.7%
excluded	3.3%

SDU  Sune Chung Jepsen and Torben Worm

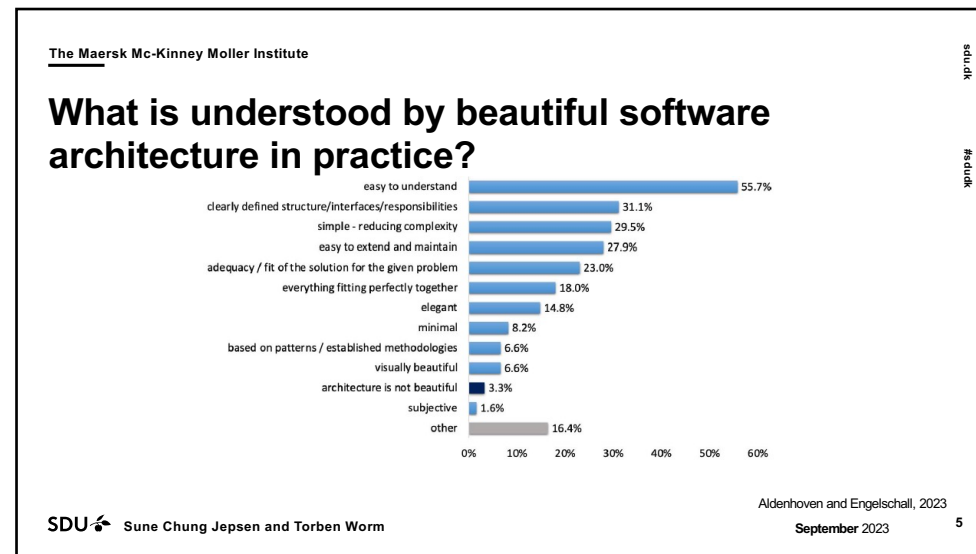
Aldenhoven and Engelschall, 2023

September 2023

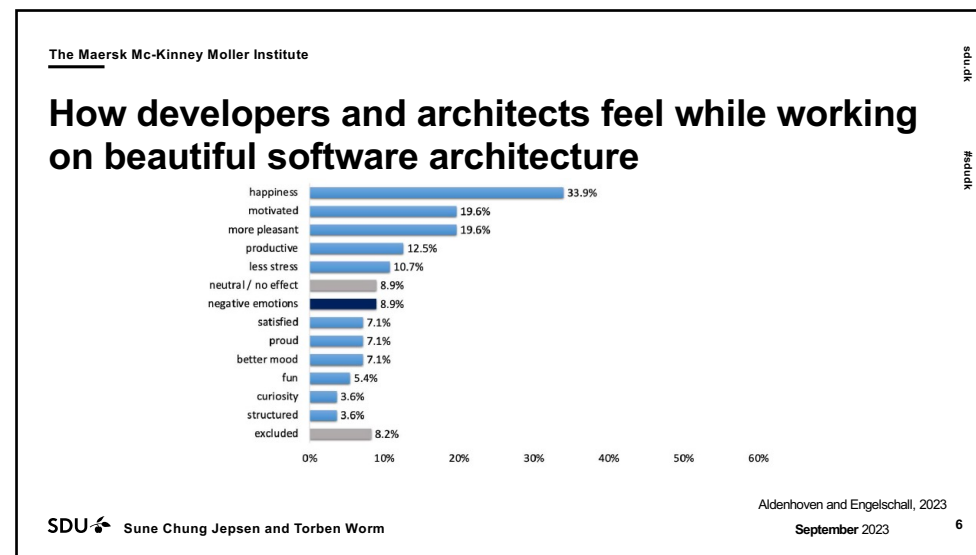
4

#sdumk

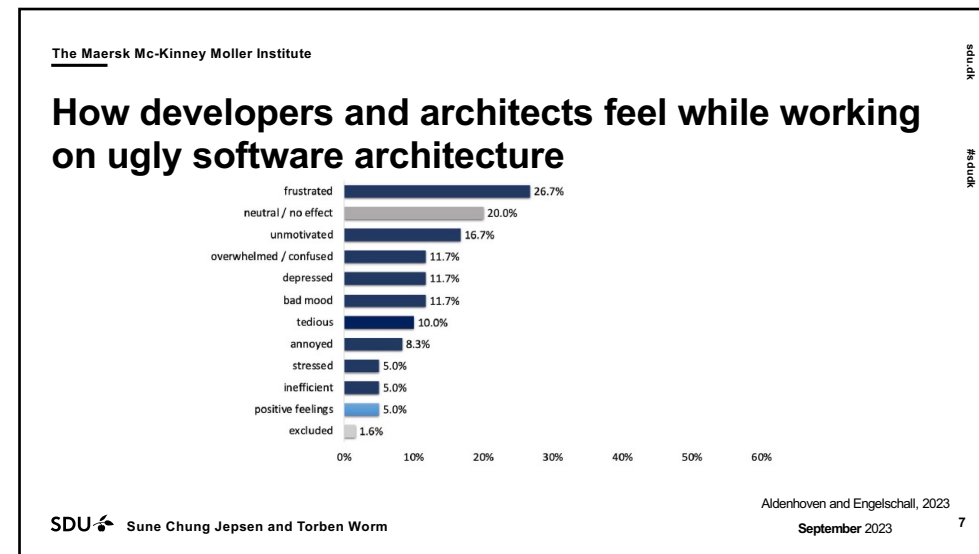
4



5



6



7

The Maersk Mc-Kinney Møller Institute

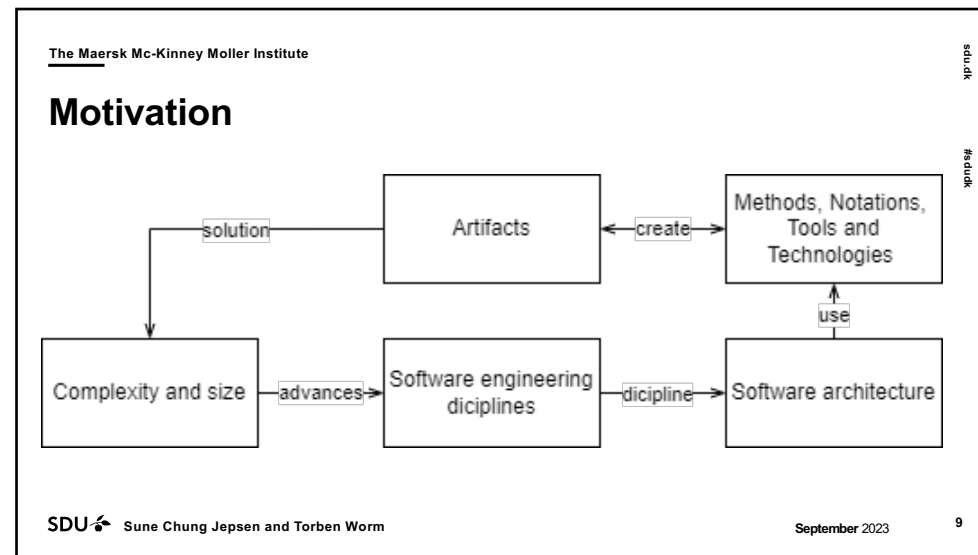
## How?

- Requirements engineering: The system requirements should be understood and clearly defined to enable the design of a beautiful software architecture.
- Focus on holistic consistency: Defining a goal for the system helps to ensure holistic consistency in the design. The “feeling of everything fitting perfectly together” can be achieved by concentrating on holistic consistency. No compromises or workarounds should be allowed in the design of the architecture.
- Abstraction: The system’s structure has to be abstracted correctly.
- Simplify: Complex solutions should be visualized and implemented as simply as possible.
- Plan for future change: This helps the system to stay easily extendable and maintainable over time.
- Iteration: Start with a first draft or prototype and iteratively refine it.
- Be creative: Use all your creative capabilities to design beautiful software architecture. Inspiration can be drawn from experience, software patterns, or other sources, if needed.

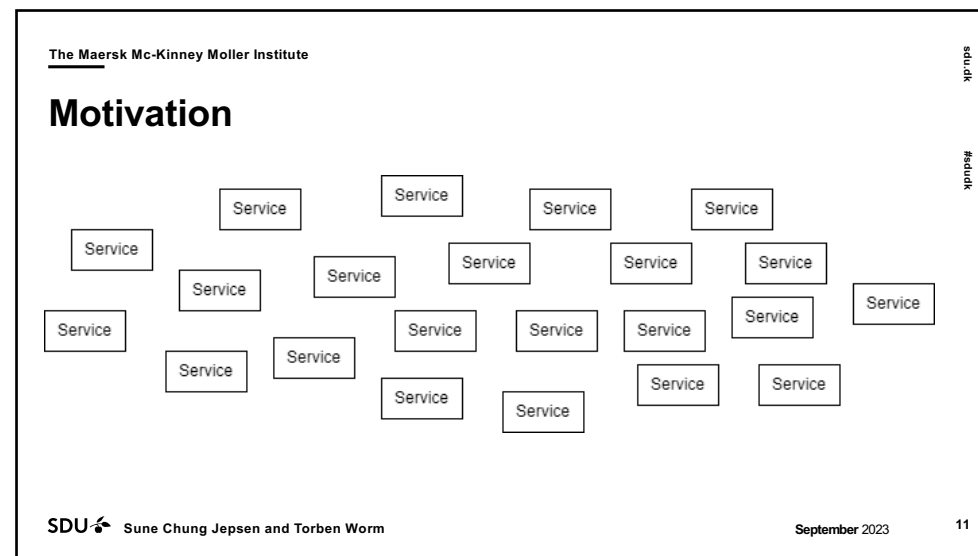
SDU Sune Chung Jepsen and Torben Worm

Aldenhoven and Engelschall, 2023  
September 2023

8



9



11

The Maersk Mc-Kinney Møller Institute

## Motivation

sdudk #sdudk

SDU Sune Chung Jepsen and Torben Worm

September 2023 12

12

The Maersk Mc-Kinney Møller Institute

## Motivation

sdudk #sdudk

SDU Sune Chung Jepsen and Torben Worm

September 2023 13

13

The Maersk Mc-Kinney Møller Institute

## Motivation

SDU Sune Chung Jepsen and Torben Worm September 2023 14

14

The Maersk Mc-Kinney Møller Institute

## Motivation

SDU Sune Chung Jepsen and Torben Worm September 2023 15

15

The Maersk Mc-Kinney Moller Institute

## Motivation

SDU Sune Chung Jepsen and Torben Worm

September 2023

16

16

The Maersk Mc-Kinney Moller Institute

## Motivation

SDU Sune Chung Jepsen and Torben Worm

September 2023

17

17



The Maersk Mc-Kinney Møller Institute

## Industrial Example

- Manufacturer of steel products
- Runs 24/7
- Dependent on stable software
- Currently production is halted for maintenance (e.g. software updates and machine adjustments) on scheduled times two times yearly
- What if we want to be able to update software outside the scheduled time-slots?

SDU Sune Chung Jepsen and Torben Worm

18

The Maersk Mc-Kinney Møller Institute

## Motivation

```

graph TD
    CS[Complexity and size] -- advances --> SED[Software engineering disciplines]
    SED -- discipline --> SA[Software architecture]
    SA -- use --> MNT[Methods, Notations, Tools and Technologies]
    MNT -- create --> ART[Artifacts]
    ART -- solution --> CS
    CS -- example --> AV[Availability]
    CS -- example --> PE[Performance]
    CS -- example --> SE[Security]
  
```

SDU Sune Chung Jepsen and Torben Worm

September 2023 20

20

The Maersk Mc-Kinney Møller Institute

#sdutk

## Topics

- Tools and technologies
- Software architecture patterns
- Software architecture documentation
- Quality attributes
- Formal software architecture
- Evaluation software architecture

SDU
Sune Chung Jepsen and Torben Worm
September 2023
21

21

The Maersk Mc-Kinney Møller Institute

#sdutk

## Why This Course

It all boils down to:

- Document software architecture
- Specify software architectural requirements
- Select software architecture
- Design and evaluate software architecture
- Implement software architecture
- Software architecture and organization

→ Reasoning about software architecture

SDU
Sune Chung Jepsen and Torben Worm
September 2023
22


22

The Maersk Mc-Kinney Møller Institute

#sdu.dk  
#sdu.dk

## This is a new course

- So please provide feedback as we move on about what works and what doesn't
- Don't hesitate to talk to me or the instruktors if you have ideas or challenges
- Use also the fora on itslearning (under "General Course Information")


SDU  Sune Chung Jepsen and Torben Worm
September 2023
23

23


The Maersk Mc-Kinney Møller Institute

#sdu.dk  
#sdu.dk

## Practical details




- Torben Worm ([tow@mmpi.sdu.dk](mailto:tow@mmpi.sdu.dk))
- Lectures
  - Fridays 8.15-10.00, U181
- Exercises
  - Fridays 10.15-11.45, U181

SDU  Sune Chung Jepsen
September 2023

24

The Maersk Mc-Kinney Møller Institute


## Teaching Assistants



Nicolai Krogager Asmussen

[niasm19@student.sdu.dk](mailto:niasm19@student.sdu.dk)


Software Engineering Master Student



Tobias Kristensen

[tokri19@student.sdu.dk](mailto:tokri19@student.sdu.dk)

Software Engineering Master Student

SDU  Sune Chung Jepsen and Torben Worm

September 2023 25


25

The Maersk Mc-Kinney Møller Institute

## Learning Objectives

Knowledge	Skills	Competences
<ul style="list-style-type: none"> <li>• Explain tools and technologies for implementing software architecture</li> <li>• Explain and discuss software architecture documentation</li> <li>• Analyze and specify architectural requirements for software architecture</li> <li>• Explain and argue for software architecture and associated qualities attributes and architectural problems</li> <li>• Explain methods for agile architecture and architectural evolution</li> <li>• Describe advanced software architecture topics to support software architecture processes and modeling</li> </ul>	<ul style="list-style-type: none"> <li>• Select and combine tools and technologies to implement software architecture</li> <li>• Analyze, design, and develop architectural prototypes of software architecture to achieve quality attributes</li> <li>• Document, describe, and communicate modern software architectures</li> <li>• Analyze existing software architectures and identify architectural problems</li> </ul>	<ul style="list-style-type: none"> <li>• Ability to apply software architectures for different quality attributes using tools and technologies</li> <li>• Ability to analyze and document software architectures and motivate the usage of adequate software architectures to obtain relevant quality attributes</li> <li>• Ability to devise and apply software architecture in agile environments</li> </ul>

<https://odin.sdu.dk/sitecore/index.php?a=fagbesk&id=130435&listid=18884&lang=en>

SDU  Sune Chung Jepsen and Torben Worm

September 2023 26

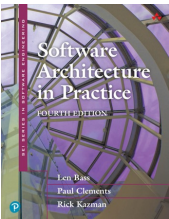
26

**The Maersk Mc-Kinney Møller Institute**

## Schedule

Course plan  
→ Lectures  
→ Literature

Located on itslearning under the "General Course Information" plan.



**SDU** Sune Chung Jepsen and Torben Worm

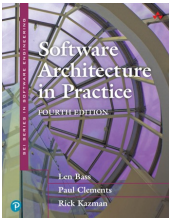
September 2023 27

27

**The Maersk Mc-Kinney Møller Institute**

## Usage of itslearning

→ Overview  
→ Plans  
→ Resources  
→ Exercises  
→ Q&A Forum  
→ Upload profile picture in itslearning



**SDU** Sune Chung Jepsen and Torben Worm

September 2023 30


30

The Maersk Mc-Kinney Møller Institute

## Preparation and exercises

Pillars of the course

- Lectures
  - Discussions and active participation, e.g. reflecting questions
  - Discussion will work better if you prepare before the lectures, i.e.
    - Read literature
    - Prepare for discussion by using your study group
- Exercises
  - Train and become familiar with the subjects
  - Exercises will build on each other
  - Pitches and presentations
  - Peer reviews

SDU  Sune Chung Jepsen and Torben Worm


September 2023 31

sdumk #sdumk

31

The Maersk Mc-Kinney Møller Institute

# Software Architecture and Technology Overview

SDU  Sune Chung Jepsen and Torben Worm

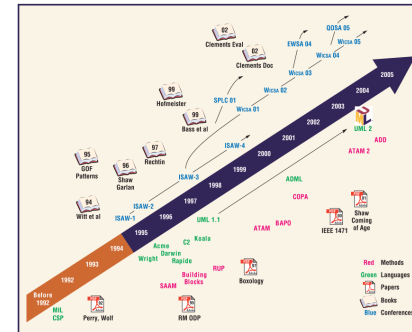
September 2023 35

sdumk #sdumk

35

The Maersk Mc-Kinney Moller Institute

## The Past, Present, and Future of Software Architecture



The past, present, and future for software architecture

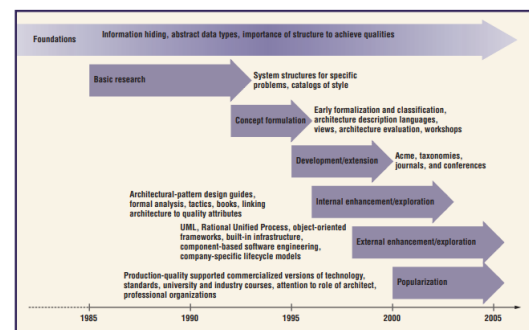
September 2023

SDU Sune Chung Jepsen and Torben Worm

36

The Maersk Mc-Kinney Moller Institute

## The Golden Age of Software Architecture



The golden age of software architecture

September 2023

SDU Sune Chung Jepsen and Torben Worm

37

## The Coming-of-Age of Software Architecture Research

→ Address different classes of problems

→ The way we work with technology

Research Setting	Sample Questions	Examples from abstract data types
Feasibility	Is there an X, and what is it? Is it possible to accomplish X at all?	Is it possible to describe the relation among components of a software system [16]?
Characterization	What are the important characteristics of X? What is X like? What, exactly, do we mean by X? What are the varieties of X, and how are they related?	What is the important information to share and to hide about a component [49]?
Method-/Means	How can we accomplish X? What is a better way to accomplish X? How can I automate doing X?	How can we incorporate abstract data types in a programming language [37][64]?
Generalization	Is X always true of Y? Given X, what will Y be?	What is a formal relation between the specification of an abstract data type and its implementation [23]?
Selection	How do I decide between X and Y?	What organization should I choose for the user interface component of a system [33]?

The coming-of-age of software architecture research

September 2023

## Programming Language

→ How do we select programming language to a given problem?

Jul 2023	Jul 2022	Change	Programming Language	Rating	Change
1	1		Python	13.42%	-0.0%
2	2		C	11.56%	-1.57%
3	4	▲	C++	10.80%	+0.79%
4	3	▼	Java	10.50%	-1.09%
5	5		C#	6.87%	+1.21%
6	7	▲	JavaScript	3.11%	+1.34%
7	6	▼	Visual Basic	2.80%	-2.07%
8	9	▲	SQL	1.48%	-0.36%
9	11	▲	PHP	1.41%	+0.21%
10	20	▲	MATLAB	1.28%	+0.53%
11	18	▲	Fortran	1.20%	+0.49%
12	21	▲	Scratch	1.07%	+0.35%
13	12	▼	Go	1.07%	-0.07%

<https://www.tiobe.com/tiobe-index/>



## Database

→ How do we select database to a given problem?

420 systems in ranking, August 2023

Rank	Rank			DBMS	Database Model	Score		
	Aug 2023	Jul 2023	Aug 2022			Aug 2023	Jul 2023	Aug 2022
1.	1.	1.	1.	Oracle	Relational, Multi-model	1242.10	-13.91	-18.70
2.	2.	2.	2.	MySQL	Relational, Multi-model	1130.45	-19.89	-72.40
3.	3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	920.81	-0.78	-24.14
4.	4.	4.	4.	PostgreSQL	Relational, Multi-model	620.38	+2.55	+2.38
5.	5.	5.	5.	MongoDB	Document, Multi-model	434.49	-1.00	-43.17
6.	6.	6.	6.	Redis	Key-value, Multi-model	162.97	-0.80	-13.43
7.	8.	8.	8.	Elasticsearch	Search engine, Multi-model	139.92	+0.33	-15.16
8.	7.	7.	7.	IBM Db2	Relational, Multi-model	139.24	-0.58	-17.99
9.	9.	9.	9.	Microsoft Access	Relational	130.34	-0.38	-16.16
10.	10.	10.	10.	SQLite	Relational	129.92	-0.27	-8.95
11.	11.	13.	13.	Snowflake	Relational	120.62	+2.94	+17.50
12.	12.	11.	11.	Cassandra	Wide column, Multi-model	107.38	+0.86	-10.76
13.	13.	12.	12.	MariaDB	Relational, Multi-model	98.65	+2.55	-15.24
14.	14.	14.	14.	Splunk	Search engine	88.98	+1.87	-8.46
15.	16.	15.	15.	Amazon DynamoDB	Multi-model	83.55	+4.75	-3.71
16.	15.	16.	16.	Microsoft Azure SQL Database	Relational, Multi-model	79.51	+0.55	-6.67
17.	17.	17.	17.	Hive	Relational	73.35	+0.48	-5.31
18.	18.	22.	22.	Databricks	Multi-model	71.34	+2.87	+16.72
19.	19.	18.	18.	Teradata	Relational, Multi-model	61.31	+1.06	-7.76
20.	20.	24.	24.	Google BigQuery	Relational	53.90	-1.52	+3.87
21.	21.	23.	23.	FileMaker	Relational	53.85	+0.53	+0.73
22.	22.	19.	19.	Neo4j	Graph	51.42	-0.64	-7.93

<https://db-engines.com/en/ranking>

September 2023

## Container

→ How do we select container to a given problem?

Technology	Type	URL
Linux Container (LXC)	SC, AC	<a href="https://linuxcontainers.org/">https://linuxcontainers.org/</a>
OpenVZ	SC, CM	<a href="https://openvz.org/">https://openvz.org/</a>
Windows Hyper-V Container (WHC)	SC	<a href="https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/">https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/</a>
Docker	AC, CM	<a href="https://www.docker.com/">https://www.docker.com/</a>
Windows Server Container (WSC)	AC	<a href="https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/">https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/</a>
rkt	CM	<a href="https://coreos.com/rkt">https://coreos.com/rkt</a>
LXD	CM	<a href="https://linuxcontainers.org/">https://linuxcontainers.org/</a>
Amazon EC2 Container Service (ECS)	CM, OF	<a href="https://aws.amazon.com/ecs/">https://aws.amazon.com/ecs/</a>
Google Container Engine (GCE)	CM, OF	<a href="https://cloud.google.com/container-engine/">https://cloud.google.com/container-engine/</a>
Microsoft Azure Container Service (ACS)	CM, OF	<a href="https://azure.microsoft.com/en-us/services/container-service/">https://azure.microsoft.com/en-us/services/container-service/</a>
Kubernetes	OF	<a href="https://kubernetes.io/">https://kubernetes.io/</a>
Swarm	OF	<a href="https://www.docker.com/">https://www.docker.com/</a>
Marathon	OF	<a href="https://mesosphere.github.io/marathon/">https://mesosphere.github.io/marathon/</a>
Cloudify	OF	<a href="http://cloudify.co/">http://cloudify.co/</a>

Type meaning: AC application container, SC system container, CM container manager, OF orchestration framework

### Container Orchestration: A Survey

## Message-oriented Middleware

→ How do we select message-oriented middleware to a given problem?

→ Recall architecture prototypes from ADD?

→ Does it achieve the needed QA?

	MOM	MQTT	AMQP 0.9	KAFKA	ZeroMQ
<b>Features</b>					
Implementation used in benchmark	Eclipse Mosquitto	RabbitMQ	Apache Kafka	ZeroMQ	
CP: Messaging types	Pub/Sub	Pub/Sub Request/Reply Point-to-point	Pub/Sub	Pub/Sub Request/Reply	Arbitrary: TCP, IPC, etc.
CP: Technical communication realization	Broker	Broker	Broker	Brokerless / Point-to-point	
CP: Transport Protocol	TCP	TCP	TCP	Arbitrary: TCP, IPC, etc.	
Packet type	Binary defined by user	Binary defined by user	Binary defined by user	Binary defined by user	
QoS: Message delivery	At least once At most once Exactly once	Exactly once, with acknowledgement settings (To be extended in 1.0)	Exactly once	At least once Exactly once	
QoS: Load balancing, clustering	Yes*	Yes*	Yes	No	
Security: Authentication	SASL PLAIN *	SASL with challenge-response	SASL PLAIN and Kerberos	SASL full	
Security: Encryption	TLS	TLS *	SSL	CurveZMQ over TCP	
Security: Authorization	ACL *	ACL *	ACL	-	
I+S: Standardization	Yes (ISO/IEC 20922)	Yes (ISO/IEC 19464)	No	No	
I+S: Open Source	Yes	Yes	Yes	Yes	
I+S: License	Royalty-free / OASIS Standard	Royalty-free / OASIS Standard	Apache License 2.0	LGPLv3	
I+S: Programming language of clients	●	●	●	●	
I+S: Commercial support	●	●	●	●	
I+S: Active community**	●	●	●	●	
I+S: Tutorial and documentation**	●	●	●	●	
I+S: Ease of implementation **	●	●	●	●	

Message-oriented Middleware for Industrial Production Systems

## The Coming-of-Age of Software Architecture Research

*Good validation entails not only showing that the specific product of the research satisfies the idealized problem of the research setting, but also that the result helps to solve the **original motivating problem**. [...] The second proposes a new technique, applies it to a toy example, and claims a contribution; this form is particularly unsatisfactory when the example is only **tenuously related to a practical problem**.*

The Maersk Mc-Kinney Møller Institute

## Arguments


→ ... *“to give an argument” means to offer a set of reasons or evidence in support of a conclusion.*

→ *An argument is not simply a statement of certain views, and it is not simply a dispute.*

→ *Arguments are efforts to support certain views with reasons.*

→ *It is not a mistake to have strong views. The mistake is to have nothing else*

Weston, 2019

SDU  Sune Chung Jepsen and Torben Worm

September 2023 44

44

The Maersk Mc-Kinney Møller Institute


## Arguments Types

→ How do one create an argument?

→ By example

→ By analogy

→ By Authority


SDU  Sune Chung Jepsen and Torben Worm

September 2023 45

45

The Maersk Mc-Kinney Møller Institute

# Complex Systems

SDU  Sune Chung Jepsen and Torben Worm

September 2023 46

#sdumk


46

The Maersk Mc-Kinney Møller Institute

## Complexity Definition

→ Cambridge Dictionary:  
→ *"the state of having many parts and being difficult to understand or find an answer to"*

→ The DevOps Handbook – How to create world-class agility, reliability, & security in technology organizations:  
→ *"defies any single person's ability to see the system as a whole and understand how all the pieces fit together"*

SDU  Sune Chung Jepsen and Torben Worm

September 2023 47

#sdumk

47

