


The Maersk Mc-Kinney Møller Institute

# Advanced Topics in Software Architecture (E23)

## Software Architecture Patterns 1

SDU  Torben Worm and Sune Chung Jepsen

February 2023

1


sdudk  
#sdudk

1

The Maersk Mc-Kinney Møller Institute

## Agenda

- Patterns
- Exercise

SDU  Torben Worm and Sune Chung Jepsen

February 2023

2


sdudk  
#sdudk

2

The Maersk Mc-Kinney Møller Institute

# What is your experience with patterns?

sdudk #sdudk

SDU  Torben Worm and Sune Chung Jepsen February 2023 3


3

The Maersk Mc-Kinney Møller Institute

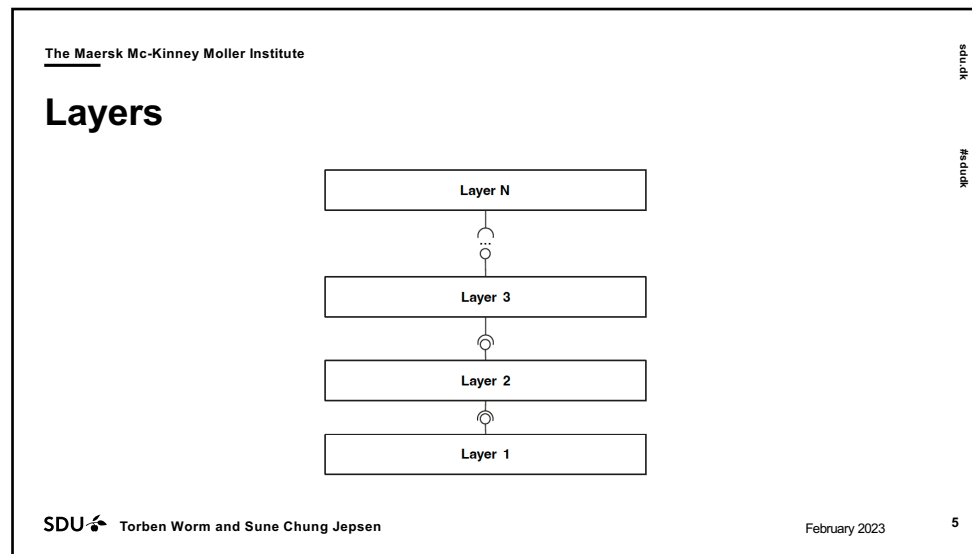
# What is a pattern

- problem-solution pairs that occur in a given context and are affected by it
  - context-problem-solution triplet
- Architectural Patterns
  - context-problem-solution triplet
- Architectural styles
  - components, connectors, and issues related to control and data flow
- It is usually not clear when a pattern is “big” enough to be considered architectural.

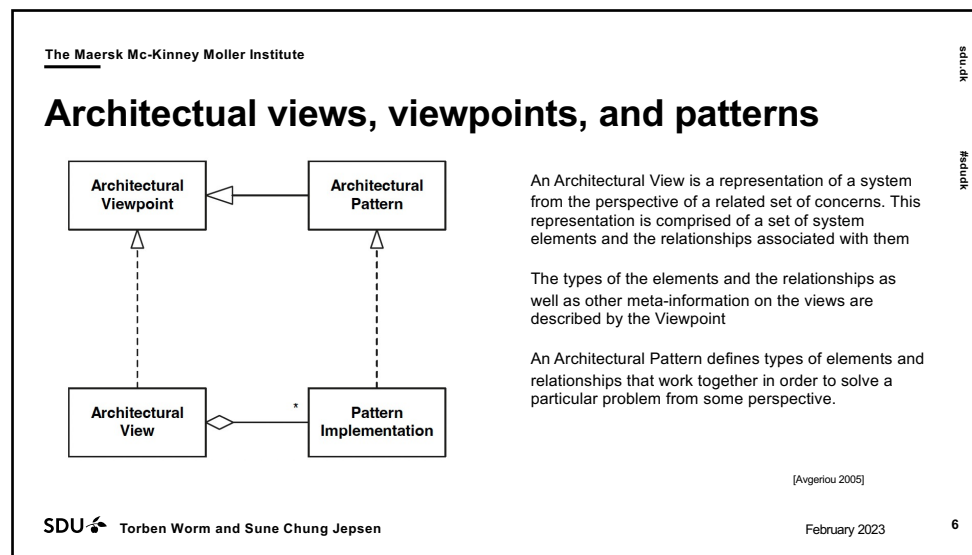
sdudk #sdudk

SDU  Torben Worm and Sune Chung Jepsen February 2023 4

4



5

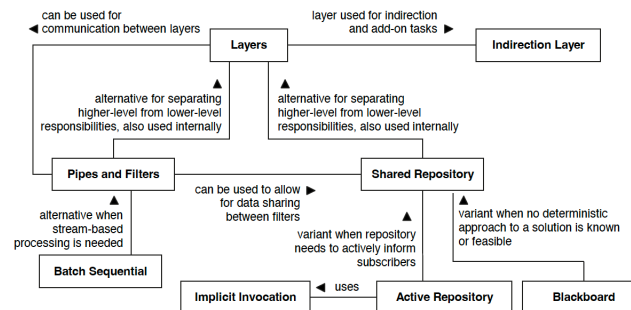


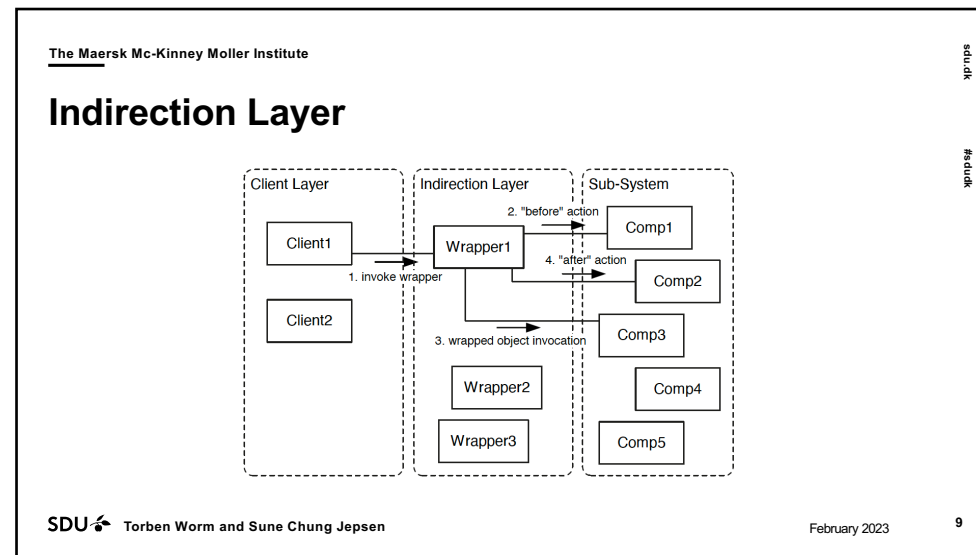
6

## Views

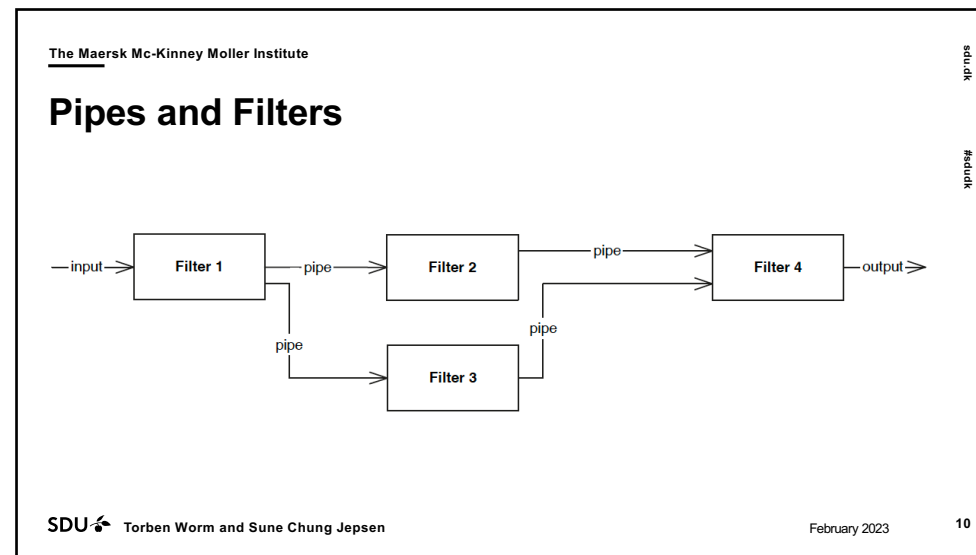
- The Layered View deals with how the system as a complex heterogeneous entity can be decomposed into interacting parts.
- The Data Flow View deals with how streams of data are successively processed or transformed by components.
- The Data-centered View is appropriate when the concerns involve how a central repository of data is accessed by multiple components.
- The Adaptation View deals with how the system adapts itself during evolution.
- The Language Extension View is concerned with how systems offer an abstraction layer to the computation infrastructure.
- The User Interaction View shows the runtime structure of components that offer a user interface.
- The Component Interaction View focuses on how individual components exchange messages but retain their autonomy.
- The Distribution View tackles concerns about disseminating components in a networked environment.

## Patterns of the Layered, Data Flow and Data-centered Views



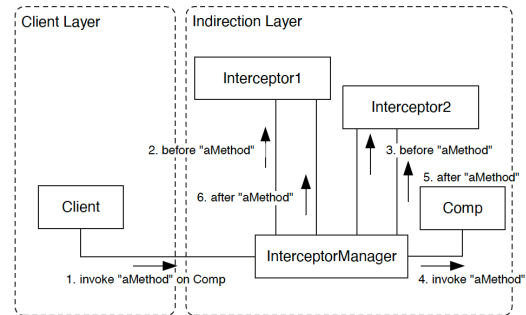


9



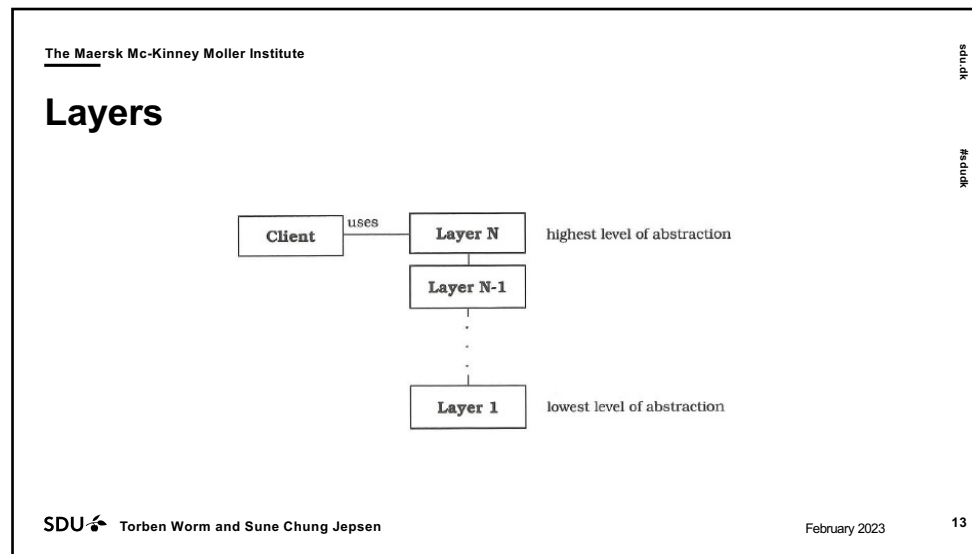
10

## Inceptor example

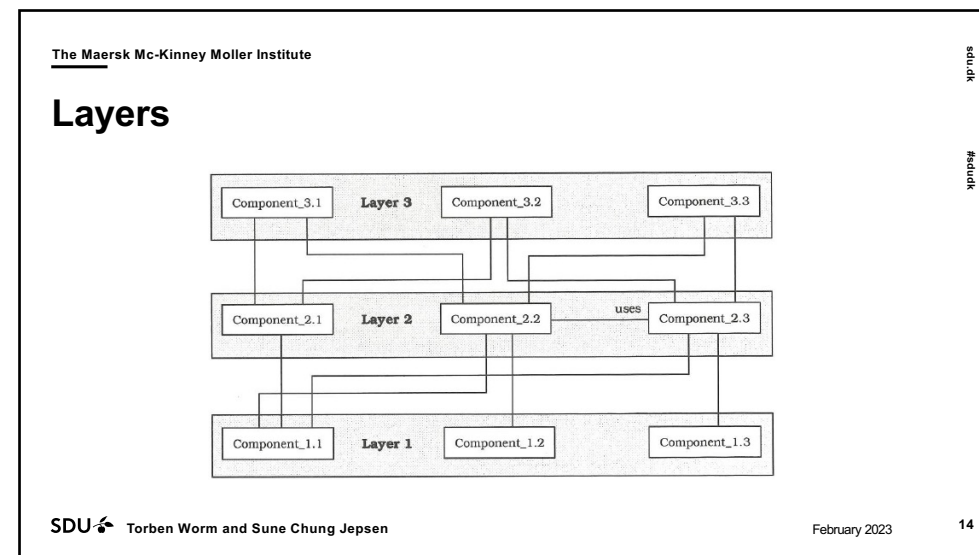


## CRC

Class	Collaborator
Layer J	• Layer J-1
<b>Responsibility</b> <ul style="list-style-type: none"> <li>• Provides services used by Layer J+1.</li> <li>• Delegates subtasks to Layer J-1.</li> </ul>	

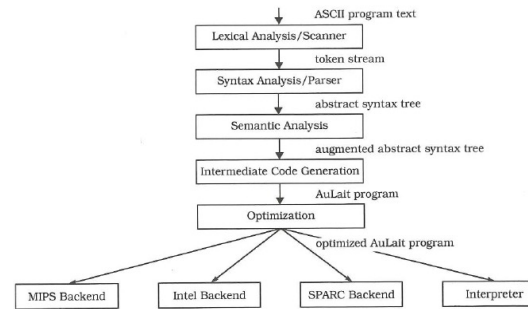


13



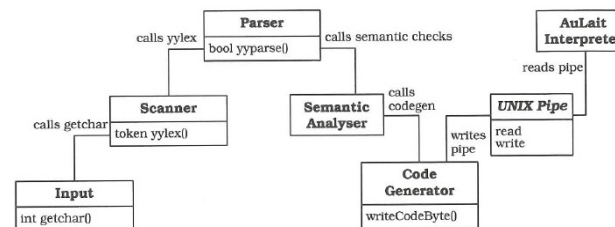
14

## Pipes and Filters



15

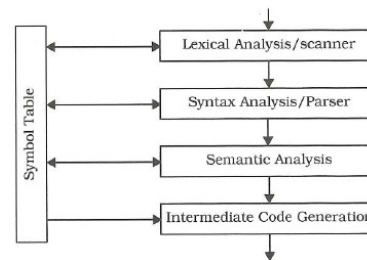
## Pipes and Filters



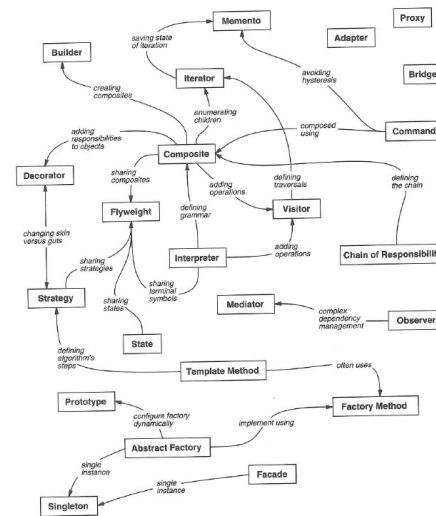
16



## Pipes and Filters



## Relationships



## Variability

Purpose	Design Pattern	Aspect(s) That Can Vary
Creational	Abstract Factory (87)	families of product objects
	Builder (97)	how a composite object gets created
	Factory Method (107)	subclass of object that is instantiated
	Prototype (117)	class of object that is instantiated
	Singleton (127)	the sole instance of a class
Structural	Adapter (139)	interface to an object
	Bridge (151)	implementation of an object
	Composite (163)	structure and composition of an object
	Decorator (173)	responsibilities of an object without subclassing
	Facade (185)	interface to a subsystem
	Flyweight (195)	storage costs of objects
	Proxy (207)	how an object is accessed, its location
Behavioral	Chain of Responsibility (223)	object that can fulfill a request
	Command (233)	when and how a request is fulfilled
	Interpreter (243)	grammar and interpretation of a language
	Iterator (257)	how an aggregate's elements are accessed, traversed
	Mediator (273)	how and which objects interact with each other
	Memento (283)	what private information is stored outside an object and when
	Observer (293)	number of objects that depend on another object, how the dependent objects stay up to date
	State (305)	status of an object
	Strategy (315)	an algorithm
	Template Method (329)	steps of an algorithm
	Visitor (331)	operations that can be applied to object(s) without changing their classes

## References

1. Paris Avgeriou and Uwe Zdun. Architectural patterns revisited - A pattern language. In Andy Longshaw and Uwe Zdun, editors, EuroPLoP' 2005, Tenth European Conference on Pattern Languages of Programs, Irsee, Germany, July 6-10, 2005, pages 431–470. UVK - Universitätsverlag Konstanz, 2005.
2. L. Bass, P. Clements, R. Kazman, and an O'Reilly Media Company Safari. Software Architecture in Practice, 4th Edition. SEI series in software engineering. Addison-Wesley Professional, 2021.
3. Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt. Past, present, and future trends in software patterns. IEEE Software, 24(4):31–37, 2007.
4. Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1 edition, 1994.