

# Software Architecture

# NENAD MEDVIDOVIĆ



# What is Software Architecture?



Perry and Wolf

SWA = {Elements, Form, Rationale}

what

how

why



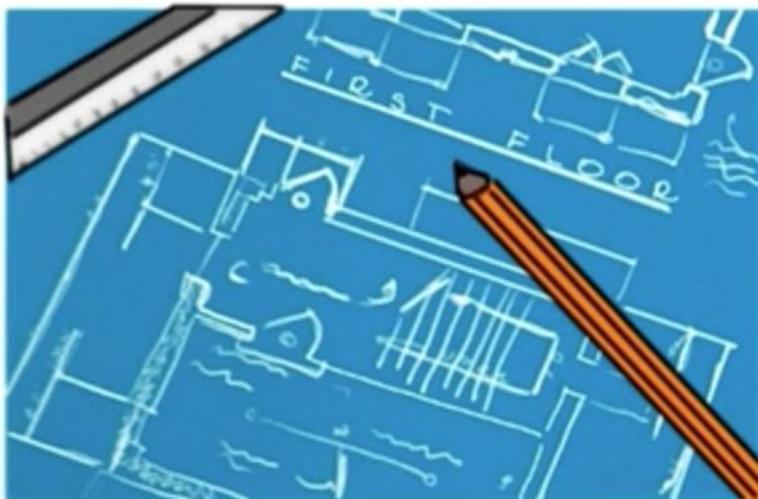
Shaw and Garland

SWA [is a level of design that] involves

- description of elements from which systems are built
- interactions among those elements
- patterns that guide their composition
- constraints on these patterns

# A General Definition of SWA

Set of principal design decisions about the system



Blueprint of a software system

Structure

Behavior

Interaction

Non-functional properties

# Temporal Aspect

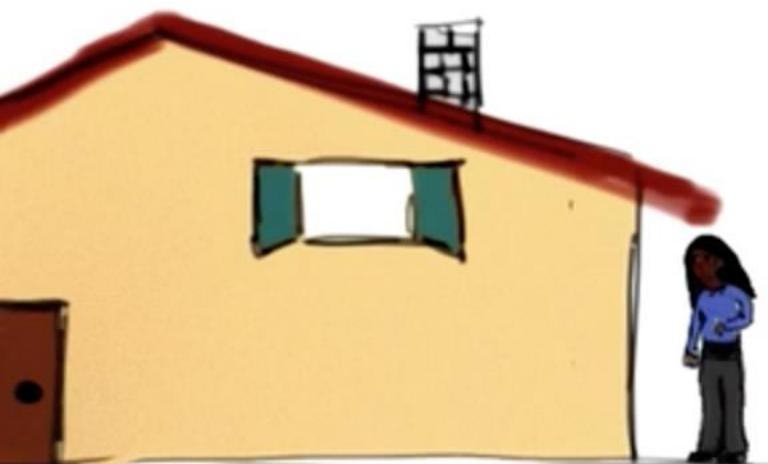
- A SWA is not defined at once, but iteratively, over time
- At any point in time there is a SWA, but it will change over time
- Design decisions are made, unmade, and changed over a system's lifetime

# Prescript vs Descriptive Architecture



A prescriptive architecture captures the design decisions made prior to the system's construction

=> as – conceived SWA



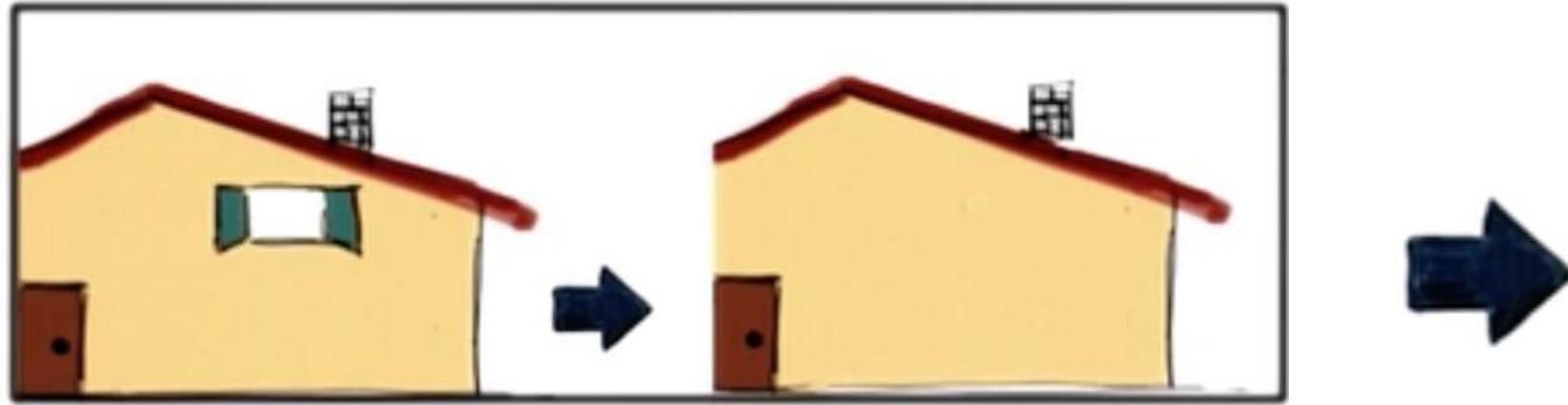
A descriptive architecture describes how the system has actually been build

=> as – implemented SWA

# Architectural Evolution



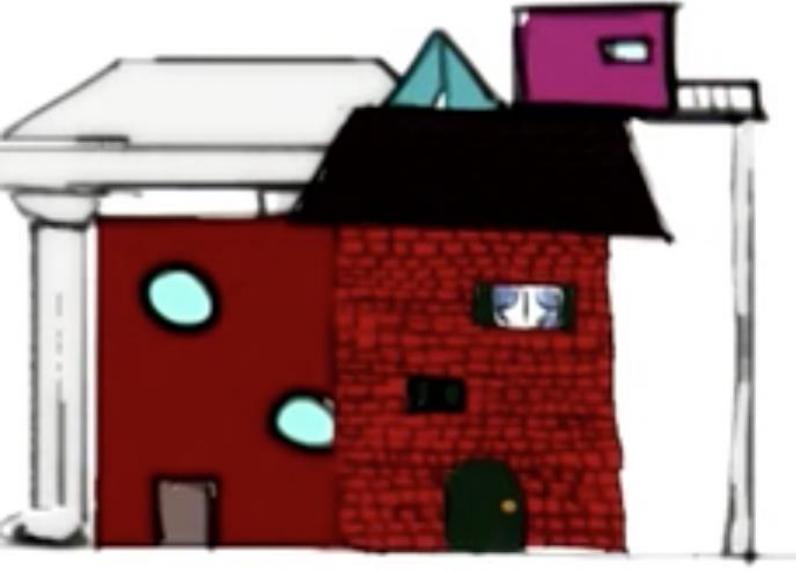
When a system evolves, ideally its prescriptive architecture should be modified first



In practice, this rarely happens

- Developer's sloppiness
- Short deadlines
- Lack of documented prescriptive architecture

# Architectural Degradation



Architectural drift – Introduction of architectural design decisions orthogonal to a system's prescriptive architecture



Architectural erosion – Introduction of architectural design decisions that violate a system's prescriptive architecture

# Architectural Recovery

Drift and erosion => degraded architecture



Keep tweaking the code  
(typically disastrous)



Architectural recovery : determine SWA  
from implementation and fix it



Which of the following sentences is true?

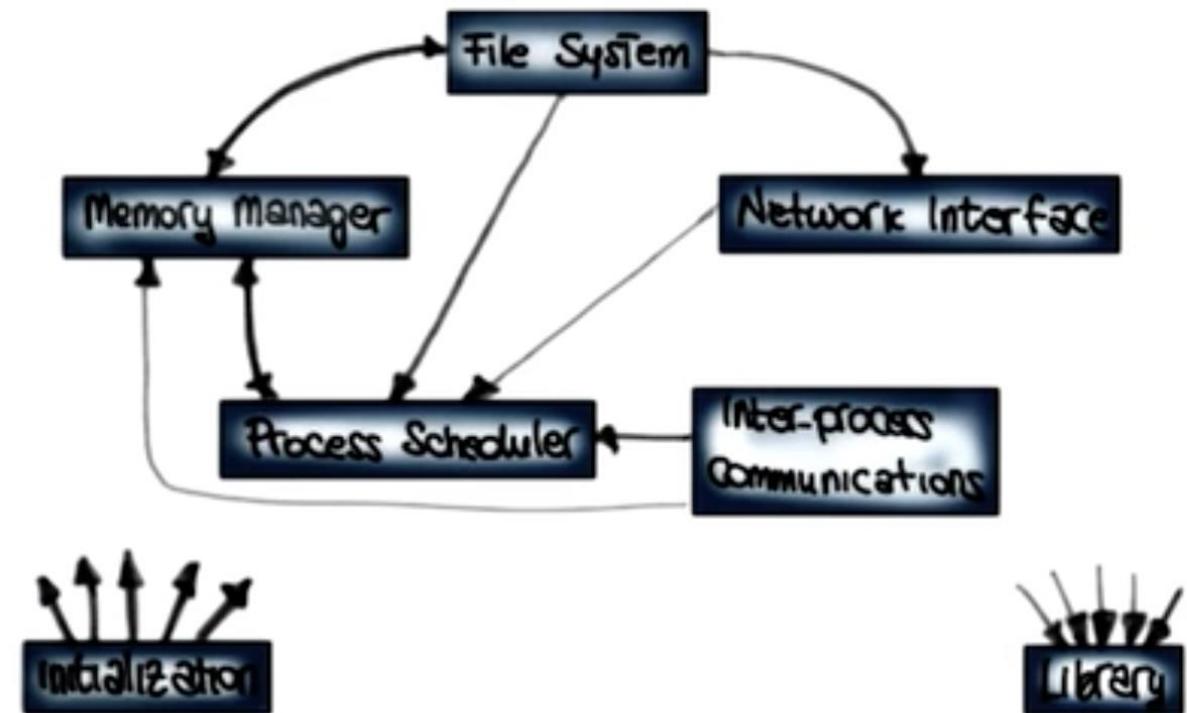
- [ ] Prescriptive architecture and descriptive architecture are typically the same
- [ ] Architectural drift results in unnecessarily complex architectures
- [ ] Architectural erosion is less problematic than architectural drift
- [ ] The best way to improve a degraded architecture is to keep fixing the code until the system starts looking and behaving as expected



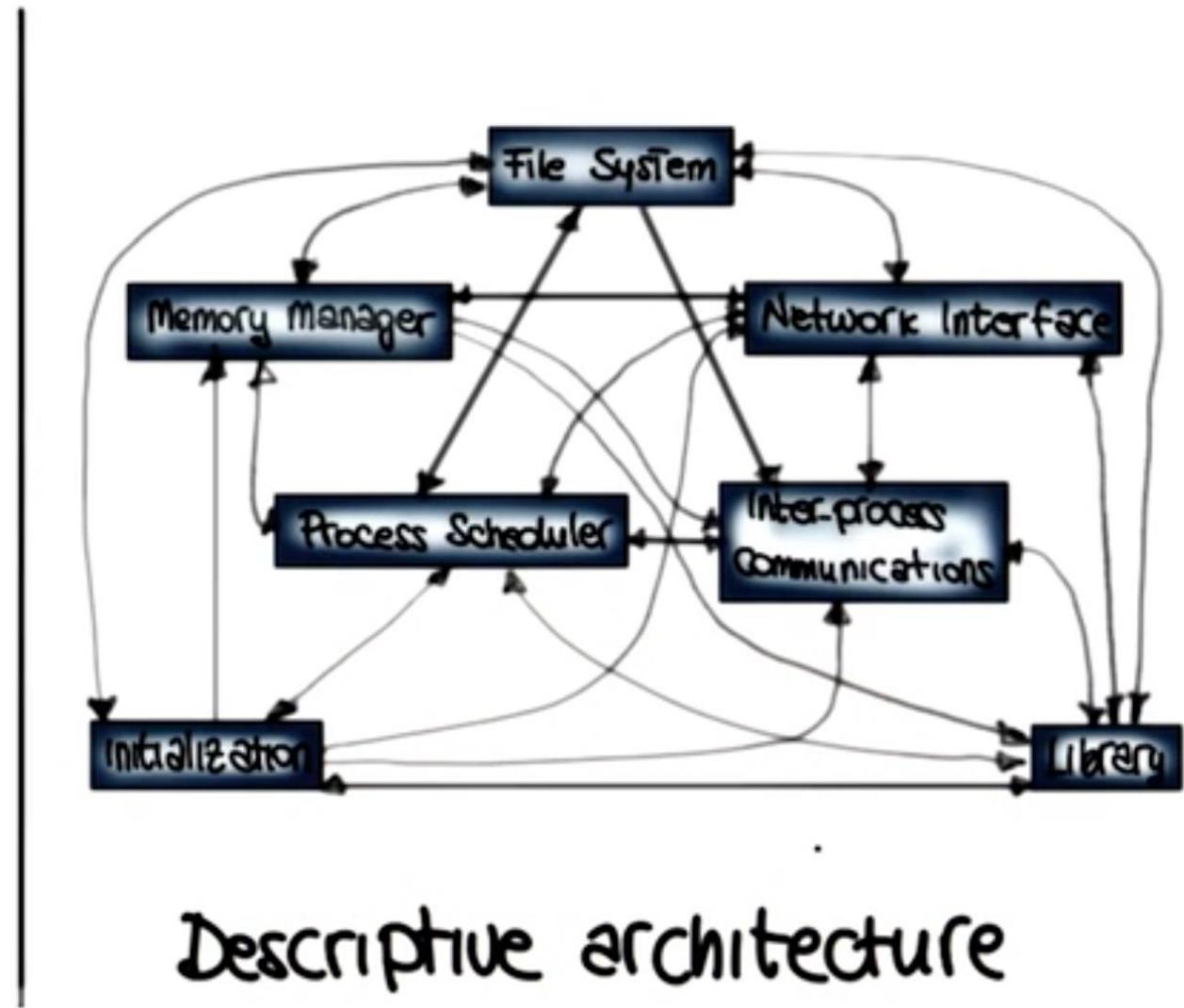
Which of the following sentences is true?

- Prescriptive architecture and descriptive architecture are typically the same
- Architectural drift results in unnecessarily complex architectures
- Architectural erosion is less problematic than architectural drift
- The best way to improve a degraded architecture is to keep fixing the code until the system starts looking and behaving as expected

# AN EXAMPLE FROM THE LINUX KERNEL

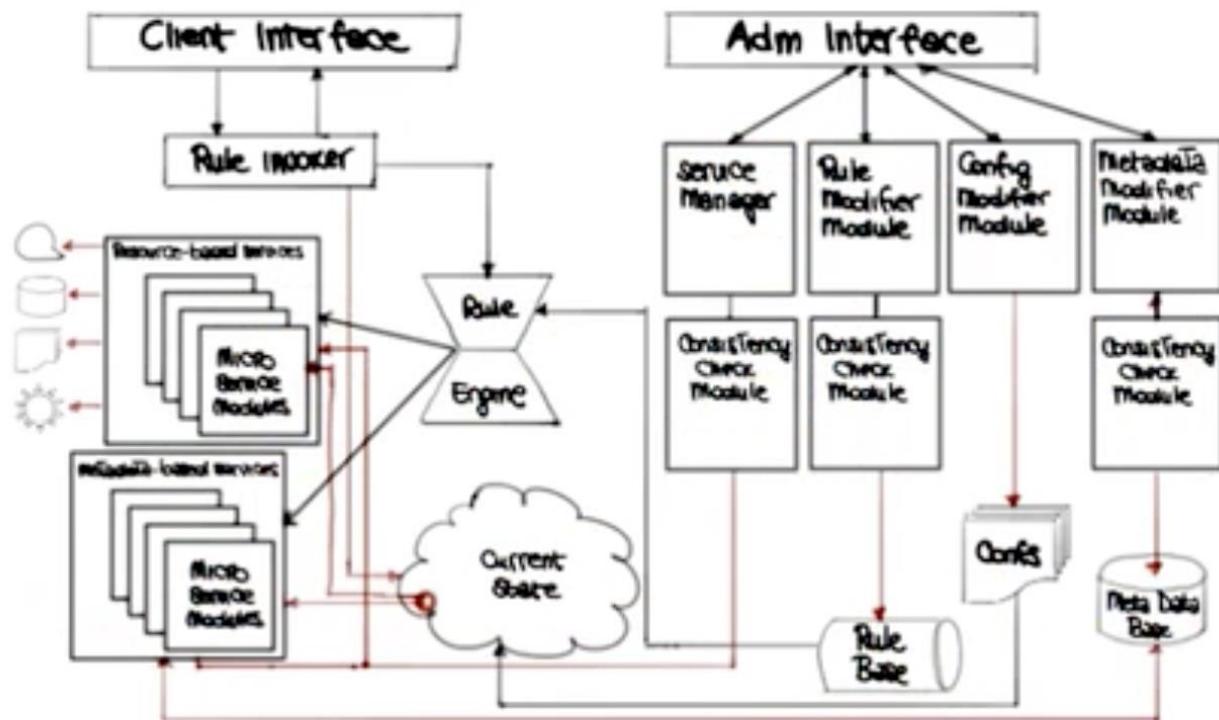


Prescriptive architecture

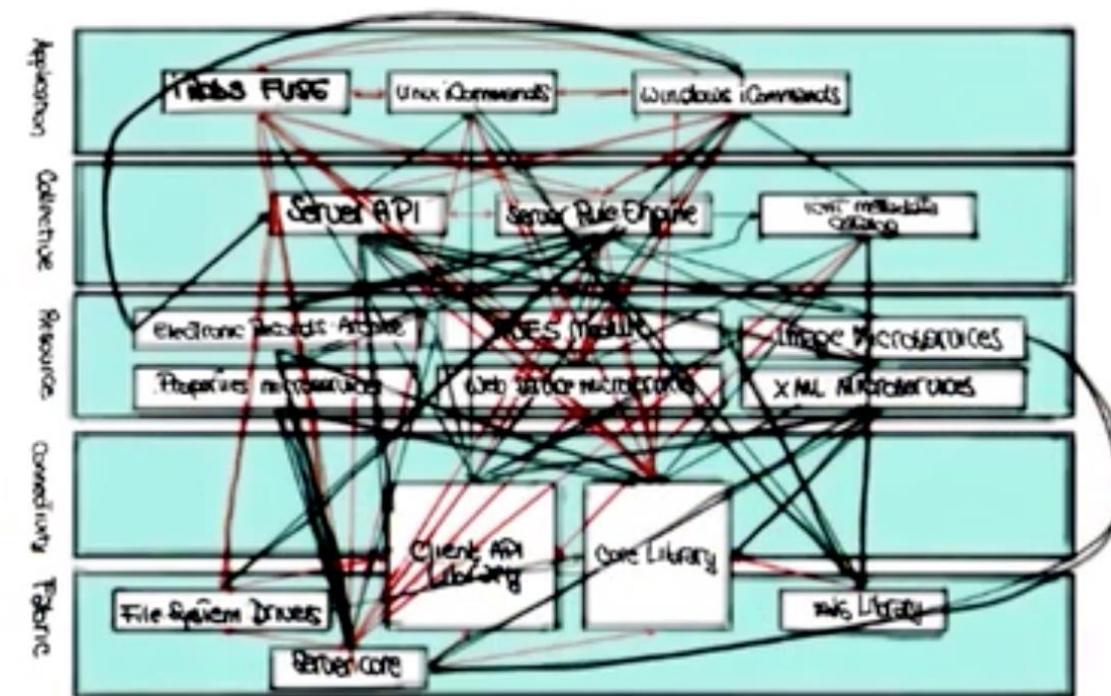


Descriptive architecture

# ANOTHER EXAMPLE: iRODS

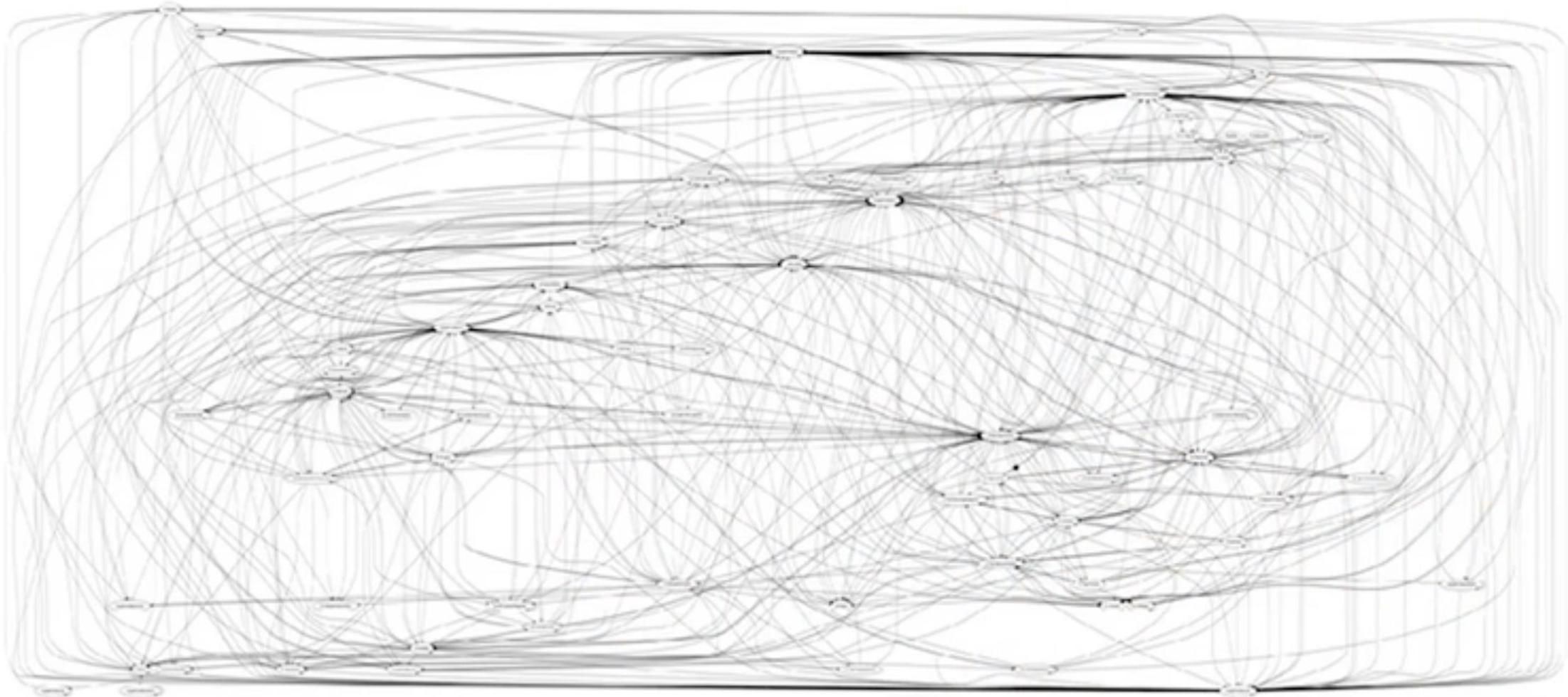


Prescriptive architecture

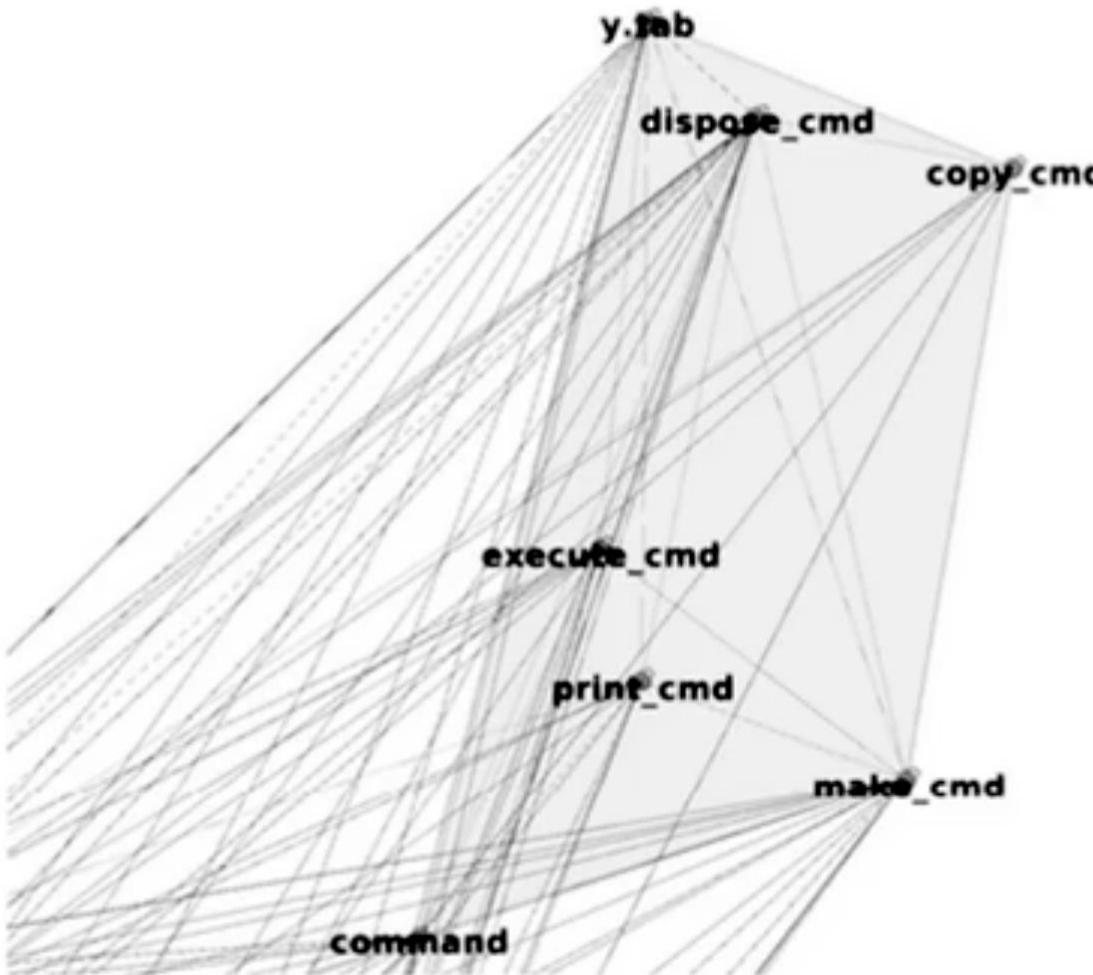


Descriptive architecture

## MORE EXAMPLES : HADOOP



# A FINAL EXAMPLE: BASH





What are ideal characteristics of an architectural design?

- Scalability
- Low cohesion
- Low coupling

# S/W Architecture's Elements

A software architecture typically is not a monolith  
Composition and inter play of different elements



Processing elements



Data elements



Interaction elements



+  ⇒ components

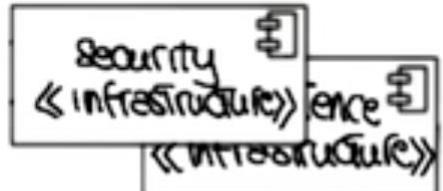


⇒ connectors

} configuration

# COMPONENTS, CONNECTORS, AND CONFIGURATIONS

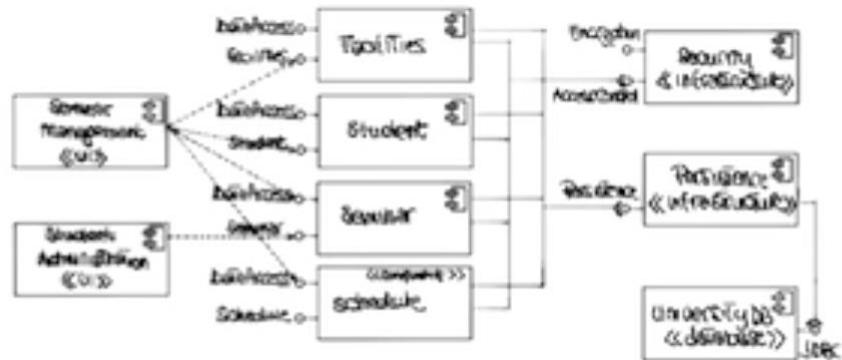
Software component : architectural entity that



- encapsulates a subset of the system's functionality and/or data
- restricts access to that subset via an explicitly defined interface

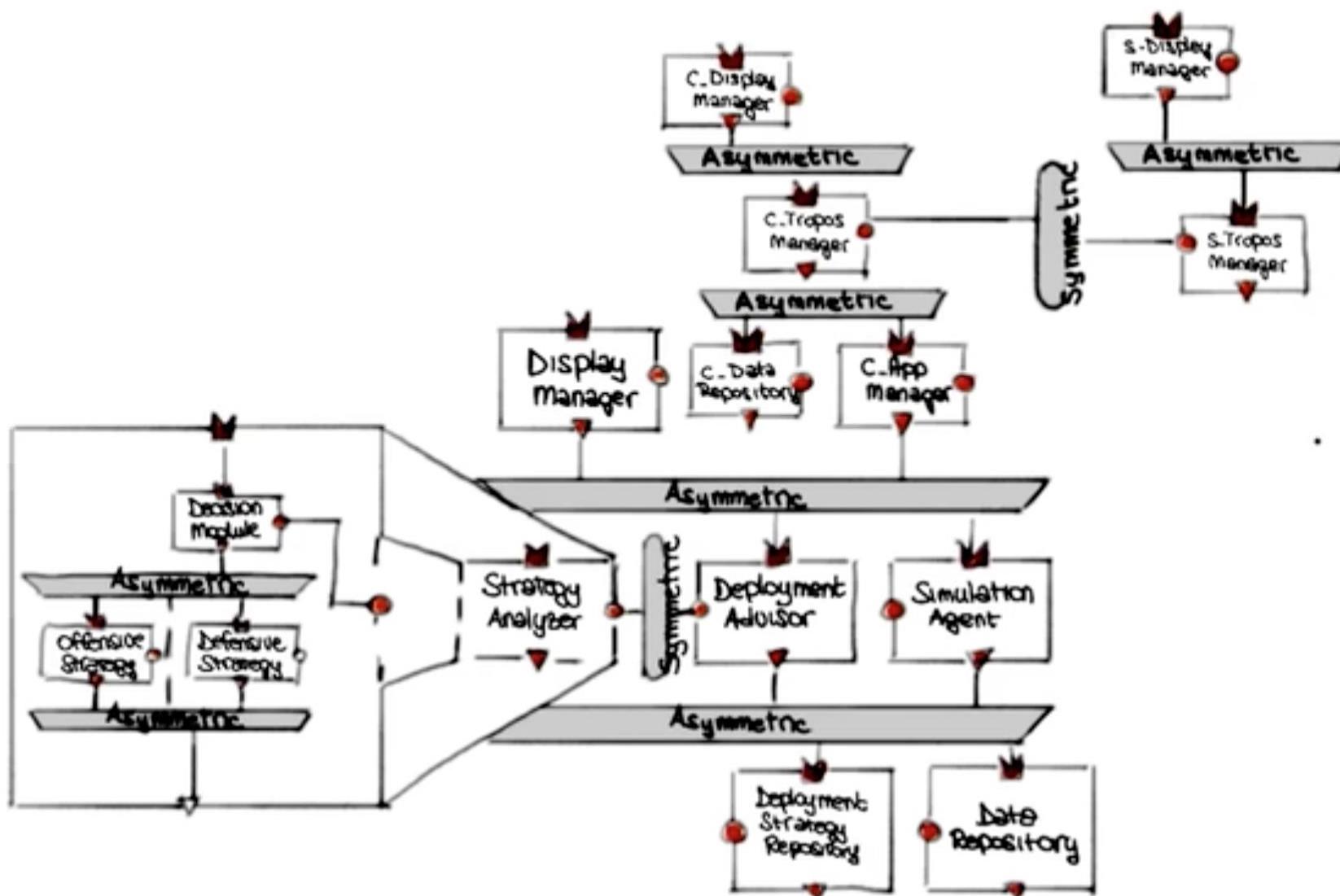


Software connector : architectural entity effecting and regulating interaction

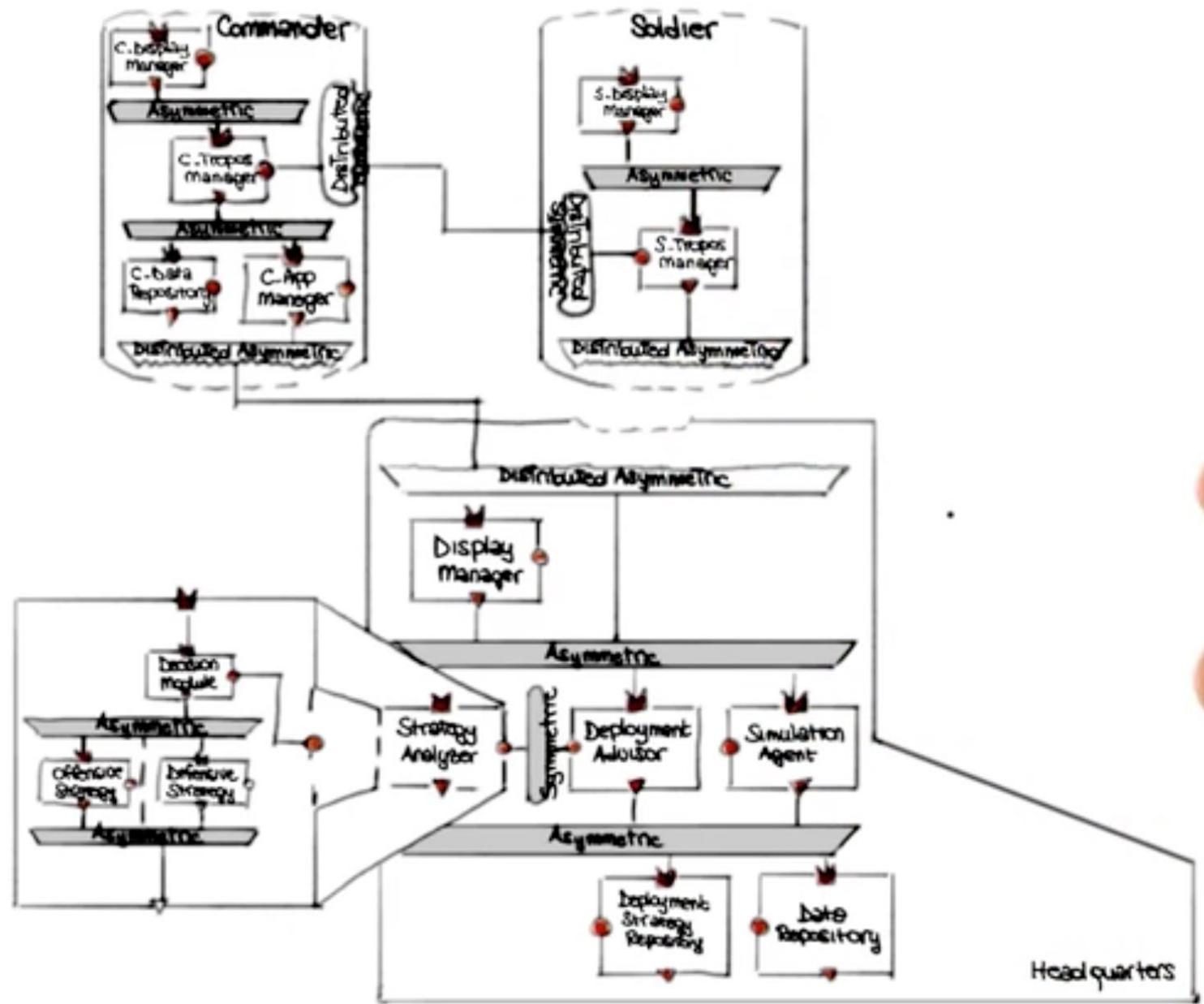


Architectural configuration : association between components and connectors of a software architecture

# AN EXAMPLE CONFIGURATION



# Deployment Architectural Perspective



# ARCHITECTURAL STYLES



An architectural style defines "a family of systems in terms of a pattern of structural organization ; a vocabulary of components and connectors, with constraints on how they can be combined"

M. Shaw and D. Garlan , 1996

Basically , named collection of architectural design decisions applicable in a given context

# ARCHITECTURAL STYLES



Pipes and filters



Event -driven



Publish -subscribe



Client -server



Peer -to -peer



Representational  
State Transfer (REST)



Consider the following architectural styles that we just saw : pipes and filters (A), event - driven (B), publish - subscribe (C), client - server (D), peer - to - peer (E), REST (F). Mark which architectural style(s) characterizes the following systems . ( Mark the corresponding letter(s) next to the systems )

Android OS	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]	F [ ]
Skype	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]	F [ ]
World-Wide Web	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]	F [ ]
DropBox	A [ ]	B [ ]	C [ ]	D [ ]	E [ ]	F [ ]



Consider the following architectural styles that we just saw : pipes and filters (A), event - driven (B), publish - subscribe (C), client - server (D), peer - to - peer (E), REST (F). Mark which architectural style(s) characterizes the following systems . ( Mark the corresponding letter(s) next to the systems )

Android OS	A [ ]	B [X]	C [X]	D [ ]	E [ ]	F [ ]
Skype	A [ ]	B [ ]	C [ ]	D [X]	E [X]	F [ ]
World-Wide Web	A [ ]	B [ ]	C [ ]	D [X]	E [ ]	F [X]
DropBox	A [ ]	B [ ]	C [ ]	D [X]	E [ ]	F [ ]

# PEER-TO-PEER (P2P) ARCHITECTURES

Decentralized resource sharing and discovery

Two representative examples



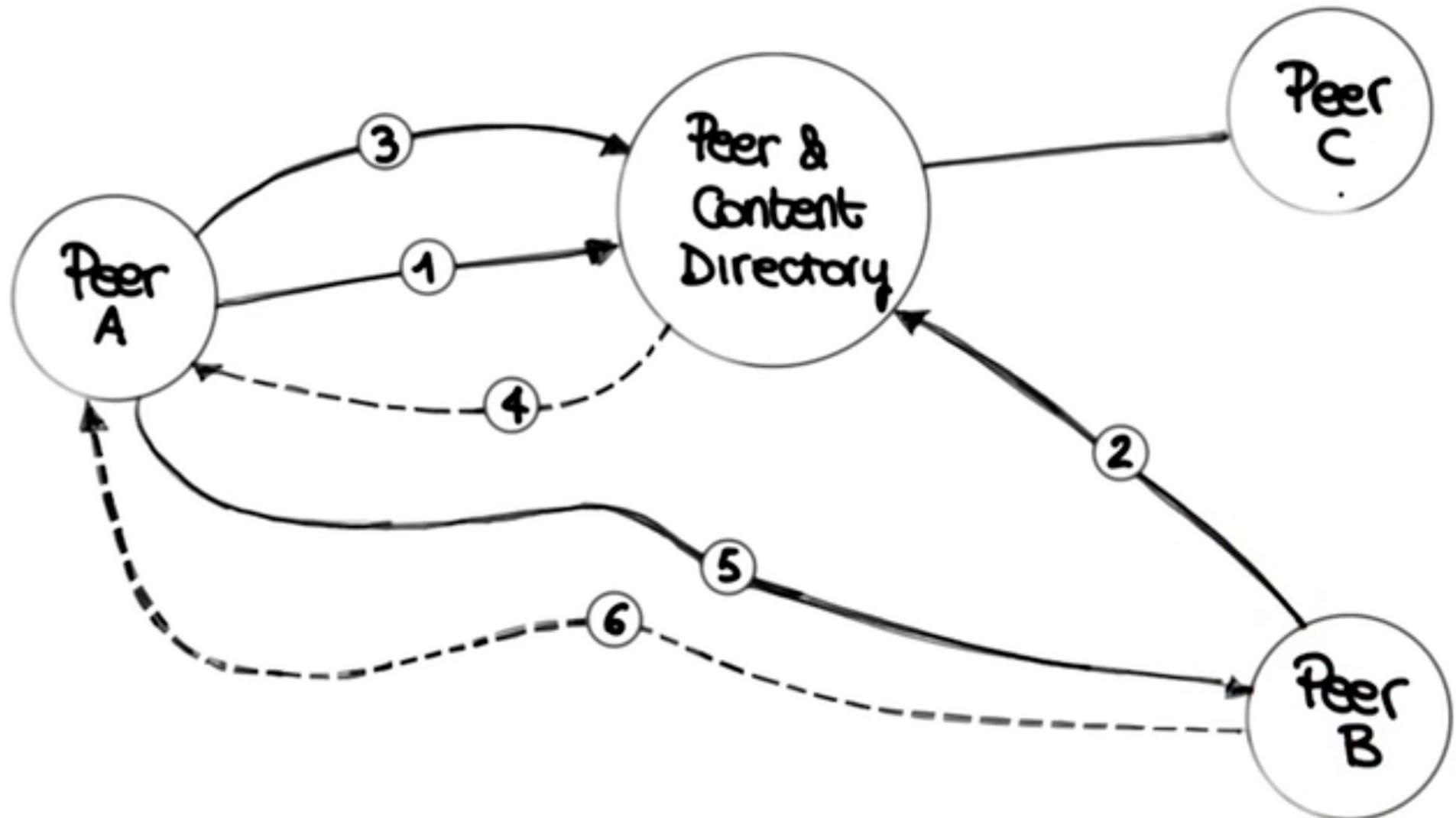
Napster



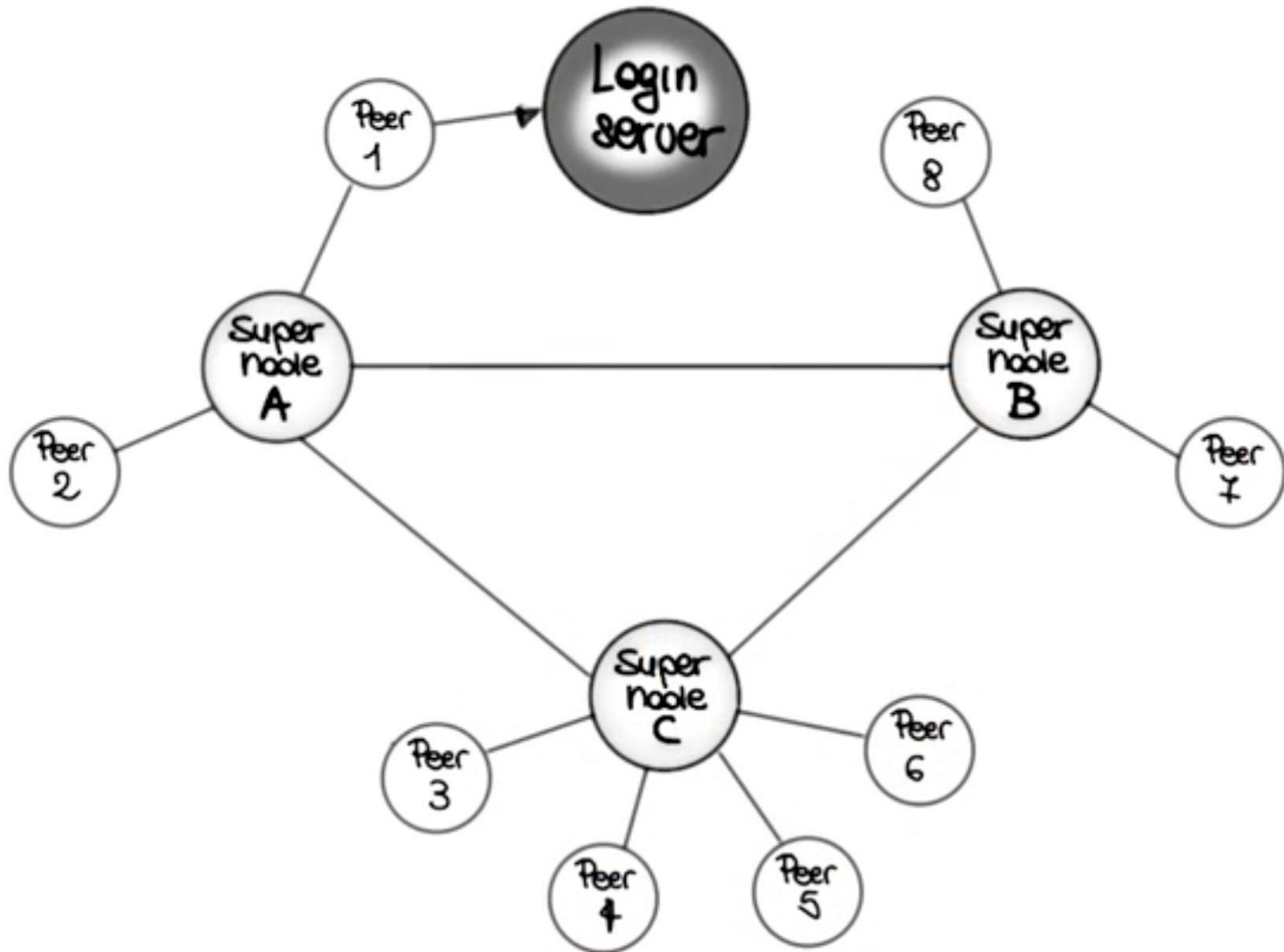
Skype



# NAPSTER



# SKYPE



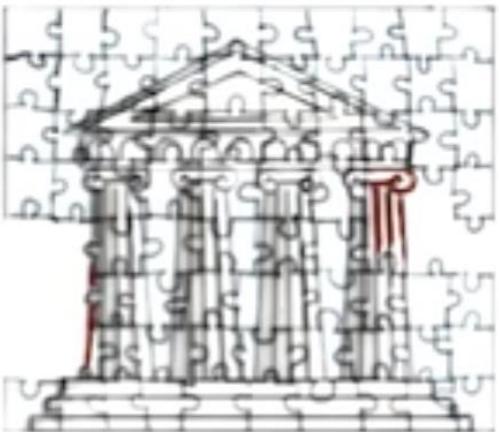
# TAKEAWAYS



A great architecture is a ticket to success



A great architecture reflects deep understanding of the problem domain



A great architecture normally combines aspects of several simpler architectures