

SOFTWARE ARCHITECTURE



PARALLEL – BUILDING CONSTRUCTION

Architects are the technical interface between customer and contractor building <the thing>

Bad architectural design cannot be rescued by good construction

Specialist types of projects require architect expertise

Schools and styles have emerged



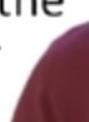
DEFINITION

What is Architecture?

Software architecture is “the structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time.” – Garlan & Perry

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. – Bass, Clements, and Kazman

Software architecture is the fundamental organization of a system, embodied in its components, their relationships to one another and the environment, and the principles governing its design and evolution. - IEEE 1471





WHY SHOULD WE CARE?

- Good things are well-architected
- Good architecture is hard
- Fixing is very difficult

Software Architecture mainly about decomposing the system into components

Each component must have individual business value

Helps organize workforce and resources

Allows for parallelization

Helps define the build vs. buy question and getting funding



1. A subsystem in an architecture must:

1 point

- be created separately and can operate individually.
- have business value.
- be integrated with one another or with existing subsystems.
- All of the above.

2. Partitioning of a large system into smaller subsystems helps the buy-or-build decision because we can examine each subsystem and reason about possible buy-or-build options for each.

1 point

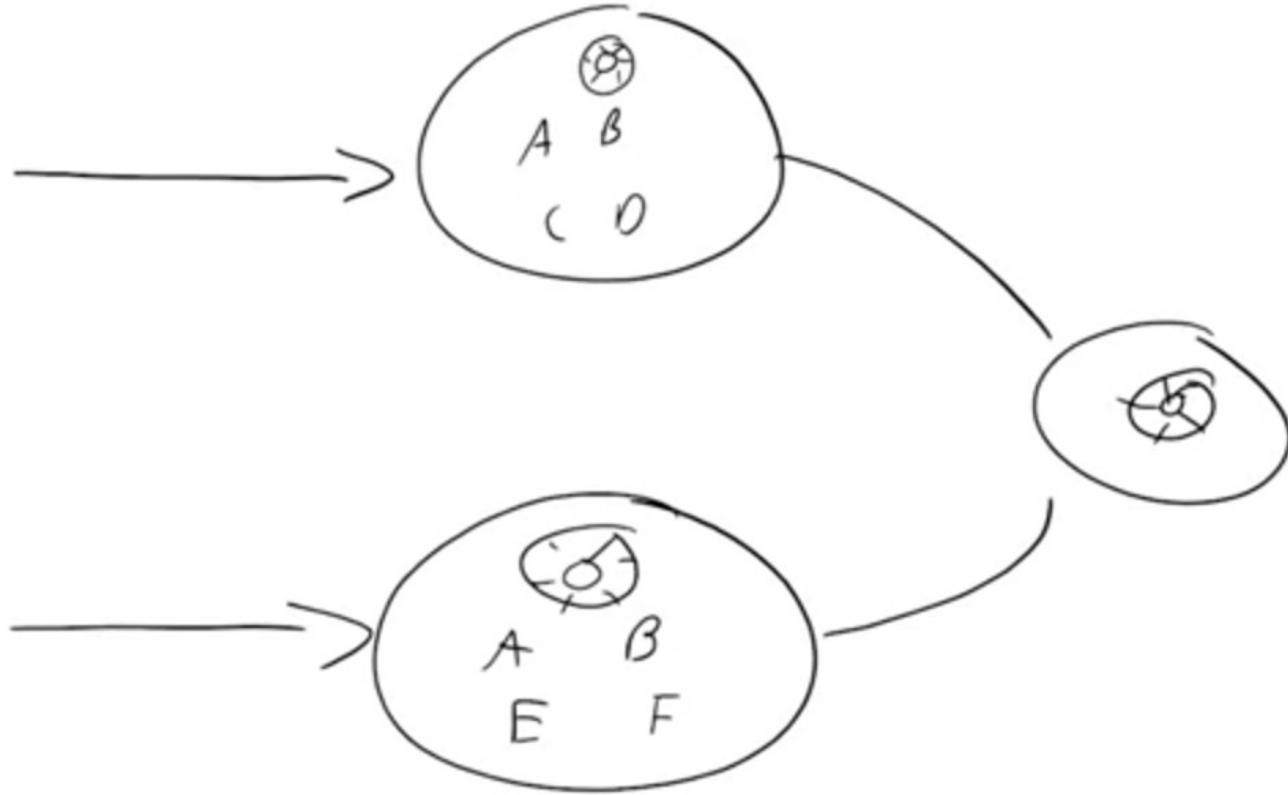
- True.
- False.

3. A good software architecture is important because:

1 point

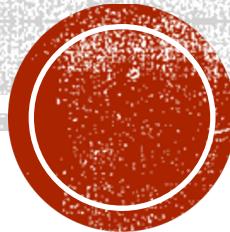
- It helps organize the workforce and resources.
- It allows for parallelization in development.
- It helps build-or-buy decisions.

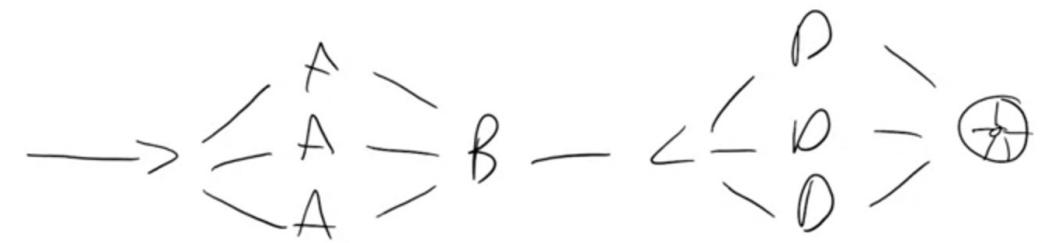
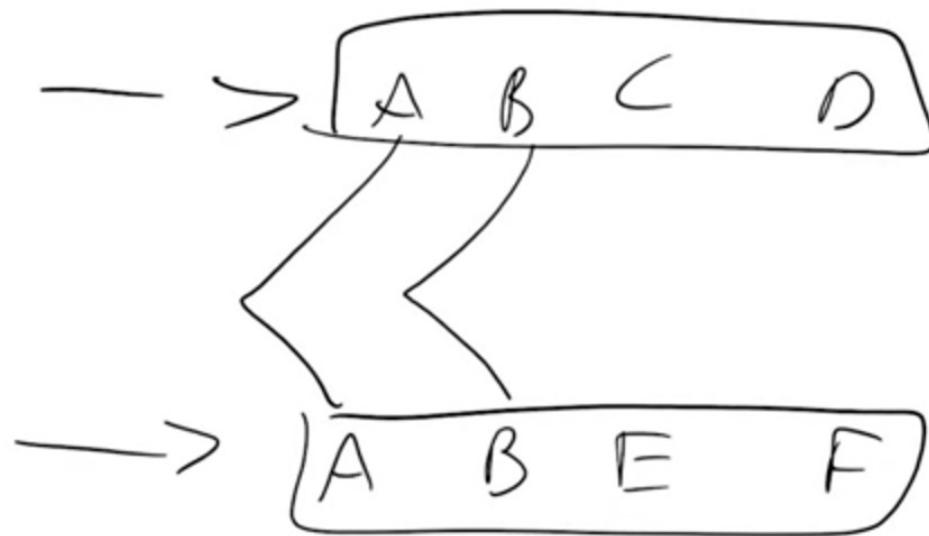




SOFTWARE ARCHITECTURE MODELS

Pipe-and-Filter

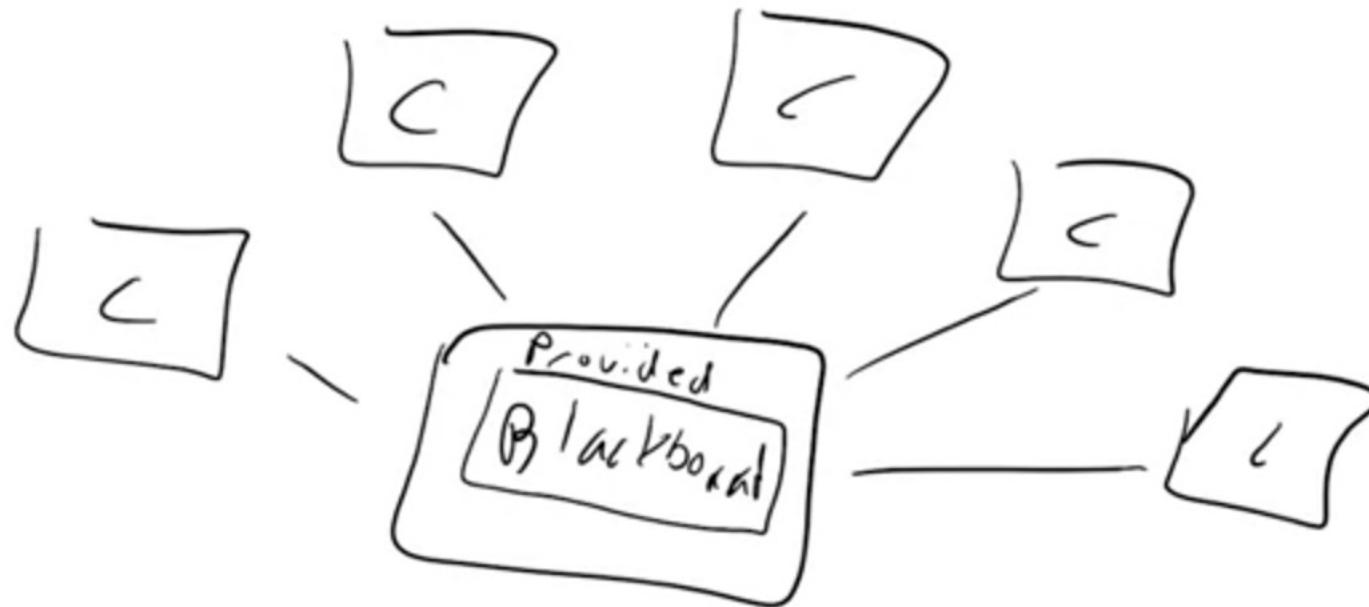




SOFTWARE ARCHITECTURE MODELS

Pipe-and-Filter

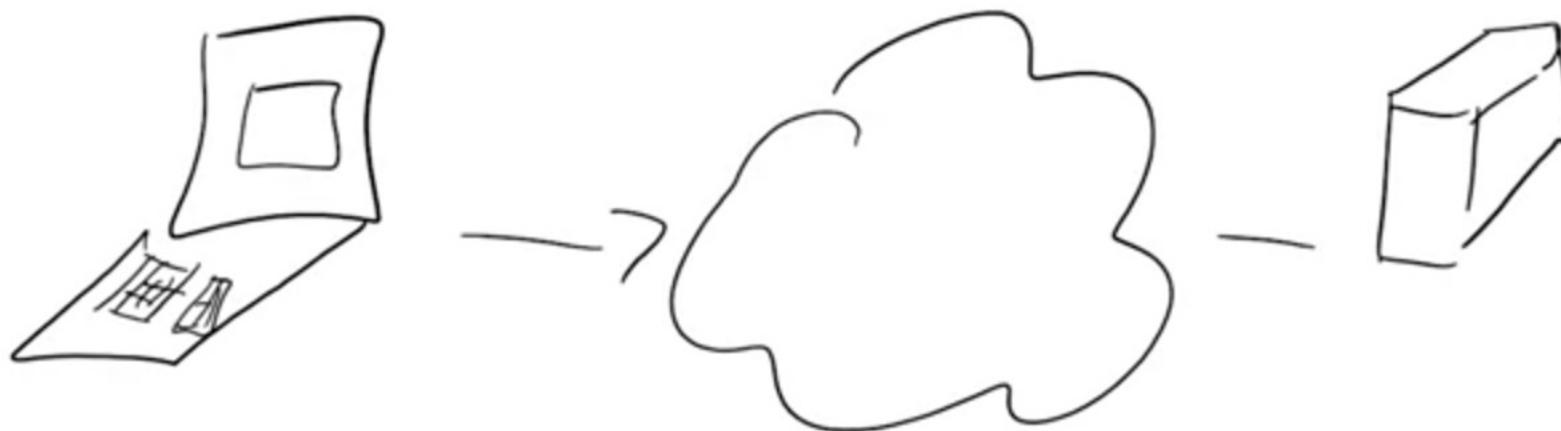
BLACKBOARD MODEL



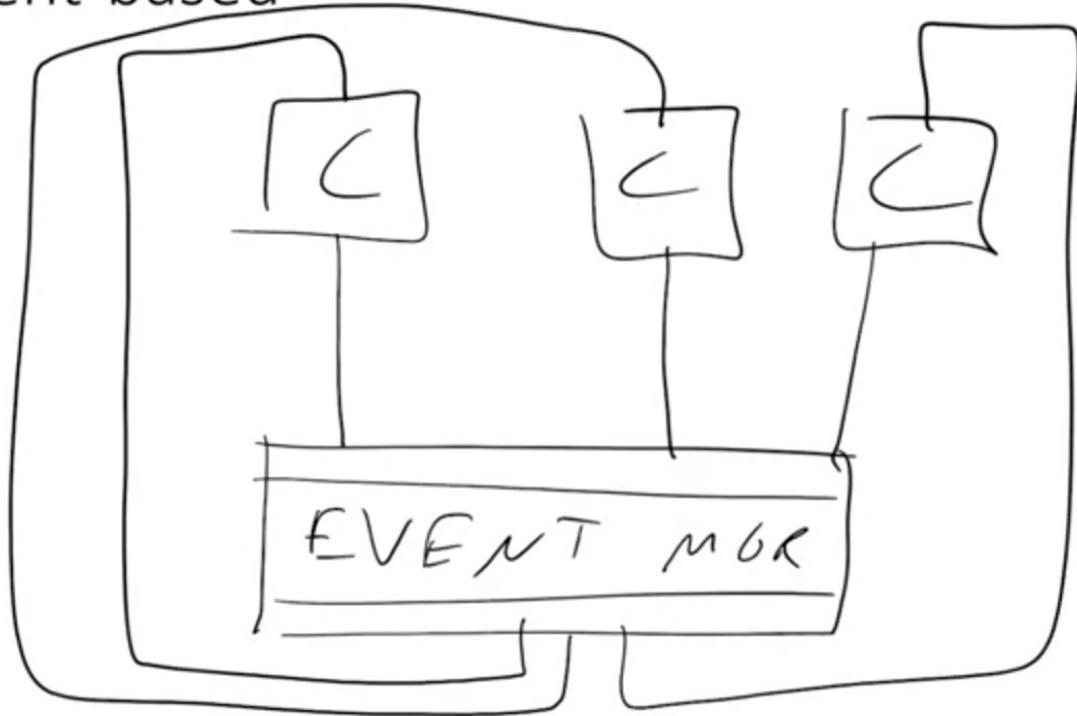
LAYERED MODEL



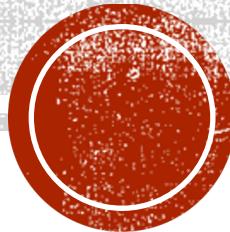
CLIENT-SERVER



Event-based



**EVENT
BASED**



SOFTWARE ARCHITECTURE – BEST PRACTICE SOLUTION

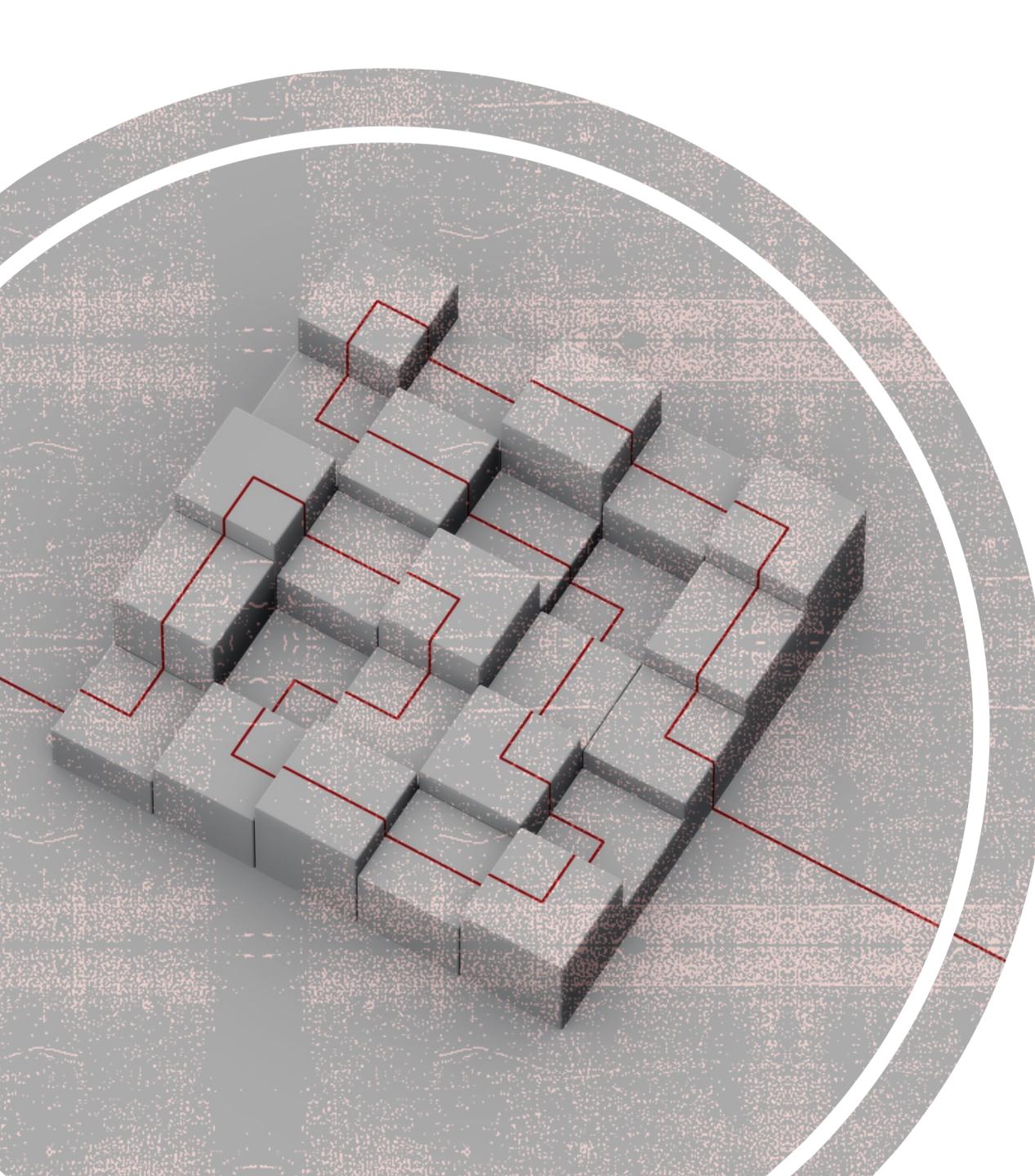
Start from the pure model, and adjust

If your goals align with the goals of the model, use it

Customize the “pure” form to your needs

Each models solves one particular large-scale problem





SOFTWARE ARCHITECTURE PROCESS

Design Process

- System Structuring
- Control Modeling
- Modular Decomposition -
maintainability, reliability, security





Sub-System : Independent system which holds the business value



Module: Component of a sub-system which can not function as standalone system

SUB-SYSTEMS VS MODULES





Performance

Reliability

Testability

Security

Usability



...IT IS ABOUT
ESTIMATION,
QUALITY AND
PARTITIONING

Partitioning the responsibilities of the software is only the beginning

Determine the costs and secure funding

Divide work and system with quality in mind

Apply software architectural models to common problems



Class 4 – Design and Implementation

Class 5 – Testing and Verification

Class 6 + 7 – Traditional S/w development models

Class 8 – Agile Fundamentals

Class 9 – UML + Process Mining