



# Software Design and Architecture

---

## Course Structure

Sajid Anwer

Department of Software Engineering,  
FAST-NUCES, CFD Campus



## Recommended Textbook

---

- System Analysis and Design in a Changing World, by Satzinger, Jackson, and Burd, 7th Edition
- System Analysis & Design by Dennis, Wixom, and Roth, 7th Edition
- Software Architecture Foundations, Theory, and Practice by Richard N. Taylor



## Evaluation Plan

---

Evaluation Type	Points
Quiz	10
Mid-I	18
Mid-II	18
Final	40
Assignments/Case Studies	04
Project	10



## Class Guidelines/Rules

---

- Raise your hand before asking any question and then WAIT for the permission.
- *Please don't miss a class.*
- Please don't use *mobile phones* in the class.
- Plagiarism in Assignments/case studies is *not allowed*.
- *No Makeup Quiz OR Assignments will be given.*
- *Best-of option* in *any evaluation* is *not applicable* in this course.



## Class Guidelines/Rules

---

- Final exam will be selective comprehensive.
- Minimum **80% attendance** is mandatory to sit in final exam.
- At least **50% Marks** are required to pass this course.
- **Project submission** will be due on **Friday 5:00pm** in the specified week.
- There will be no **formal assignment** in this, only couple of **reading assignments** will be given.
- Projects submissions weeks are specified in the outline and will **not be changed**.



## Course Contents

---

- The course is divided into two major modules
- The first module covers the introduction to software design and provides details of *software architecture and architectural styles*.
- The second module covers the *detailed design phase* with introduction to *Object Oriented Design* and various *design patterns*.

## Course Outline

---

- Fundamentals of Software Architecture
- Architecture in context of Software development life cycle
- Architectural Styles
- Architectural Views
- Model Quality Requirements in Software Architecture
- System development methodologies
- Fundamentals of object-oriented systems
- Fundamentals of Requirements engineering
  - » Requirements Modeling
    - Use case models
    - use case description

## Course Outline

---

- Conceptual models
  - » System sequence diagram
  - » Activity diagram
  - » Domain model
  - » State diagram
- Logical models for system design
  - » Class diagram
  - » Sequence diagram
  - » Data Flow Diagram
- Implementation models
  - » Deployment diagram
  - » Component diagram
  - » Package Diagram
- GoF patterns
- GRASP patterns
- Solid patterns





# Software Design and Architecture

---

## Fundamentals of Software Architecture

Sajid Anwer

Department of Software Engineering,  
FAST-NUCES, CFD Campus

## Why we need software Architecture Activity?

---



## Why we need software Architecture Activity?

---



## Why we need software Architecture Activity?

---

- What do you think is wrong in these real life scenarios?
- The requirements are correct!
  - » A staircase next to the outer wall
  - » A door on the first floor
  - » An ATM outside the bank branch
- The design is flawed!
  - » The execution based on the design results in disaster.

## What is Software Design?

---

- The literature on design methods began to appear in the 1950s and 60s.
  - Since then, design methodology has become an independent discipline of scientific study.
  - A software design is a meaningful engineering representation of some software product that is to be built.
- “The process of applying *various techniques* and principles for the purpose of defining a device, a process or a system in *sufficient detail* to permit its physical realization” [TAY59]

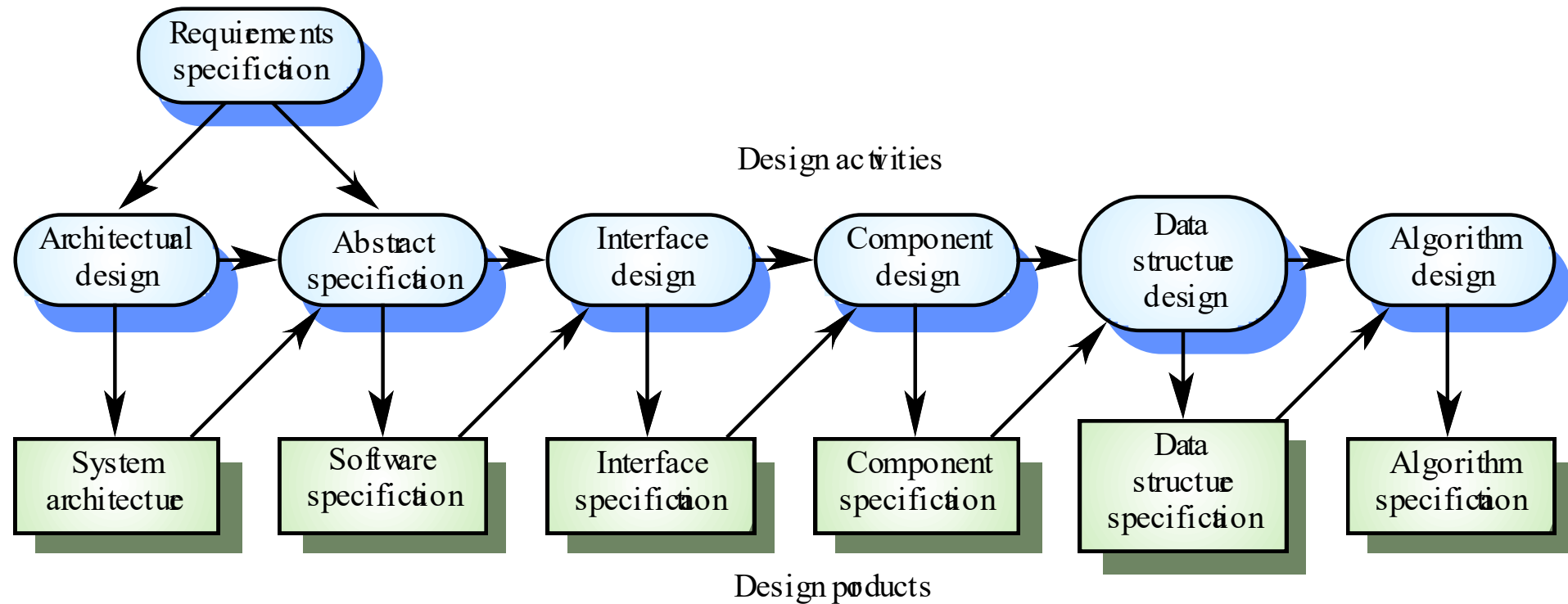


## What is Software Design

---

- Requirements specification was about the **WHAT** the system will do
- Design is about the **HOW** the system will perform its functions
  - » provides the *overall decomposition* of the system
  - » allows to *split the work* among a team of developers
  - » also lays down the groundwork for *achieving quality requirements* (performance, maintainability, reusability, etc.)
  - » takes *target technology* into account (e.g., kind of middleware, database design, etc.)

# Software Design Process



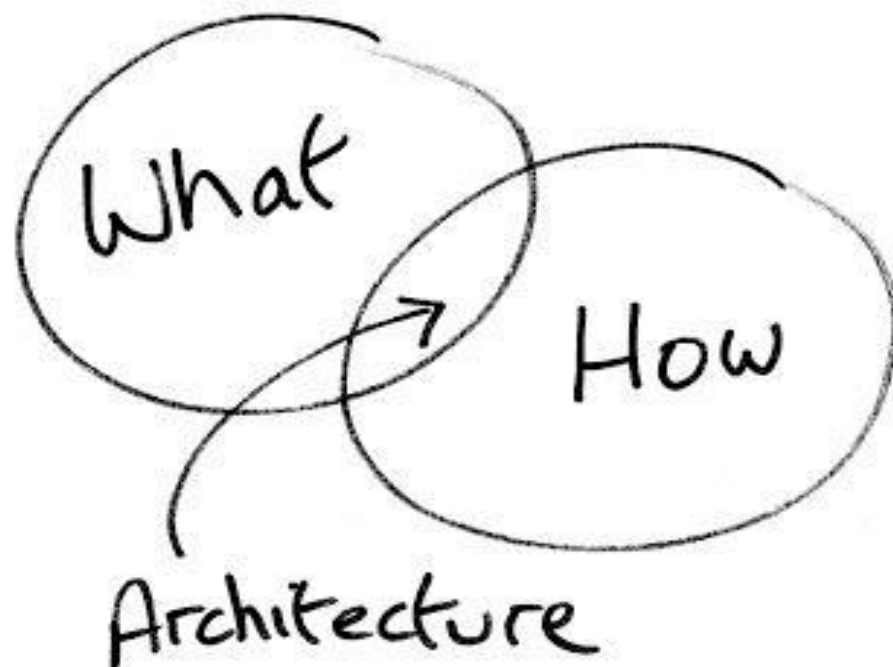
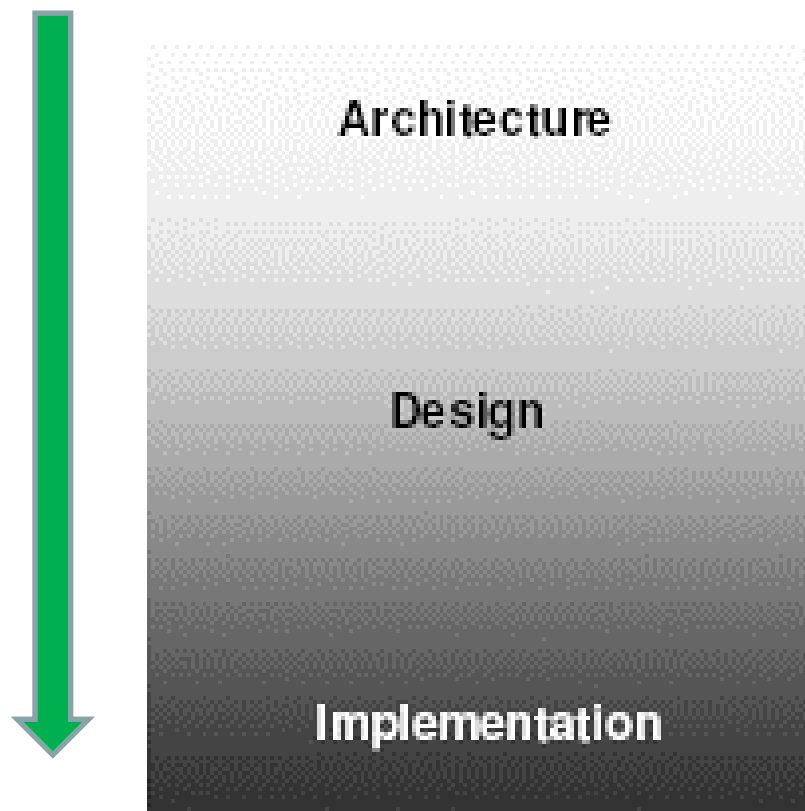
## Software Design Levels

---

- Architectural design (high-level design)
  - » architecture - the overall structure, main modules and their connections
  - » addresses the main non-functional requirements (e.g., reliability, performance)
  - » hard to change
- Detailed design (low-level design)
  - » the inner structure of the main modules
  - » detailed enough to be implemented in the programming language
  - » focuses on functional requirements



## Software Design vs Architecture

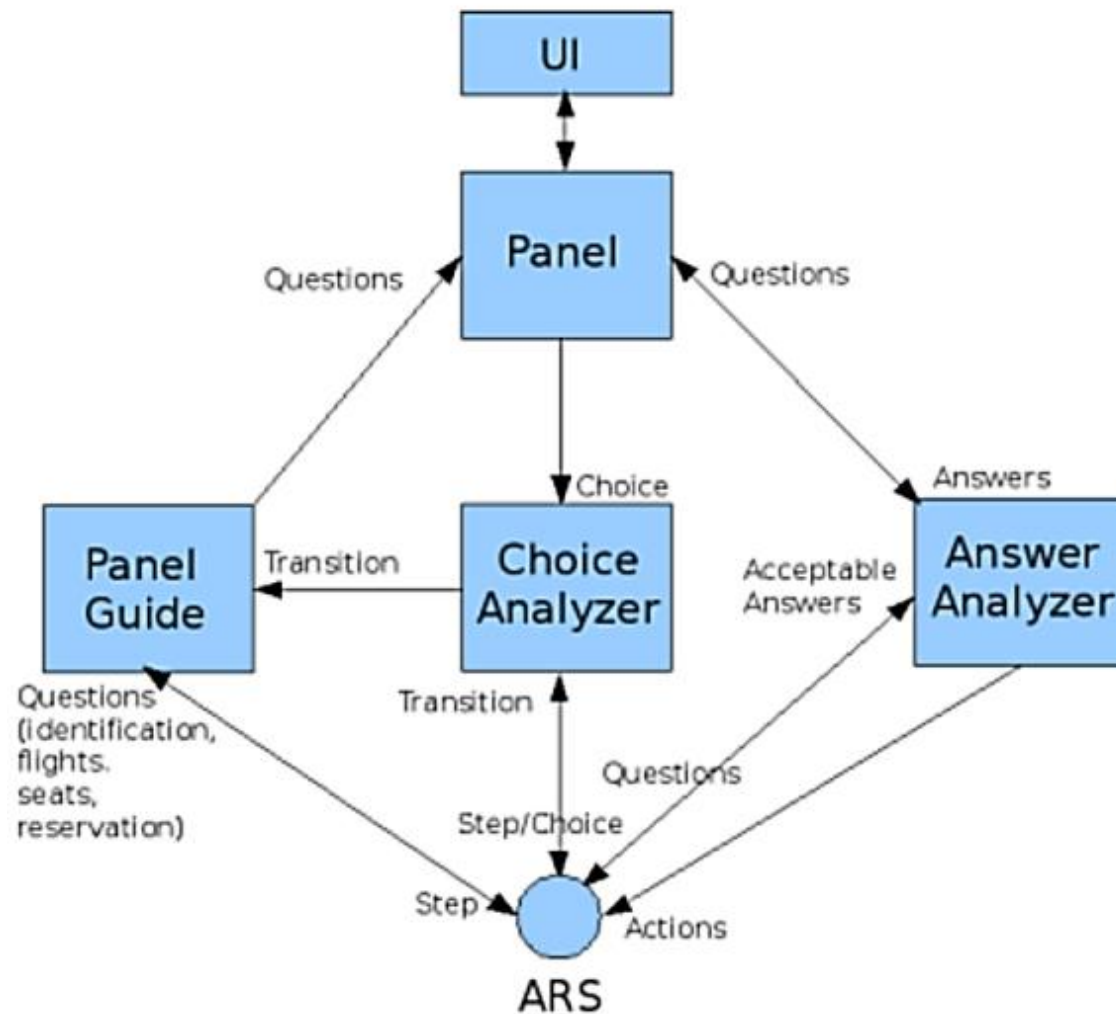


## Software Design vs Architecture

---

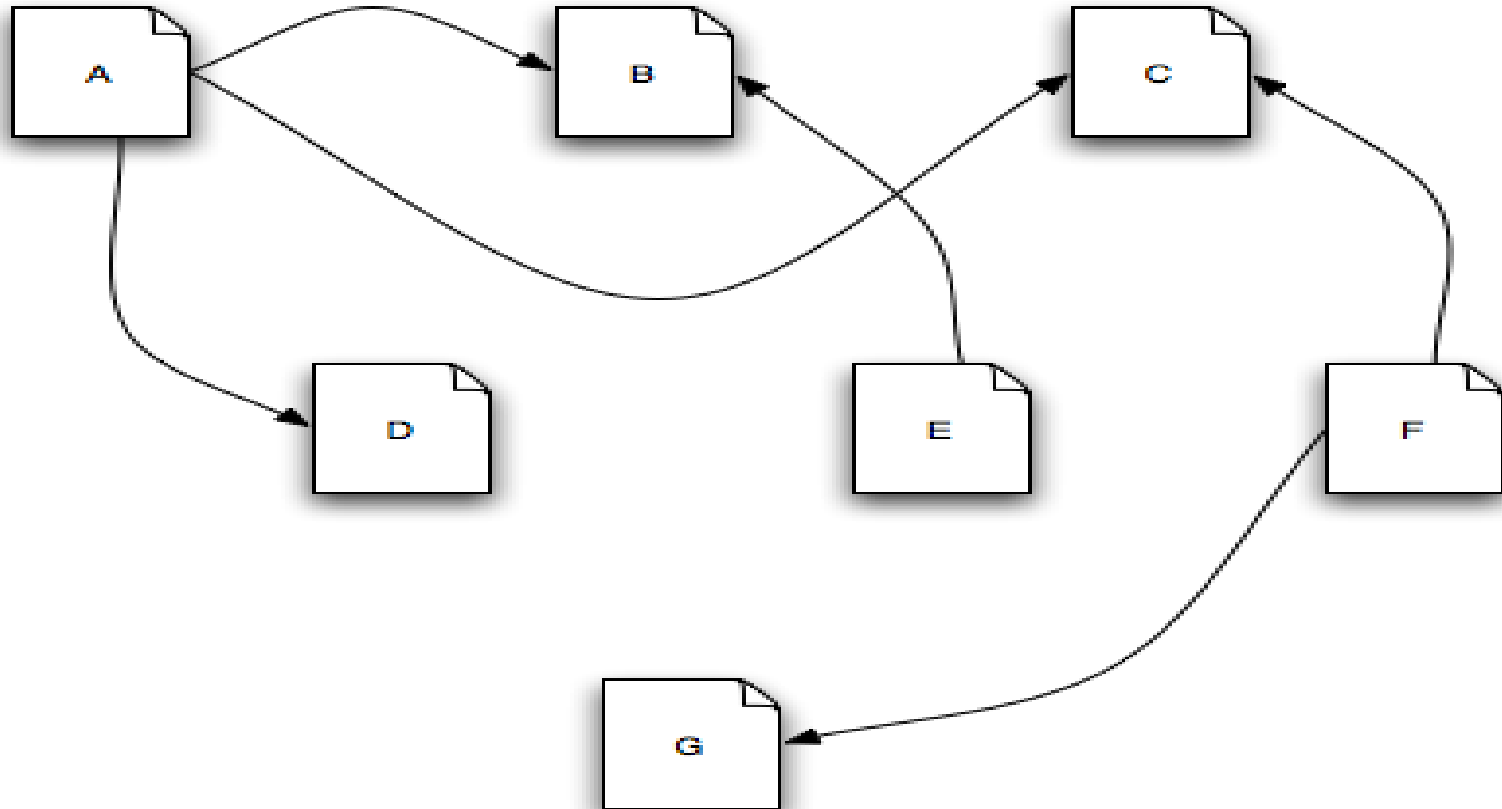
- Architecture is concerned with the *selection of architectural elements*, their *interaction*, and the *constraints* on those elements and their interactions
- Design is concerned with the *modularization* and *detailed interfaces* of the design elements, their *algorithms and procedures*, and the data types needed to support the architecture and to satisfy the requirements.
- Architecture...is specifically not about...details of implementations (e.g., algorithms and data structures.)

# Software Architecture Diagram

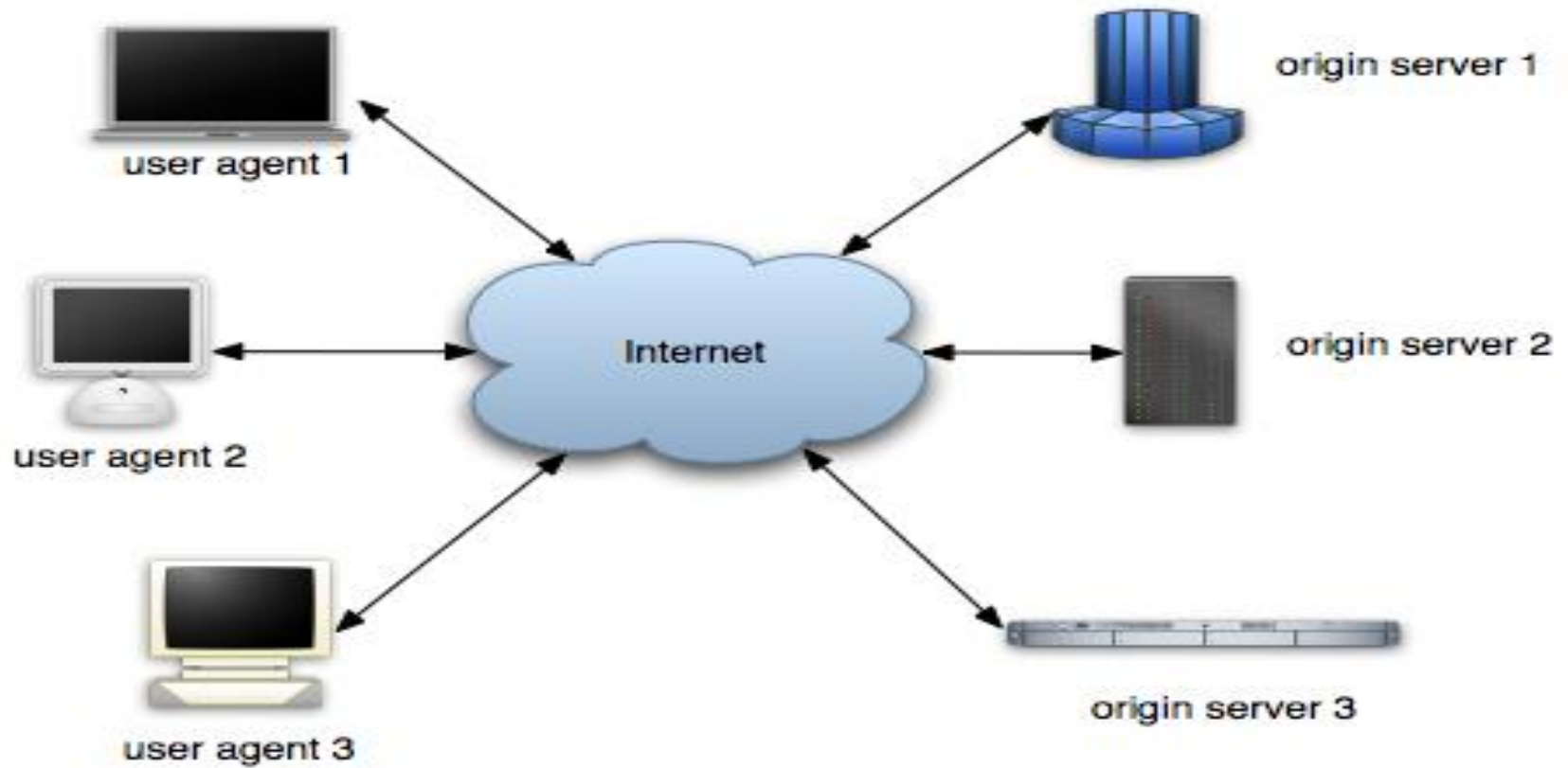


## Software Architecture Diagram--WWW

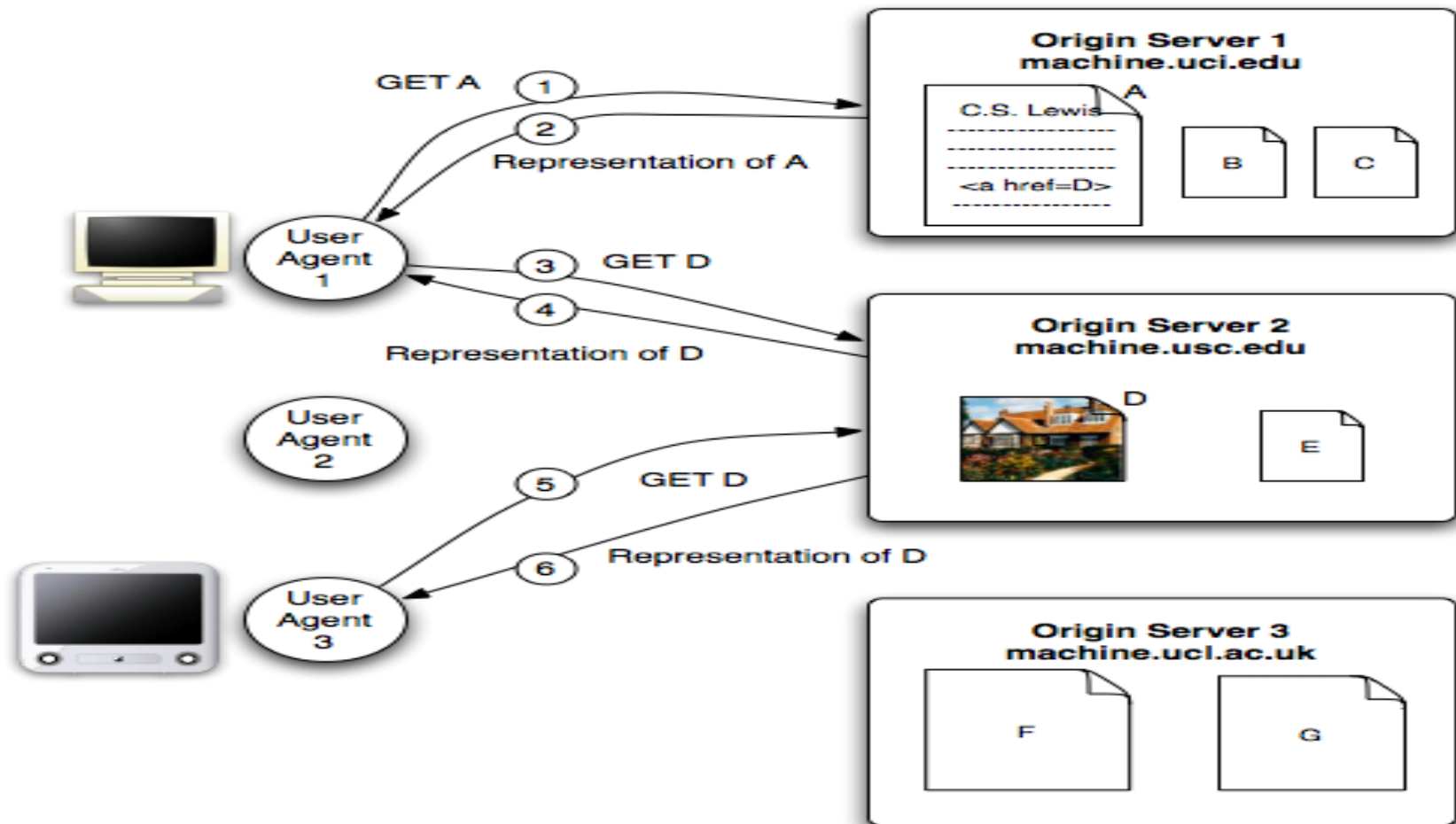
---



## Software Architecture Diagram--WWW



## Software Architecture Diagram--WWW



## Software Architecture Diagram--WWW

---

- Architecture of the Web is wholly separate from the code
- There is no single piece of code that implements the architecture.
- There are multiple pieces of code that implement the various components of the architecture.
  - » E.g., different Web browsers