

ENGINEERING ONLINE

Lecture Notes

Course Number: CSC 513

Instructor: Dr. Singh

Lecture Number: 6



Enterprise Architecture Observations

Continual squeeze on funds, staffing, and time available for IT resources

- ▶ Demand for rapid development and deployment of applications
- ▶ Demand for greater ROI (RETURN ON INVESTMENT)
- ▶ Essential tension
 - ▶ Need to empower users and suborganizations to ensure satisfaction of their local and of organizational needs
 - ▶ Ad hoc approaches with each user or each suborganization doing its own IT cause failure of interoperability

Avoid

Enterprise Architecture Principles

Business processes should drive the technical architecture

- ▶ Define dependencies and other relationships among stakeholders (including users) and suborganizations of an organization
- ▶ Message-driven approaches are desirable because they decouple system components
- ▶ Event-driven approaches are desirable because they help make a system responsive to events that are potentially visible and significant to users

STAKEHOLDERS : INTERESTED PARTIES (THAT YOU CARE ABOUT)
WITH A STAKE IN
THE SYSTEM

NCSU

LIST STAKEHOLDERS

- ✓ FACULTY
- DEANS (ADMINISTRATORS)
- RESEARCH SPONSORS
- ✓ STUDENTS
- STAFF
- BUS TRANSPORTATION DEPT
- STATE GOVT
- RESTAURANTS (ON CAMPUS)
- RESEARCH ORGS
- OTHER UNIVS.
- ALUMNI
- STUDENT GOVT
- PROSPECTIVE EMPLOYERS
- FINANCE

BUSINESS PROCESSES

COURSE
SCHED

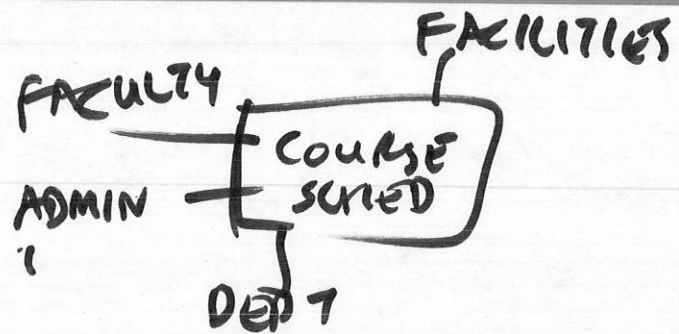
ENROLLMENT

SPACE
RENTAL

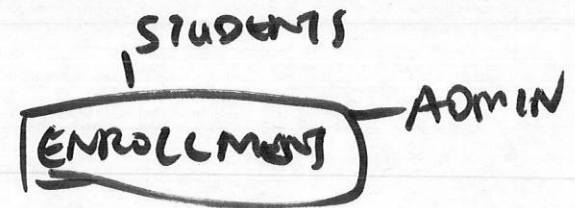
PAYMENTS

OUTREACH
(PROVIDE
INFO)





PRE AWARD



POST AWARD



Architecture Modules: Applications

(FRONTENDS
OF BUSINESS
PROCESSES)

- ▶ Often directly visible to users
 - ▶ Application deployment
 - ▶ Data modeling and integrity
 - ▶ Business intelligence: decision support and analytics
- ▶ More technical but indirectly visible to users
 - ▶ Interoperation and cooperation
 - ▶ *Ontologies*: representations of domain knowledge
 - ▶ Component and model repositories
 - ▶ Business process management

Architecture Modules: Systems

Functionality used by multiple applications

- ▶ Middleware: enabling interoperation, e.g., via messaging
- ▶ Identity management, e.g., ID across a system to support Single Sign On
- ▶ Security and audit
- ▶ Accessibility
- ▶ Policy repositories and engines

Architecture Modules: Infrastructure

- ▶ Connectivity
- ▶ Platform: hardware and operating systems
- ▶ Storage
- ▶ System management

MANY APPS
.....
FEW
SYSTEMS
(POTENTIALLY)
MANY
INFRASTRUCTURE

A WAY TO ADDRESS
THE TENSION
OF P53

CONTRAST

.....
IP
.....



ENROLLMENT

APP

PORTAL, ELIGIBILITY
COURSE DATA
STUDENT

SYS

DBMS; ID(WRAP)

WEB SERVER
(APACHE)
APP SERVER

INF

NETWORK
SERVER - 003

PHYSICAL
SERVER

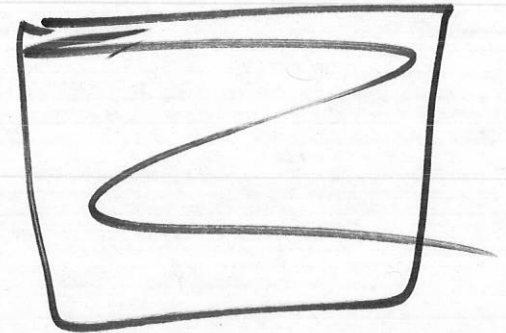
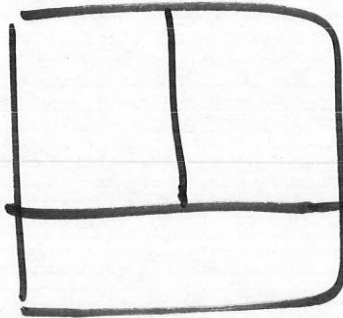
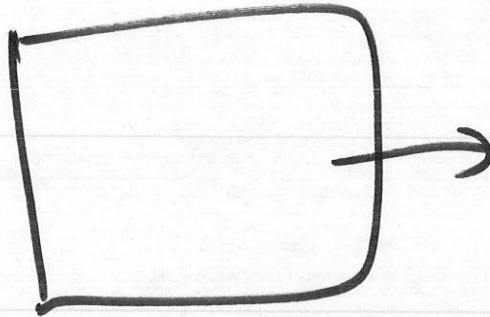
COURSE SCHED CATALOG

SERVER 04

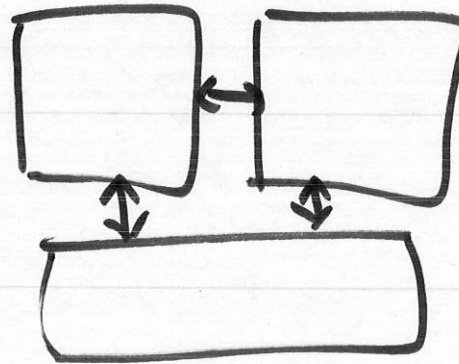


TRND

EMERGING
STRUCTURE



PAPER
ARCH



Functionalities in a Working Enterprise System

precursor to three-tier

- ▶ Presentation: user interaction
 - ▶ A large variety of concerns about device constraints and usage scenarios
- ▶ Business logic
 - ▶ Application-specific reasoning
 - ▶ General rules
- ▶ Data management
 - ▶ Ensuring integrity, e.g., entity and referential integrity (richer than storage-level integrity)
 - ▶ Enabling access under various kinds of problems, e.g., network partitions
 - ▶ Supporting recovery, e.g., application, operating system, or hardware failures

Enterprise Functionalities

Bases for choosing the above three-way partitioning as opposed to some other

- ▶ Size of implementations
- ▶ Organizational structure: who owns what and who needs what
- ▶ Staff skill sets
 - ▶ User Interface: usability and design
 - ▶ Programming
 - ▶ Database
 - ▶ Policy tools
- ▶ Products available in the marketplace

*Emerge hand-in-hand with markets
↳ Based on architecture*

Outline

Challenges of Electronic Business

Architecture in IT

- Enterprise Architecture

- Tiered Architecture

- Web Architecture

- Middleware

- Deployment Architecture

Contracts and Governance

XML Concepts and Techniques

XML Modeling and Storage

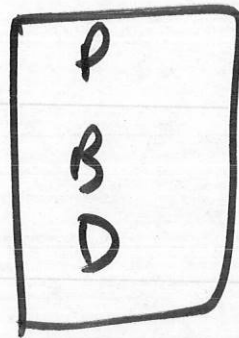
Summary and Directions

One-Tier and Two-Tier Architectures

- ▶ One tier: monolithic systems; intertwined in the code base
 - ▶ Historically the first (usually the case)
 - ▶ Common in legacy systems
 - ▶ Difficult to staff, maintain, and scale up
- ▶ Two-tier: separate data from presentation and business logic
 - ▶ Classical client-server (or fat client) approaches
 - ▶ Mix presentation with application business rules
 - ▶ Change management

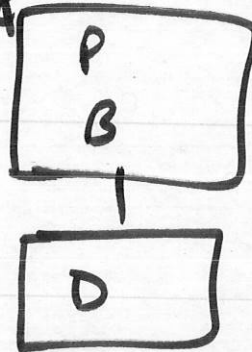
Three-Tier Architecture

- ▶ Presentation tier or frontend
 - ▶ Provides a view to user and takes inputs
 - ▶ Invokes the same business logic regardless of interface modalities: voice, Web, small screen, ...
- ▶ Business logic tier or middle tier
 - ▶ Specifies application logic
 - ▶ Specifies business rules
 - ▶ Application-level policies
 - ▶ Inspectable
 - ▶ Modifiable
- ▶ Data tier or backend
 - ▶ Stores and provides access to data
 - ▶ Protects integrity of data via concurrency control and recovery

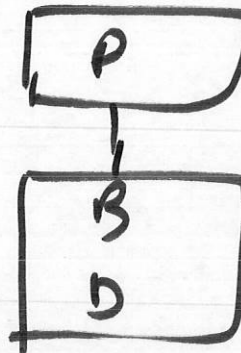


ONE TIER

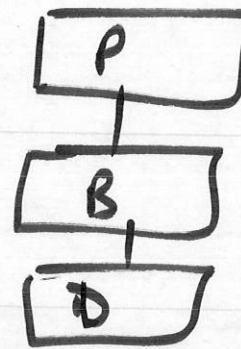
THICK CLIENT
FAT



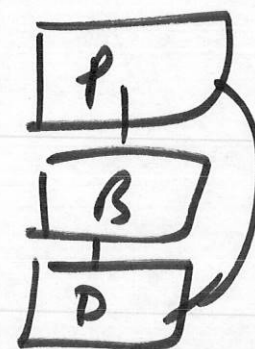
TWO TIER



?

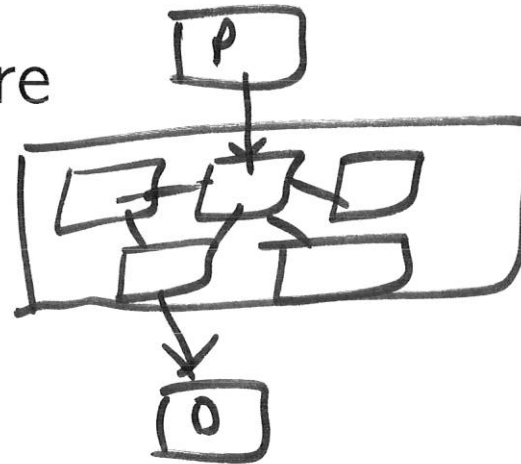


THREE TIER



Multitier Architecture

Also known as n-tier



COURSES
STUDENT
ELIGIBILITY
EVENTS (TWO FEW
COURSES)

- ▶ Best understood as a componentized version of three-tier architecture where
 - ▶ Functionality is assembled from parts, which may themselves be assembled
 - ▶ Supports greater reuse and enables greater dynamism
 - ▶ But only if the semantics is characterized properly
- ▶ Famous subclass: service-oriented architecture

(ENTERPRISE VERSION AS ABOVE)
COULD ALSO USE EXTERNAL
SERVICES)