

Enterprise Application Architectures

- Overview
- Component-based application architectures
- Platforms for component-based architectures
- How to build component-based architectures

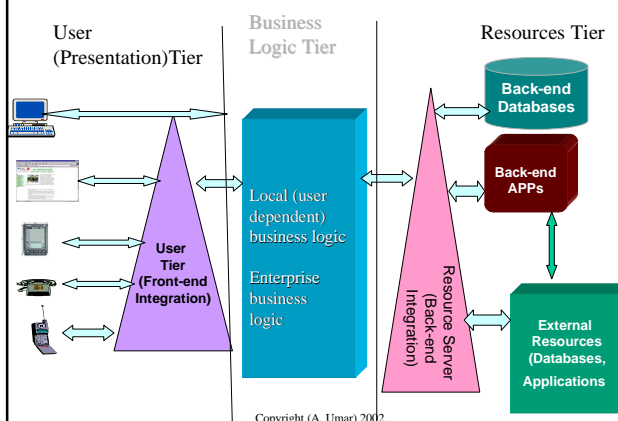
Amjad Umar

Introduction

- Need to develop application architectures for flexible, portable, and interoperable enterprise applications.
- Focus is on building solutions that translate logical application architectures into a software architecture that can be implemented by using appropriate IT infrastructure.
- Software architecture:
 - components (software components)
 - what they do
 - how do they interface/interact with others

Copyright (A. Umar) 2002

General View



Needed Architectures

- System architectures that provide
 - Increased support for a variety of users
 - Flexibility in business logic so that new features can be added quickly by using off the shelf products
 - Integration with back-end systems with emphasis on trading partners .
In particular:
 - legacy systems from a variety of players in EB
 - new systems from new business partners and suppliers
 - Systems on trading networks
- Additional requirements
 - Internet scale (thousands of users)
 - Internet connectivity (*unpredictable open Internet*)
 - Multiple customers with multiple interests
 - Multiple configurations and user profiles
 - High-volume infrastructure
 - Commercial service provision

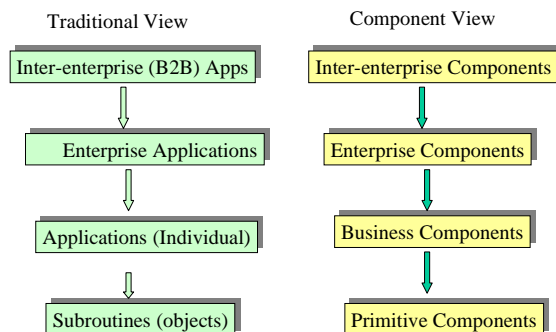
Copyright (A. Umar) 2002

Components- Revisited

- **Primitive Component.**
 - the lowest level component that performs one specified function such as display a stock quote.
 - Resides on one machine
- **Business Component.**
 - implements a single business concept such as purchasing, payment, billing, etc.
 - usually consists of one or more primitive components and can be thus distributed across machines to satisfy business needs.
- **Enterprise Component.**
 - implements an enterprise wide business functionality and consists of one or more business components.
- **Inter-enterprise Components**
 - very large grained federated systems that handle, for example, B2B functionalities.

Copyright (A. Umar) 2002

Component Versus Traditional Views



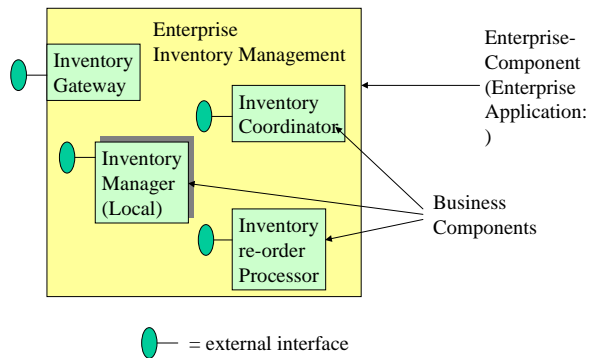
Copyright (A. Umar) 2002

Business components - Closer Look

- A business component (BC)
 - implements a business concept
 - has a well defined interface
 - is self-contained.
- BCs are key to component-based architectures -- can compose large scale systems by assembling BCs
- BC may rely on other components, but must not rely on implementation choices of those components
- BC roughly represents a business application

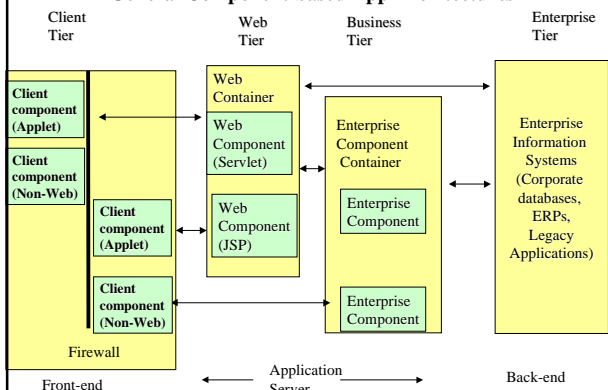
Copyright (A. Umar) 2002

Enterprise Inventory Management as BCs

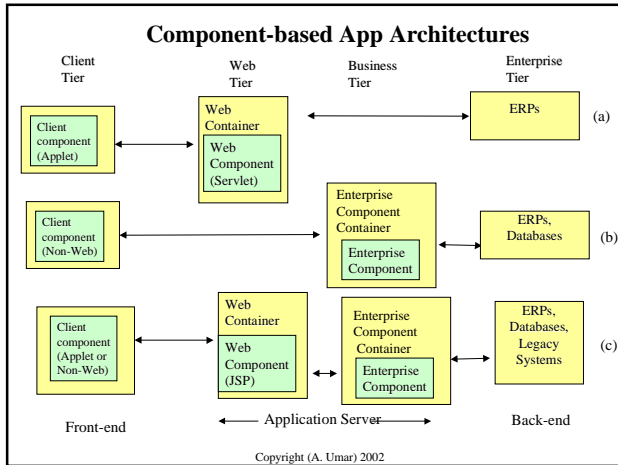


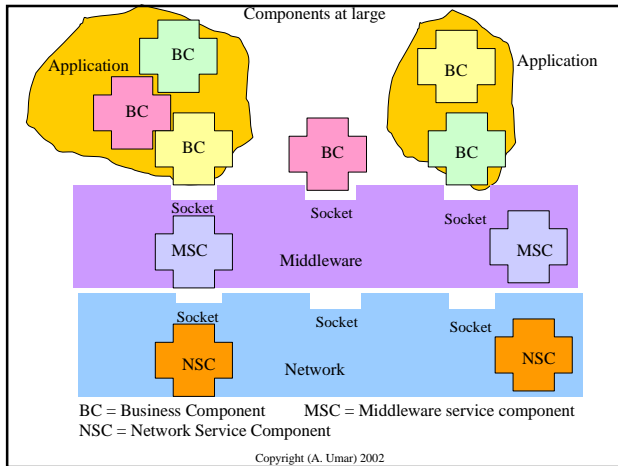
Copyright (A. Umar) 2002

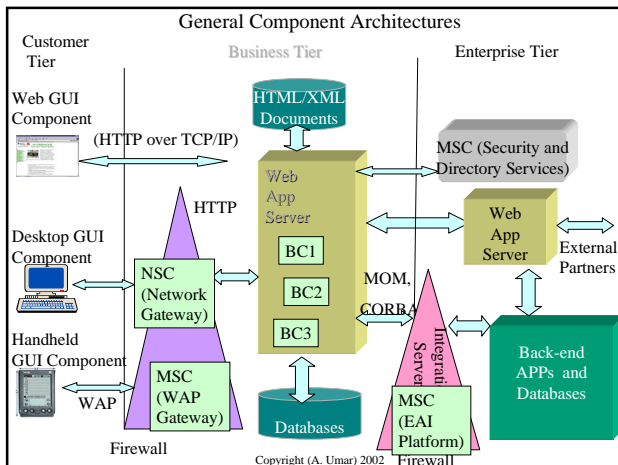
General Component-based App Architectures



Copyright (A. Umar) 2002





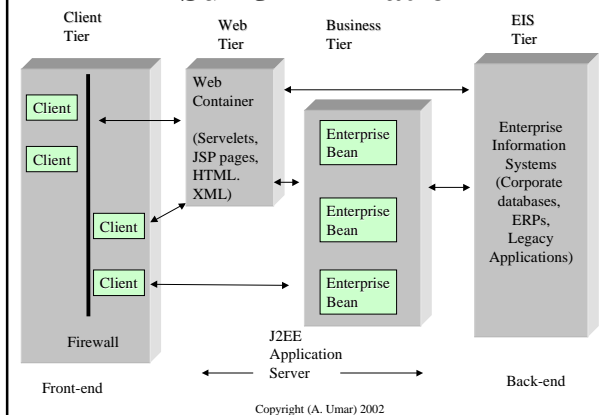


Component-based Platforms

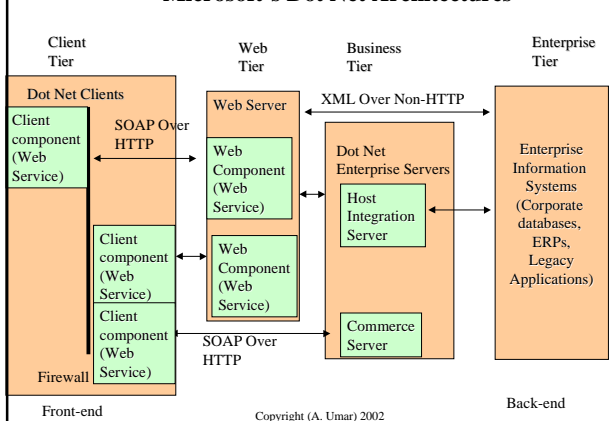
- Component-based platforms are commercially available for deployment of enterprise applications
- They provide the “plumbing” to build, deploy, and run component-based software
- The two principal examples:
 - Sun's J2EE
 - Microsoft's Dot Net.
- Many EC/EB platforms, such as IBM websphere, are also becoming component-based platforms

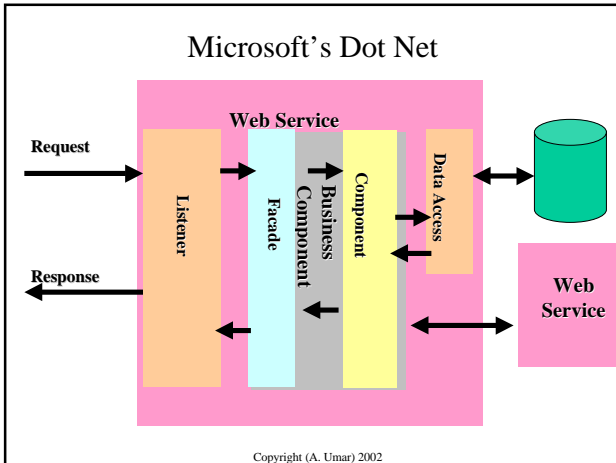
Copyright (A. Umar) 2002

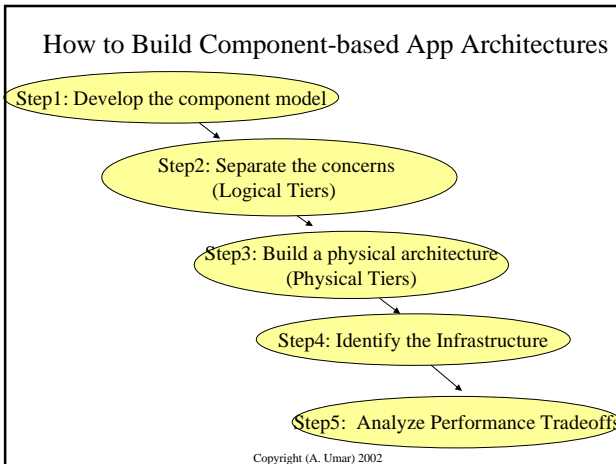
Sun J2EE Platform

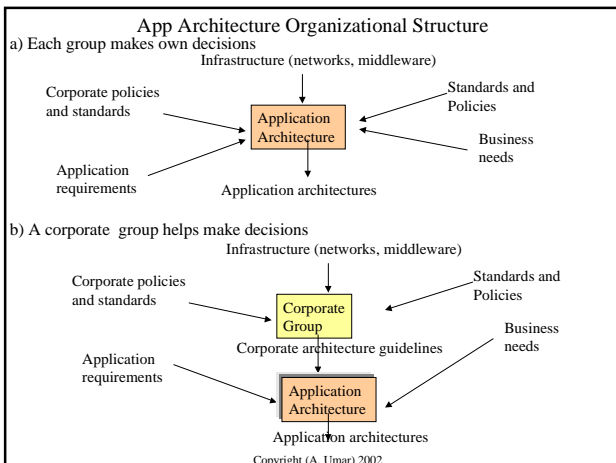


Microsoft's Dot Net Architectures



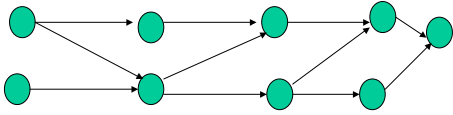




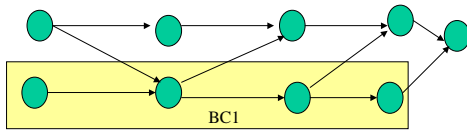


Step1:Build the component Model

A) Identify Primitive Components

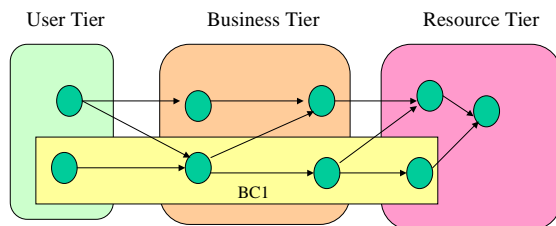


B) Identify Business Components (BCs)



Copyright (A. Umar) 2002

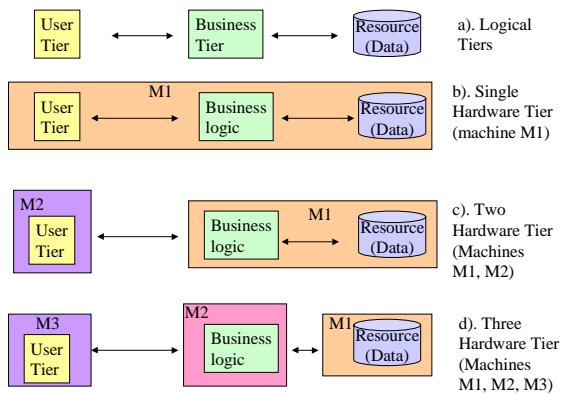
Step2: Separation of Concerns



A given BC may span user, business and resource tiers
(e.g., an off the shelf package)

Copyright (A. Umar) 2002

STEP 3: Logical Versus Hardware Tiers



Copyright (A. Umar) 2002

