



## Chapter 7 : Defining the System Architecture

Tannaz R.Damavandi  
Cal Poly Pomona

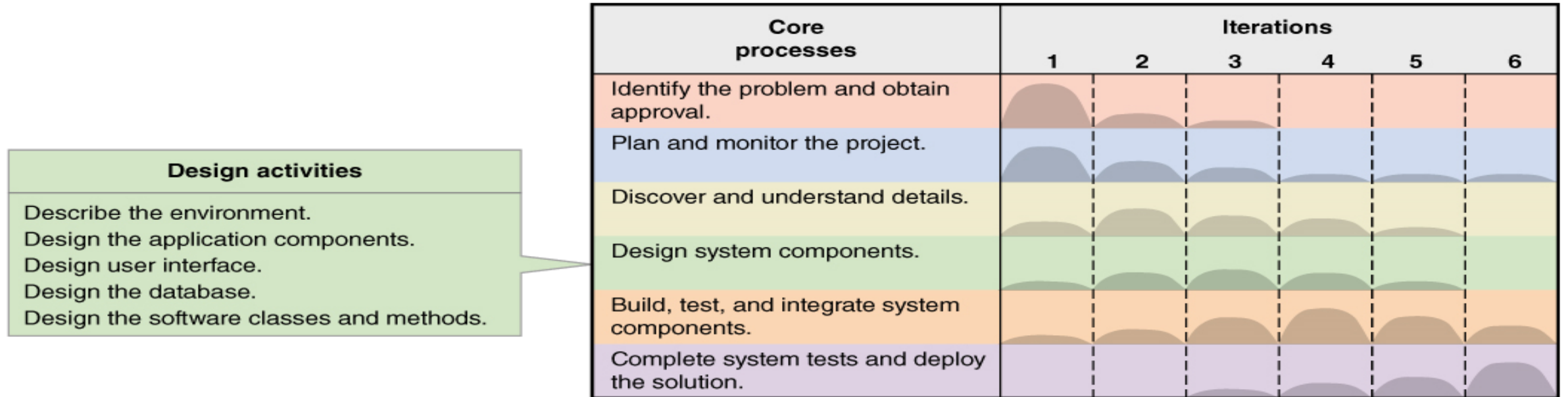
# Outline

- Anatomy of a Modern Information System
- Architectural Concepts
- Interoperability
- Architectural Diagrams
- Describing the Environment
- Designing Application Components

# Overview

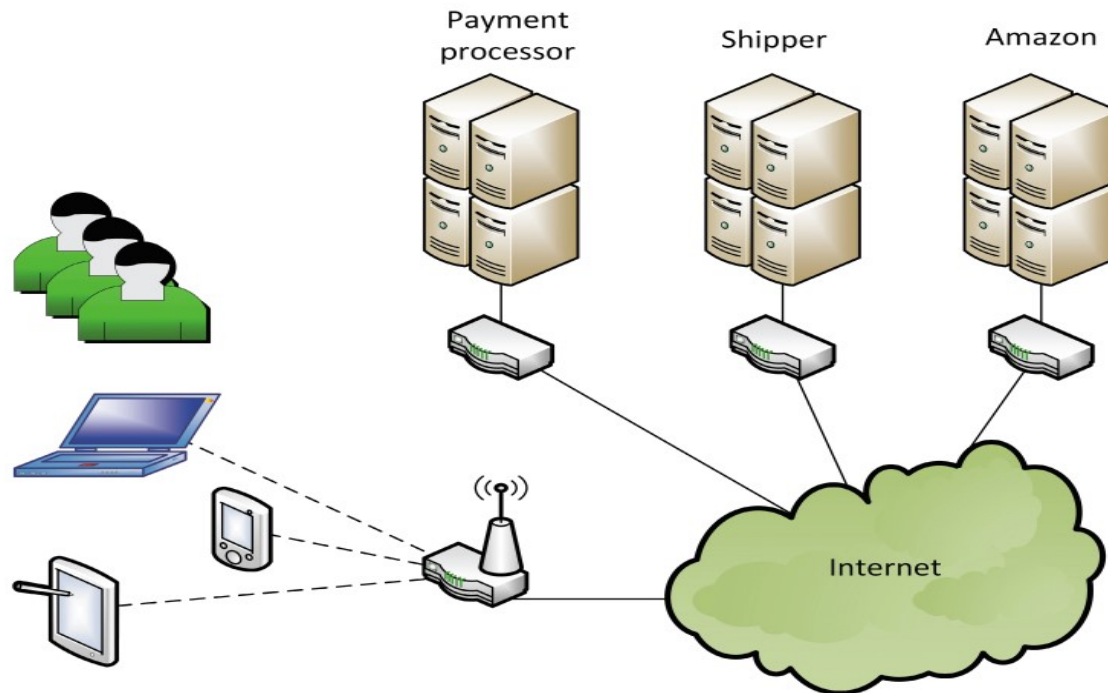
- An important part of new system development is choosing appropriate technologies
- Explain and provide a summary of technology and architectural concepts
- describe the details for the activity – *Describe the Environment*
- Describe the details for the activity – *Design the application components*

# Activities of “Design System Components”



# Anatomy of Modern System- Computing Devices

- **Server** – manages shared resources and enables users and other computers access to these resources
  - **Server Farm:** Very large databases and very high use
- **Personal computing devices or clients**
  - Desktops, laptops, tablets, smartphones...



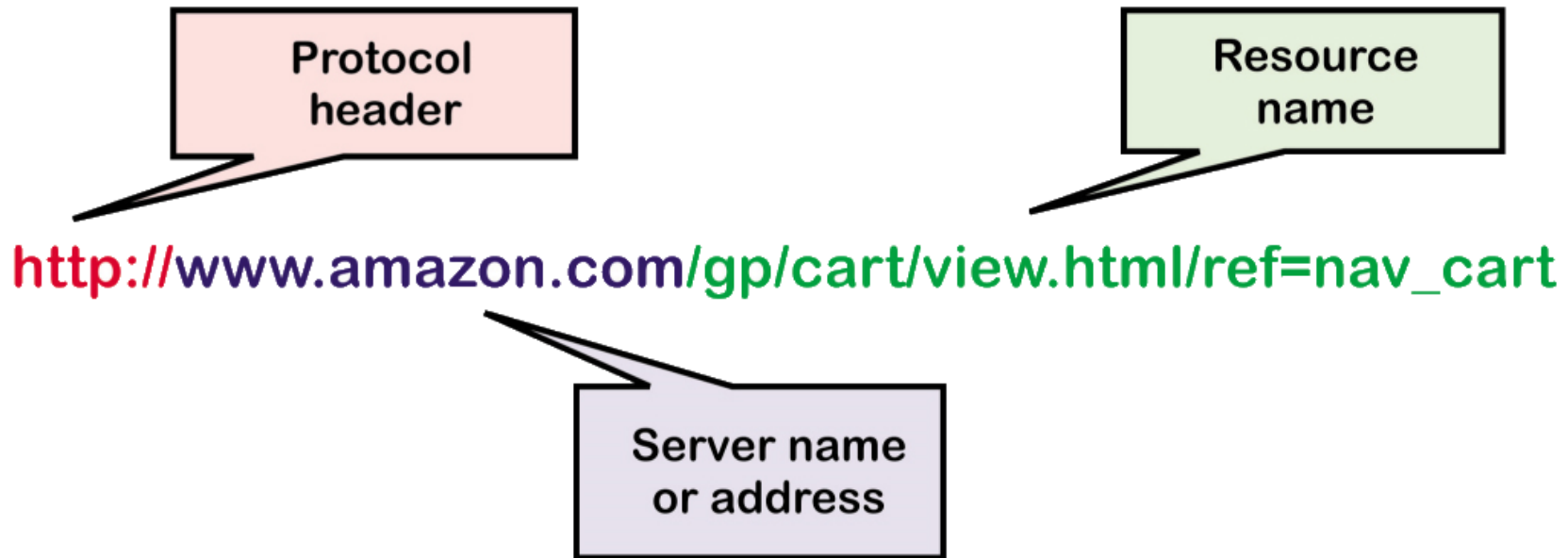
Simplified architecture for application (Amazon.com)

# Anatomy - Networks

- **Computer network** – hardware, software, transmission media
- **Internet backbone** – High-capacity with high-bandwidth trunk lines and large high-speed computers .
  - Owned by governments and telecom companies
- **Local area network (LAN)** –Small network for a single site
- **World Wide Web (WWW)** - All the interconnected resources accessed through the Internet

# Anatomy – Networks (Cont'd)

- **Uniform Resource Locator (URL)** - The identifier for the Web to locate a particular resource
- **Hyperlink** – The URL of a resource embedded within another resource



# Class Discussion 1

What are the pros and cons of networking specifically WWW?



# Anatomy – Software (Cont'd)

- Web-Based Applications

- Uses a web browser
- Accessed through a URL
- Resides on a Web server
- Uses standard IP protocols

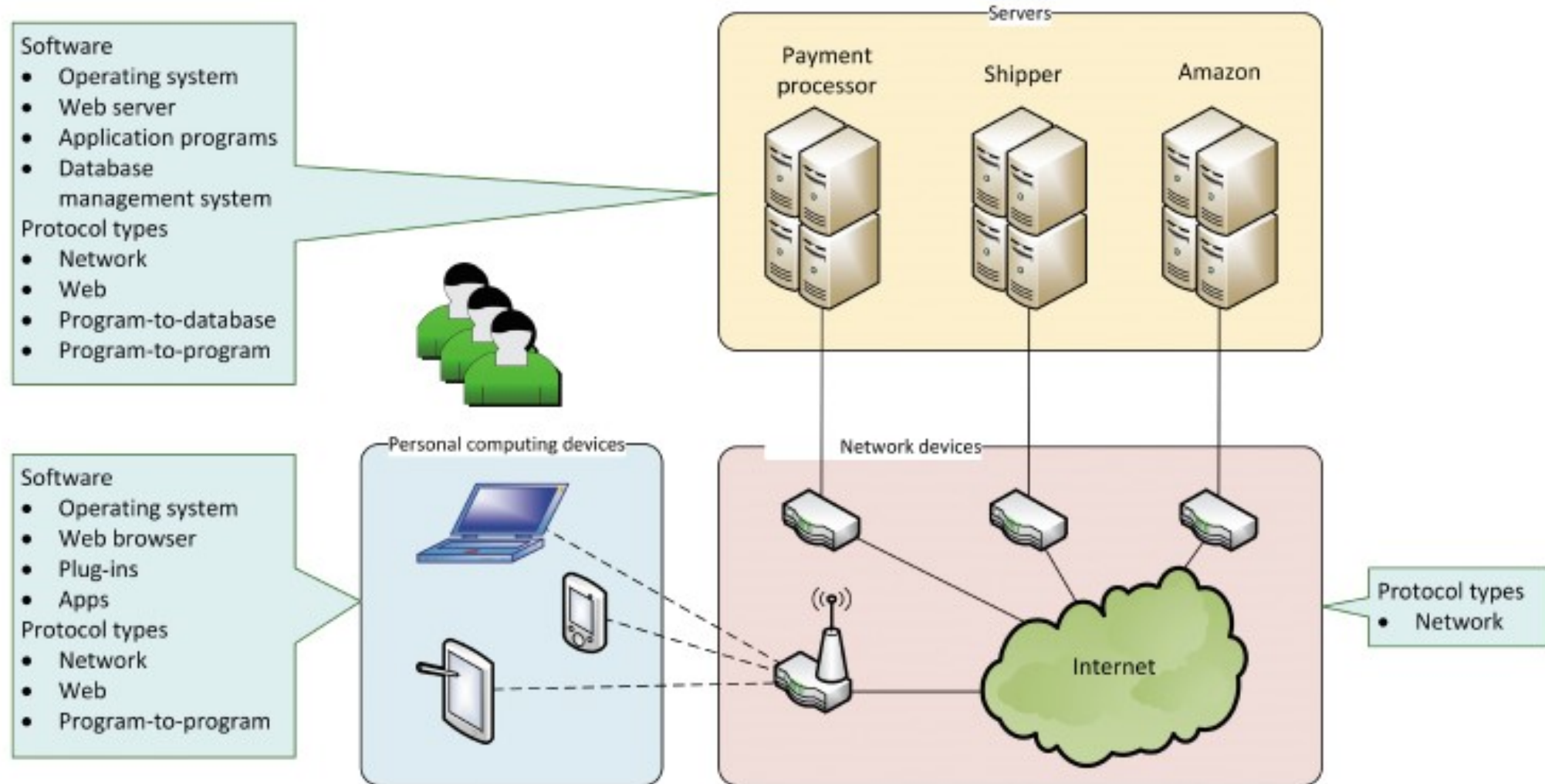
- Embedded Software

- Software apps or functions embedded within another app, such as within a browser or O/S
  - Toolbars: A set of links to Web resources or installed apps that extend the capabilities of a Web browser . i.e. Acrobat PDF toolbar
  - Plug-ins: are often used to correctly display certain types of web content. i.e : Amazon plug-ins to preview 30-second music clips.
  - Widgets: stock market widget is an example for a Web browser widget. It is always displayed in one corner of the browser no matter what Web page is being displayed.

# Anatomy – Protocol

- **Protocol**- A set of languages and rules to ensure accurate and efficient communication and data exchange between hardware and software
- **Network protocols**
  - Virtual Private Network (VPN) Creates a private network but on the Internet by using secure technologies and encryption

# Anatomy – Software and Protocols



# Anatomy – Web Protocols

- **HTML** (hypertext markup language)
  - Protocol for the structure and content of a Web page
- **XML** (extensible markup language)
  - An extensions of HTML that enables defining semantics of tags
- **HTTP** (hypertext transfer protocol)
  - Defines format and content for transfer of Web documents
- **HTTPS** (hypertext transfer protocol secure)
  - Encrypted and secure http transfers

# Software as a Service (SaaS)

- SaaS
  - No software is installed on the user's device
  - Application services is accessed remotely
  - User data is isolated and stored on common servers

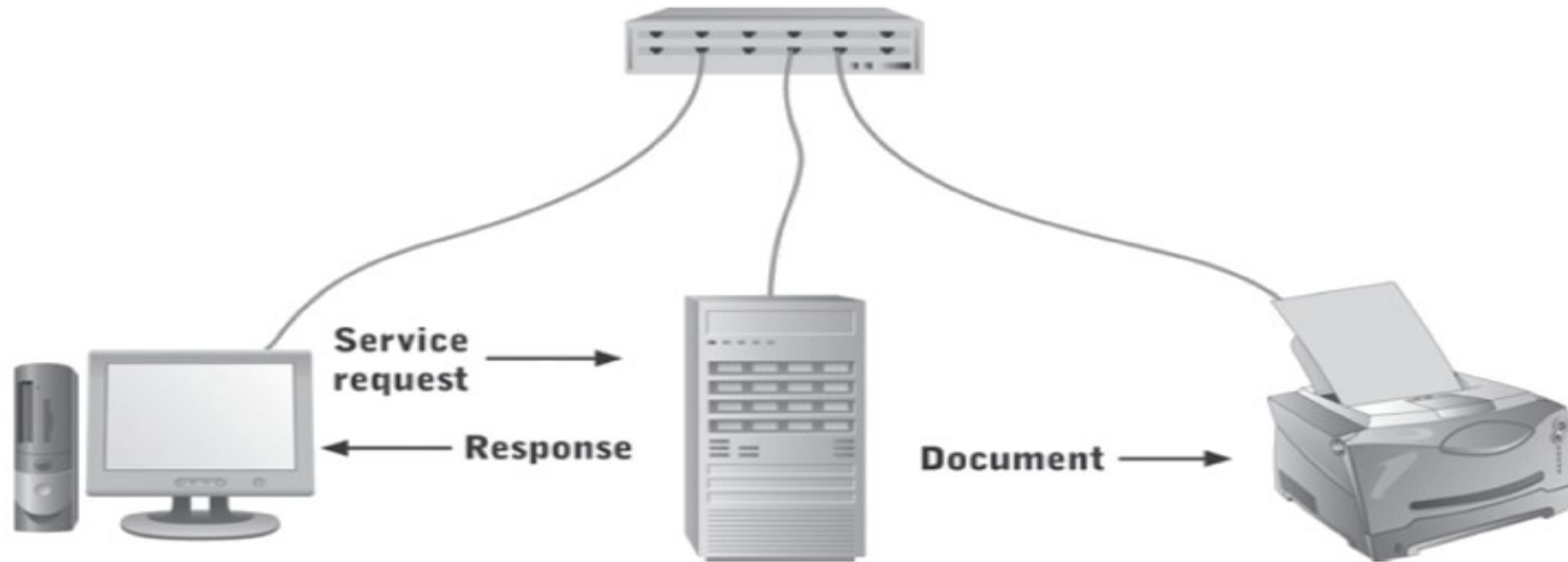
# Web Services

- Web service
  - Software function that is executed with Web standards
    - Access via a URL
    - Inputs sent via the URL
    - Executes remotely
    - Data returned within a Web page
      - Example : Shipping options for Amazon.com

# Distributed Architecture

- Client/Server Architecture

- Software design with part of the application on a server and part on the client

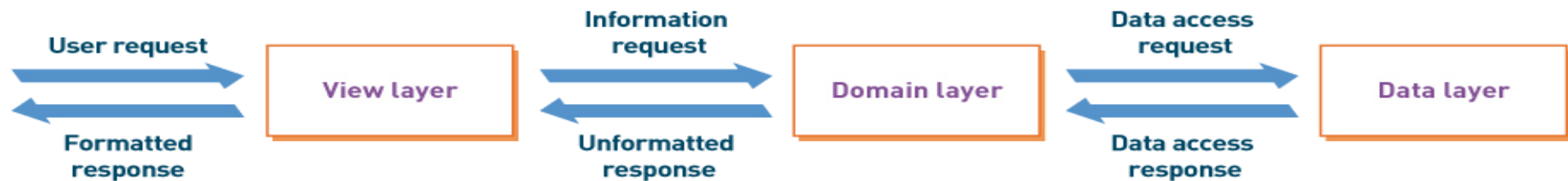


# Distributed Architecture

- Three-Layer architecture

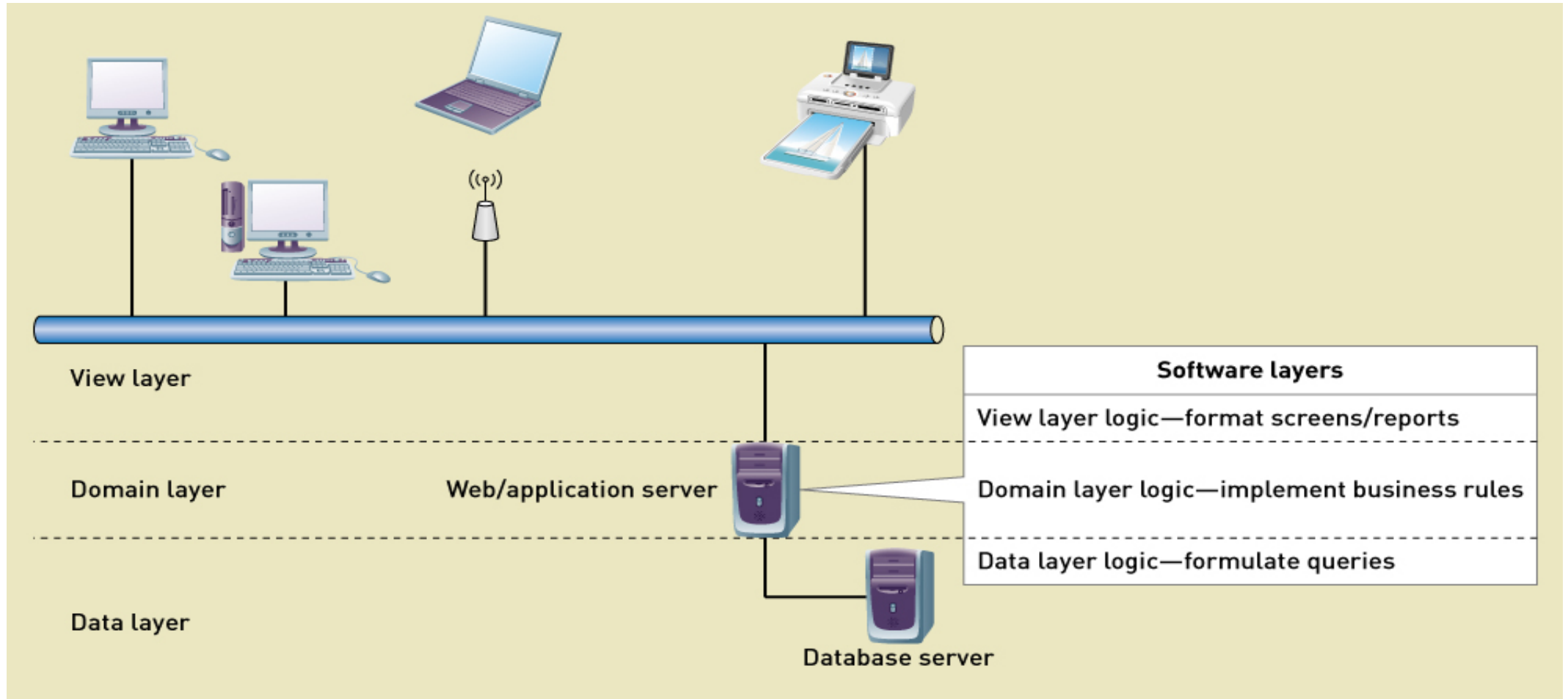
- Client/server architecture with application divided into view layer, logic layer, and data layer

- View layer – which accepts user input and formats and displays processing results
- logic layer – which implements the rules and procedures of business processing
- data layer – which manages stored data, usually in one or more databases





# Three Layer Architecture



# Interoperability

- Interoperability
  - The ability of an application to interact with other software
- Important characteristic in current development projects
  - Understand the environment
  - Reuse software existing components (purchased or in-house)
  - Build components considering interoperability
  - Combine all components into a solution system

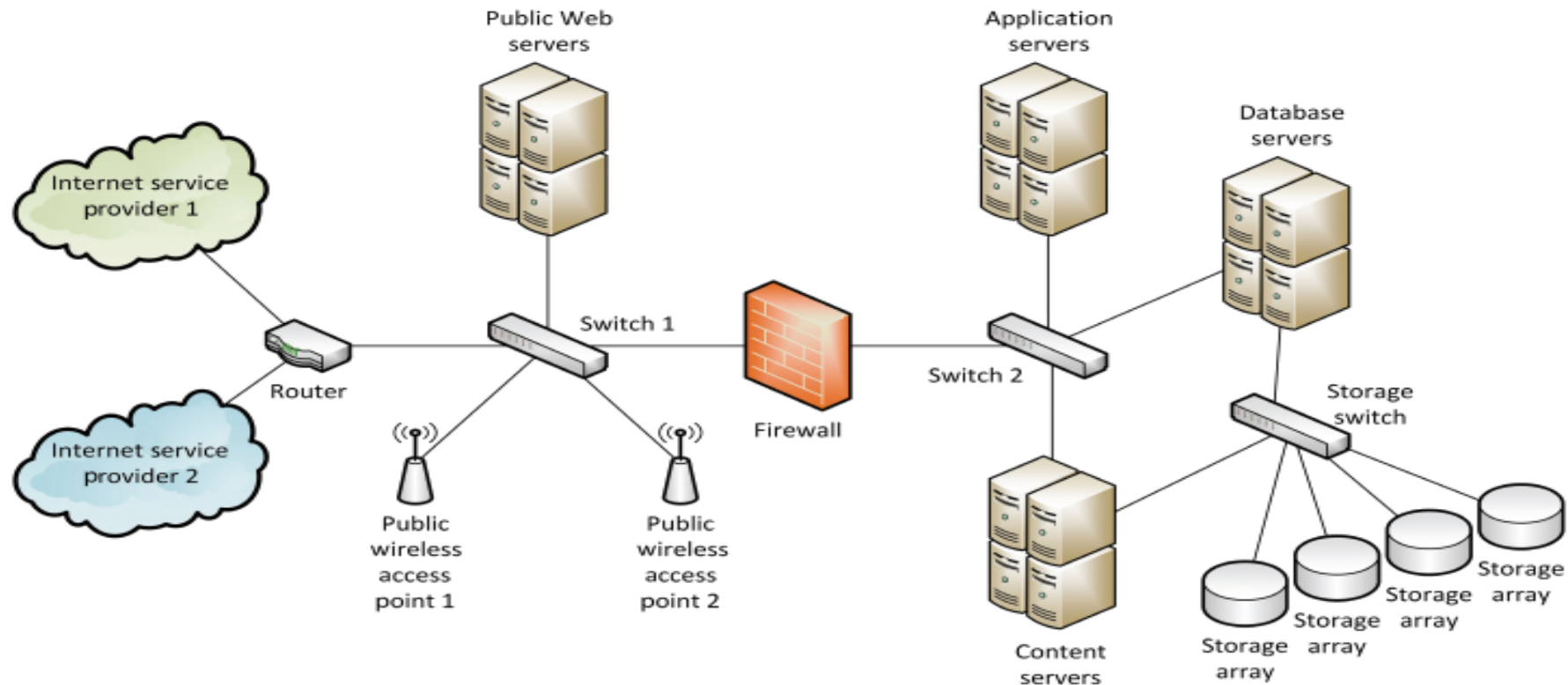
# Diagrams for System Architecture

- Location Diagrams
  - Identify geographical placement of hardware, software, and users



# Diagrams for System Architecture

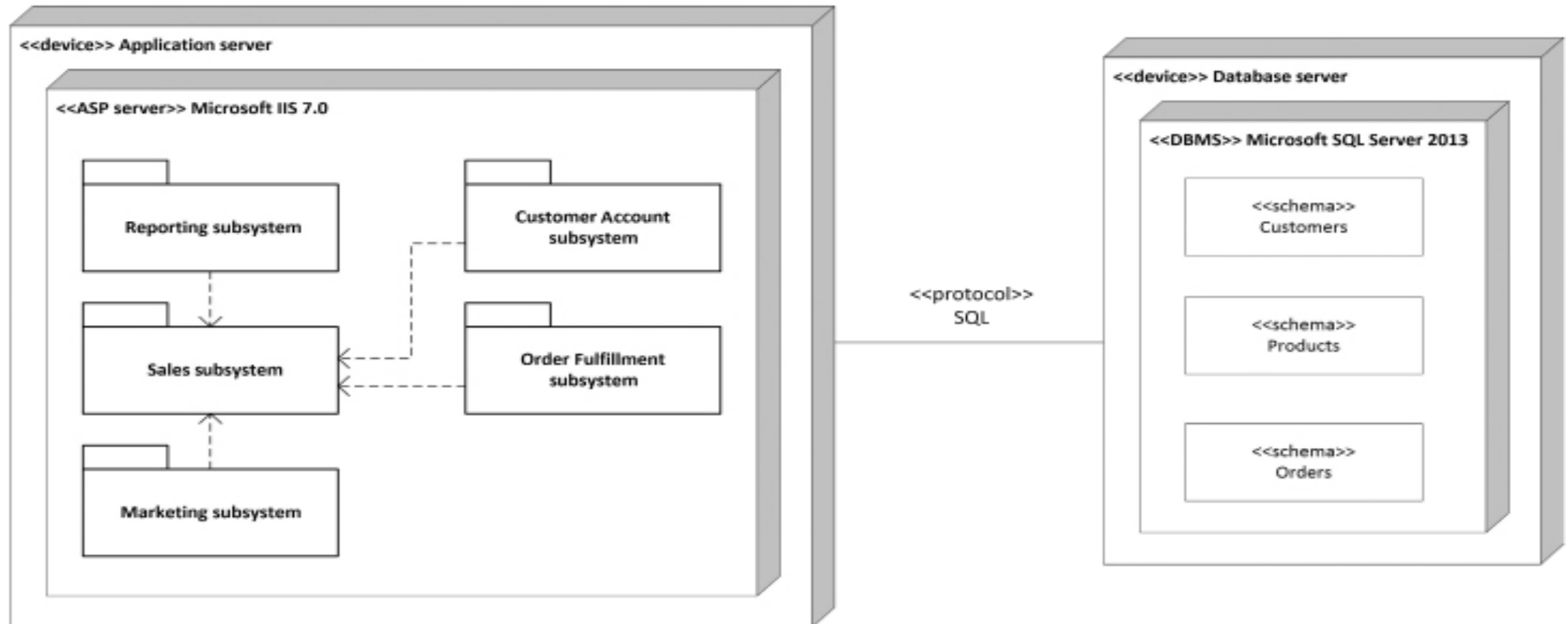
- Network Diagrams
  - How the application software is deployed across the hardware and system software



# Diagrams for System Architecture

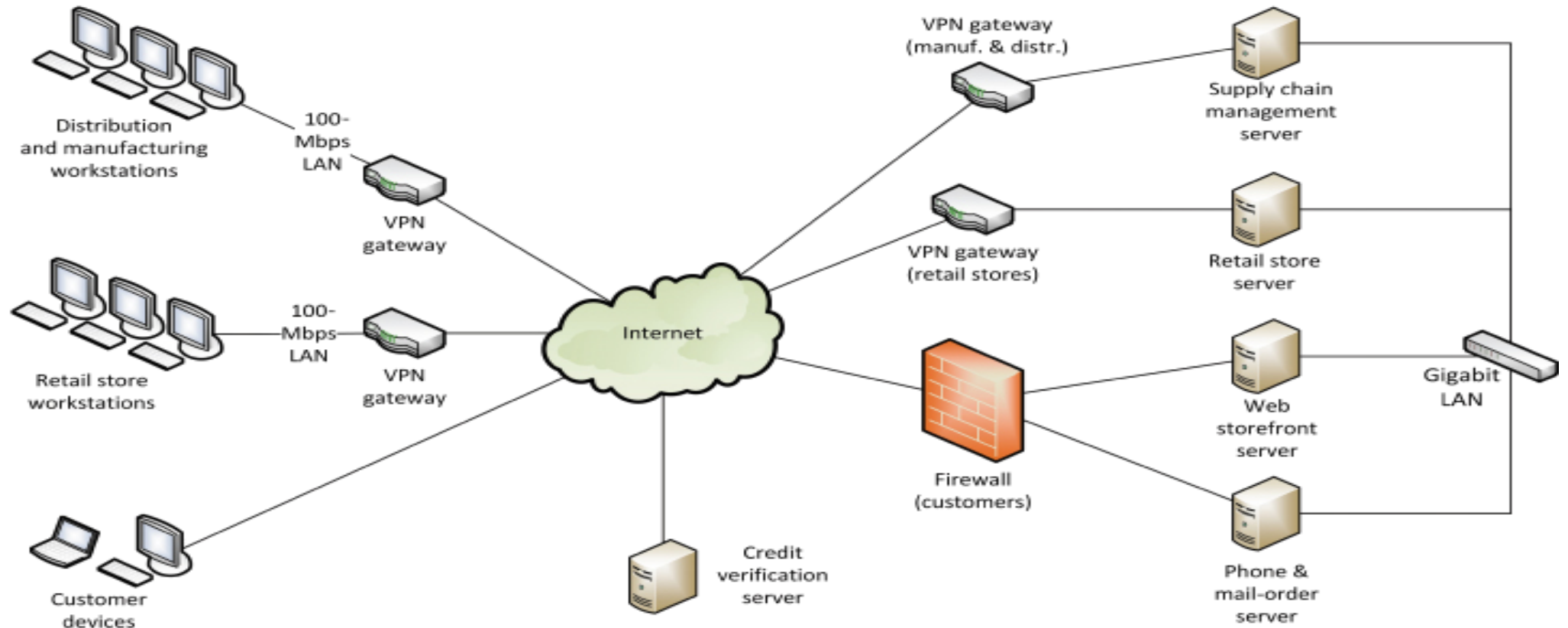
- Deployment Diagrams

- describes how software components are distributed across hardware and system software components.



# RMO Environment- Existing

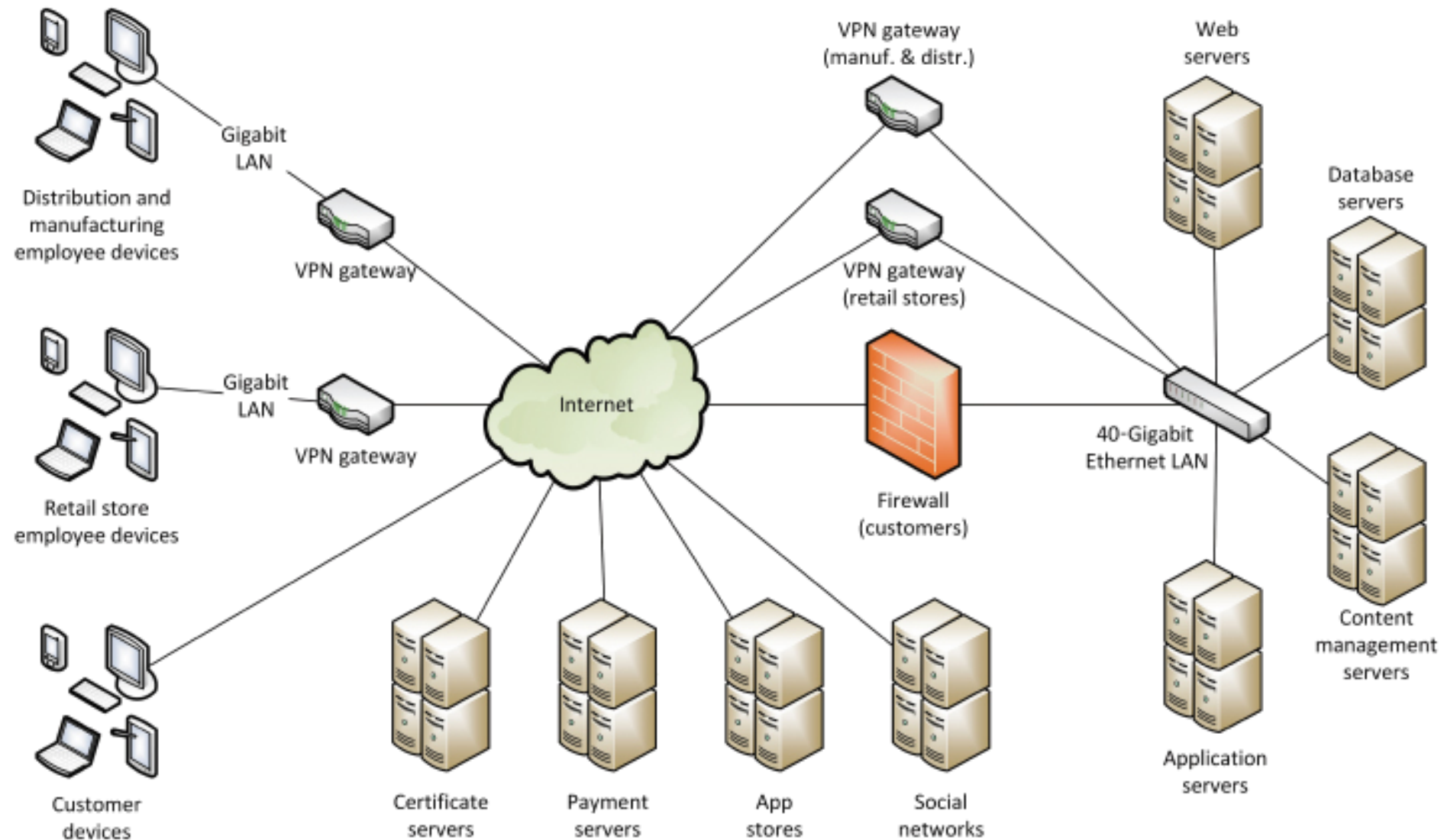
- Current environment prior to new development



# RMO Environment- Proposed

- More mobile devices and apps
- Web application software and content
- Social networking applications
- Security issues

# RMO Environment- Proposed



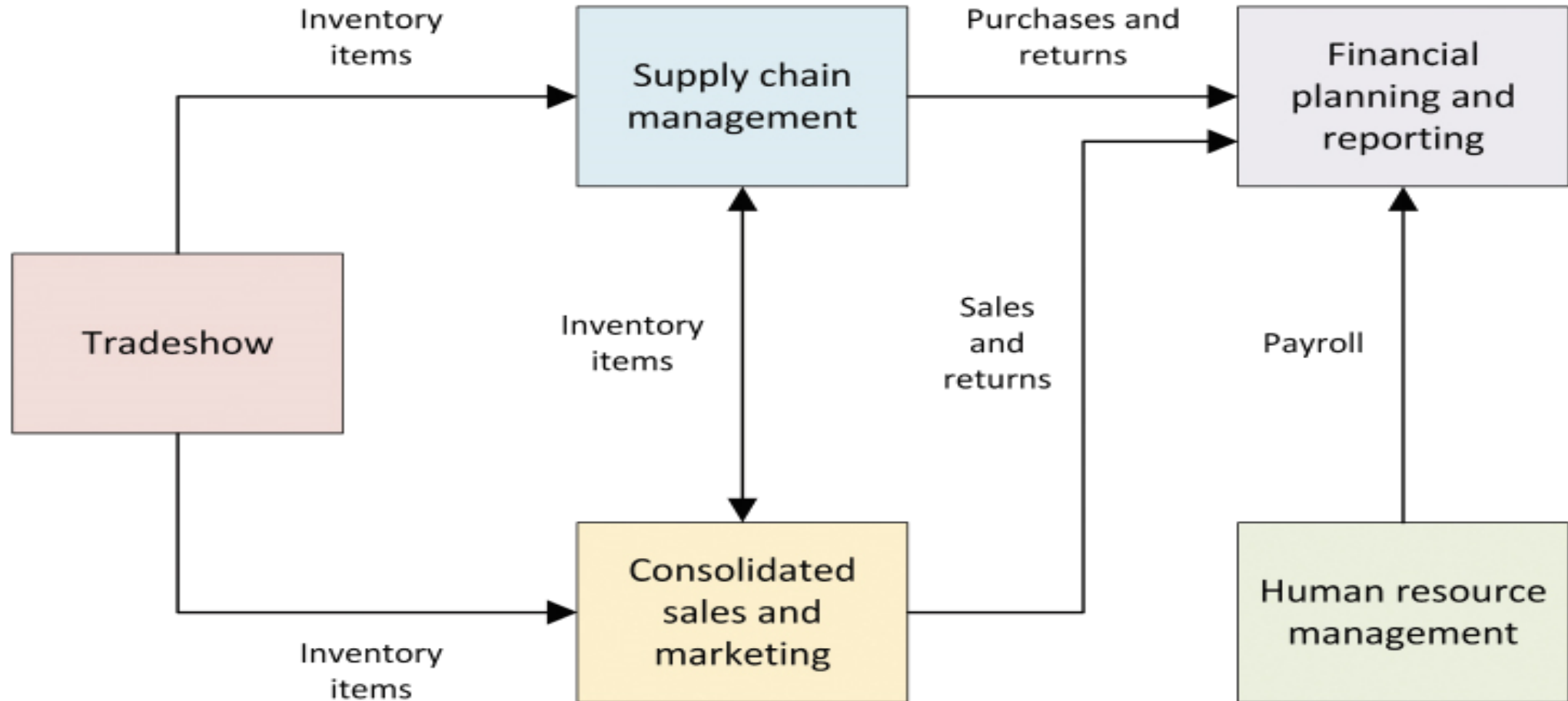


# Designing Application Components

- Application Component Boundaries
  - Which components perform which functions
    - Top-down approach ;factoring
    - Bottom-up approach; grouping
  - How to group functions to build components
    - Events and use cases incorporate attributes that can be used to measure similarity, including the following:
      - **Actors:** Each use case identifies one or more specific actors. Software for use cases that interact with the same actors is a candidate for grouping into a **single application component**.
      - **Shared data:** Use cases that interact with the same domain class(es) are candidates for grouping into a **single application component**.
      - **Events:** Use cases that are triggered by the same external, temporal, or state event are candidates for grouping into a **single application component**.

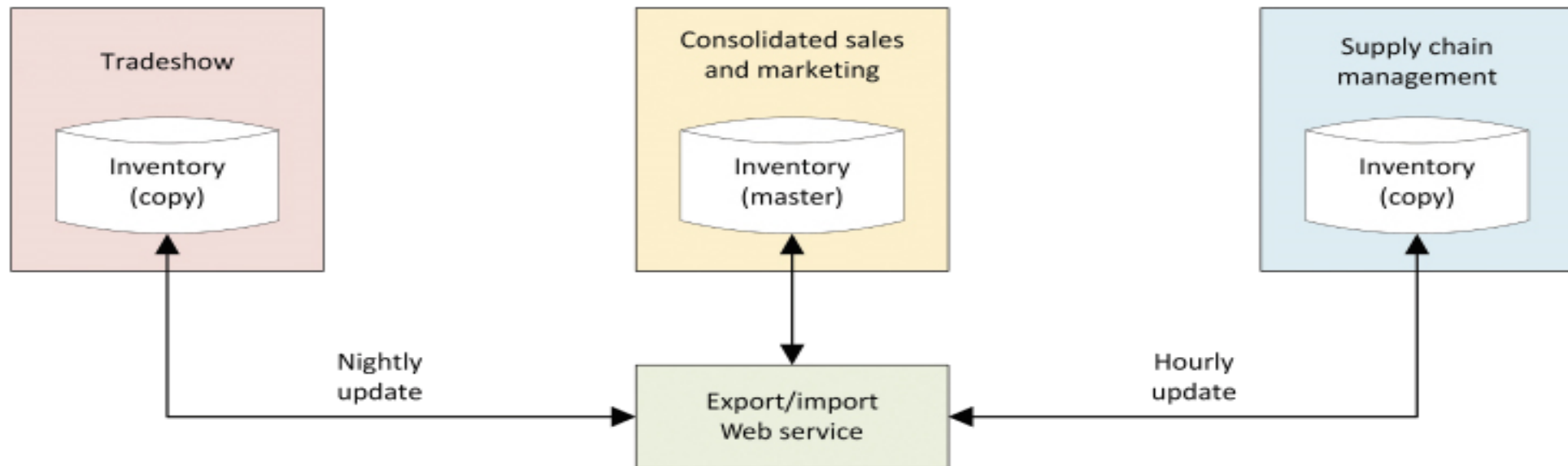
# RMO CSMS Component Integration

- Subsystem integration and data flows



# RMO CSMS Data Ownership

- Who “owns” the data
  - System of record : a system or application component that maintains the current and correct master copy of one or more data items
    - What system is responsible to maintain the data
    - What system has a copy or can access the data



# Summary

- Anatomy of a Modern Information System
  - Consist of computing devices, networks, software, and protocols
  - Deployed as stand-alone software, network based, Web based
- Architectural Concepts
  - SaaS – software as a service
  - Web services
  - Distributed architectures
    - Client/server and three-layer architecture
- Interoperability
  - Getting all the components to work together
- Architectural diagrams
  - Location diagrams
  - Network diagrams
  - Deployment diagrams

# Summary (Cont'd)

- Describing the Environment
  - External systems
  - Technology architecture
  - Key questions requiring answers
- Designing Application Components
  - Application component boundaries
  - Grouping functions into components
  - System of record – who owns the data