Murdoch
UNIVERSITY

Designing the system architecture

Topic 7

ICT284 Systems Analysis and Design

About this topic

System design proceeds from a high level, architectural design, to the detailed specifications for software and databases that enable the solution to be built. In this topic we'll begin to look in more detail at the activities involved in systems design, starting with defining the system architecture and components, and designing the database.

Murdoch UNIVERSITY

Unit learning outcomes addressed in this topic

1. Explain how information systems are used within organisations to fulfil organisational needs
2. **Describe the phases and activities typically involved in the systems development life cycle**
3. Describe the professional roles, skills and ethical issues involved in systems analysis and design work
4. Use a variety of techniques for analysing and defining business problems and opportunities and determining system requirements
5. Model system requirements using UML, including use case diagrams and descriptions, activity diagrams and domain model class diagrams
6. **Explain the activities involved in systems design, including designing the system environment, application components, user interfaces, database and software**
7. **Represent early system design using UML, including sequence diagrams, architectural diagrams and design class diagrams**
8. Describe tools and techniques for planning, managing and evaluating systems development projects
9. Describe the key features of several different systems development methodologies
10. Present systems analysis and design documentation in an appropriate, consistent and professional manner

Murdoch UNIVERSITY

Topic learning outcomes – 1

After completing this topic you should be able to:

- Explain how designing the system architecture involves decisions from a high level of abstraction to a low level
- Interpret location, network and deployment diagrams used to describe the system architecture
- Describe the three-layer architecture and explain why it is important in system design
- Describe the activities involved in the systems design activity ‘describe the system environment’
- Explain how application components can be defined and integrated

Murdoch UNIVERSITY

Topic learning outcomes - 2

After completing this topic you should be able to:

- Describe in general terms what is involved in designing the database and incorporating it into the system design
- Differentiate between the responsibilities of the data administrator and database administrator
- Discuss the different options for distributing a database
- Briefly explain the importance of protecting the database and some methods used

Murdoch UNIVERSITY

Resources for this topic

READING

- Satzinger, Jackson & Burd, Chapter 7

  We will cover much of this only briefly. The main sections are:

  'Architectural Concepts', p195-200

  'Describing the Environment', p203-207 (skim) and

  'Designing Application Components', p208-213.

- Satzinger, Jackson & Burd, Chapter 9

  The main topic we will cover from this chapter is section 'Distributed Database Architectures', p279-284.

  Skim sections 'Protecting the Database' and 'Databases and Database Management Systems' p258-262 for background.

  Omit section 'Relational Databases' entirely.

Murdoch UNIVERSITY

Resources for this topic

Except where otherwise referenced, all images in these slides are from those provided with the textbook: Satzinger, J., Jackson, R. and Burd, S. (2016) *Systems Analysis and Design in a Changing World*, 7th edition, Course Technology, Cengage Learning: Boston. ISBN-13 9781305117204

Murdoch UNIVERSITY

Tutorial 7 –

From analysis to design

In the first of several tutorials covering system design, we'll review the main activities and concepts involved in systems design, and apply them to the Conference Coordinator Information System project.

As one of these decisions, we'll examine the 'Buy vs Build' question. As much of the functionality of the CCIS looks fairly standard, buying rather than building the system from scratch has some attractive aspects. To make an informed decision, however, we need to carefully evaluate what the packaged options can offer, and compare this with the requirements we have specified for the new system.

.

Murdoch UNIVERSITY

Topic outline

- Introduction
- Architectural patterns
- Some architectural concepts
- Design activity: describe the system environment
- Design activity: design the application components
- Design activity: design the database

Murdoch UNIVERSITY

Introduction

Murdoch UNIVERSITY

Architectures

Murdoch UNIVERSITY

- *Architecture* is the way the IT in a business is organised and put together
- Can be defined from the highest level **enterprise architecture**, which focuses on the defining and supporting the business goals of the organisation
- To the lowest level of **software architecture**, specifying how the software components are defined and interact
- In this topic we will focus on the intermediate level of **systems** architecture

System architecture

Two interdependent aspects:

- **Technology** architecture
  - Computer hardware, network computers and hardware, and system software
  - Defines the infrastructure that supports the application software
- **Application** architecture
  - The software applications
  - Components are distributed across the various devices and connected via networks and protocols

Example: Amazon.com

Murdoch UNIVERSITY

The following slides use Amazon.com as an illustration of what is involved in a large, distributed web-based system:

1. Computing devices
2. Their connection across networks
3. The software components that are used, and how they are distributed over the devices
4. The protocols that are required to ensure coordination and accurate, efficient data exchange among components

Example: Amazon.com Simplified architecture

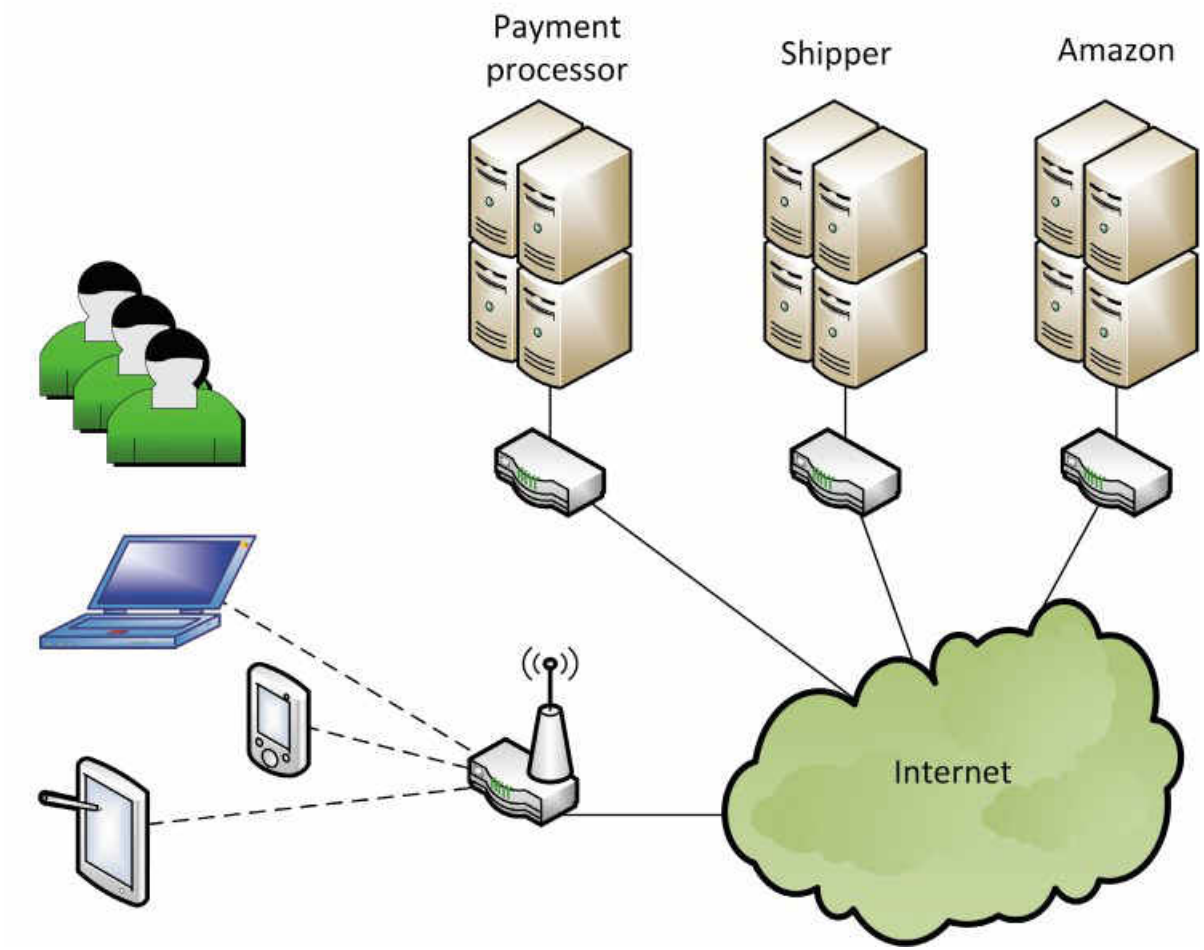


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Example: Amazon.com Software

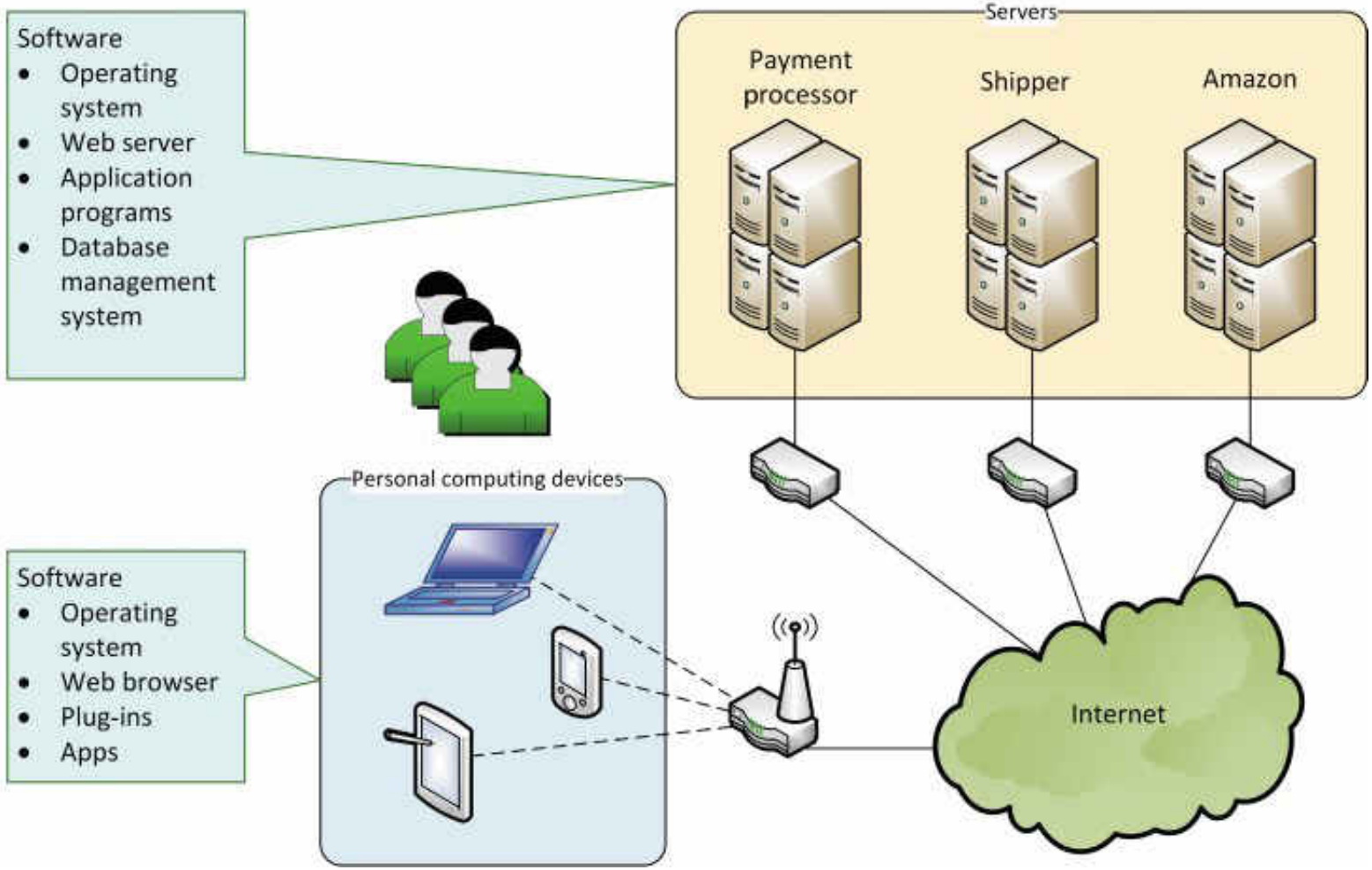


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Example: Amazon.com Software and protocols

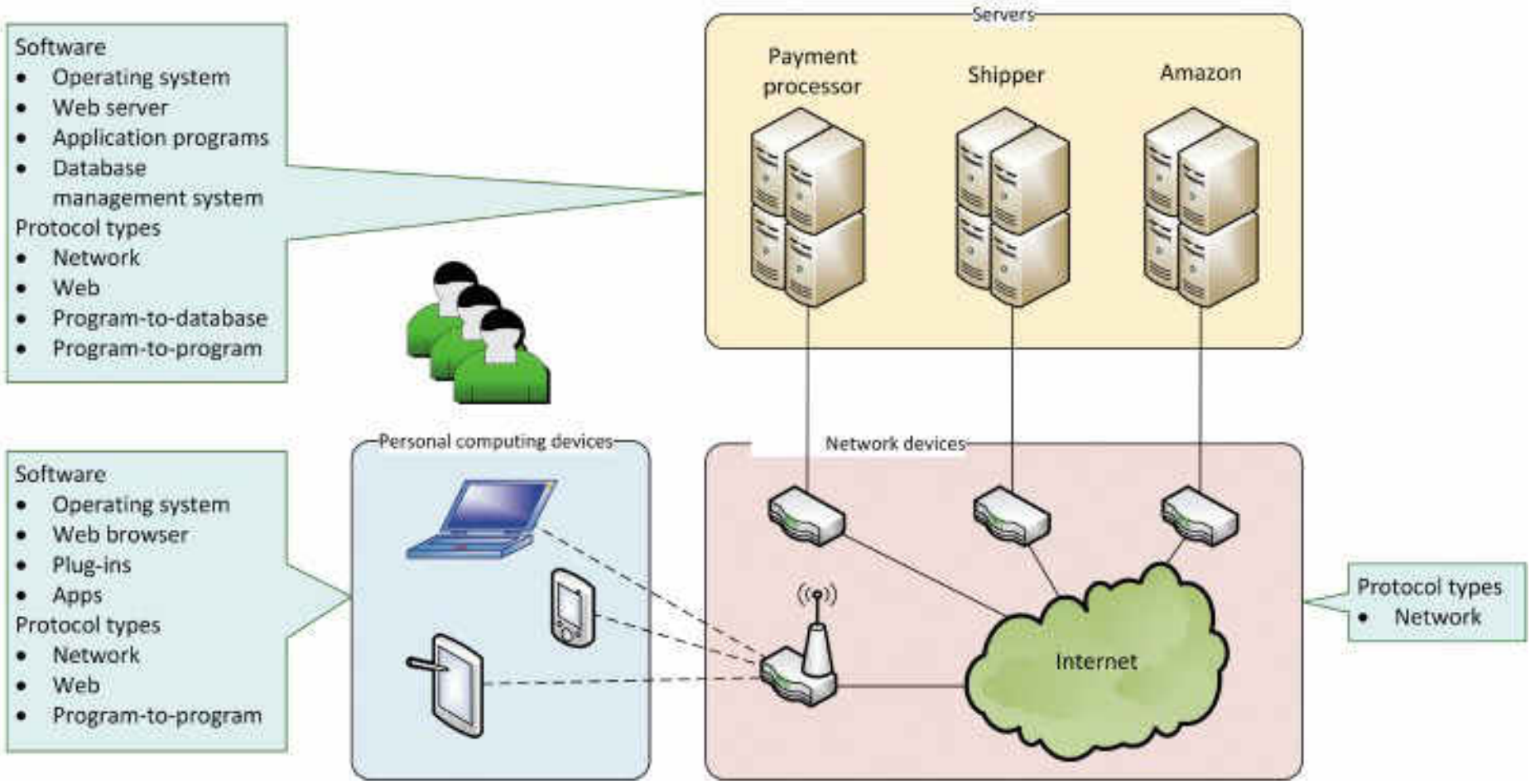


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

… So how do we decide what we need and what goes where?

Murdoch UNIVERSITY

- As with most problems, we can consider from the top down – from the highest level of abstraction (which can also be readily communicated to users) to the lowest level of coding
- The ideas of predesigned and reusable **architectural patterns** are relevant here

A useful summary here: https://msdn.microsoft.com/en-us/library/ee658117.aspx

Architectural styles or patterns

Murdoch UNIVERSITY

- These are the highest level of abstraction in software design
- The architectural *style* (now usually called *pattern*) describes in broad terms how the parts of the system will be organised and interact

  event-driven, client-server, data-centric, etc

- *"an architectural style determines the vocabulary of components and connectors that can be used in instances of that style, together with a set of constraints on how they can be combined"**

* David Garlan and Mary Shaw, January 1994, CMU-CS-94-166, see "*An Introduction to Software Architecture*" at http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf]

Architectural patterns

Murdoch UNIVERSITY

Each pattern has properties that define it:

- A vocabulary of **design elements** – what the components and connectors are
- A set of **configuration rules** – how these components can be put together
- A **semantic representation** – a meaning that constrains what it can and can't do
- Built-in analysis within the pattern

Benefits:

- Help **understand and communicate** the system-level organisation
- Design level and code level **reuse**
- Help provide **interoperability** of system components

From Verma, V. (2009). Software architecture: a case based approach. Pearson. pp23.

Architectural patterns

- There are a number of different patterns, each suited to solving a different, recurring, type of problem
- For example in **pipes and filters** the output of one processing element (filter) becomes the input (through the pipe) to the next

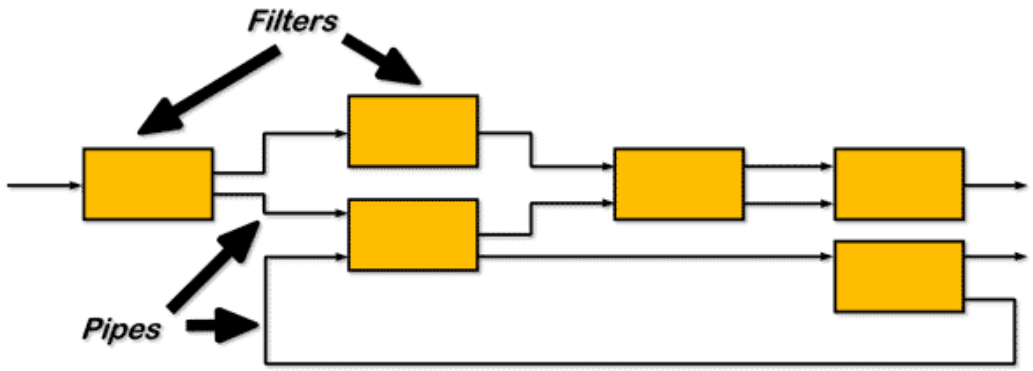
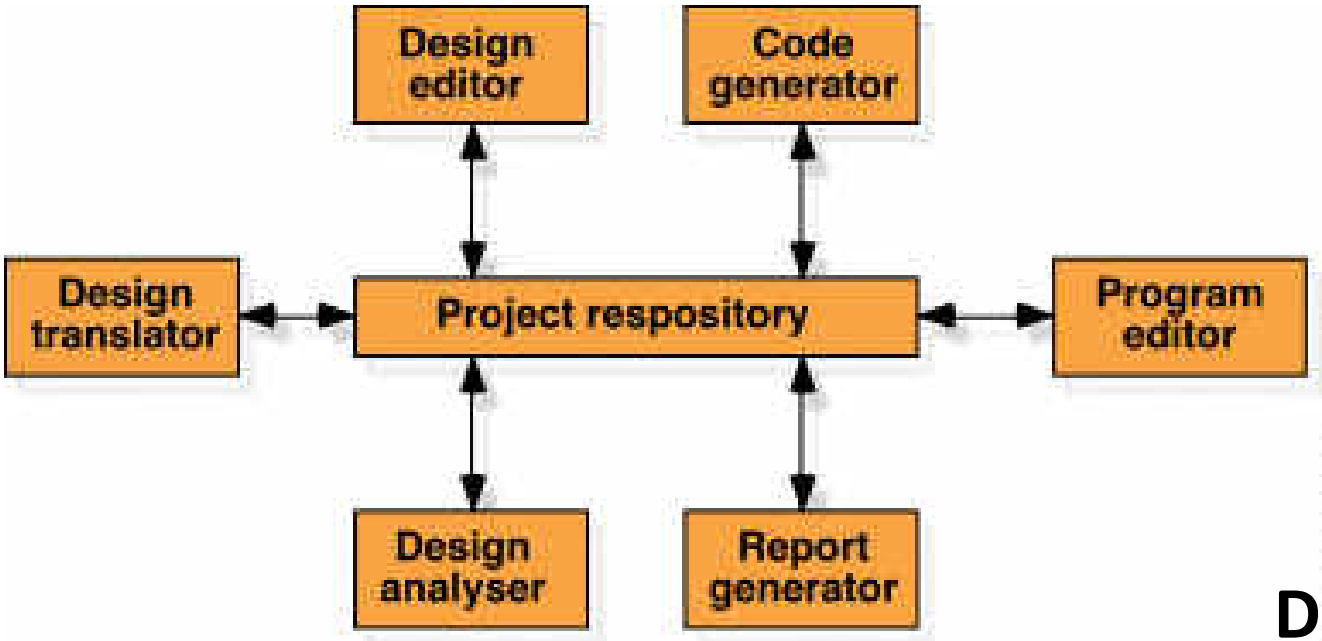


Diagram by J. Armarego

Architectural patterns - examples

Murdoch UNIVERSITY

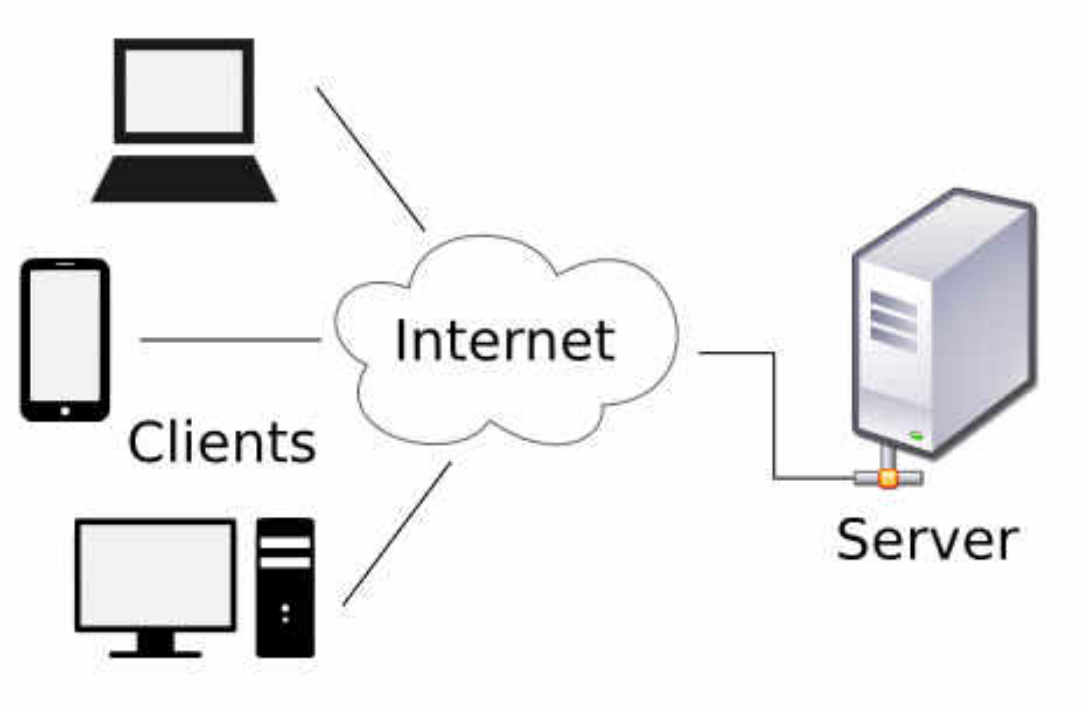


Data-centric

After: Sommerville

Diagram by J. Armarego

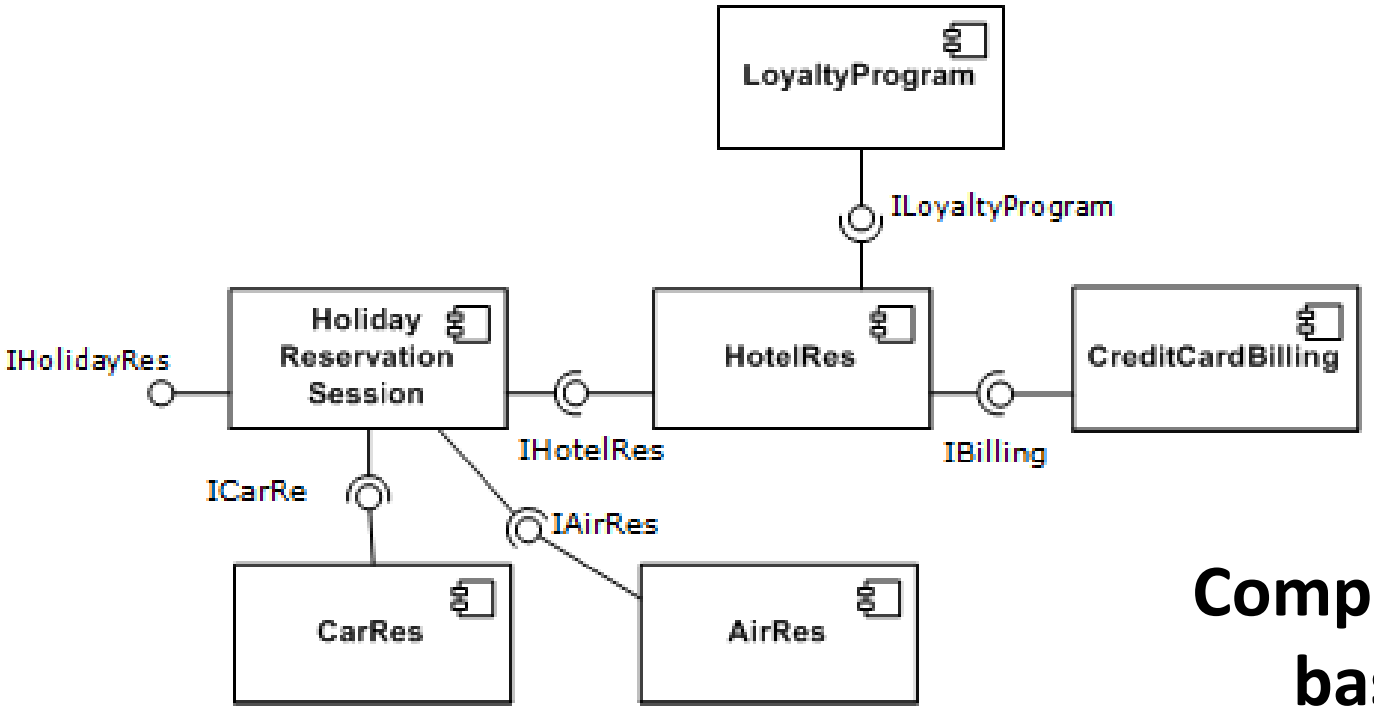
Architectural patterns - examples



Client-server

Image from Wikipedia

Architectural patterns - examples

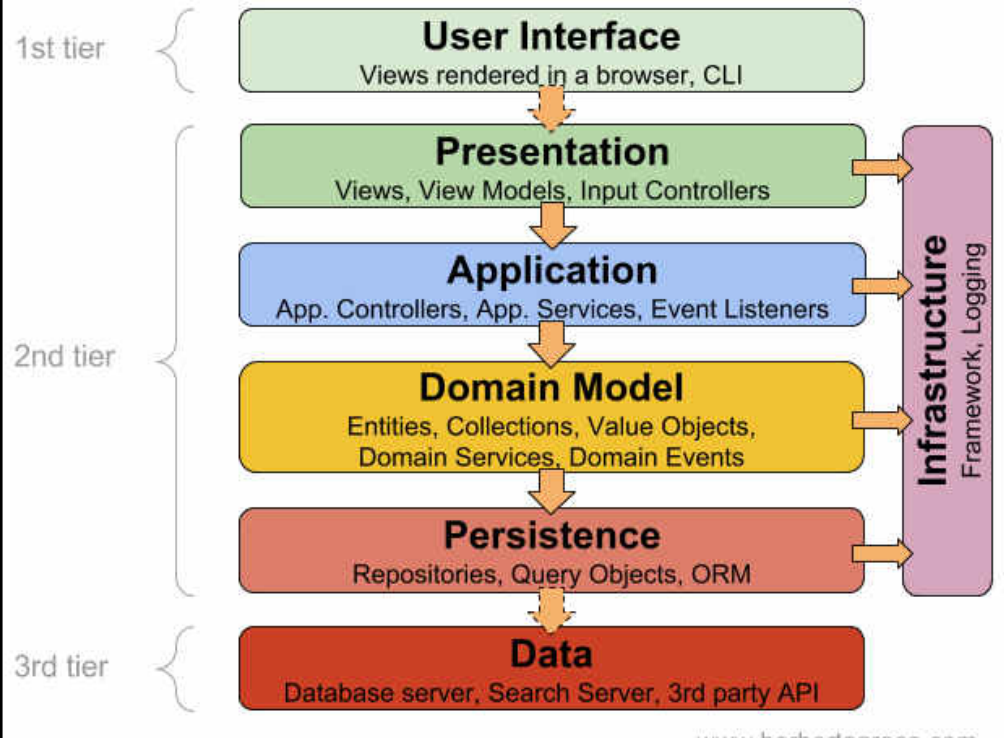


Component-based

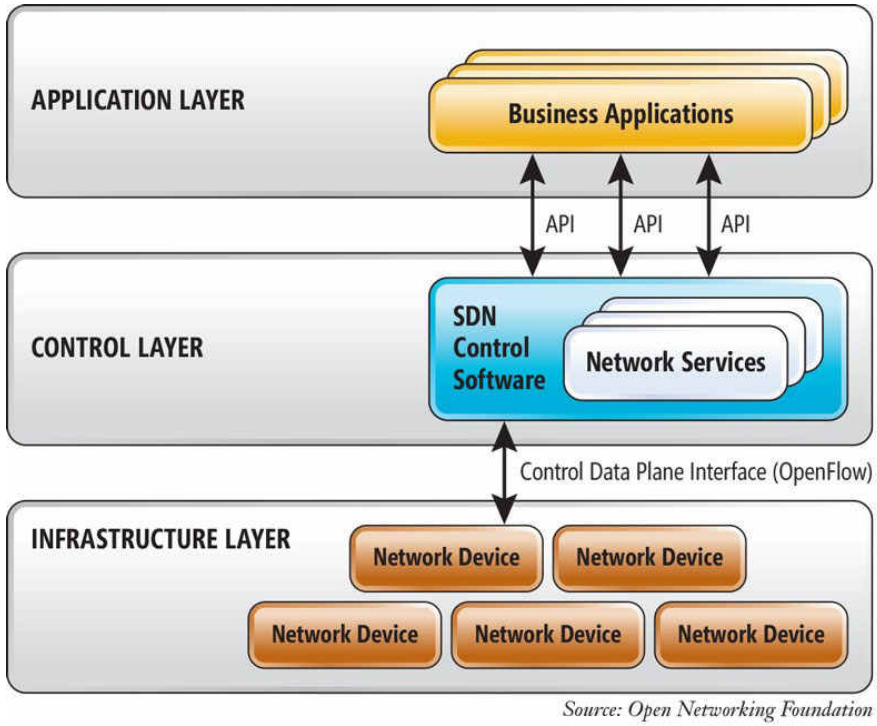
Image by Cmendes at English Wikipedia

Architectural patterns - examples

Murdoch UNIVERSITY



Layered



Images from https://herbertograca.com/2017/08/03/layered-architecture/

Catalogue of architectural styles/patterns

Murdoch UNIVERSITY

Structure
Component-based
Monolithic application
Layered
Pipes and filters

Shared memory
Database-centric
Blackboard
Rule-based

Messaging
Event-driven aka implicit invocation
Publish-subscribe
Asynchronous messaging

Adaptive systems
Plug-ins
Microkernel
Reflection
Domain specific languages

Distributed systems
Client-server (2-tier, 3-tier, n-tier)
Shared nothing architecture
Space-based architecture
Object request broker
Peer-to-peer
Representational state transfer (REST)
Service-oriented
Cloud computing patterns

List from Wikipedia

Combining architectural patterns

Murdoch UNIVERSITY

- "The architecture of a software system is almost never limited to a single architectural style, but is often a combination of architectural styles that make up the complete system.
- "For example, you might have a SOA design composed of services developed using a layered architecture approach and an object-oriented architecture style.
- "If you are building a desktop application, you may have a client that sends requests to a program on the server. In this case, you might deploy the client and server using the client/server architecture style, and use the component-based architecture style to decompose the design further into independent components that expose the appropriate communication interfaces. "

This section from https://msdn.microsoft.com/en-us/library/ee658117.aspx

Some architectural concepts

Murdoch UNIVERSITY

Web-based applications

- Use a web browser as user interface
- Are accessed via a URL
- Server-side software is called from a web server
- Use of web standards to communicate between browser and server

Very common - easy to access and use, BUT target for security breaches as we standards are open and widely known

Software as a Service (SaaS)

Murdoch UNIVERSITY

- Similar concept to a utility – software is accessed and used without being installed locally
- -- the organisation only pays for the service and doesn’t have to install or maintain anything
- Little or no software is installed on the user’s device
- User data is isolated and stored on common servers

Web Services

Murdoch UNIVERSITY

- Software function that is executed using Web standards:
  - Access via a URL
  - Inputs sent via the URL
  - Executes remotely
  - Data returned within a Web page
- Enables software functions developed by one organisations to be embedded within another organisation's systems, like a subroutine
  - e.g. calculate shipping charges, process credit card
- Designers must decide what functionality can be incorporated as Web services, and if any of their own software should be made available as Web services to other systems

Interoperability

Murdoch UNIVERSITY

- The ability of a component of a system to interact with other components or systems
- Applies to all components whether built, purchased, or used as a service

- -- vital to have an understanding of how components will (or won't) play together

Distributed architectures

- Although systems can be designed for a standalone, single machine…
- … most systems will be *distributed* across many machines and locations, sometimes across the world
- Distribution can also improve performance and reliability
- There are various ways of organising software to access distributed data and resources, e.g.
  - Client-server architecture
  - Three-layer architecture

Distributed architectures: client-server

- Software design and deployment method that divides software into component 'server' (managing resources) and 'client' (using those resources)
- Client requests resource from server, server returns it
- Server software can be on single machine or distributed across multiple (e.g. server farms)

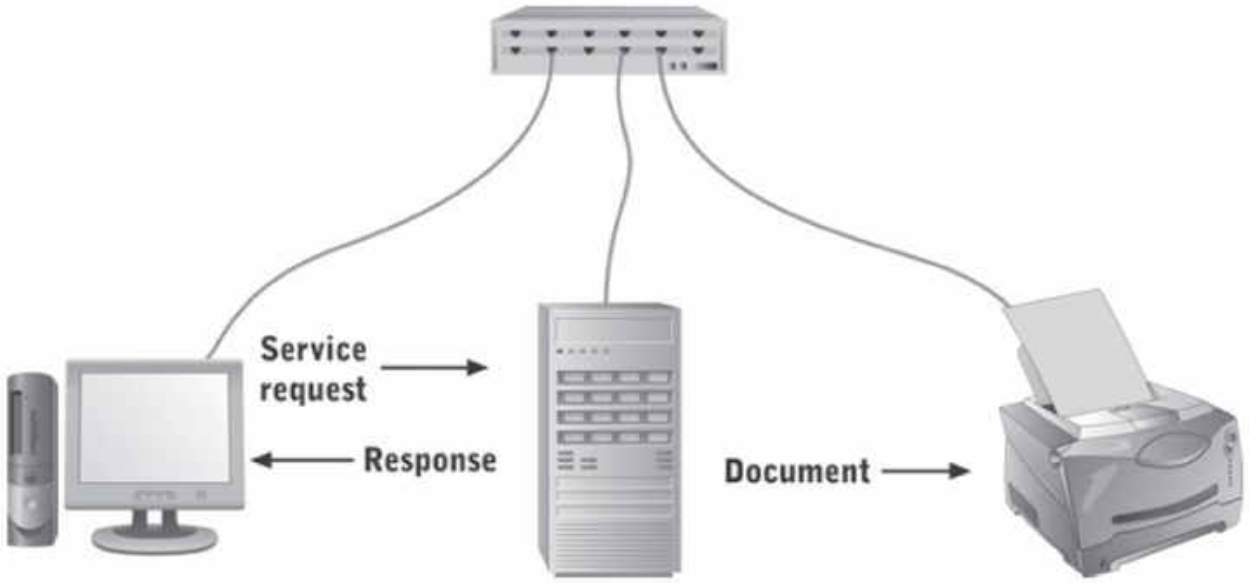


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Distributed architectures: Three–layer architecture

A client/server architecture that divides an application into view layer, business logic layer, and data layer

View layer

- contains the user interface

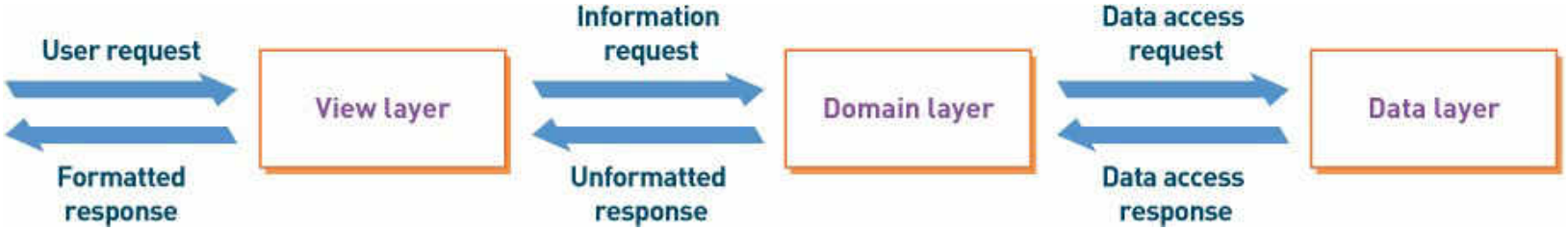
Business logic layer or domain layer

- contains the programs that implement the business rules and processes

Data layer

- interacts with the data

Three Layer Architecture



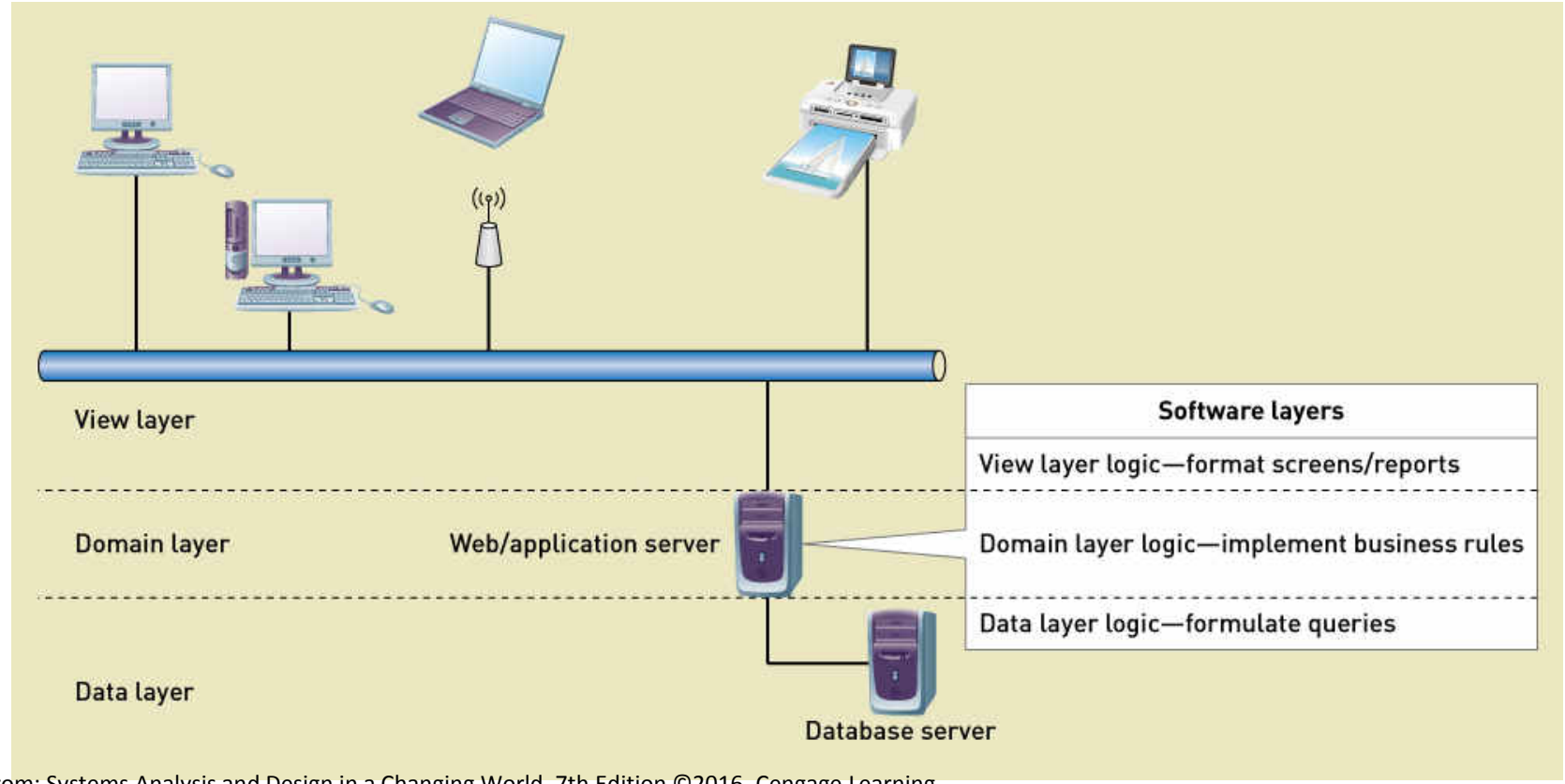


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Distributed architectures: Three–layer architecture

- A client/server architecture that divides an application into view layer, business logic layer, and data layer
- Very commonly used architecture
- Layers can execute on the same computer, separate computers or split across computers
- Flexible – layers can be relatively independent of one another – the only dependences are through the requests and responses
- - can re-deploy resources in response to changing conditions

Internet deployment: Advantages and potential problems

Murdoch UNIVERSITY

Advantages

- Accessibility of web based applications
- Low-cost communication through high capacity internet backbone
- Widely implemented standards

Potential problems

- Security
- Throughput
- Changing standards

Design activities - reminder

Murdoch UNIVERSITY

Design activities - reminder

Design activities
Describe the environment. Design the application components. Design user interface. Design the database. Design the software classes and methods.

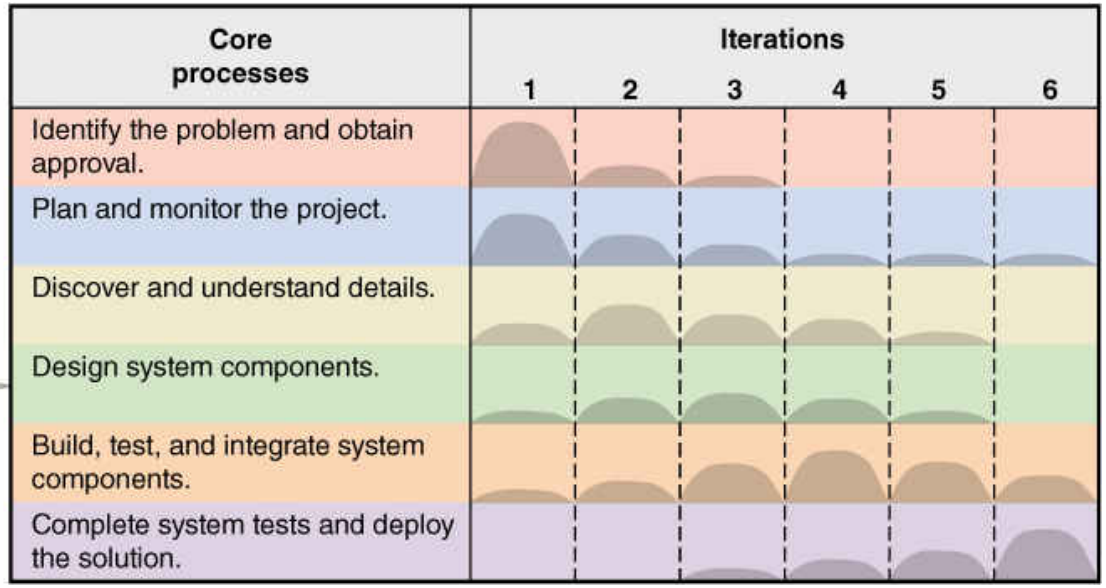


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Key design questions for each activity

Murdoch UNIVERSITY

Design activity	Key question
Describe the environment	How will this system interact with other systems and with the organization's existing technologies?
Design the application components	What are the key parts of the information system and how will they interact when the system is deployed?
Design the user interface	How will users interact with the information system?
Design the database	How will data be captured, structured, and stored for later use by the information system?
Design the software classes and methods	What internal structure for each application component will ensure efficient construction, rapid deployment, and reliable operation?

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Design activity: Describe the system environment

Murdoch UNIVERSITY

Describe the environment

How will this system interact with other systems and with the organisation's existing technologies?

- A system will rarely be completely new – it will have to fit with what already exists in the organisation, so this can limit design choices
- Two key elements of environment:
  - External systems
  - Technology architecture

Describe the system environment: questions to ask

1. What are the key features of the existing or proposed technology environment that will support or constrain the system?
2. What external systems and databases will the new system interact with?
3. What devices will be used for automated inputs and outputs?
4. What user interface technology will be used?

- Once these questions are answered, the systems designer knows what s/he is working with and can commence designing and documenting in architectural diagrams

Architectural diagrams used in systems design

Murdoch UNIVERSITY

- **Location diagrams** show where all the various system components (hardware, buildings, users) are located geographically
- **Network diagrams** show how locations and hardware components are interconnected with network devices and wiring
    - Different aspects and levels of detail can be emphasised
- **Deployment diagrams** show how software components are distributed across hardware and system software

Location diagrams

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Network diagrams

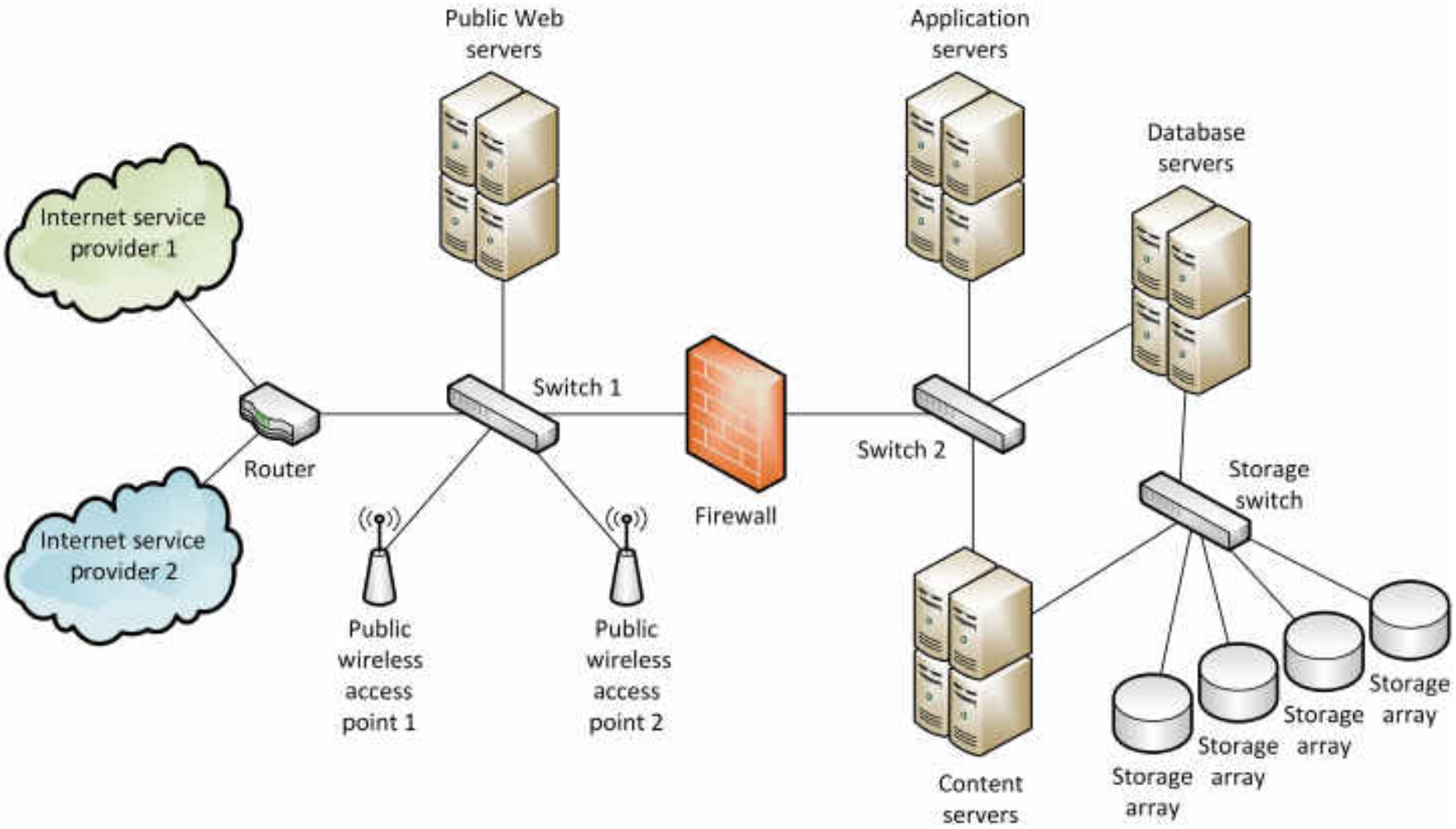
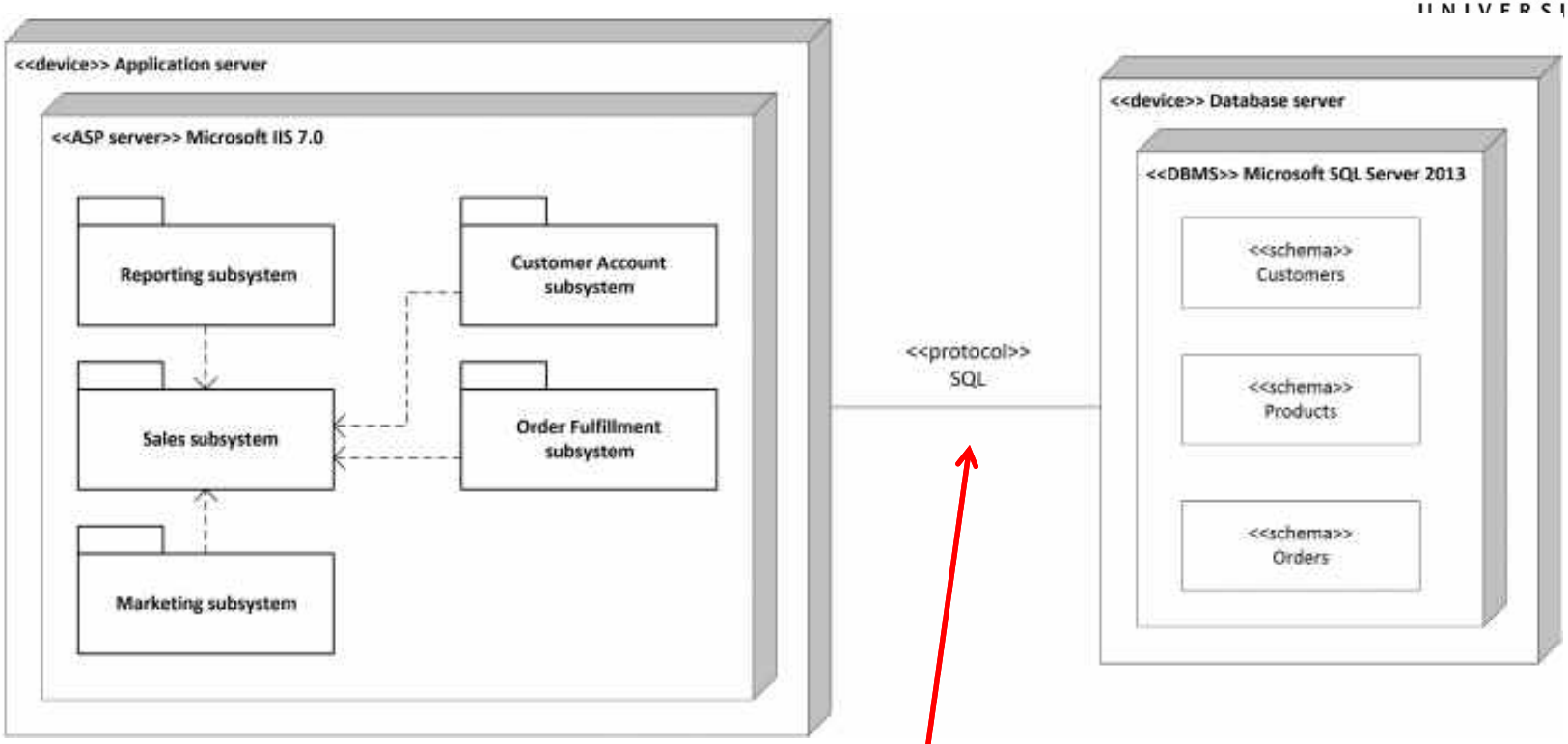


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Deployment diagrams



Application server hosting **Microsoft IIS**, which hosts several of the subsystems of the RMO CSMS

Database server hosting **Microsoft SQL Server,** which hosts the customer, products, and orders data

Communication via **SQL**

Example: RMO CSMS environment

Murdoch UNIVERSITY

- RMO currently has a data centre in Park City, with internal networks in every office, warehouse, manufacturing plant and retail store
- The distribution centres and manufacturing plants are connected via a VPN to the data centre, as are the retail stores
- There are separate servers for each primary system and a LAN connects all servers and users in the data centre and corporate offices

Example: Existing RMO CSMS environment

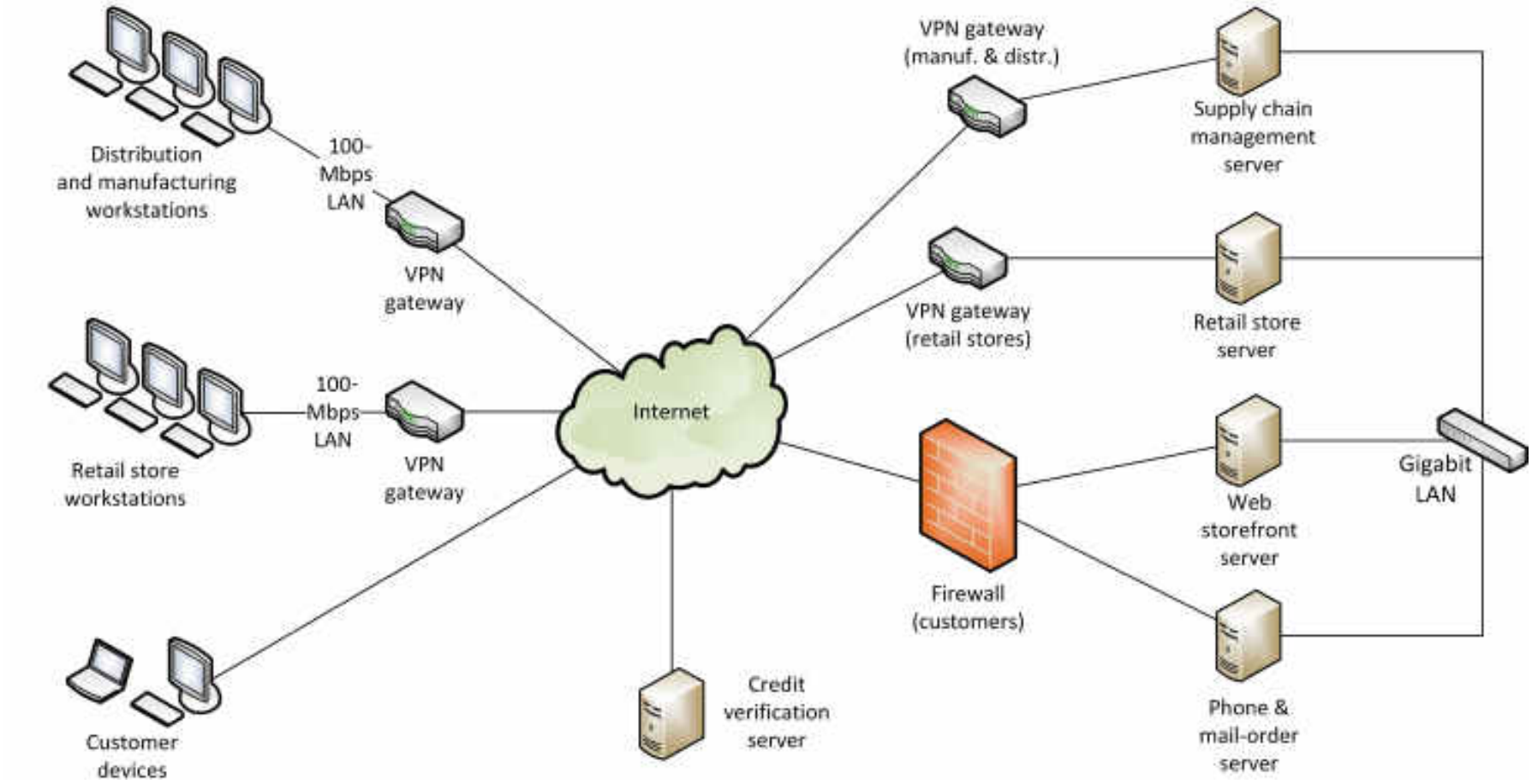


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Example: Proposed RMO CSMS environment

Murdoch UNIVERSITY

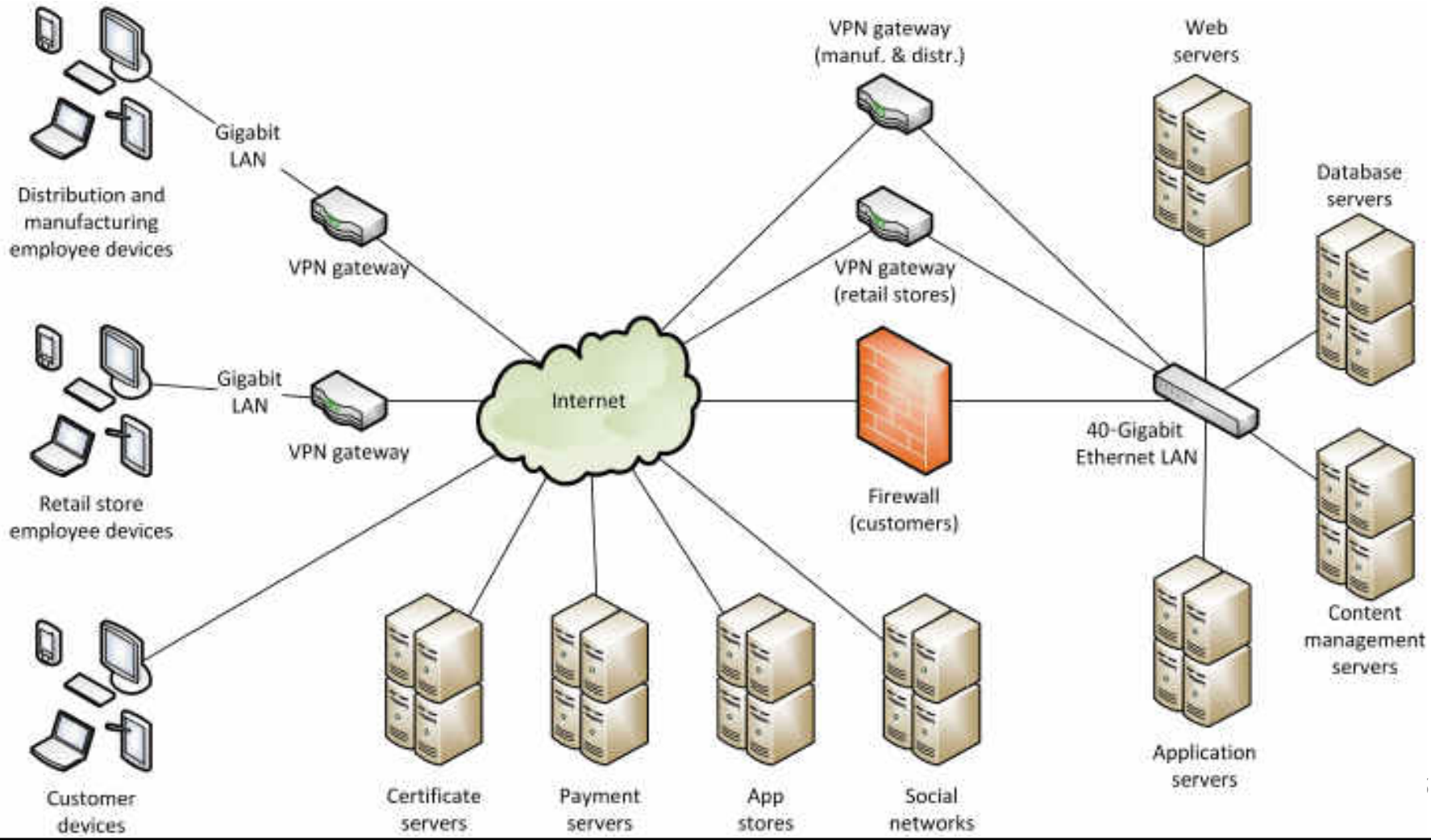
Updates for the new system would now involve:

- Support for mobile devices and apps
- Updated support for Web technologies
- Social networking
- And security considerations in all of the above

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Example: Proposed RMO CSMS environment

Murdoch UNIVERSITY



Example: Updated architecture diagram shows:

Murdoch UNIVERSITY

- Additional user interface devices
- Upgraded network speeds
- More servers, and servers grouped by service type (web, database, content management, application)
  - External hosting could be an alternative
- External service providers (payment, app stores, social networks)

Design activity: Define the application components

Murdoch UNIVERSITY

Design the application components

What are the key parts of the information system and how will they interact when the system is deployed?

- Application component is a well defined unit of software that performs some function(s)
- Need to decide how the functionality is to be packaged into components, and how they will interact, e.g.
  - Subsystems
  - Database
  - Separation into layers
- Application component design is precursor to defining the software classes and methods

Design the application components: boundaries

Murdoch UNIVERSITY

- We need to design the system so that the application components form logical units (perform one or more specific tasks)
- Can be top-down or bottom-up, but need to look for similarities among system functions to guide grouping:
  - **Actors** – group software for use cases that interact with the same actors
  - **Shared data** – group use cases that interact with the same domain classes
  - **Events** – group use cases that are triggered by the same event

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Example: RMO CSMS application components

Murdoch UNIVERSITY

- Grouping by customer actor – partial

Use case	User/actor	Domain class(es)	Event(s)	Group
Create phone sale	Phone sales representative	ProductItem, InventoryItem, SaleItem, Sale, SaleTrans	Customer request while shopping by phone	A
Create store sale	Store sales representative	ProductItem, InventoryItem, SaleItem, Sale, SaleTrans	Customer request while shopping in store	B
Create/update customer account	Customer, phone or store sales representative	Customer, Account, Address	Customer request or sale to a new customer	C
Look up order status	Shipping, customer, management, phone or store sales representative	ProductItem, InventoryItem, SaleItem, Sale, SaleTrans, Shipment, ReturnItem	Customer, representative, shipping, or management request	
Track shipment	Shipping, customer, management, phone or store sales representative	Shipment, Shipper, SaleItem	Customer, representative, shipping, or management request	
Create item return	Customer, phone or store sales representative	SaleItem, ReturnItem	Customer requests return	
Search for item	Customer, phone or store sales representative	ProductItem	Customer request while shopping online, by phone, or in store	
View product comments and ratings	Customer, phone or store sales representative	ProductItem, ProductComment	Customer request while shopping online, by phone, or in store	
View accessory combinations	Customer, phone or store sales representative	ProductItem, AcessoryPackage	Customer request while shopping online, by phone, or in store	

(See text for complete example)

Example – CCIS use case list

Grouping by actor

Use Case Name	Actors
Submit paper	Author
Reject late paper [temporal]	Author
Reject incomplete paper	Editor
Submit camera-ready paper	Author
Withdraw paper	Author, editor
Record reviewer details	Reviewer
Assign number to paper [state]	
Allocate paper	Editor
View paper	Reviewer
Create paper review	Reviewer
Remind reviewer [temporal]	Reviewer
Set reviewer to unavailable [state]	
View reviewer comments	Editor
Record decision	Editor
Notify of decision	Editor, Author
Create/update schedule	Editor
View schedule	Editor, Chair, Author, Reviewer
Create summary of papers	Editor
View summary of papers	Editor, Chair, Author, Reviewer
Create author list	Editor
View author list	Editor, Chair, Author, Reviewer

Example: RMO CSMS application components

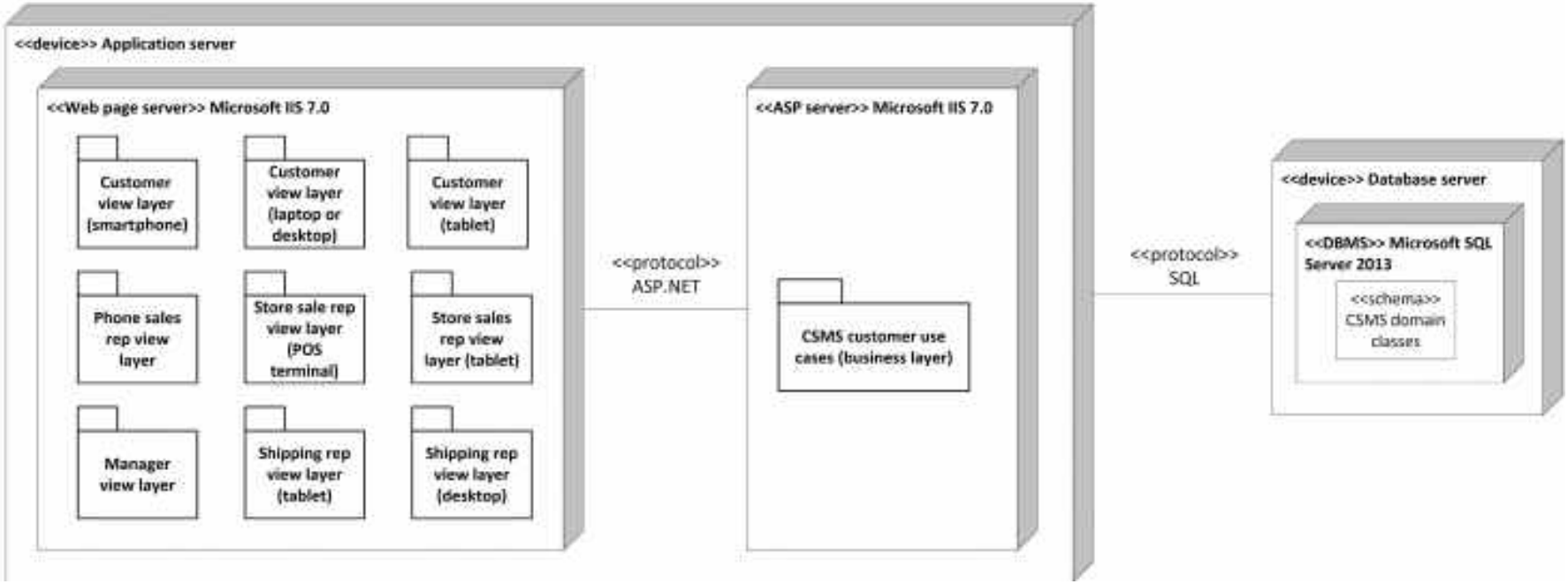
Murdoch UNIVERSITY

Once the groupings have been analysed they can be allocated to different layers of the 3-layer architecture, e.g.

- **Data layer** – contains the database representing the entire data model
- **Business (domain) layer** – here could include all of the use cases, as there are significant overlaps in actors, domain classes and events
- **View layer** – several, each customised and optimised for a different user

This can be shown in a deployment diagram (next)

Example: Deployment diagram for RMO CSMS showing distribution of application components



View layer

Business layer

Data layer

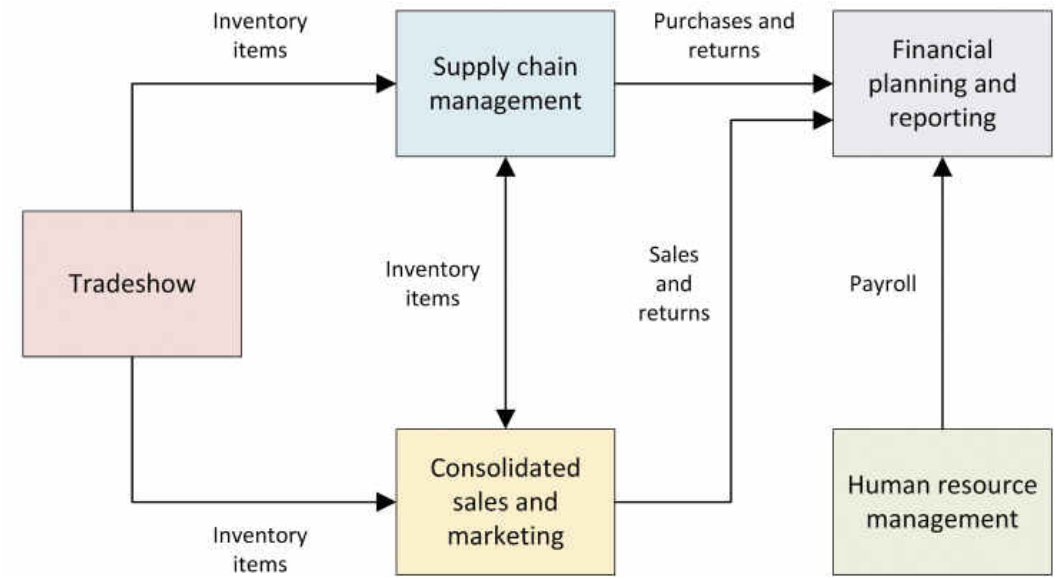
Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Design the application components: integration

Murdoch UNIVERSITY

- The new system will seldom be completely standalone and will need to integrate with existing systems
- Similarly, the new system itself may not be entirely custom built, so will need to integrate purchased or SaaS components with developed ones
- Need to define how these components will communicate, e.g.
  - Batch data export and import
  - Data import via Web services
  - Direct access via utilities

Example: RMO CSMS subsystem integration showing data flows



And synchronisation of data copies

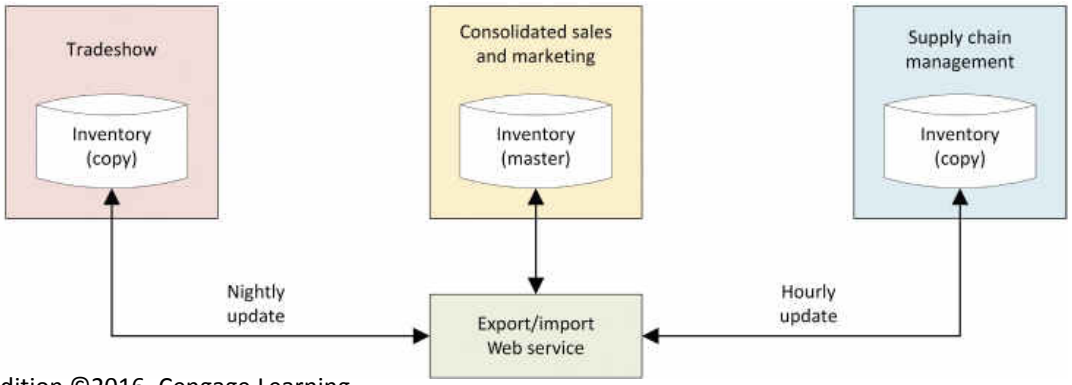


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Design activity: Design the database

Murdoch UNIVERSITY

Design the database

How will data be captured, structured an stored for later use by the information system?

- By definition, an information system requires data – usually (though not always) in a database
- Current technology frequently uses Relational Database Management Systems (RDBMS)
- Requires converting the data model (from the DMCD or ERD) to a relational database
- Database design also involves addressing many other technical issues, such as performance and security

Database and DBMS

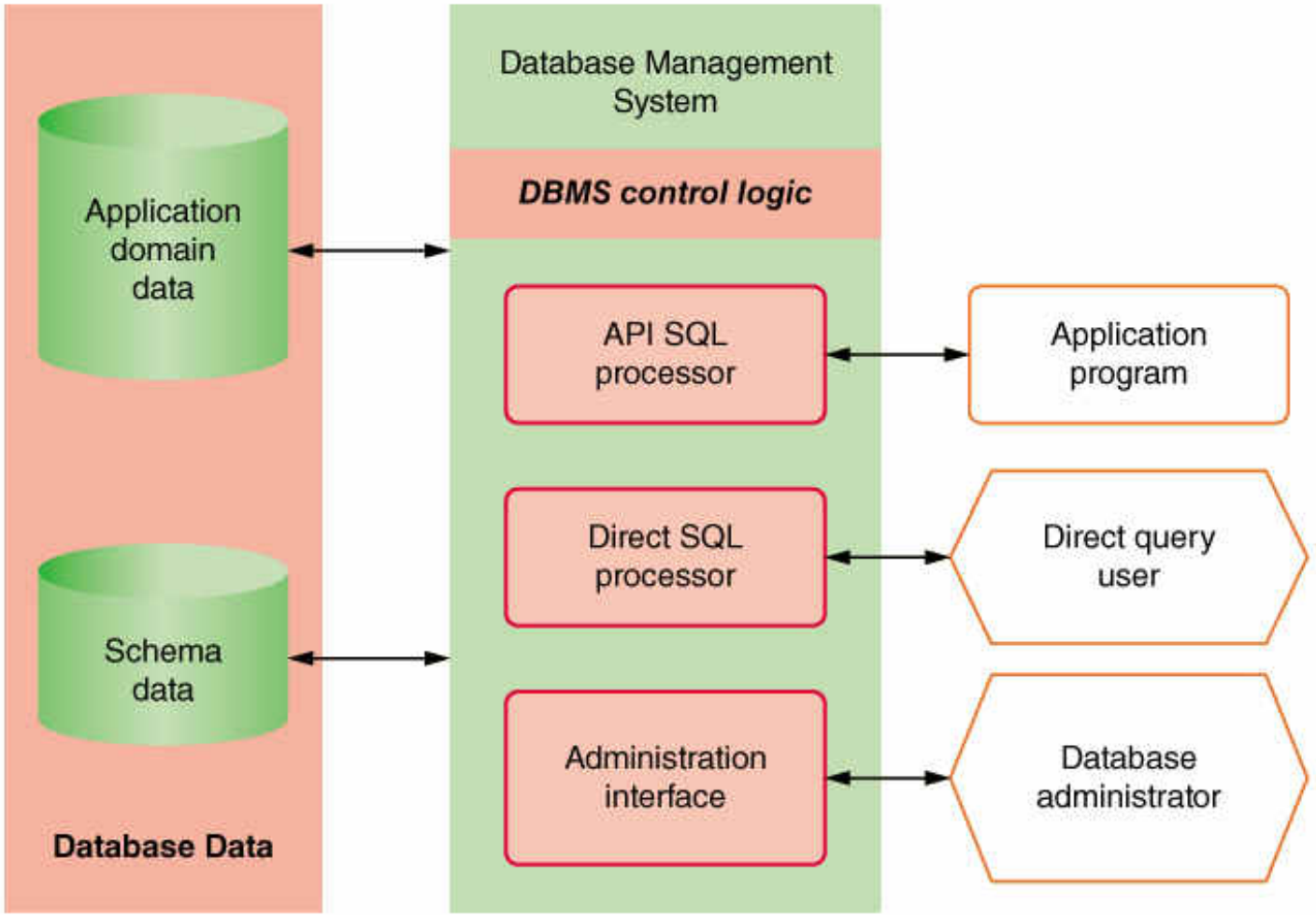


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Database and DBMS

Murdoch UNIVERSITY

- Database (DB) -- an integrated collection of stored data that is centrally managed and controlled
  - A **distributed database** is one that is physically located across different locations
- Database management system (**DBMS**) -- a system software component that manages and controls one or more databases
- **Schema** -- database component that contains descriptive information about the structure of the data stored in the physical data store
- Structured Query Language, **SQL** -- the standard query language to access and update data in a relational DBMS

Data administrator and database administrator

Permanent technical positions in the IT department responsible for aspects of the database:

- **Data administrator** – responsible for structure and integrity of the *data* itself:
  - Data standards
  - Data ownership and access
  - Data quality
- **Database administrator** – responsible for the correct and efficient functioning of the *DBMS*
  - Managing multiple users
  - Implementing security
  - Performance
  - Backup and recovery

Design the database

Murdoch UNIVERSITY

In iterative development database is foundational: early iterations need to focus on data and key portions of the database

Starting with the domain model class diagram (or ERD)…

- Choose database structure
  - Usually relational (RDBMS)
- Design database schema
  - Convert data model to tables
- Define constraints
  - primary keys; foreign keys; domain constraints
- Design architecture (distributed, etc) according to business requirements

Relational database structure – 'tables' of rows and columns

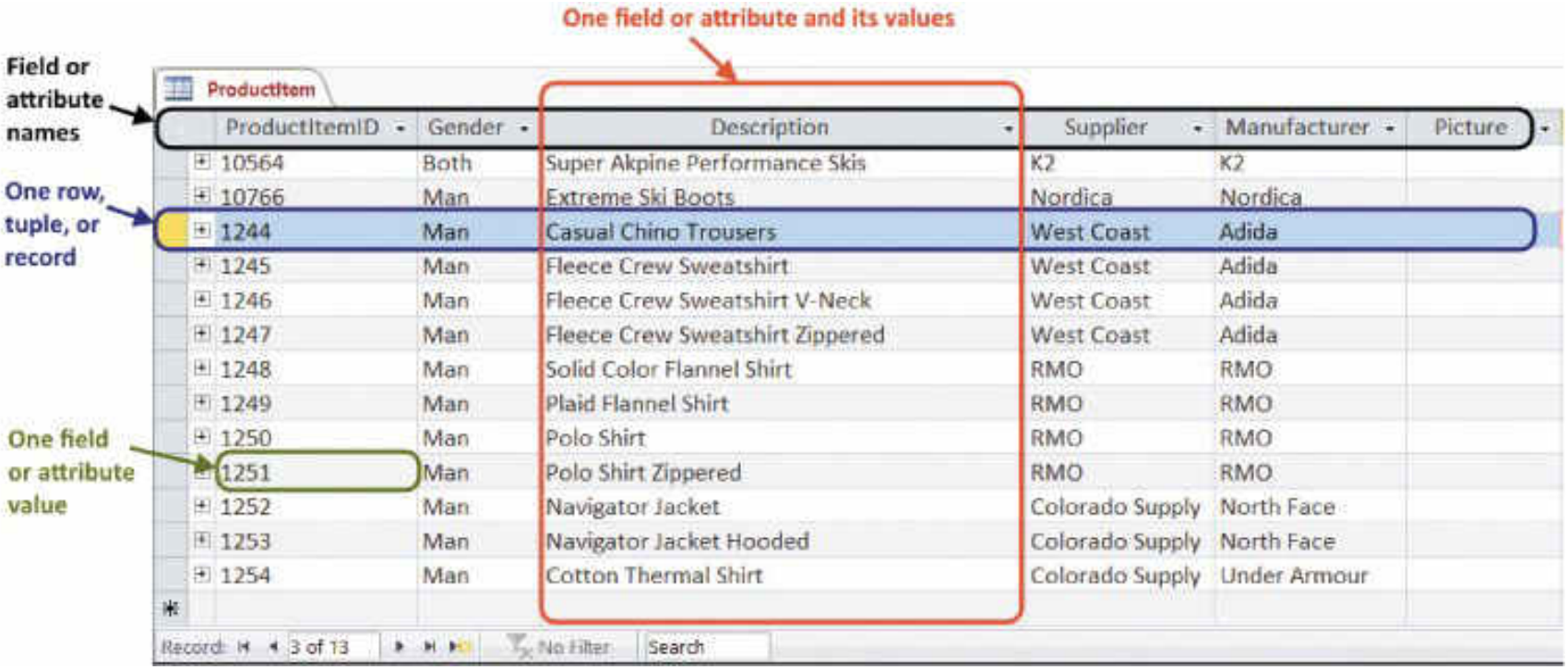
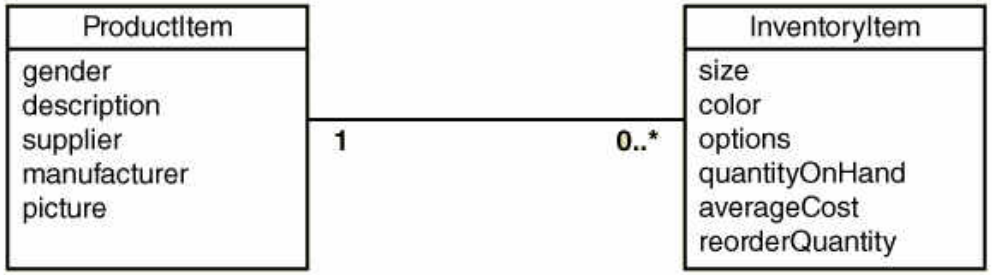


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Representing associations between classes in the database



ProductItemID, the **key** attribute in ProductItem, is included in Inventory Item as **foreign key** to make the link

ProductItem

ProductItemID	Gender	Description	Supplier	Manufacturer	Picture
10564	Both	Super Akpine Performance Skis	K2	K2	
10766	Man	Extreme Ski Boots	Nordica		
1244	Man	Casual Chino Trousers			
1245	Man	Fleece Crew Sweatshirt			
1246	Man	Fleece Crew Sweatshirt V-Neck			
1247	Man	Fleece Crew Sweatshirt Zippered			
1248	Man	Solid Color Flannel Shirt			
1249	Man	Plaid Flannel Shirt			
1250	Man	Polo Shirt			
1251	Man	Polo Shirt Zippered			

Record: 3 of 13 No Filter Search

InventoryItem

InventoryItem	ProductItem	Size	Color	Options	QuantityOnHand	Average Cost	RecorderQuantity
86779	1244	30/30	Khaki		45	$12.75	100
86780	1244	30/30	Slate		10	$12.75	100
86781	1244	30/30	LightTan		17	$12.75	100
86782	1244	30/31	Khaki		22	$12.75	100
86783	1244	30/31	Slate		6	$12.75	100
86784	1244	30/31	LightTan		31	$12.75	100
86785	1244	30/32	Khaki		120	$12.75	100
86786	1244	30/32	Slate		28	$12.75	100
86787	1244	30/32	LightTan		21	$12.75	100
86788	1244	30/33	Khaki		7	$12.75	100

Record: 1 of 12 No Filter Search

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Database architectures

- Databases and their DBMSs may be deployed in a number of ways:
  - Single desktop system
  - Shared database on a LAN
  - Large database requiring several servers to host
  - Multiple databases distributed globally
- In a large, widely spread organisation it makes sense to *distribute* parts of the database to where it is most used
  - Performance, security, reliability, local management

Distributed database architectures

- A **distributed relational database** distributes or duplicates tables to multiple database servers located in geographically important locations, according to business needs

Various possibilities:

- Vertical partitioning (by columns of the database)
- Horizontal partitioning (by rows of the database)
- Replication (of parts or all of the database)
- Combinations of these

Distributed database design

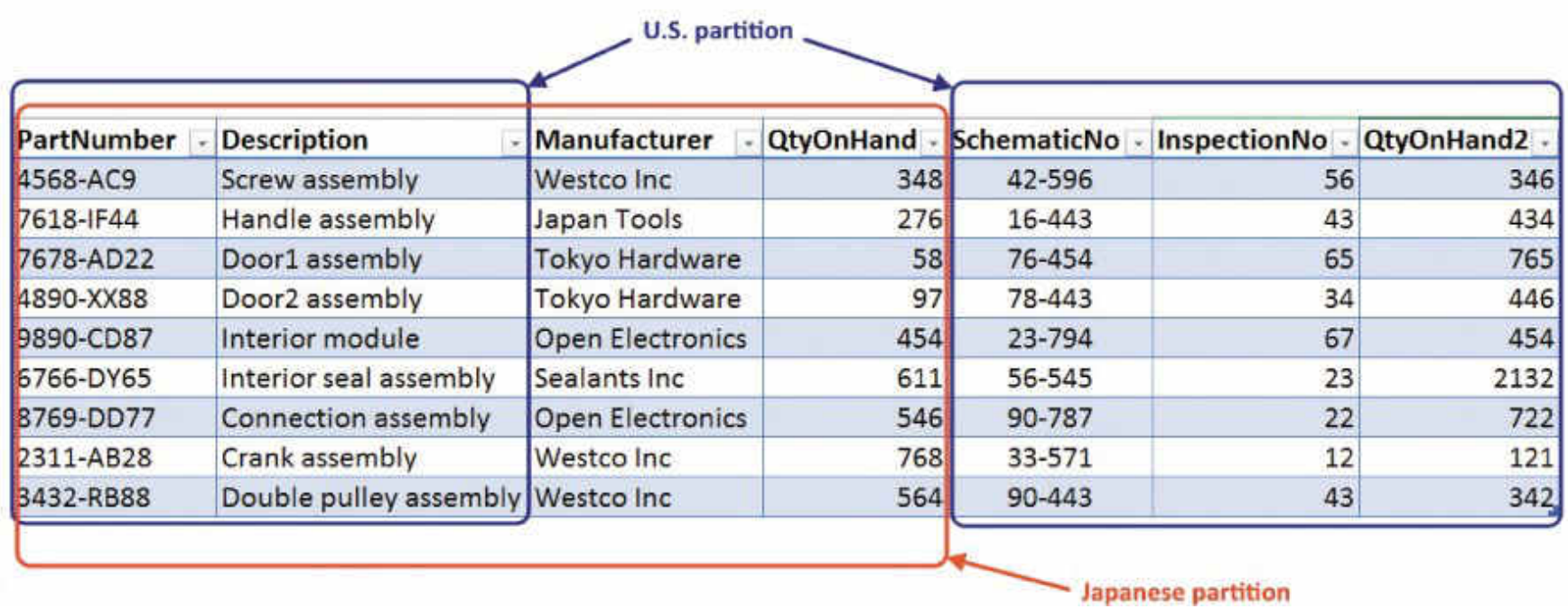
- Horizontal Partitioning – different *rows* are stored at different locations.

AcctNumb	LastName	FirstName	SSN	TypeOfAcct	Balance	DateLastActivity	
01-85562-1	Jones	Bill	878-77-9890	Checking	$ 7,908.39	5/9/2014	U.S. accounts
01-85444-2	Johnson	Harold	676-44-3433	Checking	$25,698.33	5/2/2013	
02-45443-2	Williams	Jonathon	343-44-2322	Checking	$ 3,938.77	4/4/2012	
01-34999-1	Redd	Mary	898-79-3487	Savings	$12,898.71	12/2/2013	
01-23989-2	Chun	Tun	233-59-6765	Savings	$ 8,932.67	1/8/2014	Hong Kong accounts
01-87889-4	Gang	Bao	322-48-3545	Checking	$ 568.33	3/4/2014	
01-32339-2	Jiang	Rui	550-43-5454	Savings	$35,788.23	7/8/2014	
02-39988-1	Ma	Shuo	343-98-2345	Checking	$ 1,893.55	8/23/2014	

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Distributed database design

- Vertical Partition – Different *columns* are stored at different locations



PartNumber	Description	Manufacturer	QtyOnHand	SchematicNo	InspectionNo	QtyOnHand2
4568-AC9	Screw assembly	Westco Inc	348	42-596	56	346
7618-IF44	Handle assembly	Japan Tools	276	16-443	43	434
7678-AD22	Door1 assembly	Tokyo Hardware	58	76-454	65	765
4890-XX88	Door2 assembly	Tokyo Hardware	97	78-443	34	446
9890-CD87	Interior module	Open Electronics	454	23-794	67	454
6766-DY65	Interior seal assembly	Sealants Inc	611	56-545	23	2132
8769-DD77	Connection assembly	Open Electronics	546	90-787	22	722
2311-AB28	Crank assembly	Westco Inc	768	33-571	12	121
3432-RB88	Double pulley assembly	Westco Inc	564	90-443	43	342

- Combinations of replication, horizontal, and vertical are possible

Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Image from Whitten, J.L., & Bentley, L.D, (2007), Systems Analysis and Design Methods, 7th Edition, McGraw-Hill. p495

Data Partitioning and Data Replication

Murdoch UNIVERSITY

Logical Data Store	Physical Data Stores using Partitioning	Physical Data Stores using Replication
1 CUSTOMERS	1P.# Oracle 7: REGION 1 CUSTOMERS 1P.# Oracle 7: REGION 2 CUSTOMERS	Not applicable. Branch offices do not need access to data about customers outside of their own sales region.
2 PRODUCTS	Not applicable. All branch offices need access to data for all products, regardless of sales region.	2M Oracle 8i: PRODUCTS (Master) 2R Oracle 8i: PRODUCTS (Replicated Copy)

Example: Database architecture for RMO CSMS - Replicated and partitioned database

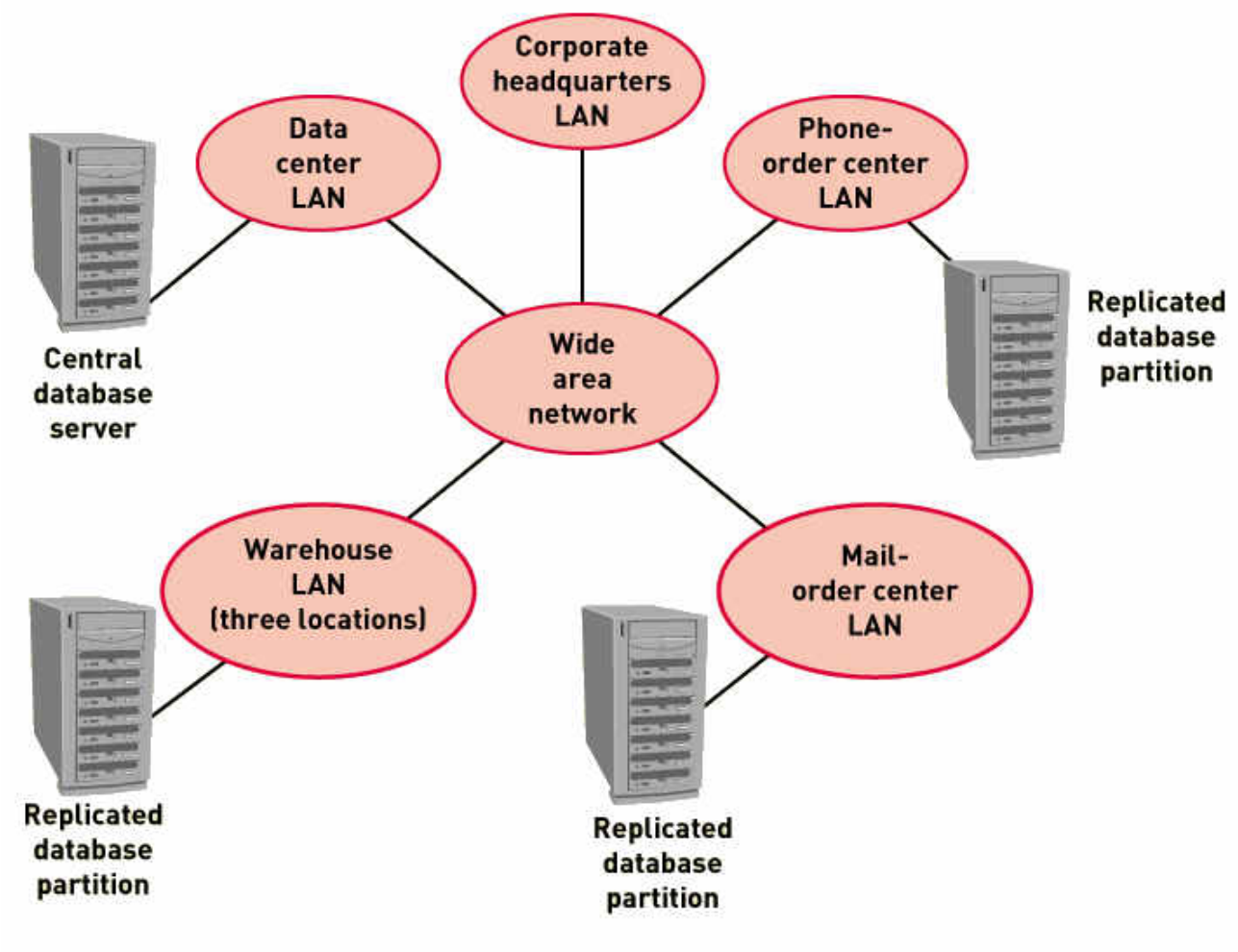


Image from: Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning

Protecting the database

Murdoch UNIVERSITY

- **Security and access** - usually via access privileges
- **Transaction Logging** – a technique to record all updates including change, date, time, user
  - Helps to prevent fraud
  - Recovery mechanism for failures
- **Concurrency and Update Controls**
  - *Transaction* – a piece of work with several steps, either all must complete or none must be accepted
  - *Locking* is used to control transaction sequencing so that it is correct even when multiple transactions occur at the same time

Topic learning outcomes revisited

After completing this topic you should be able to:

- Explain how designing the system architecture involves decisions from a high level of abstraction to a low level
- Interpret location, network and deployment diagrams used to describe the system architecture
- Describe the three-layer architecture and explain why it is important in system design
- Describe the activities involved in the systems design activity 'describe the system environment'
- Explain how application components can be defined and integrated
- Describe in general terms what is involved in designing the database and incorporating it into the system design
- Differentiate between the responsibilities of the data administrator and database administrator
- Discuss the different options for distributing a databases
- Briefly explain the importance of protecting the database and some methods used

Murdoch UNIVERSITY

What’s next?

In the next topic, we’ll continue our coverage of systems design activities, by considering the user interface.

Murdoch UNIVERSITY