Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Purposefully different, consistently excellent

Welcome to EEN1037: Web Applications Development

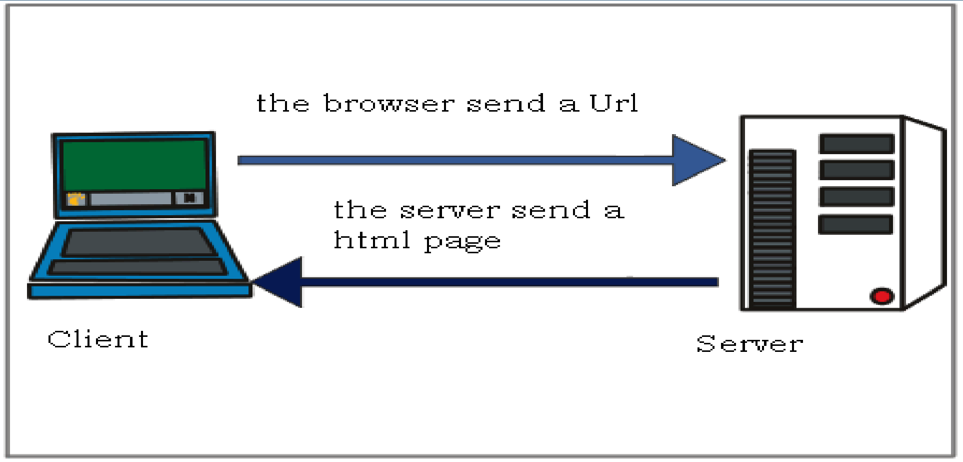
Web Applications Development

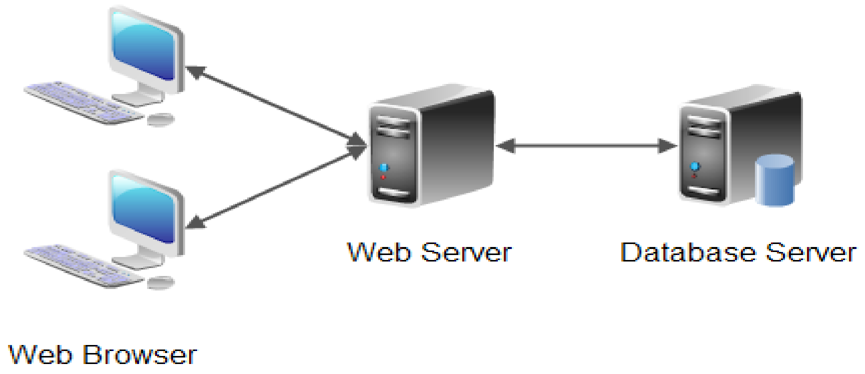
Week 1

Joseph Mullally

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

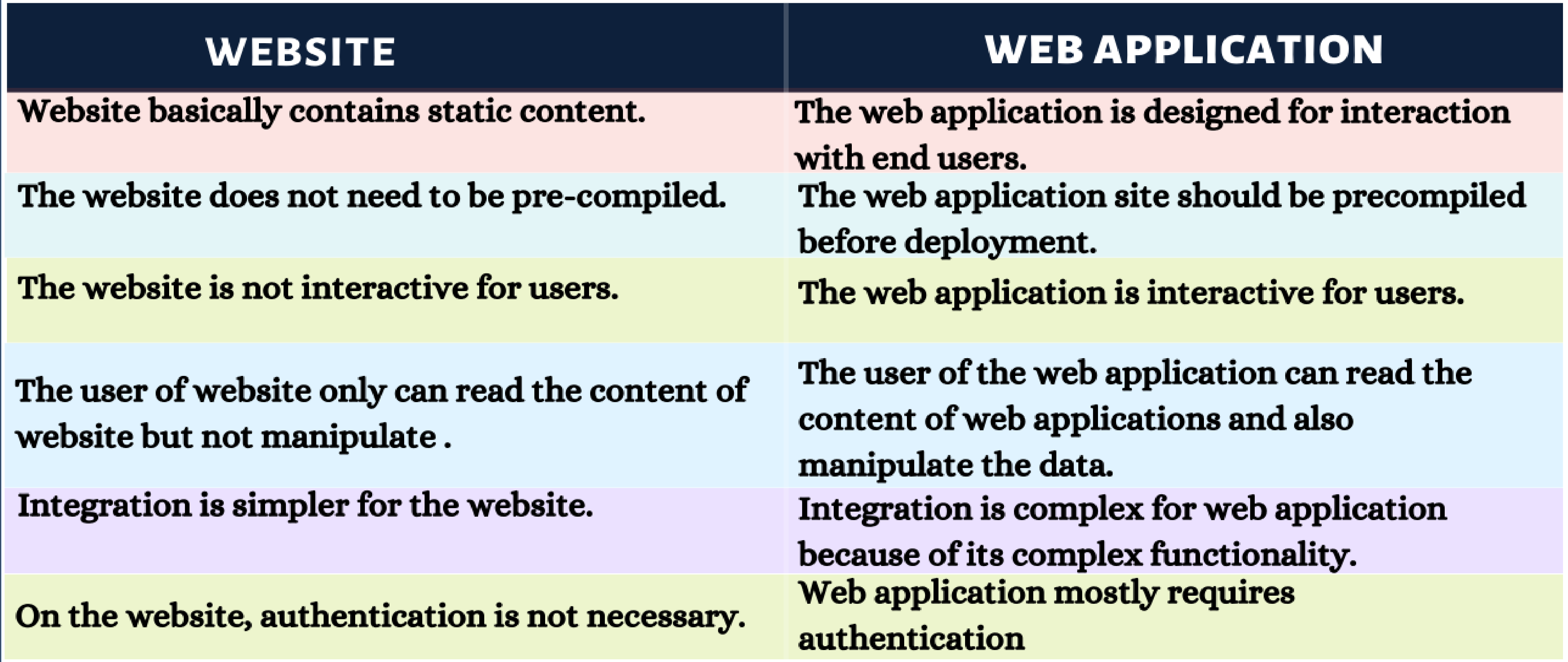
Website Vs Web Application





Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

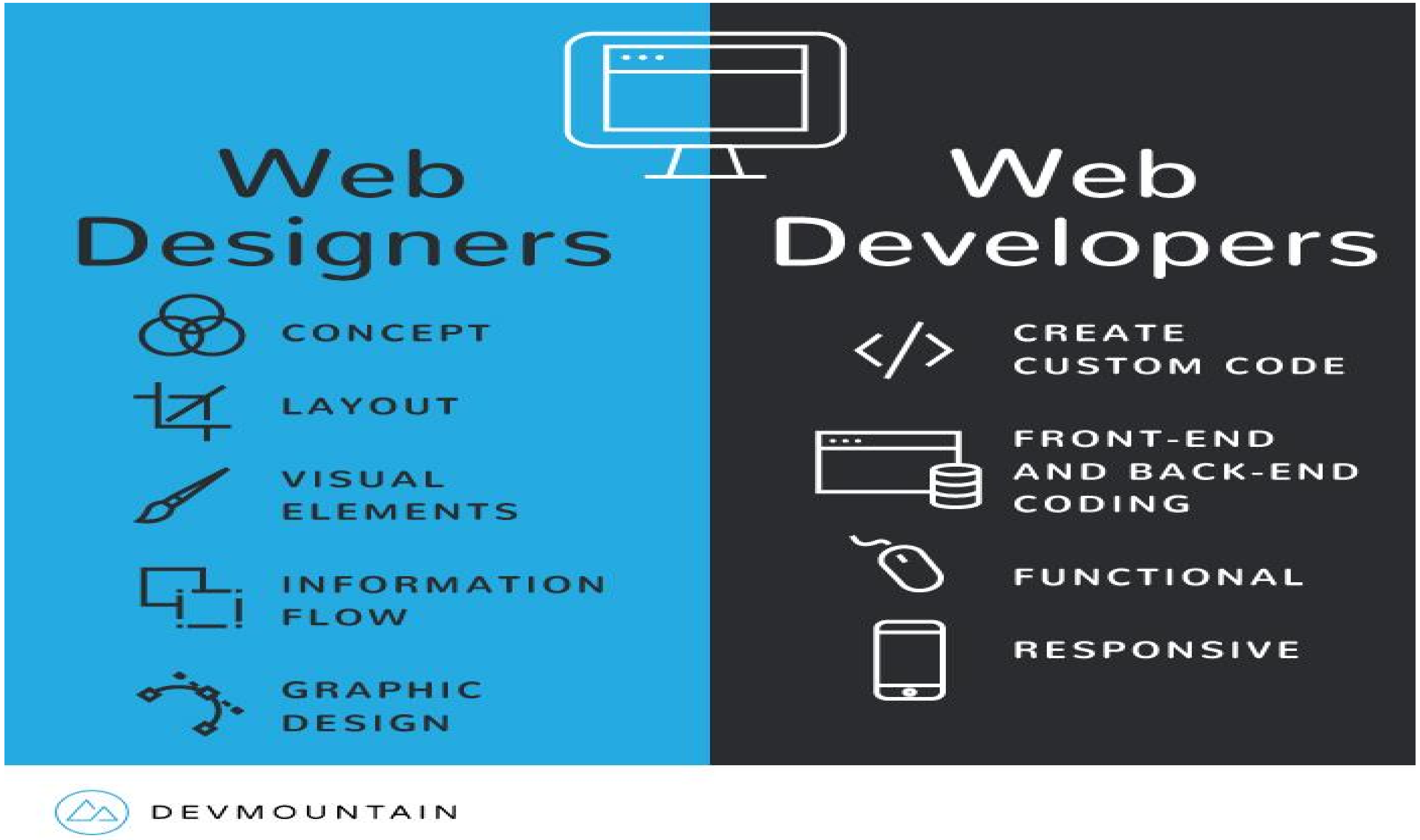
Website Vs Web Application



WEBSITE	WEB APPLICATION
Website basically contains static content.	The web application is designed for interaction with end users.
The website does not need to be pre-compiled.	The web application site should be precompiled before deployment.
The website is not interactive for users.	The web application is interactive for users.
The user of website only can read the content of website but not manipulate .	The user of the web application can read the content of web applications and also manipulate the data.
Integration is simpler for the website.	Integration is complex for web application because of its complex functionality.
On the website, authentication is not necessary.	Web application mostly requires authentication

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Web Designers Vs Web Developer



Web Designers

- CONCEPT
- LAYOUT
- VISUAL ELEMENTS
- INFORMATION FLOW
- GRAPHIC DESIGN

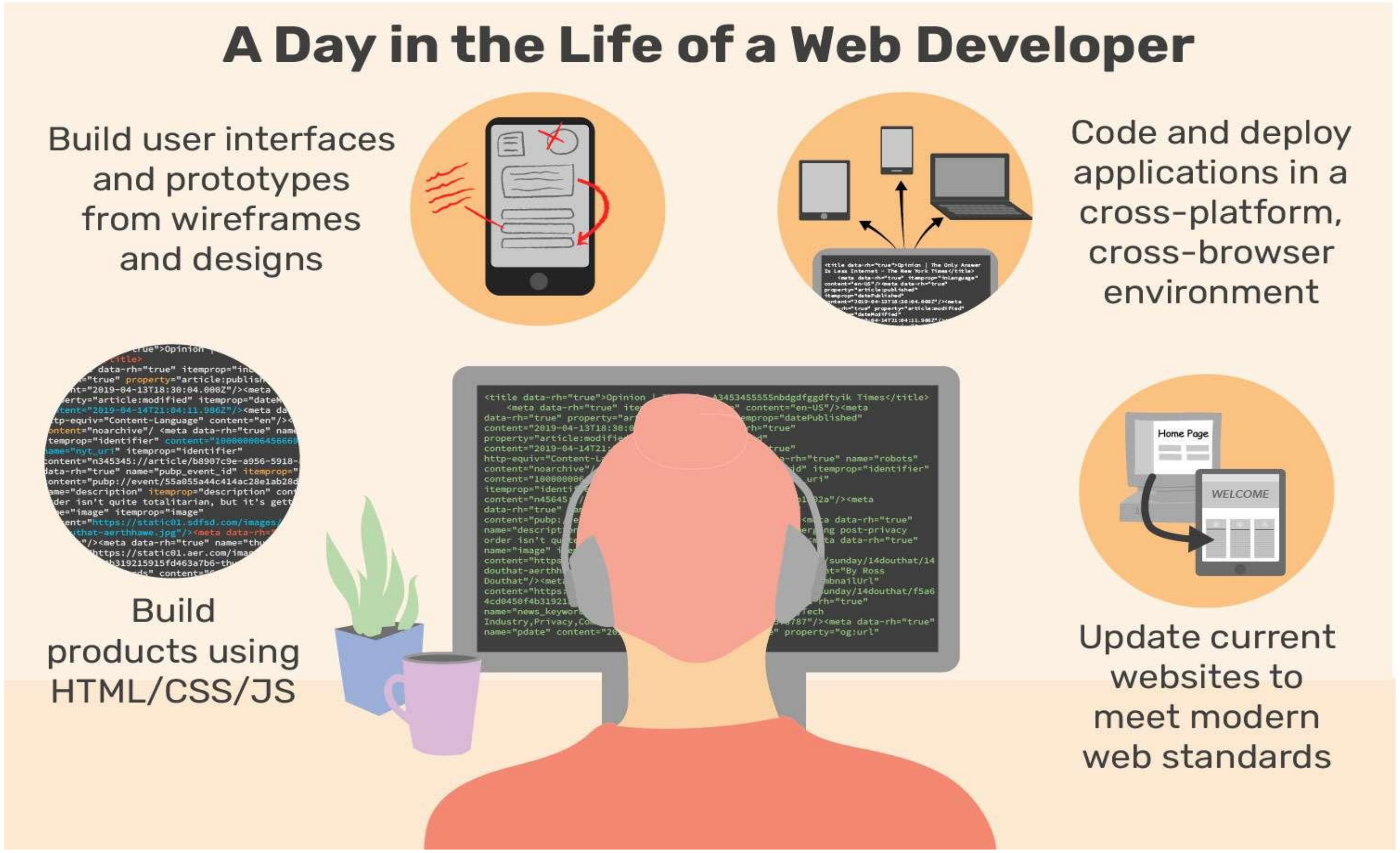
Web Developers

- CREATE CUSTOM CODE
- FRONT-END AND BACK-END CODING
- FUNCTIONAL
- RESPONSIVE

DEVMOUNTAIN

DCU
Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Web Developer Tasks


A Day in the Life of a Web Developer
Build user interfaces and prototypes from wireframes and designs
Code and deploy applications in a cross-platform, cross-browser environment
Build products using HTML/CSS/JS
Update current websites to meet modern web standards
Home Page
WELCOME

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Web Developers Roles

- Web Development is very often a team-based task.
- Roles
  - Project Manager
  - Product Manager/System Architect
  - Front-end Developer(UI/UX Designer)
  - Back-end Developer (Programmer)
  - Database Designer/Developer
  - Quality Assurance/Tester
  - Web Deployment Manager/Web Hosting/Cloud Management
  - Full Stack Developer

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Web Application Development

- Client-Side Development
  - Static Web site
  - Everything runs on the client machine
  - Legacy systems

- Server-Side Development
  - Dynamic Web applications
  - Application runs on a dedicated server
  - Complex tasks execution
  - Database connectivity

Server-Side vs Client-Side

- Advantages:
  - Browser Independent
  - Application Updates
  - Code Protection
  - Local file access (security)
- Disadvantages:
  - Server Resources

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Server-Side Introduction

- Web Sites can harness both client-side and server-side
- Example: Form validation/javascript front-end
- Many client issues, such as browser independence:
  - Concentrating on Server-side rather than client-side
- Previous web development was only Server-side Systems
- Most of the Modern Web Applications use the Client/Server Systems
  - Client-side growing in recent years due to HTML5 and Javascript.

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Client/Server Systems

- “Client/Server represents a logical relationship between an entity (client) that requests a service from another entity (server) which provides a shared resource”

- Same or Distinct Machines
- Client -> multiple servers
- Server -> multiple clients
- Relationship conducted by means of ‘Transactions’
- Well defined requests and responses

-> Client/Server = Cooperative Processing

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Client Attributes

- Client process is “proactive”
- Issues requests to the server
- Typically begins and ends with the user’s session
- Responsible for performing dialog with the user
    - Screen Handling
    - Menu/Command Interpretation
    - Data Entry/Validation
    - Help Processing
    - Error Recovery
    - Graphical Applications also: window handling, mouse, keyboard entry, sound/video
- Client-side Validation vs Server-side Validation
- Sometimes server-side better/required (e.g. Username availability)

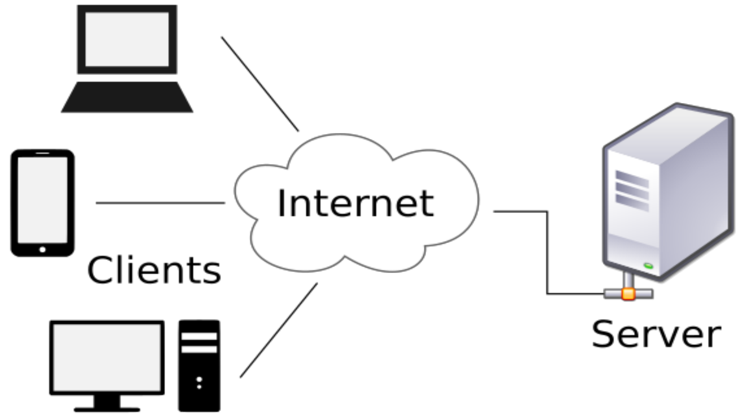
Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

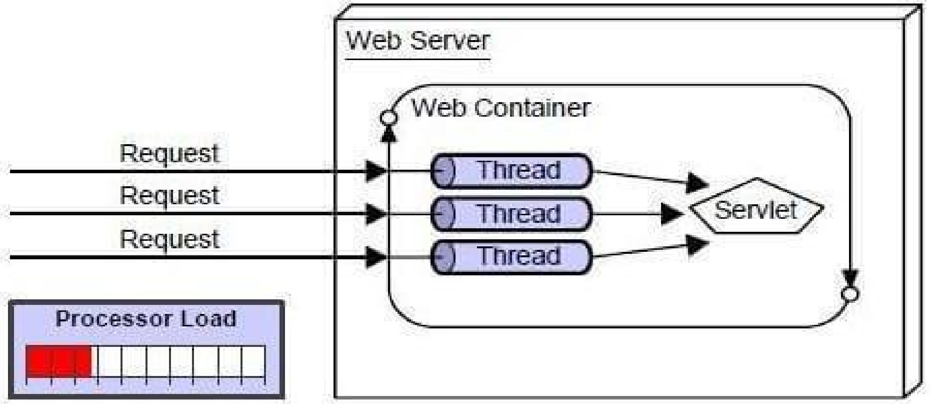
Server Attributes

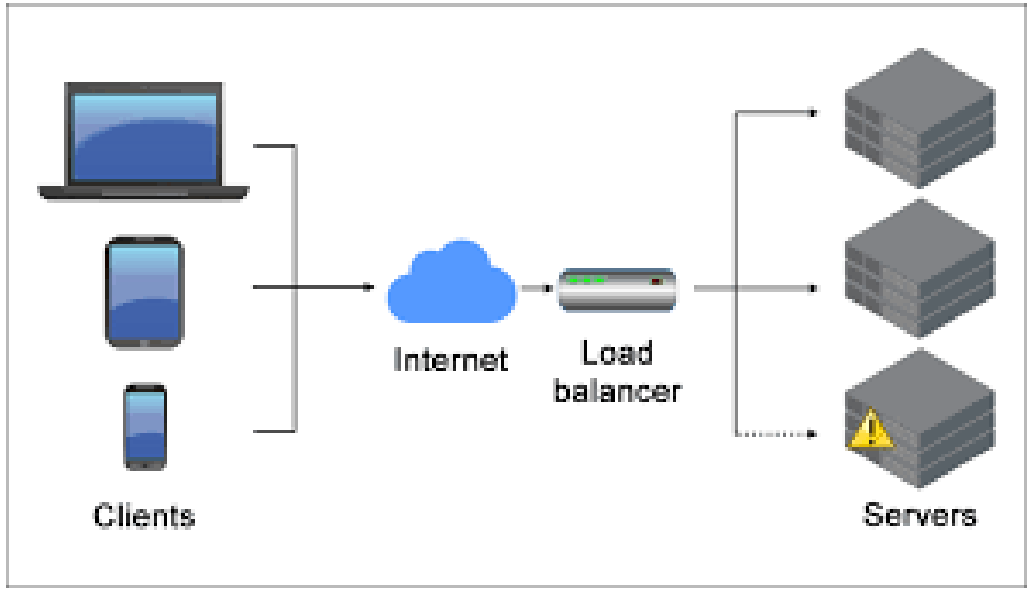
- Server process is “reactive”
- Triggered by the arrival of requests from its clients
- Typically runs regardless of whether clients are active
- Reliability Major Issue – what happens if the server goes down?
- Served by Server Process itself or spawned slave process
- Spawning allows the master process to receive/handle multiple requests simultaneously
- Server is “function-specific” – performs a set of predefined transactions.
- Server takes request and performs required logic to process the request → Transaction

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

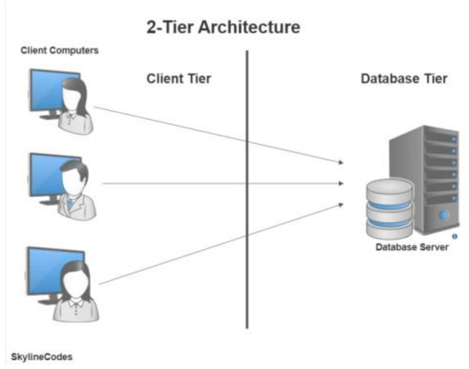
Client Server Interaction

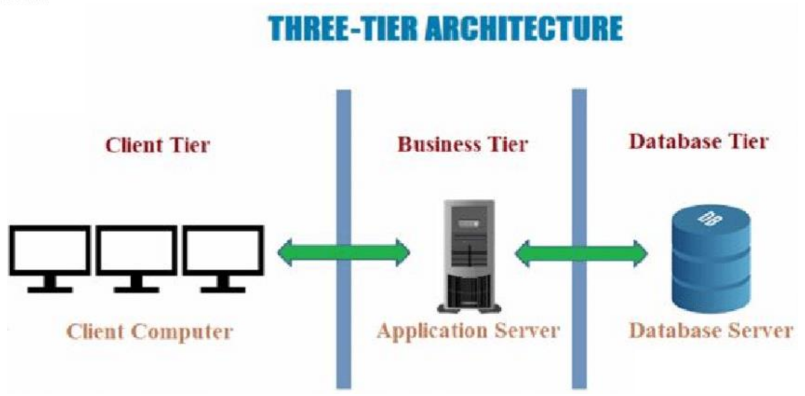


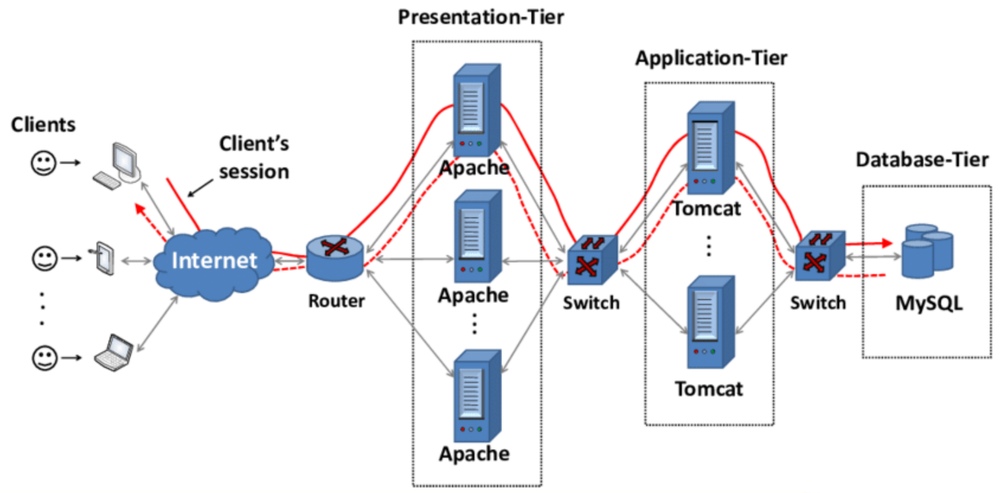




2-Tier → 3-Tier → n-Tier Architecture

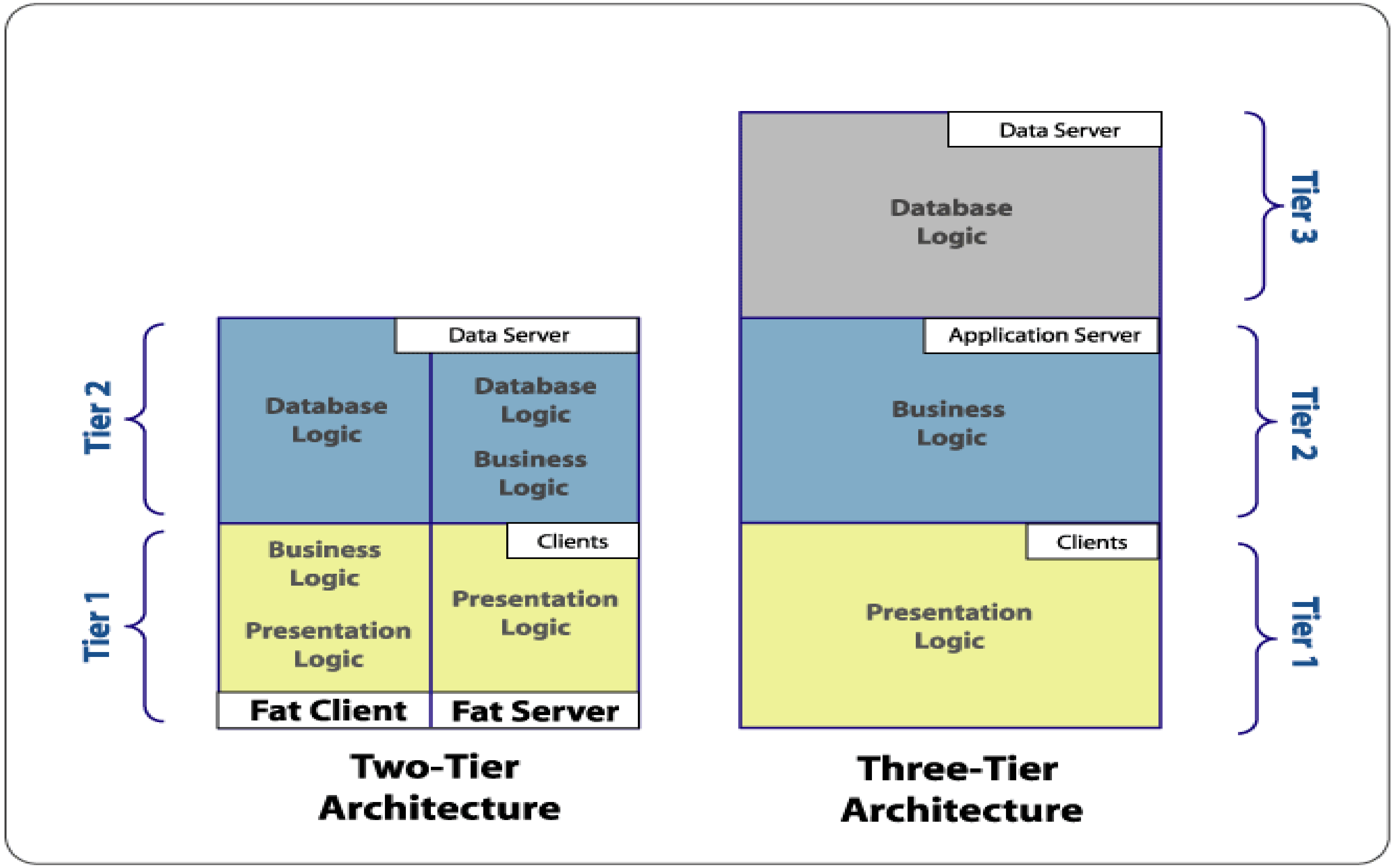






Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

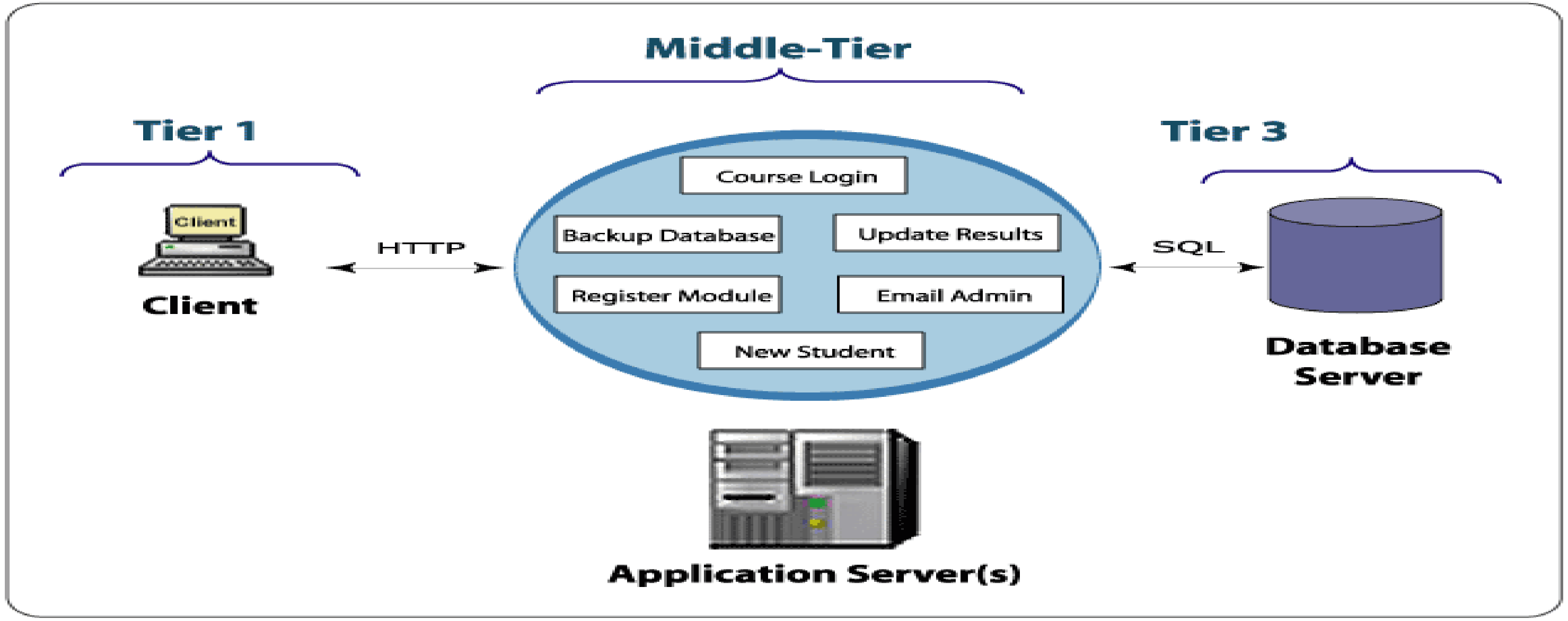
2-Tier vs 3-Tier Architecture



Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

n-Tier Architecture

– 3-Tier often referred to as n-tier client/server architectures
– Typically, the middle tier is not a single application – rather a collection of components
– Each component implements a relatively small business function
-> Each transaction is the product of several middle-tier components



Ollscoil Chathair Bhaile Átha Cliath Dublin City University

n-Tier Architecture

Component-Based

- Applications can be written in smaller stages and released as functionality becomes available.

- No application need be considered “final” (add components later!)

- Different programmers can work on individual components and “plug” them together
  - can use “off-the-shelf” components

- Individual components can be reused for different functionality or applications
  - Java source code reuse
  - Compiled “binary black boxes”

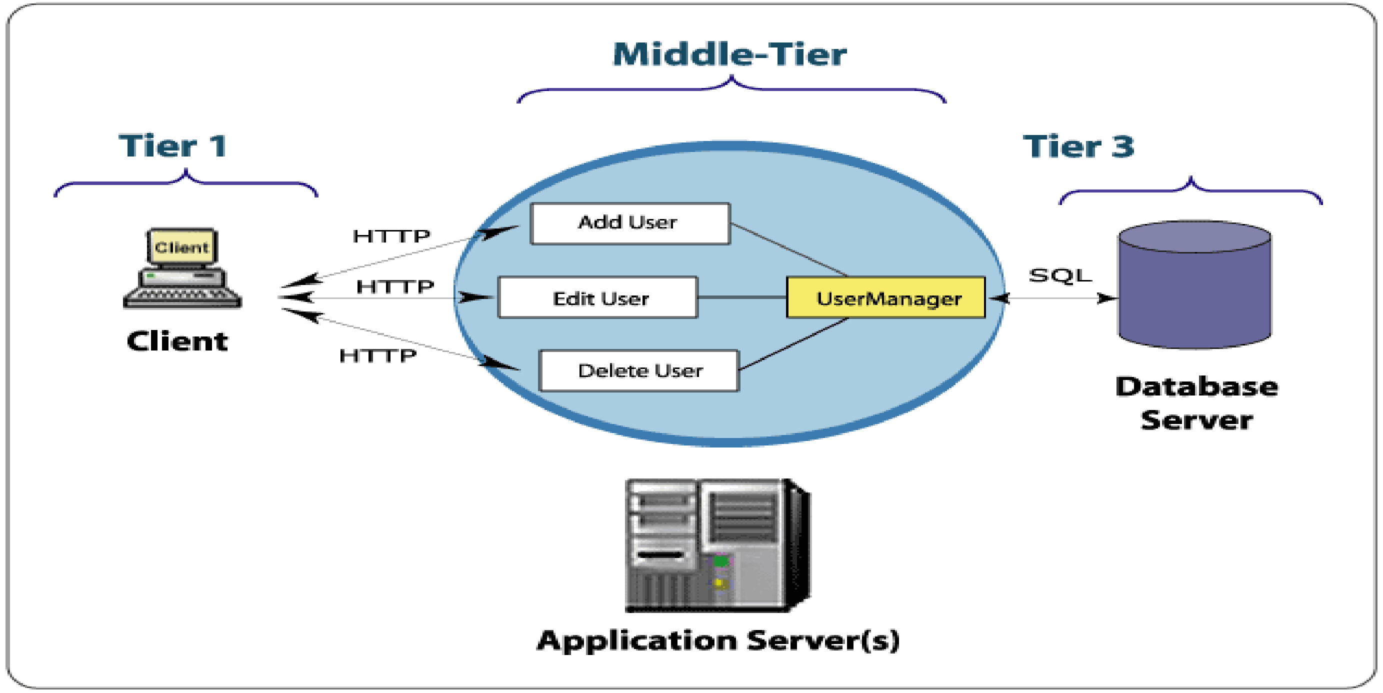
- Components provide a new level of abstraction. Clients send requests to components to execute functions on their behalf. Databases security and schemas are hidden from clients

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

n-Tier Architecture

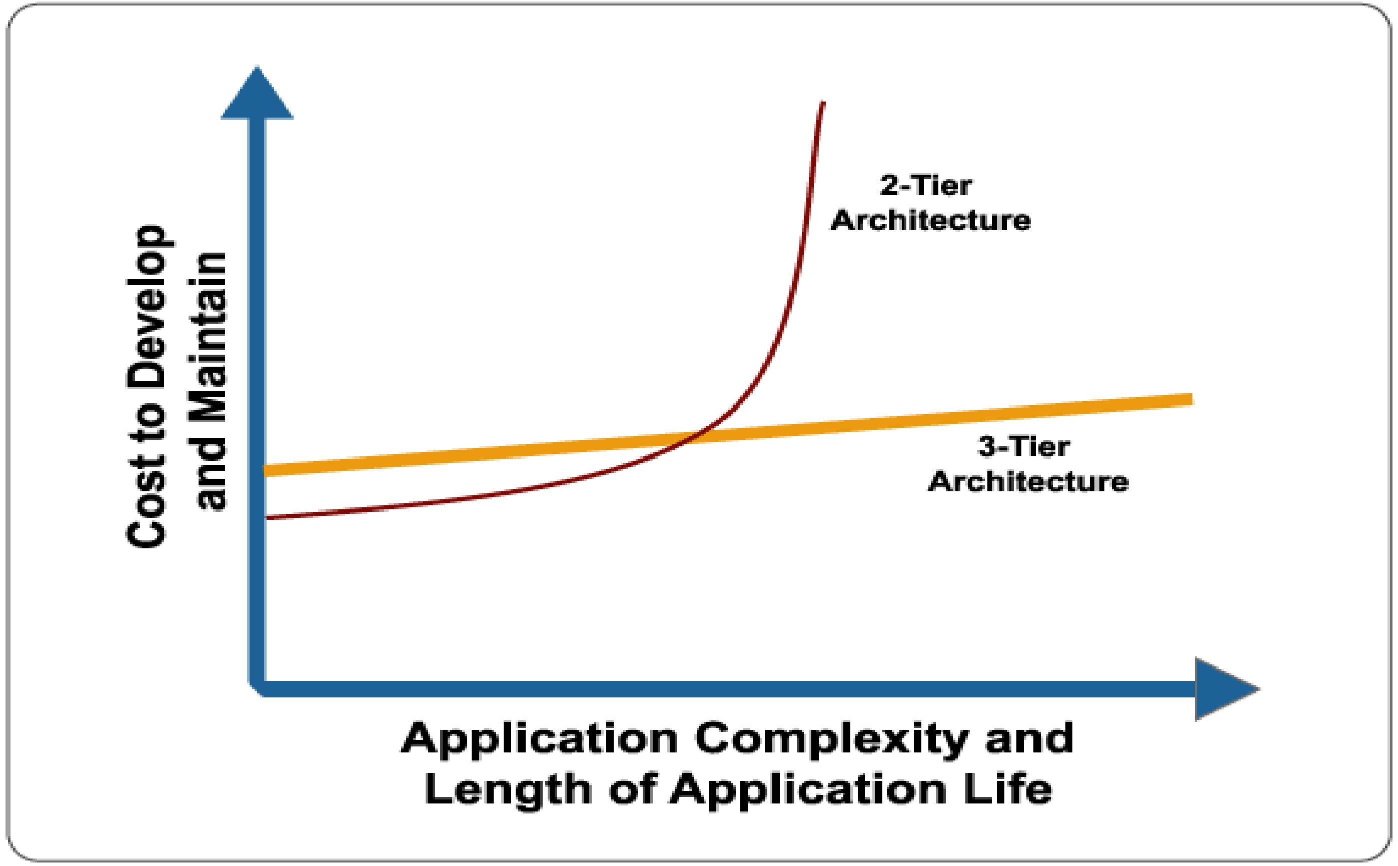
Component-Based

- Component architecture provides consistent, secure access to data and eliminates random, uncontrolled updates from many applications at once. For example:

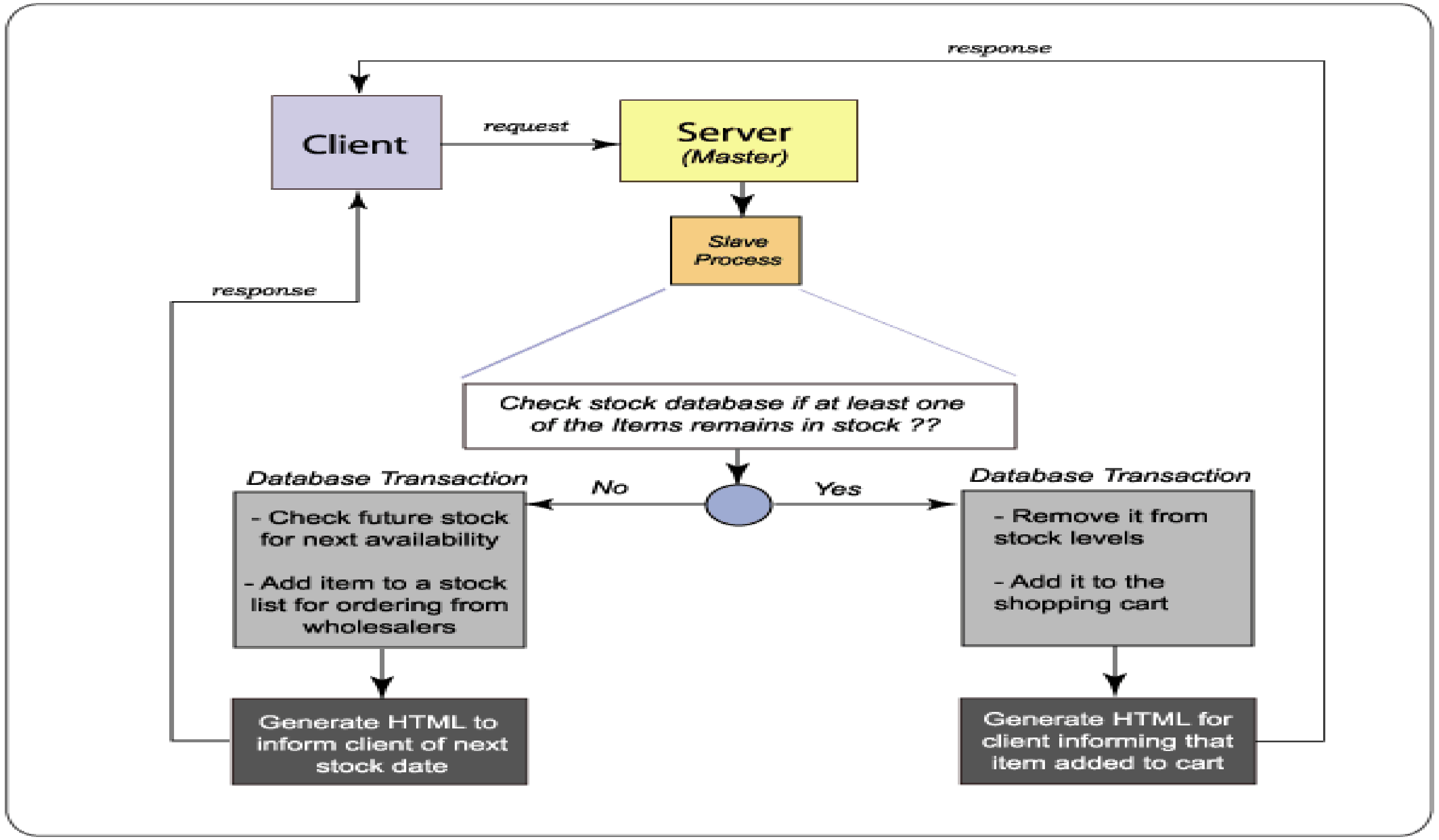


Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

2-Tier vs 3-Tier Architecture



Web Server Transaction Example


response
Client
request
Server
(Master)
Slave Process
response
Check stock database if at least one of the Items remains in stock ??
Database Transaction
No
Yes
Database Transaction
- Check future stock for next availability
- Add item to a stock list for ordering from wholesalers
- Remove it from stock levels
- Add it to the shopping cart
Generate HTML to inform client of next stock date
Generate HTML for client informing that item added to cart

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Detailed Transaction Example

1. Log the transaction

2. Decrease the stock levels of this item by 1

3. Check if the stock levels have dropped below a certain level

4. If stock has dropped below this level, email notify an employee or automatically place an order with the wholesaler

5. Log this stock shortage

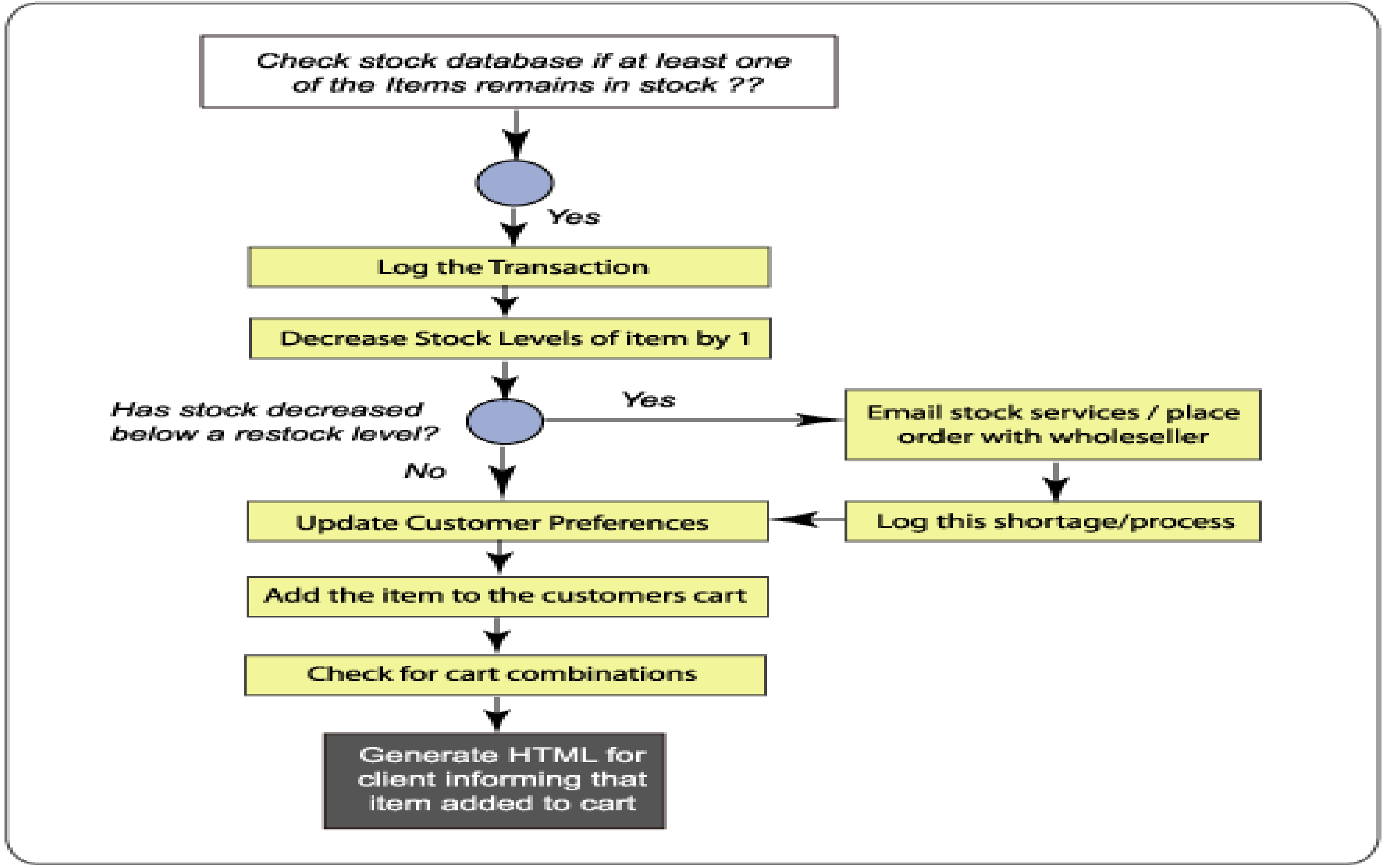
6. Update database information on this customer, so that the "system" knows that the customer is interested in this genre of film

7. Add the item to the user's cart

8. Perform checks on the user's cart to check for 2 for 1 or reduced-price combinations

9. Generate the web page to return to the client

Detailed Transaction Example


Check stock database if at least one of the Items remains in stock ??
Yes
Log the Transaction
Decrease Stock Levels of item by 1
Has stock decreased below a restock level?
Yes
Email stock services / place order with wholeseller
No
Update Customer Preferences
Log this shortage/process
Add the item to the customers cart
Check for cart combinations
Generate HTML for client informing that item added to cart

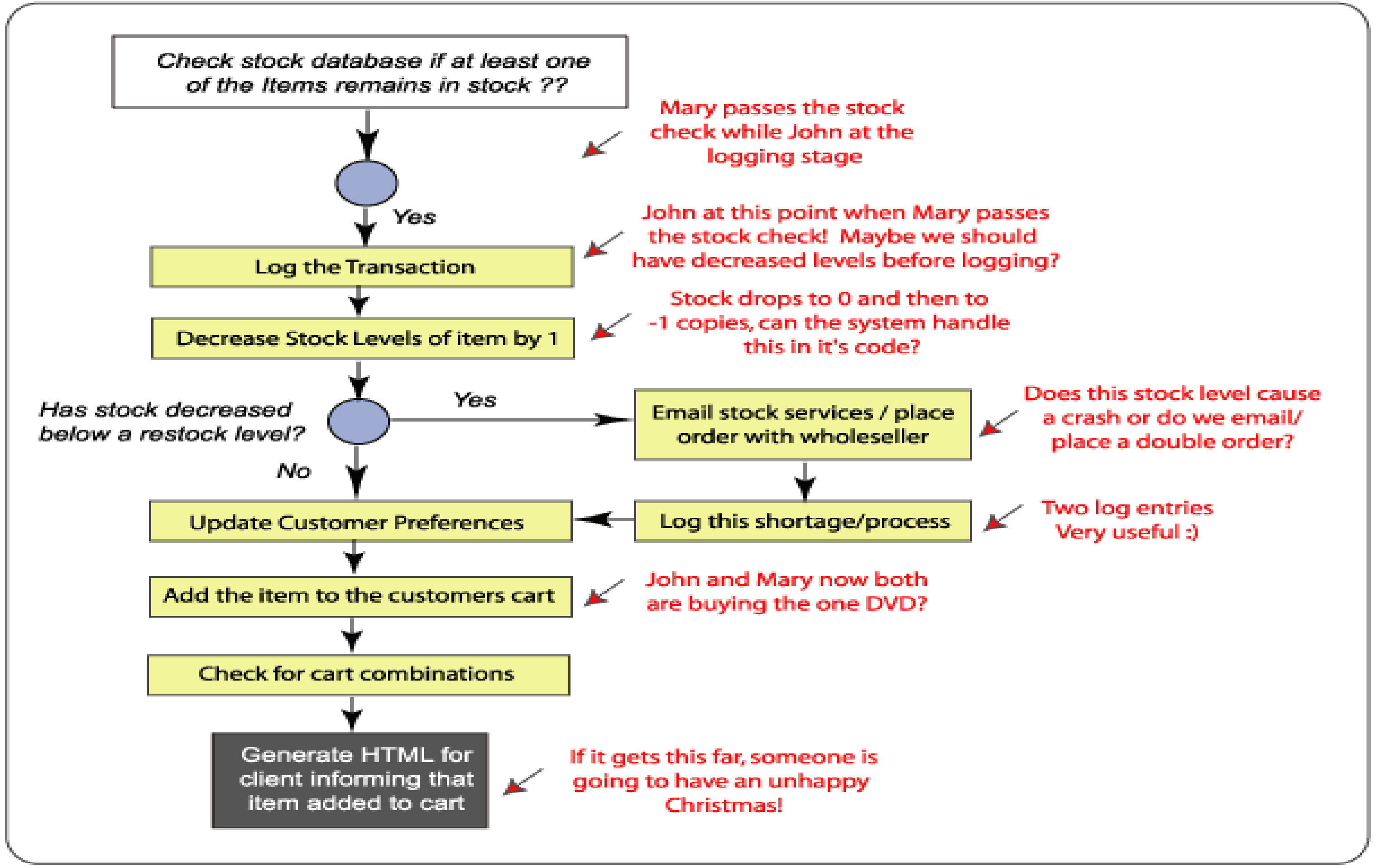
Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Mutual Exclusion Issues

- Server may simultaneously service numerous requests
- Server must resolve mutual exclusion issues
- Otherwise, corrupt transactions/results/data
- Server must ensure that either all or no updates occur
- Consider example where two people attempt to buy the last remaining DVD in stock

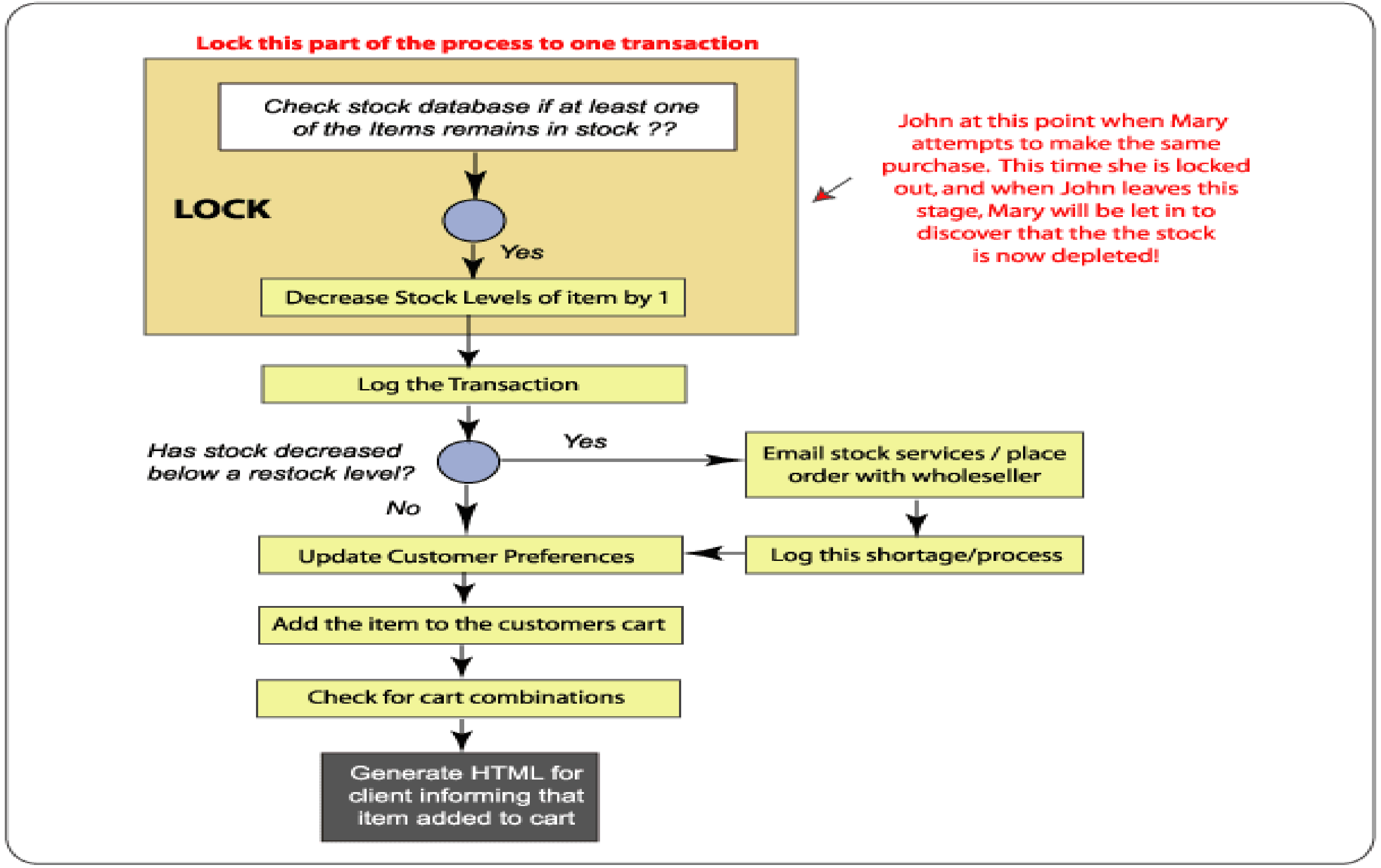
Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Concurrent Purchase Example


Check stock database if at least one of the Items remains in stock ??
Mary passes the stock check while John at the logging stage
Yes
Log the Transaction
John at this point when Mary passes the stock check! Maybe we should have decreased levels before logging?
Decrease Stock Levels of item by 1
Stock drops to 0 and then to -1 copies, can the system handle this in it's code?
Has stock decreased below a restock level?
Yes
Email stock services / place order with wholeseller
Does this stock level cause a crash or do we email/ place a double order?
No
Update Customer Preferences
Log this shortage/process
Two log entries Very useful :)
Add the item to the customers cart
John and Mary now both are buying the one DVD?
Check for cart combinations
Generate HTML for client informing that item added to cart
If it gets this far, someone is going to have an unhappy Christmas!

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Concurrent Purchase - Solution



Lock this part of the process to one transaction
Check stock database if at least one of the Items remains in stock ??
LOCK
Yes
Decrease Stock Levels of item by 1
Log the Transaction
Has stock decreased below a restock level?
Yes
No
Email stock services / place order with wholeseller
Log this shortage/process
Update Customer Preferences
Add the item to the customers cart
Check for cart combinations
Generate HTML for client informing that item added to cart
John at this point when Mary attempts to make the same purchase. This time she is locked out, and when John leaves this stage, Mary will be let in to discover that the the stock is now depleted!

Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Scalability

- Capacity of an application, software or hardware product to adapt to the changing workload applied by consumers of these systems
- If the number of users doubles, can your server handle the load? (response time, crashes, memory shortages)
- How much money should be spent on server resources on a new project?
  - – Too little: server problems
  - – Too much: waste of money / Moore’s Law wastage

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Scalability

Two traditional approaches to scalability:

- Vertical Scaling (scaling up): means adding resources to a single hardware component, such as application or database server. Typically involves the addition of CPUs or increasing memory capacity.

- Horizontal Scaling (scaling out): means adding new hardware components to an existing distributed system. For example, buying an additional server and placing it in parallel. Servers are then “load balanced” whereby each server handles every second request.

Both approaches can be used to address a consistent growth in server demand. However, neither respond particularly well to certain scenarios. Let’s consider two of these scenarios.

Scenario #1

A small company has a website demonstrating their new hand-gesture interface for desktop interaction. This is very cool and someone places an article on www.slashdot.org describing the gesture system and linking to the company website. The resulting huge number of visitors was never anticipated when creating the specification for the web server and as a result the web server falls over under the increased load.

This scenario actually has a name, the "**slashdot effect**", also known as "**slashdotting**". This is not limited to slashdot.org and in recent years would be more likely to be caused by newer social domains such as Twitter or Reddit.

Scenario #2

A university has a web application server and a base application where students can log into their "portal" to view their registration record, timetables, library loans etc. This is used sporadically by students, with occasional log-ins.

Typical Scenario
Student Population: 10,000
Log-ins: Once every 14 days (two weeks) = 714 log-ins per day
Day-time Hours (16) = 714/16 = 45 log-ins per hour
University vertically specs a server to handle 10 times this number!

Problem Situation
The university decides (rightly) that this portal would be an appropriate vehicle for the distribution of examination results. Students should be allowed to log in and view their results when they are released at 12:00 on February 10th.

Students eager for their results, **all try to log in simultaneously.** So now we have 10,000 students all attempting to log in at 12:00 to check their results.

Spamming of server starts! Server falls over!

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Cloud Computing

- Cloud used everywhere today: email, social networking, websites, file storage and even applications (Google Docs, 365 Live)
- For developers, there are several cloud computing platforms offered
- Amazon Web Services (AWS)
- Buy no hardware -> instead effectively rent hardware from Amazon
- AWS hosting big players NetFlix, Foursquare, Pinterest and Amazon itself
- Hardware Virtualisation : virtual machines can be created, acting like a real computer with an operating system. AWS have vast server farms of high-end machines (host machines), from which they can virtualise servers (guest machines) on behalf of their client users.

Note: Only covering Cloud Computing in relation to deployment and scaling of web applications – This is a massive topic!

DCU
Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Cloud Computing

• Users of Cloud Service can log into Cloud Platform and choose types of servers (CPU, memory, OS, kernel etc.)

• Within minutes, servers are set up, mapped to IPs with login/encryption keys

• Users pay for what they use – costs can seem quite considerable.

• Traditional Local Server Costs: specifying, building, installing software, electricity, patching, hardware failures, server-room cooling, security etc.

• Virtualised servers introduce elasticity dynamically provisioning resources on a near real-time basis. No need to engineer for peak loads!

• If the load on the server becomes excessive, Cloud Platform will automatically scale your application onto the other server instances -> pay by usage of these instances.

Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

Cloud Computing

Two final considerations on moving to cloud infrastructure:

• Downtime: are these systems sufficiently reliable. AWS has had some crashes over the past few years bringing down some major sites

• Data Protection/Security/Privacy: Is it suitable to store, for example, your student academic record on a cloud server? Who can view your data – are there legal issues?

DCU
Ollscoil Chathair Bhaile Átha Cliath
Dublin City University