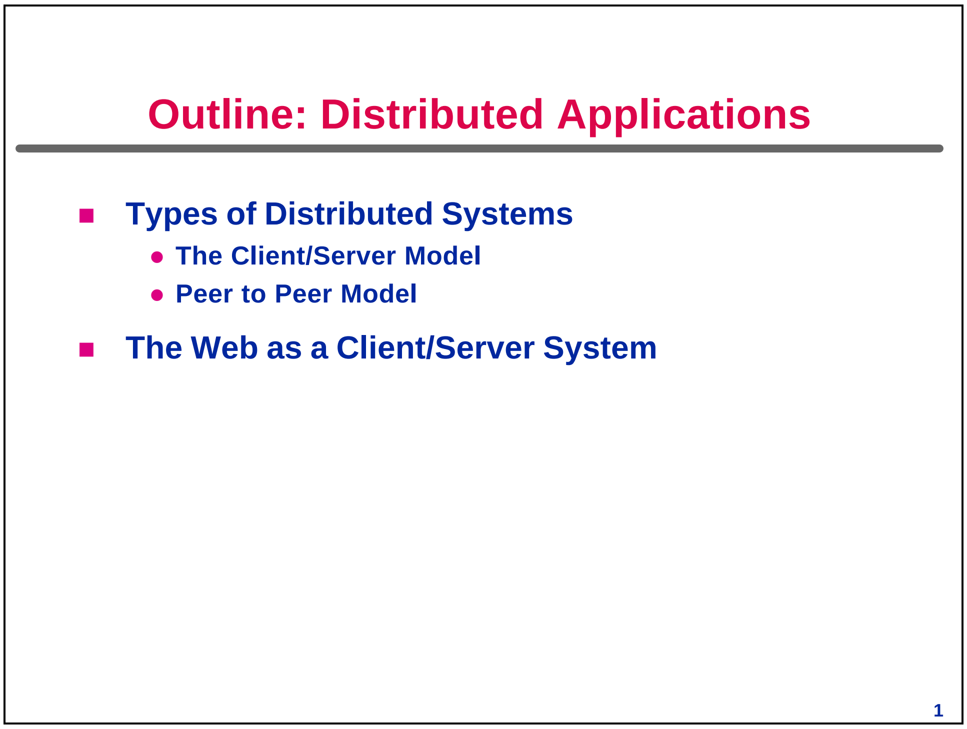
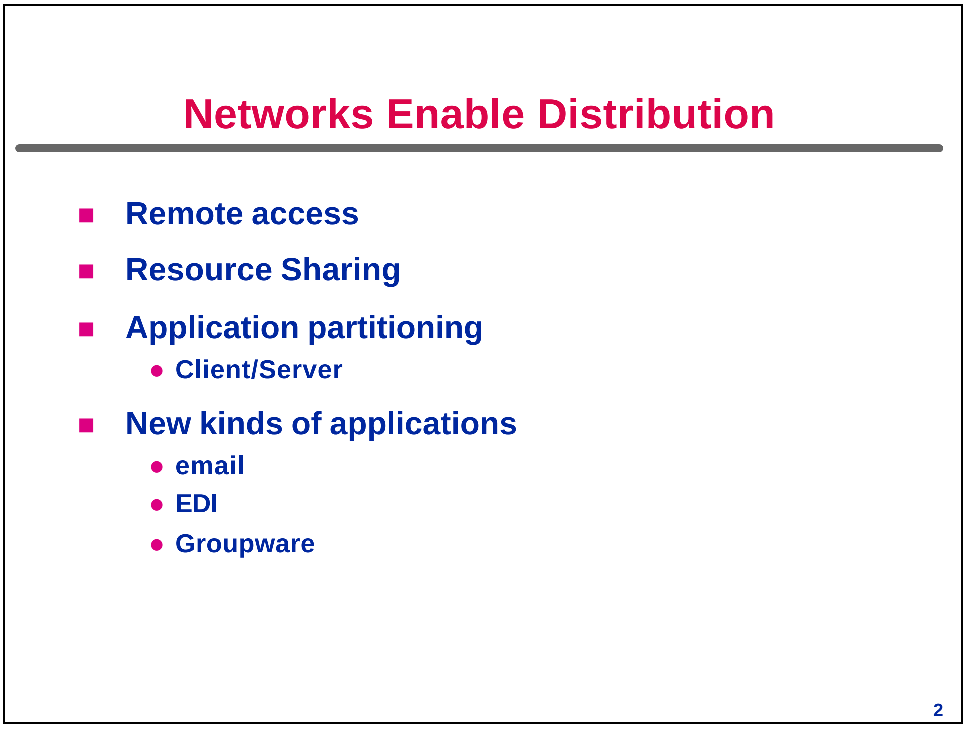


Outline: Distributed Applications

- Types of Distributed Systems
  - The Client/Server Model
  - Peer to Peer Model
- The Web as a Client/Server System

1



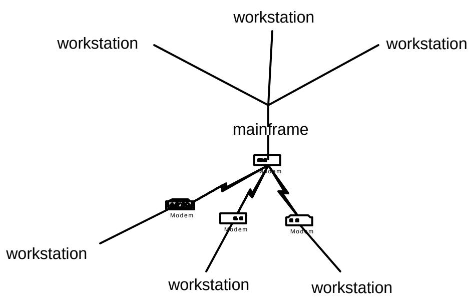
Networks Enable Distribution

- Remote access
- Resource Sharing
- Application partitioning
  - Client/Server
- New kinds of applications
  - email
  - EDI
  - Groupware

2

Remote access

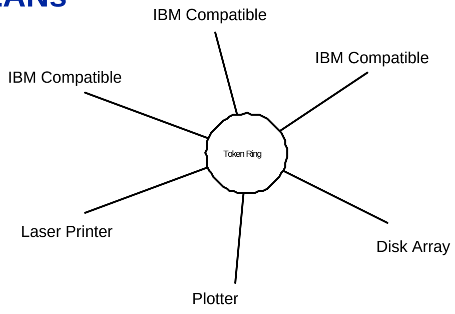
- Centralized computer power
- Several remote “dumb” terminals
- Example: Early airline reservation systems



3

Resource sharing

- Several stand-alone computers share expensive peripherals
  - e.g. printers, plotters, scanners
- Example: Office LANs



4

Application Partitioning

- Split the application functionality in several pieces
- Place each piece in the machine where it can be handled most efficiently

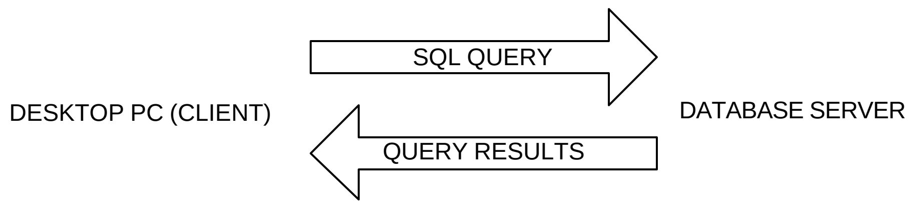
5

The Client/Server Model

- Split application functionality into two pieces:
  - CLIENT
    - Sends requests to server to access network resources
    - Usually (but not always) is the piece that interfaces with user
    - Usually a medium-end PC
  - SERVER
    - Furnishes clients with application-specific resources
      - Databases
      - Huge disk drives
      - Connections to network
    - Accepts and responds to requests from several clients
    - Usually a high-end PC, minicomputer, or mainframe

6

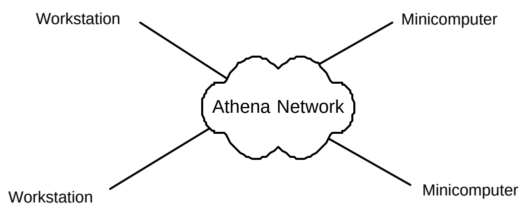
Example: Database Servers



- Runs user interface
- Formulates queries and displays results

- Contains all data
- Runs queries, updates tables, and returns results

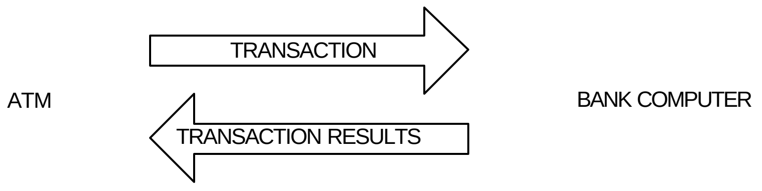
Example: File Servers



Diskless Athena workstations:
- Provide user interface
- Execute programs

Athena file servers:
- Store all programs
- Store all user files

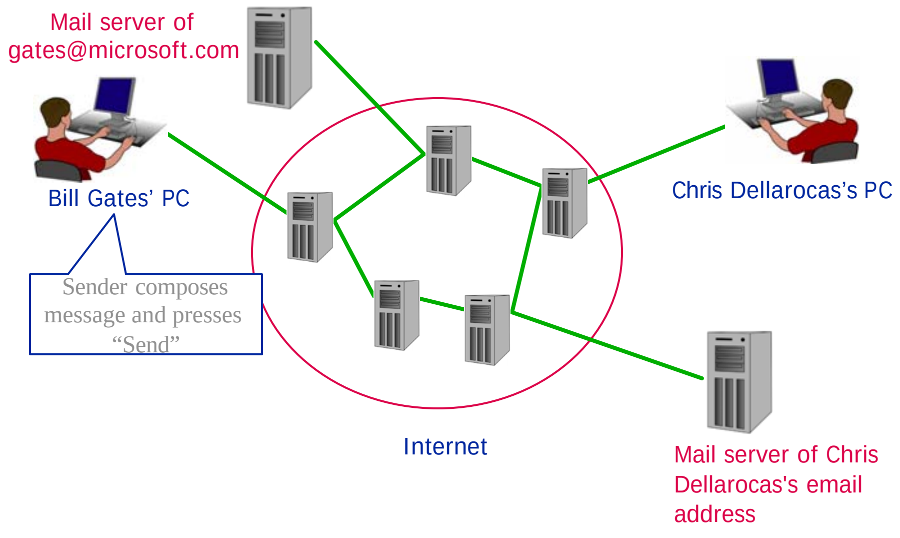
Example: ATM machines (Transaction Servers)

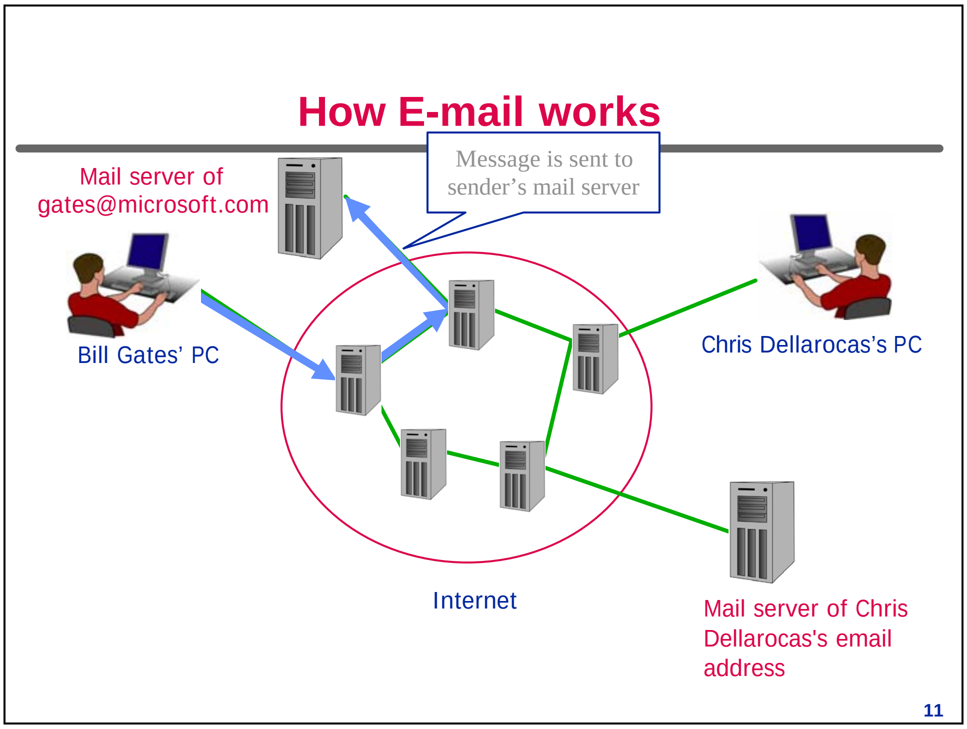


- Executes local dialog with customer
- Control mechanical parts of ATM

- Checks account balances
- Credits or debits customer account

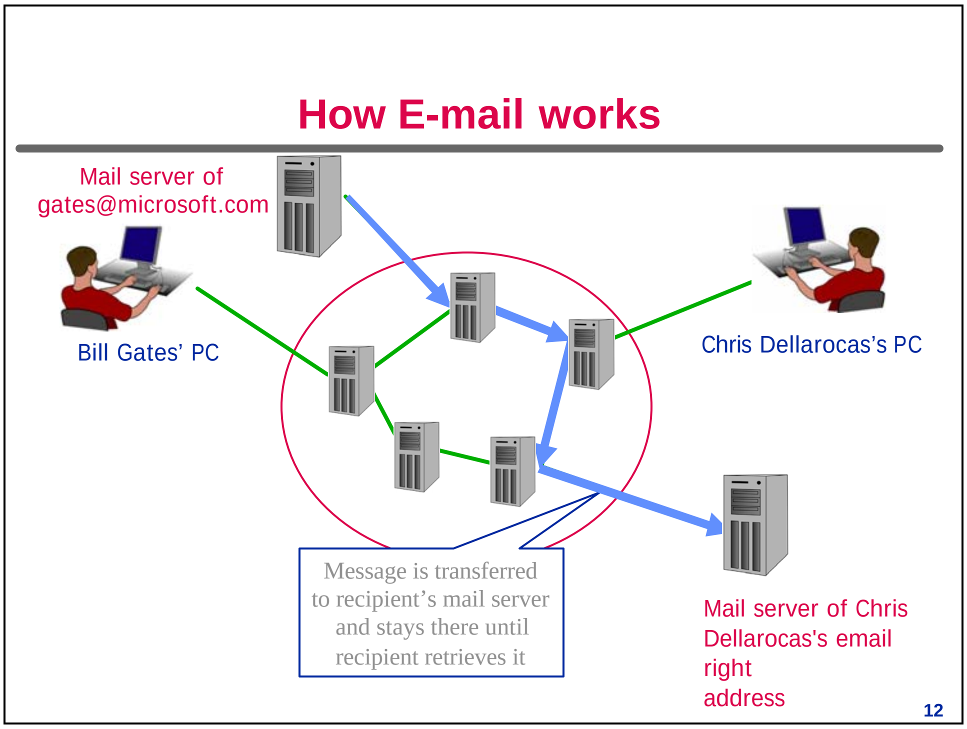
How E-mail works


Mail server of gates@microsoft.com
Bill Gates' PC
Sender composes message and presses "Send"
Chris Dellarocas's PC
Internet
Mail server of Chris Dellarocas's email address



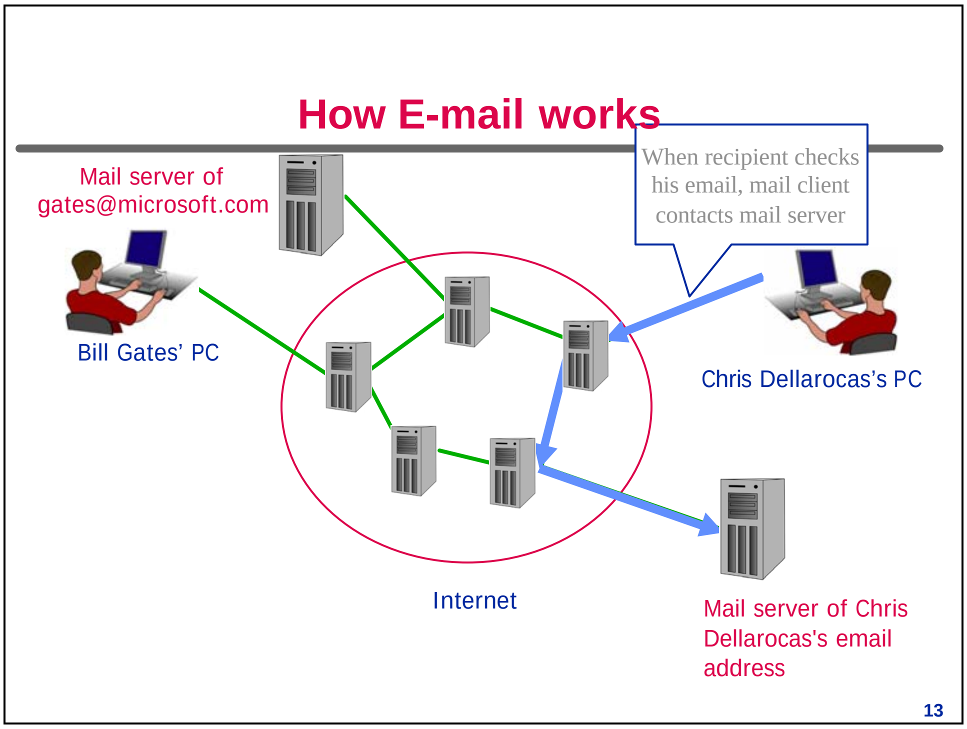
How E-mail works

Mail server of gates@microsoft.com
Message is sent to sender's mail server
Bill Gates' PC
Chris Dellarocas's PC
Internet
Mail server of Chris Dellarocas's email address
11



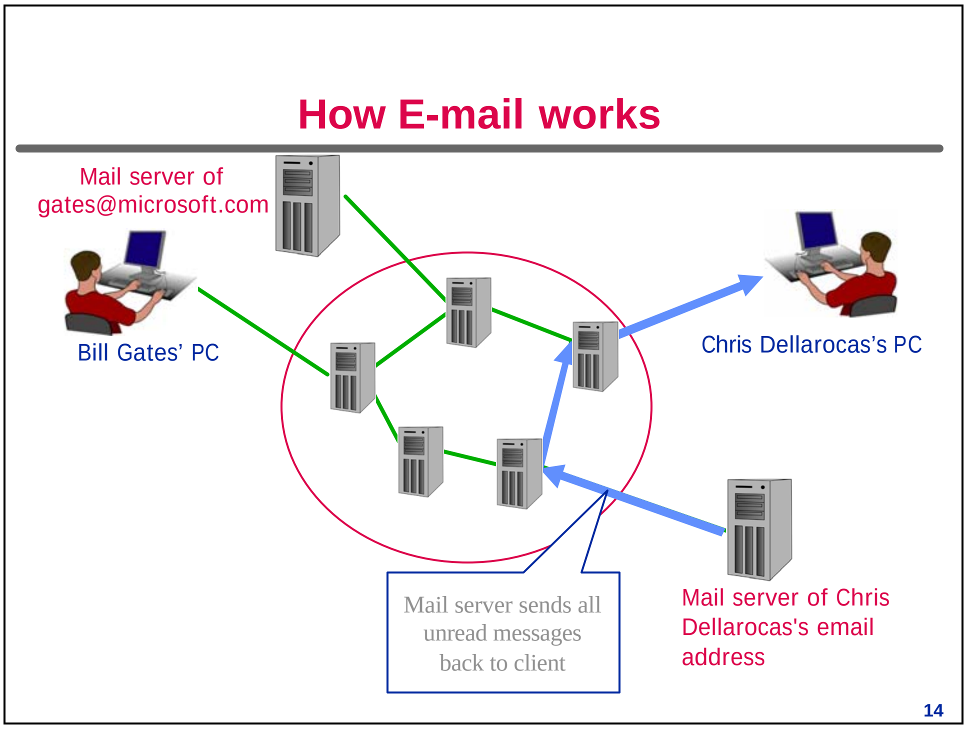
How E-mail works

Mail server of gates@microsoft.com
Bill Gates' PC
Chris Dellarocas's PC
Message is transferred to recipient's mail server and stays there until recipient retrieves it
Mail server of Chris Dellarocas's email right address
12



How E-mail works

Mail server of gates@microsoft.com
When recipient checks his email, mail client contacts mail server
Bill Gates' PC
Chris Dellarocas's PC
Internet
Mail server of Chris Dellarocas's email address
13

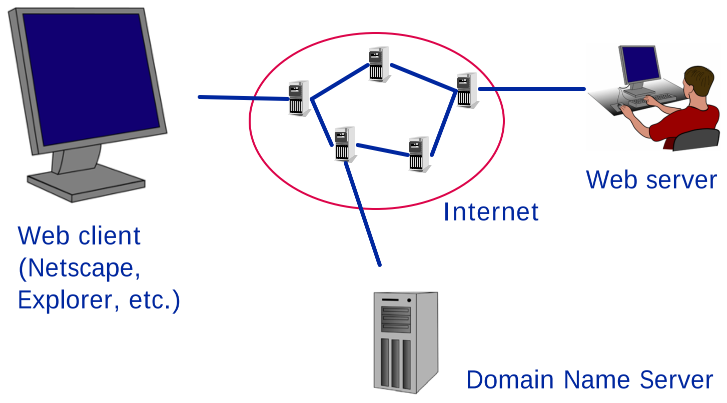


How E-mail works

Mail server of gates@microsoft.com
Bill Gates' PC
Chris Dellarocas's PC
Mail server sends all unread messages back to client
Mail server of Chris Dellarocas's email address
14

How the Web works

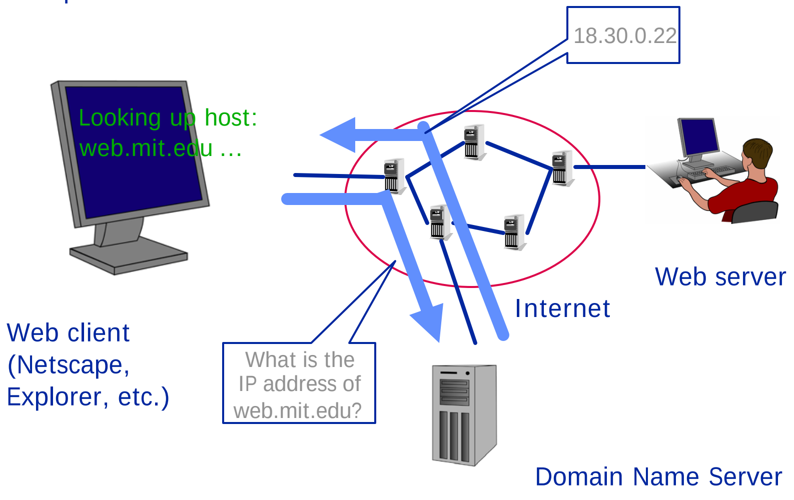
Open Location:
http://web.mit.edu/sloan/www/index.html



15

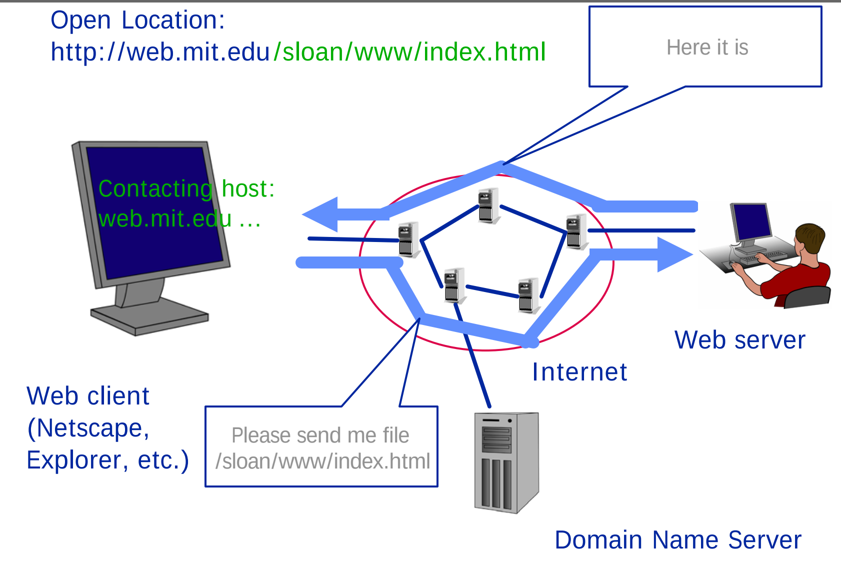
How the Web works

Open Location:
http://web.mit.edu/sloan/www/index.html



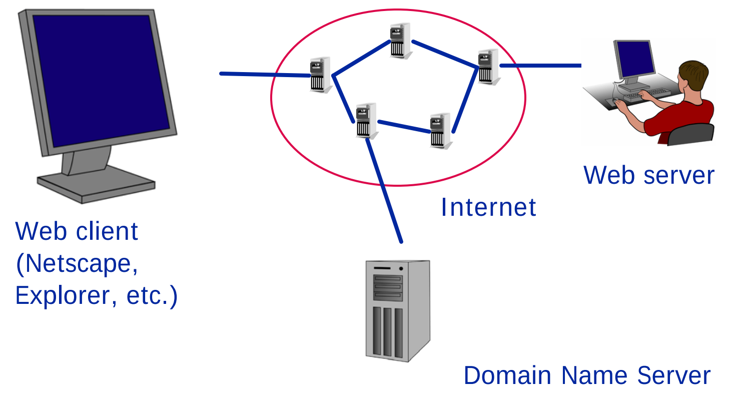
16

How the Web works



17

How the Web works



18

The WWW as a Client/Server System

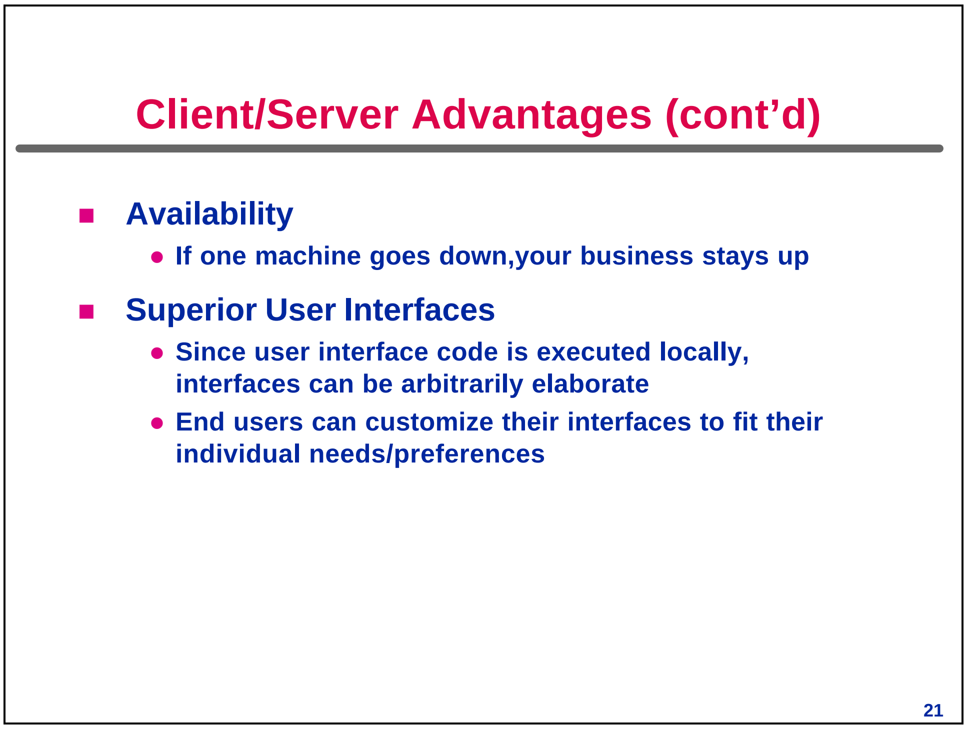
- Web Clients
  - Use HTTP protocol to connect to servers
  - Request and display Web pages stored in servers
  - Typical clients: Web browsers
- Web Servers
  - Listen for incoming connections from clients
  - Use HTTP protocol to converse with clients
  - Store and transmit Web pages to clients
- Any machine connected on the Internet can be a Web client and/or a Web server
  - all It takes is the right software

19

Client/Server Advantages

- Price
  - PC networks much cheaper than mainframes of equivalent computing power
- Scalability
  - Easy to grow/modernize system as needs change
    - Add clients
    - Upgrade/add servers
- Vendor-Independence
  - Different system components can come from different vendors

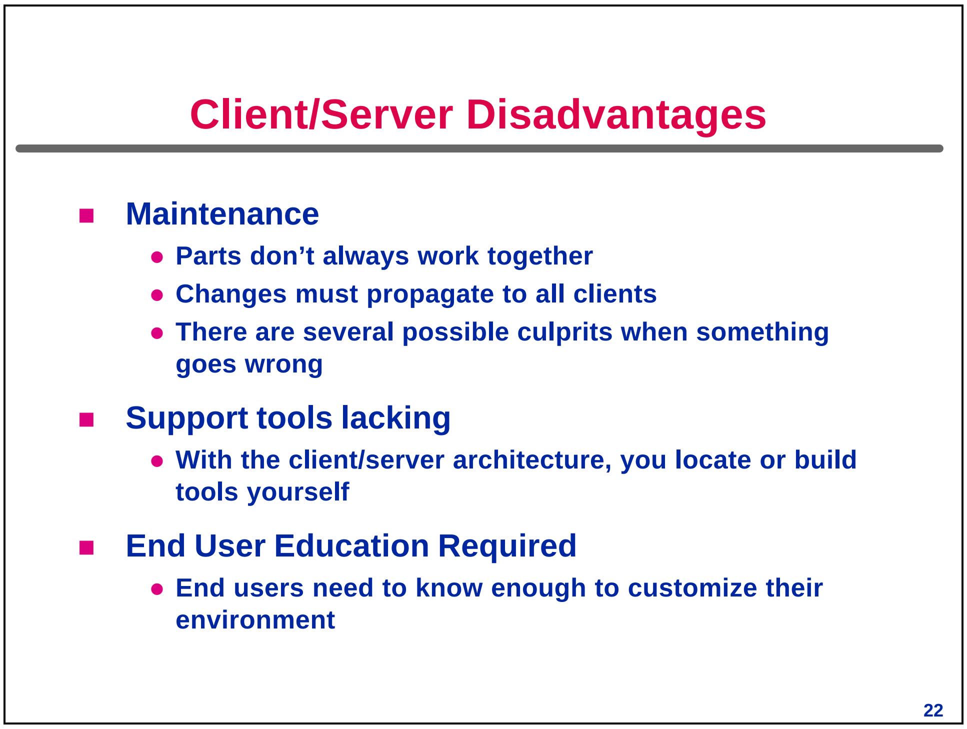
20



Client/Server Advantages (cont'd)

- Availability
  - If one machine goes down,your business stays up
- Superior User Interfaces
  - Since user interface code is executed locally, interfaces can be arbitrarily elaborate
  - End users can customize their interfaces to fit their individual needs/preferences

21



Client/Server Disadvantages

- Maintenance
  - Parts don't always work together
  - Changes must propagate to all clients
  - There are several possible culprits when something goes wrong
- Support tools lacking
  - With the client/server architecture, you locate or build tools yourself
- End User Education Required
  - End users need to know enough to customize their environment

22

Types of Client/Server Systems

- Issues:
  - How much processing to do locally vs. in the server
    - PRESENTATION LAYER: User Interface
    - APPLICATION LAYER: Application-specific processing
    - DATA MNG LAYER: Actual storage of data

    - THIN CLIENTS: Only presentation layer
  - In how many pieces to split the application
    - 2-tier, 3-tier and multi-tier architectures

Types of Client/Server Systems

Fat Client Systems

- Client implements presentation and application layer
  - local processing at client side
- Example: Lotus Notes, Quicken
- Advantages
  - better server scalability -- server needs to do less work
  - less network traffic
- Disadvantages
  - client is more complex; difficult to port to different platforms
  - changes in server architecture are more likely to require changes in client

25

Thin Client Systems

- Client only implements presentation layer
  - All processing is done at server side
- Example: WWW
- Advantages
  - easy to port client to different architectures
  - client is decoupled from changes in the application
- Disadvantages
  - server does all the work; might get easily saturated
  - potentially long network delays

26

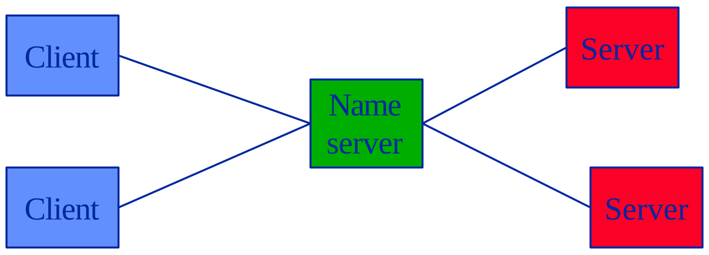
2-tiered vs. multi-tiered architectures

- Limitations of 2-tiered client/server
  - single server
  - server location fixed (otherwise clients need to change)
- What if…
  - we want to add a second server to share the load
  - we want to move the network location of a server
  - we want to change our database from Sybase to Oracle
  - … but do not want to modify all clients

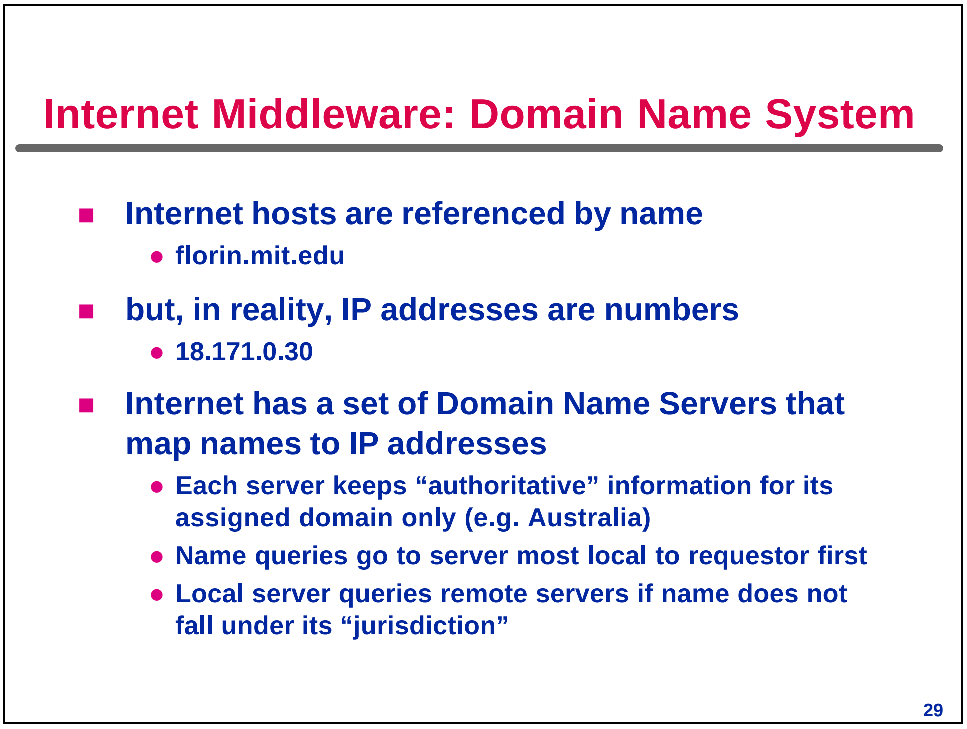
27

Enter Middleware

- Set of technologies that "glue" together clients and servers
- Examples:
  - Name servers
  - Load balancers



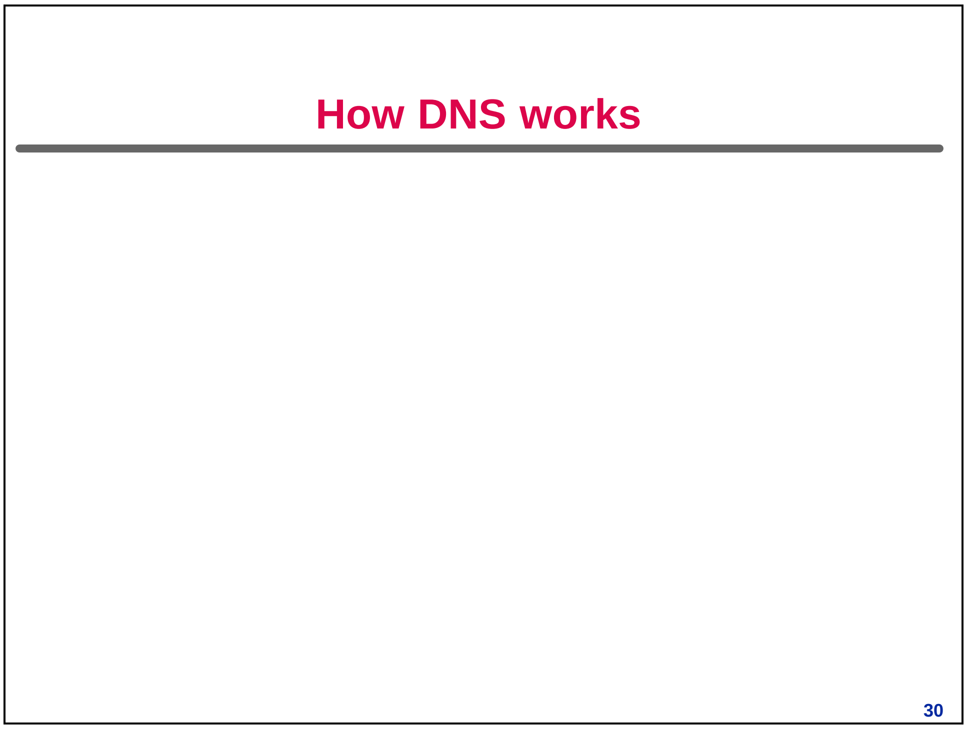
28



Internet Middleware: Domain Name System

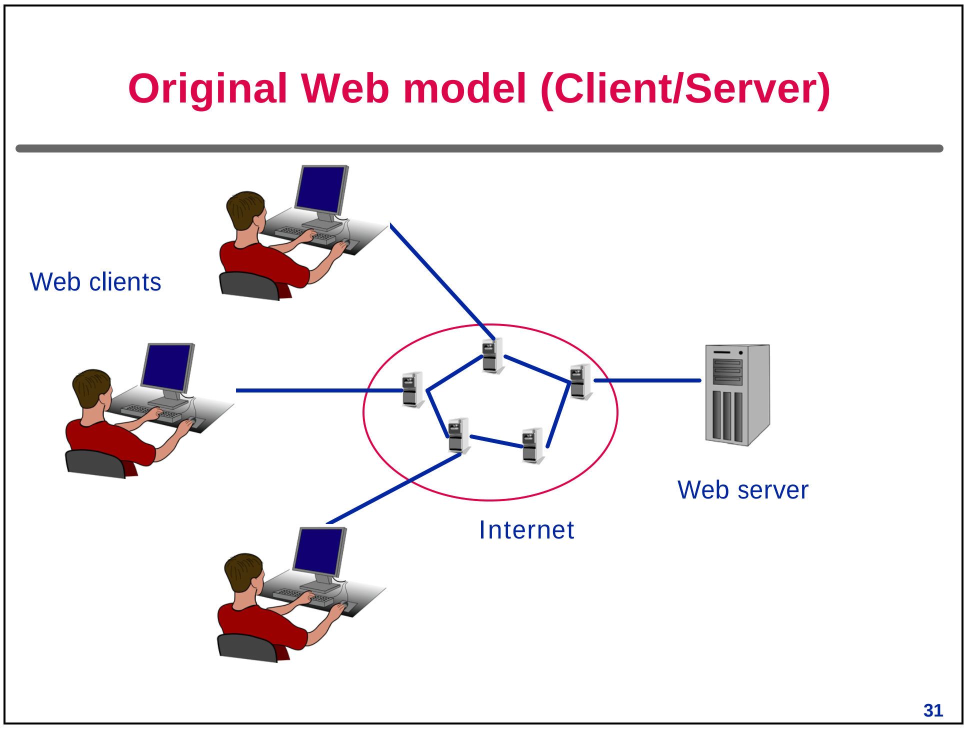
- Internet hosts are referenced by name
  - florin.mit.edu
- but, in reality, IP addresses are numbers
  - 18.171.0.30
- Internet has a set of Domain Name Servers that map names to IP addresses
  - Each server keeps “authoritative” information for its assigned domain only (e.g. Australia)
  - Name queries go to server most local to requestor first
  - Local server queries remote servers if name does not fall under its “jurisdiction”

29

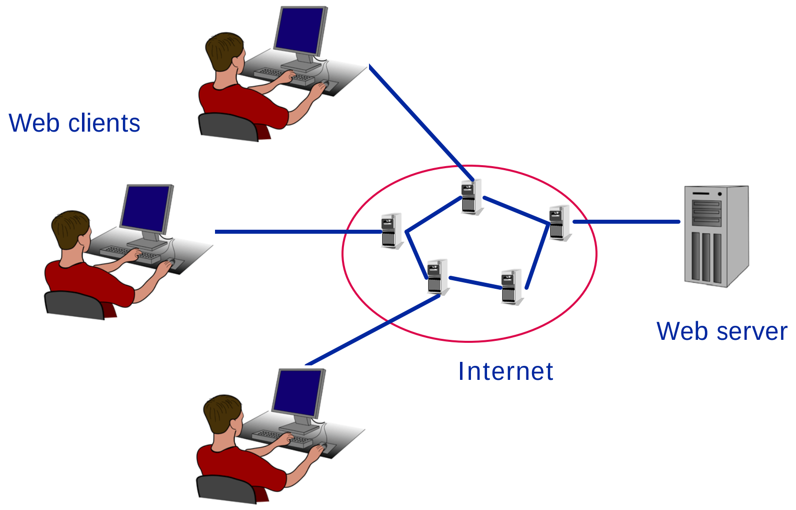


How DNS works

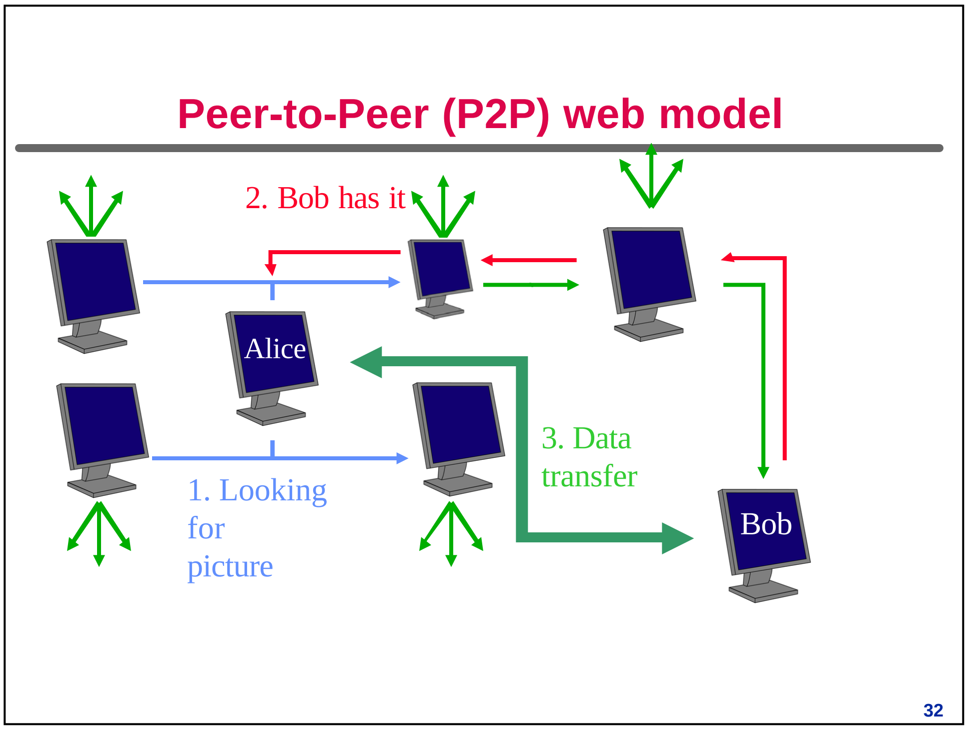
30



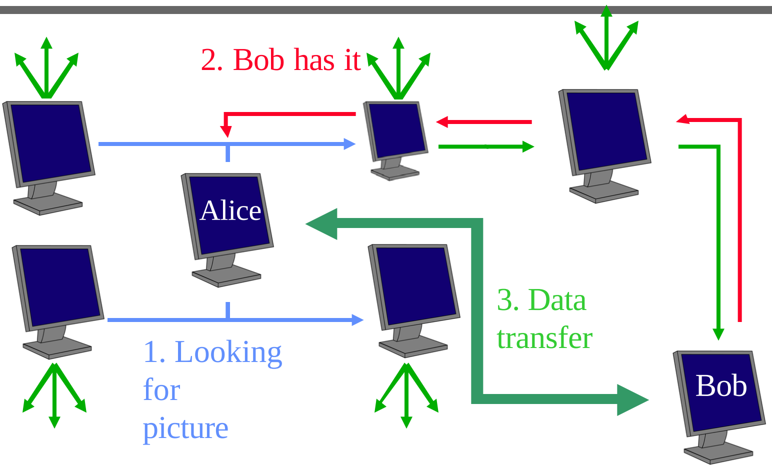
Original Web model (Client/Server)



31



Peer-to-Peer (P2P) web model



32

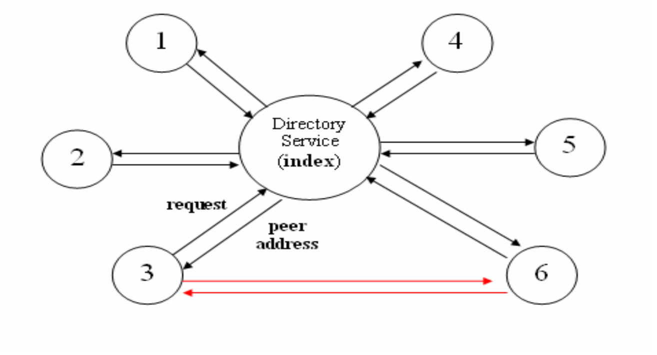
Applications of P2P model

- File sharing
  - Napster, Gnutella
- Utilization of spare computing power
  - Auctioning of machine cycles
  - SETI@home
- Better information search
- Highly robust distributed computing
  - No single point of failure

33

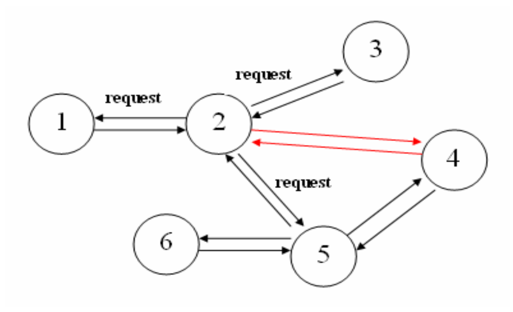
P2P Algorithms (1)

- Centralized directory model
  - peers connect to a central directory to publish what information they offer for sharing



P2P Algorithms (2)

- Flooded requests model
  - a peer floods the request to directly connected peers (no central directory) which they further flood their peers etc.
  - until the request is answered or a number of flooding steps are performed



Issues with P2P

- Everybody wants your files
- … but few people want to give you theirs!
  - Possible solution: Reputation mechanisms
- What is the business model?
  - Who can make money through P2P?

36