Web Client Languages
Summer 2017

Thomas A. Powell
tpowell@pint.com

http://classes.pint.com/cse134b

Housekeeping

- Class page http://classes.pint.com/cse134b
- Slack Channel - https://cse134summer2017.slack.com
  - Updated syllabus, notes, homework specifications, etc. will always be posted in these places
- History of this class
- My goals for this class
- Participation and Attendance
- Overview of this quarter
  - Syllabus, Groups, Slack, Tools, Etc.
- Standard academic issues (esp. cheating)
- Demo and Intro

Introduction

Important Truism

Being human is pretty constant, technology and society are variables though.

Does Technology change us?

- So let me get this straight…in a single generation (maybe two) due to the stimulus of the Internet
- we have generation wide environmental adaption that has changed humans to be
  - hyper multitasking
  - super friend making
  - crazy tech savvy
- We are Homo Internet Superior!

Seriously though getting the Web is hard

- The technology of the Web will turn out to be ~~easy~~ not as hard as dealing with the people
- We must always remember we build for people who are
  - Not always logical, rational, consistent, right, …
- We are people too, so it helps
  - We can embrace a study of people for gain or ignore it at our peril

So once again the Web/Internet really is…

- ….the real world filled with nonsense as well
  - It is still early in the evolution of the Internet
  - Yes some people don't believe and do dumb things and others take advantage
  - Certainly things are not always what they seem
  - Obviously society hasn't digested all the changes yet…but it will if history is any guide
- We have to be students of all this to navigate Web dev/design effectively since its use (or misuse) is quite powerful (see current events)

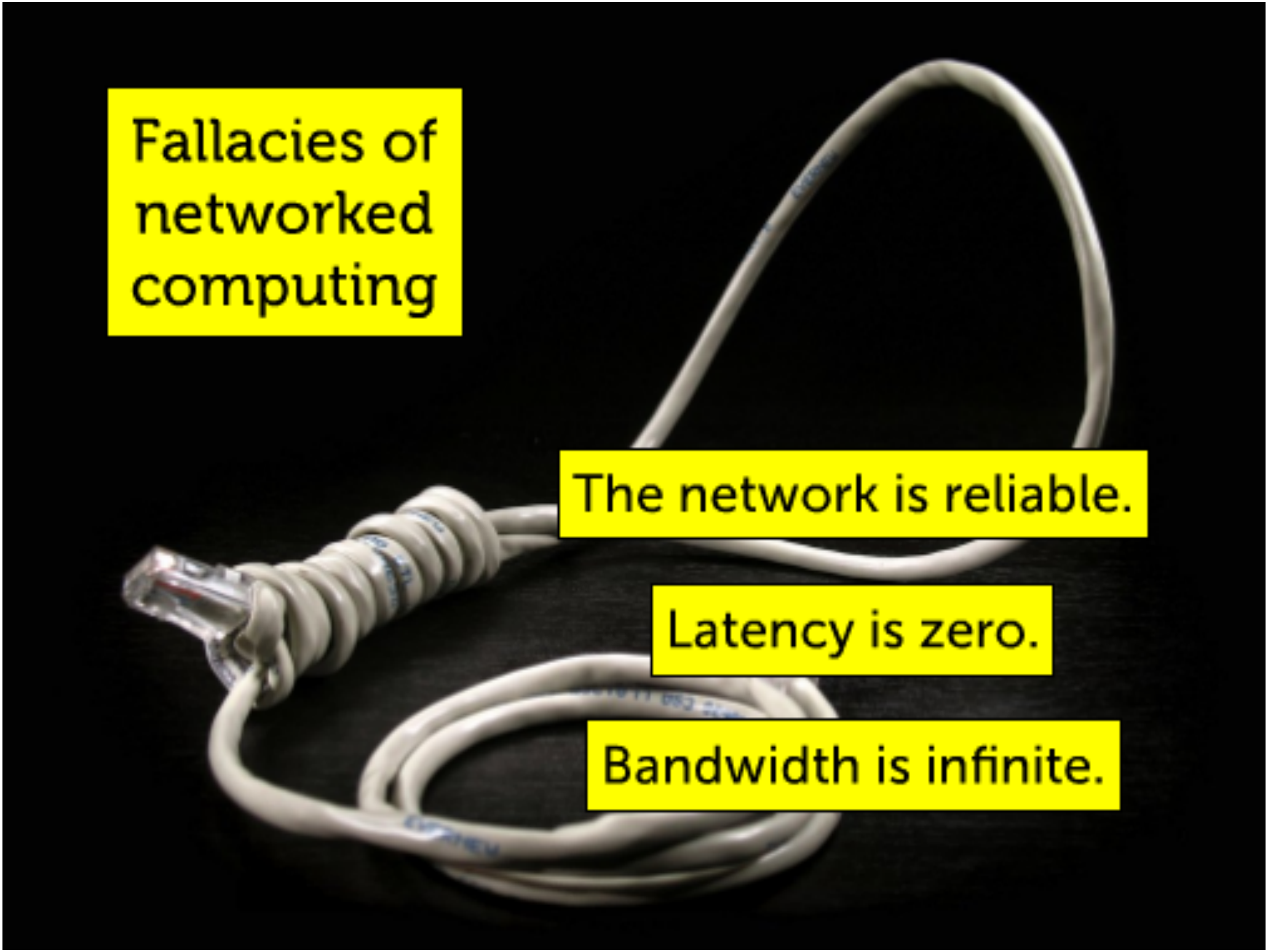
Things changed … but they haven't

- Browsers – since the dawn
- HTTP – since the dawn
- HTML – since the dawn
- Images – GIFS*, JPG, later PNG, WebP even
- CSS – since 1997
- JavaScript – since 1995 (though we clued in later)
- Web Servers – NCSA, Apache, IIS, Ngnix
- Server Side Code - since 1993/1994
  - CGI, PHP, Java, Ruby, JS, …
- DB driven sites – since 1993/4

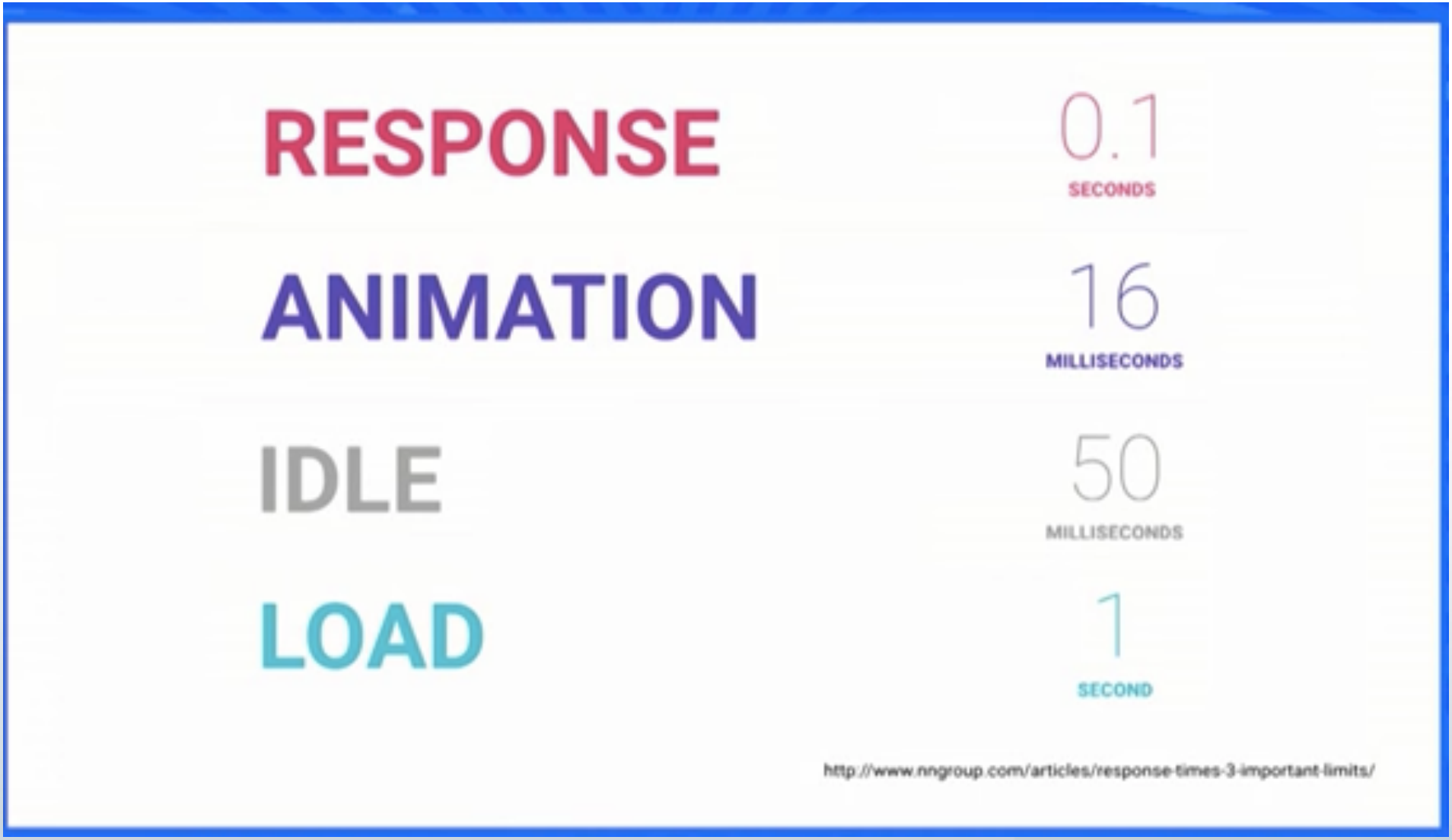
Web Development is…

- …just a form of client-server programming!
- Yes and a very hostile one…
  - Huge issues with an insecure and potentially unreliable network running software built often by those lacking formal large systems design being used by end users who we don't pay enough attention to and assume are just like us!
- Unfortunately lots of belief out there that this is different!


Douglas Crockford on browsers:
“The most hostile software development environment imaginable.”


Fallacies of networked computing
The network is reliable.
Latency is zero.
Bandwidth is infinite.

Unfortunately These Constraints are TOUGH



Modern Web Dev Practices

- Software Engineering without _____
- Release cycles
  - You mean when we change code?
- QA
  - Always doing or never doing?
  - Let the users tell us!

Client-Side … No problem

knowledge areas: 7

dimensions: x 4

platforms: x 3

browsers per platform: x 4

rendering modes: x 2

=672

Source: http://nate.koechley.com/talks/2008/at-media-2008-pro-frontend-engineering.pdf

So what we will learn in the class

- Of course tech
  - Browsers
  - HTML
  - CSS
  - JavaScript – ES5, ES6, Frameworks, Tools, Oh My!
  - HTTP & Network concerns
- Human stuff – usability
- Engineering thinking
- Some architecture ideas
- General medium concerns


Stuff you know
Stuff you know
you don't know
Stuff you don't know
you don't know

Some Core Ideas

Class Demo – How Browsers Work Overview

- Premise: Type www.ucsd.edu
- Actions
  - URL decode, Domain Look-up (DNS Resolution), Network Connection
  - HTTP Request
  - Server Decode Request (Return object … or not)
  - Data received (in base case here it is HTML)
  - Browser parses HTML, sees more requests to make for example CSS, JS, images, etc.
  - Repeat process until all objects finished
- Final page composed of HTML+CSS+Media Objects (images)+Executed JavaScript

Big Challenge

- What is Web design / development is different things to different people
- Depending on the person Web “D” includes many things such as:
  - Visual design
  - Programming
  - HTML
  - Navigation issues
  - Usability
  - Business issues (marketing, commerce, etc.)
- Depending on the project Web design actually may draw from any of these areas so it truly can be a very multidisciplinary field.

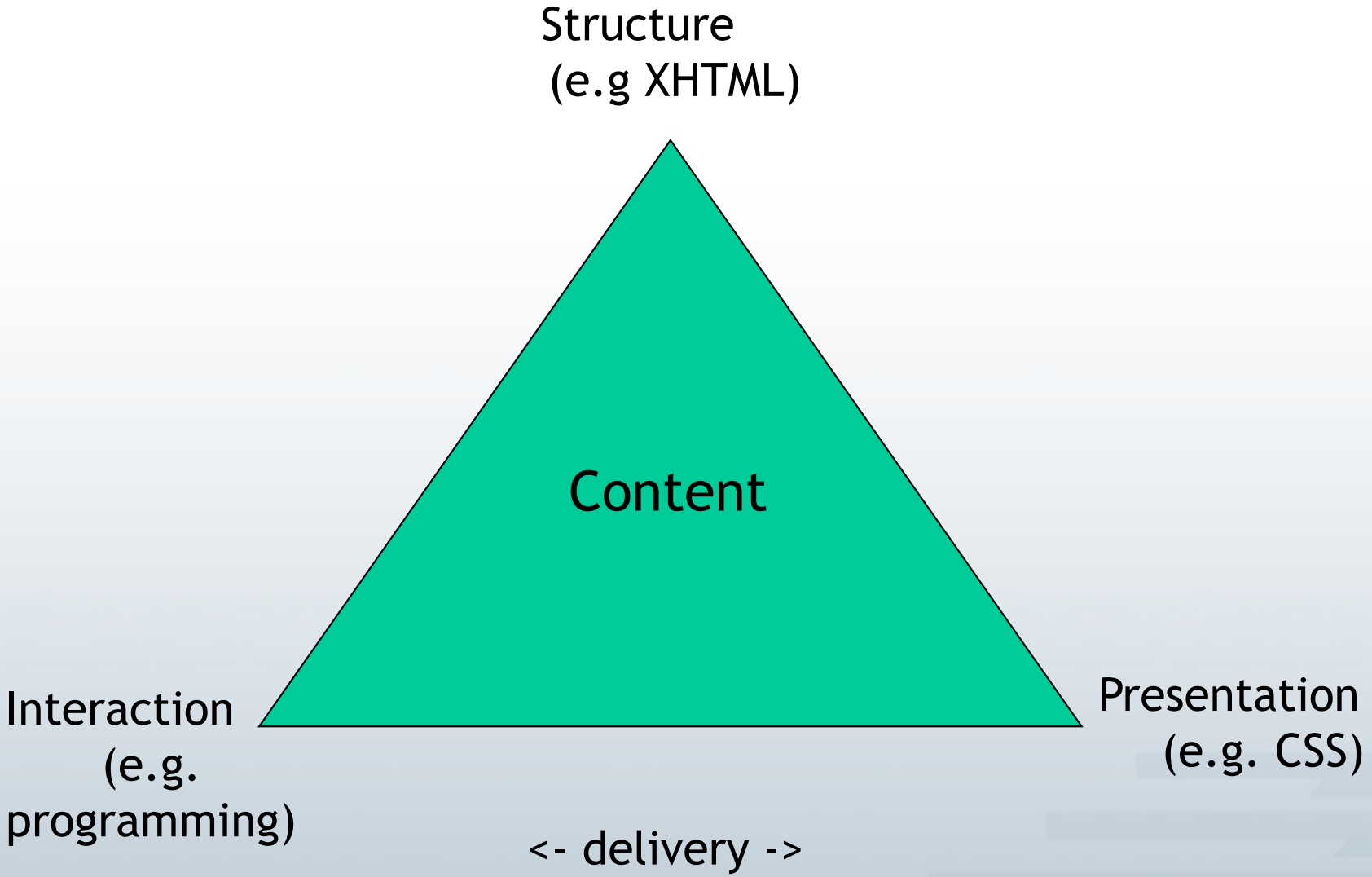
Best Sites?

- Given a "best practices" approach to Web design and development answer the following:
  - Question: What are the "best" sites you know on the Web? In other words who does it right? Who is really cleaning up with money, growth, name, execution, etc.
  - My answers _____, _____, _____, _____
  - Now what did those look like?

The 5 Pillars

1. Content
2. Structure
3. Technology (Implementation)
4. Delivery
5. Design

Another Way to View Web Development



2 Participants

1. Site Owners
   - Developers, people who pay for the site, etc.

2. Users

Rule: You must balance between what the users wants/needs are and the realities of the site owners' wants/needs

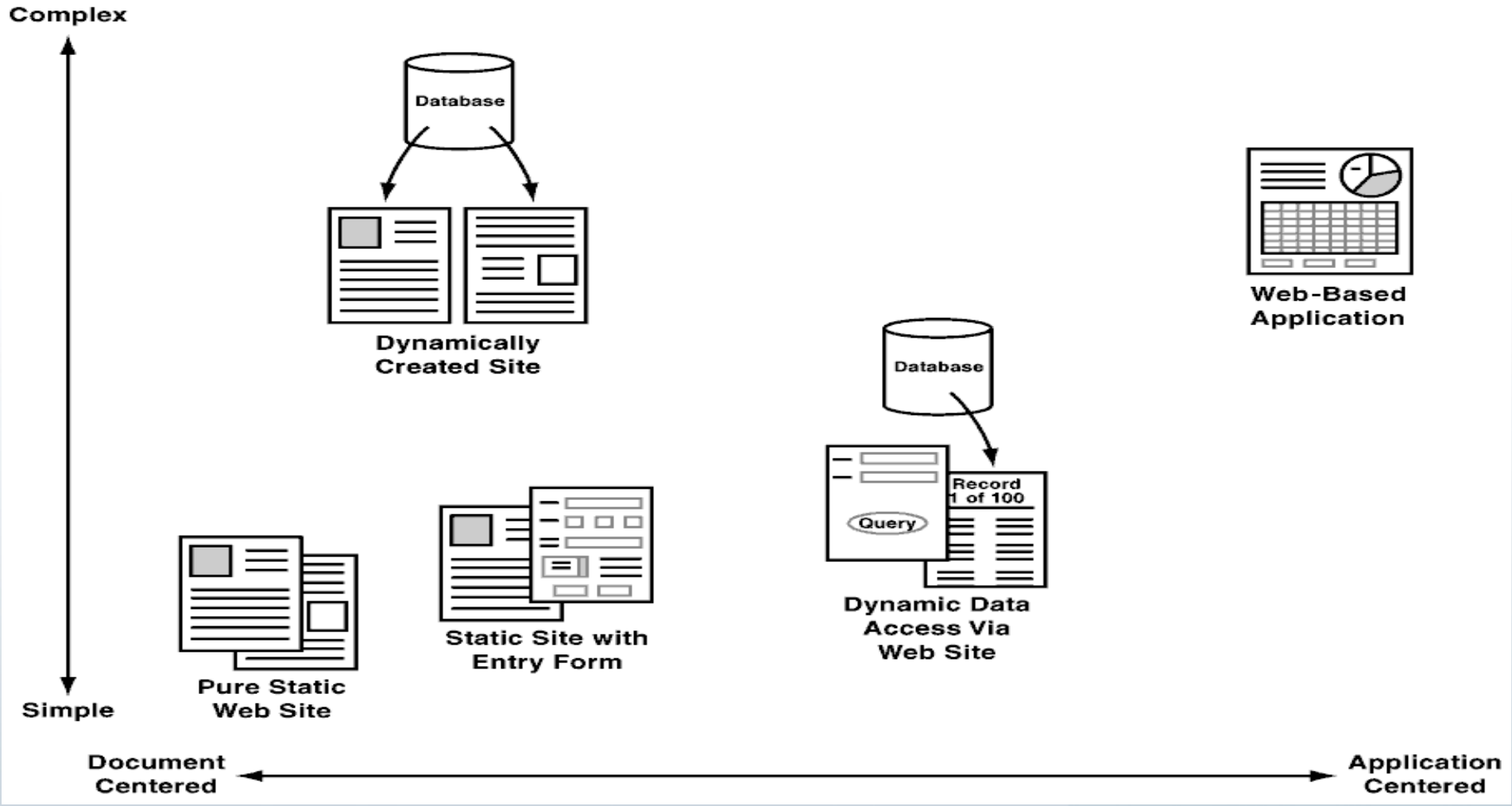
2 Participants – A Balance of power

- A balance of power
  - User in control – mistakes made?
  - Too much developer control – feel restrictive
- "Las Vegas" or "Disneyland" design
- *Seen it before the old Macintosh vs. command line argument*

Web Site Types

	Intranets	Extranets	Public
Info about Users	High	Medium	Low
Capacity Planning	Possible	Usually possible	Difficult to impossible
Bandwidth	High	Varies	Varies greatly
Ability to set technology	Yes	Sometimes	Rarely

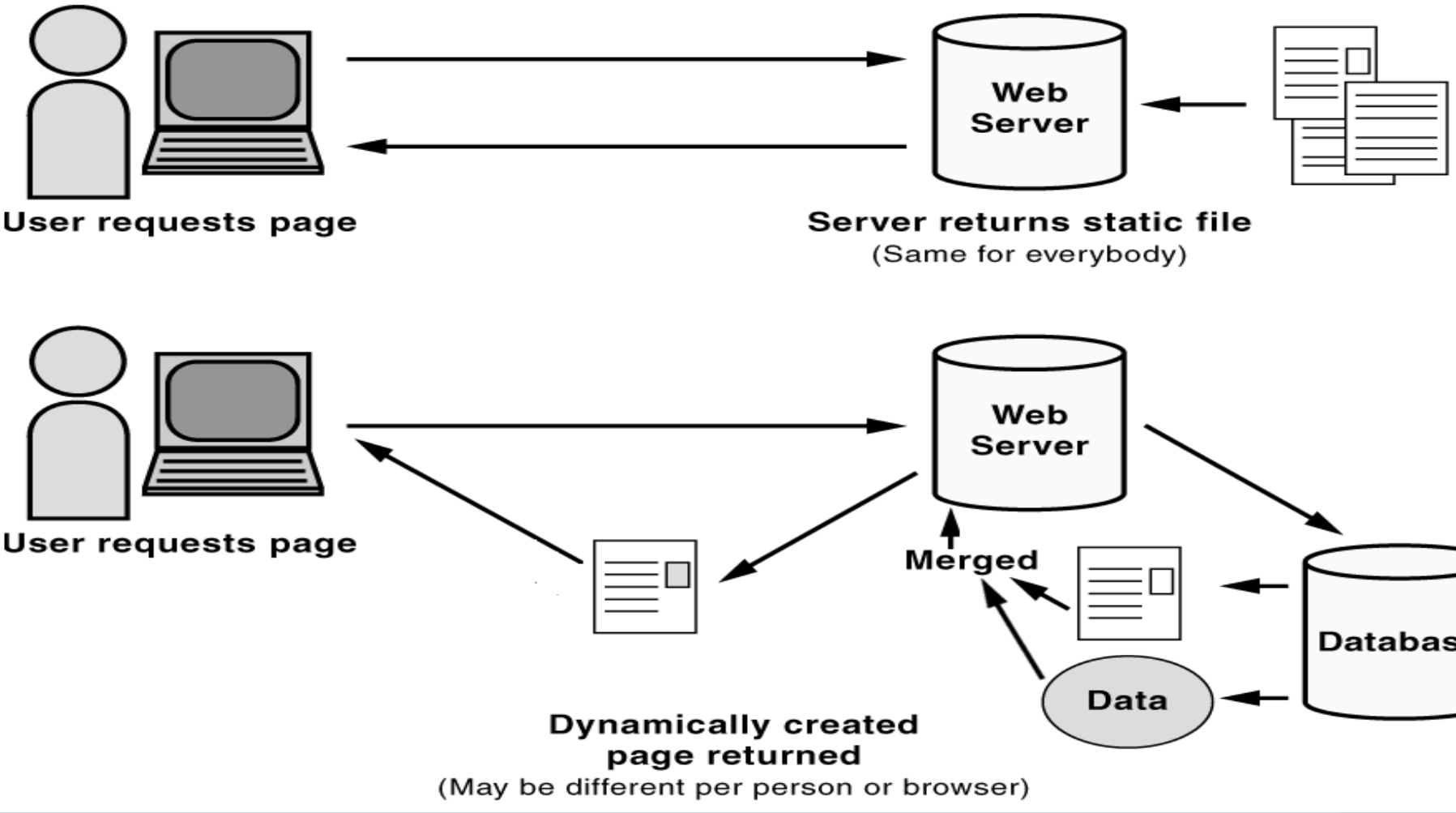
Simple Range – Outside of Purpose Even



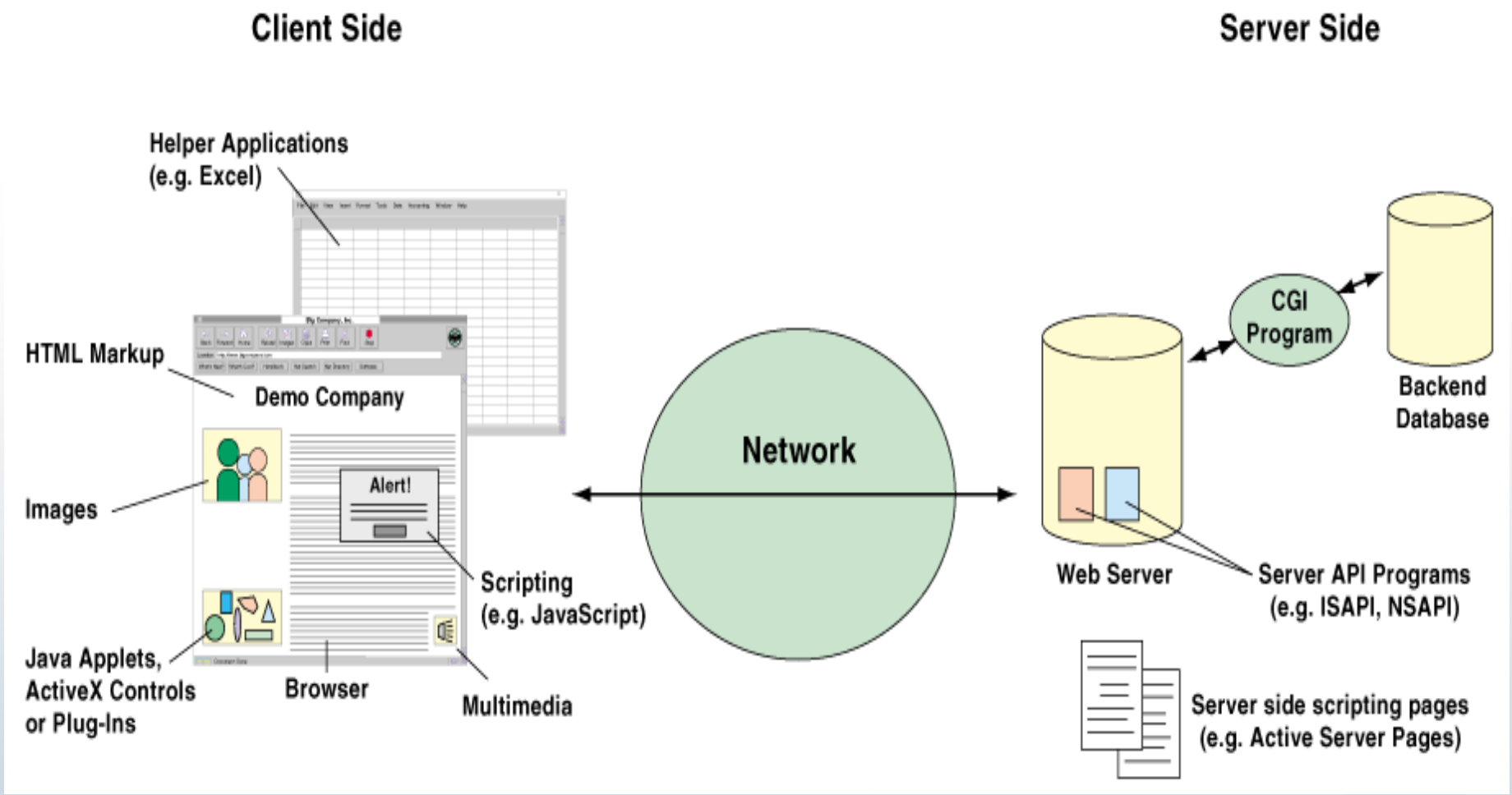
Site Types Another Take

- Static Sites
  - Most common
  - Pages don't change per visitor and are built to fit users generically
- Dynamic Sites
  - Built on the fly for users
  - Personalized sites fall into this category (myYahoo)
  - Usually stored in a database
  - Beware – The Static Dynamic
- Interactive sites
  - Those that allow the user to interact with content or site features in a significant fashion beyond simple selection

Dynamic Site Overview



The Medium of Development & Context



But but but…

- This last slide assumes browsers are making the requests and processing the response.

- Is this always the case? NO!
  - Native App (ObjC / Java) + Embedded Web View
  - Native App + Raw HTTP / JSON

  - Both mobile apps and desktop apps can work either way

- Should we count these types of apps as Web apps?

But but but Part 2

- The picture also assumes that we are talking to a traditional Web server like Apache, IIS, Ngnix, etc. and that software invokes and manages the execution of programs and retrieval of static assets
- Is this always the case? No!
- In the Node pattern in particular the program itself acts as the server
  - Not unique at all … been done since Perl
  - Admittedly while common this is crazy dangerous without proxying if you've been around the Web awhile

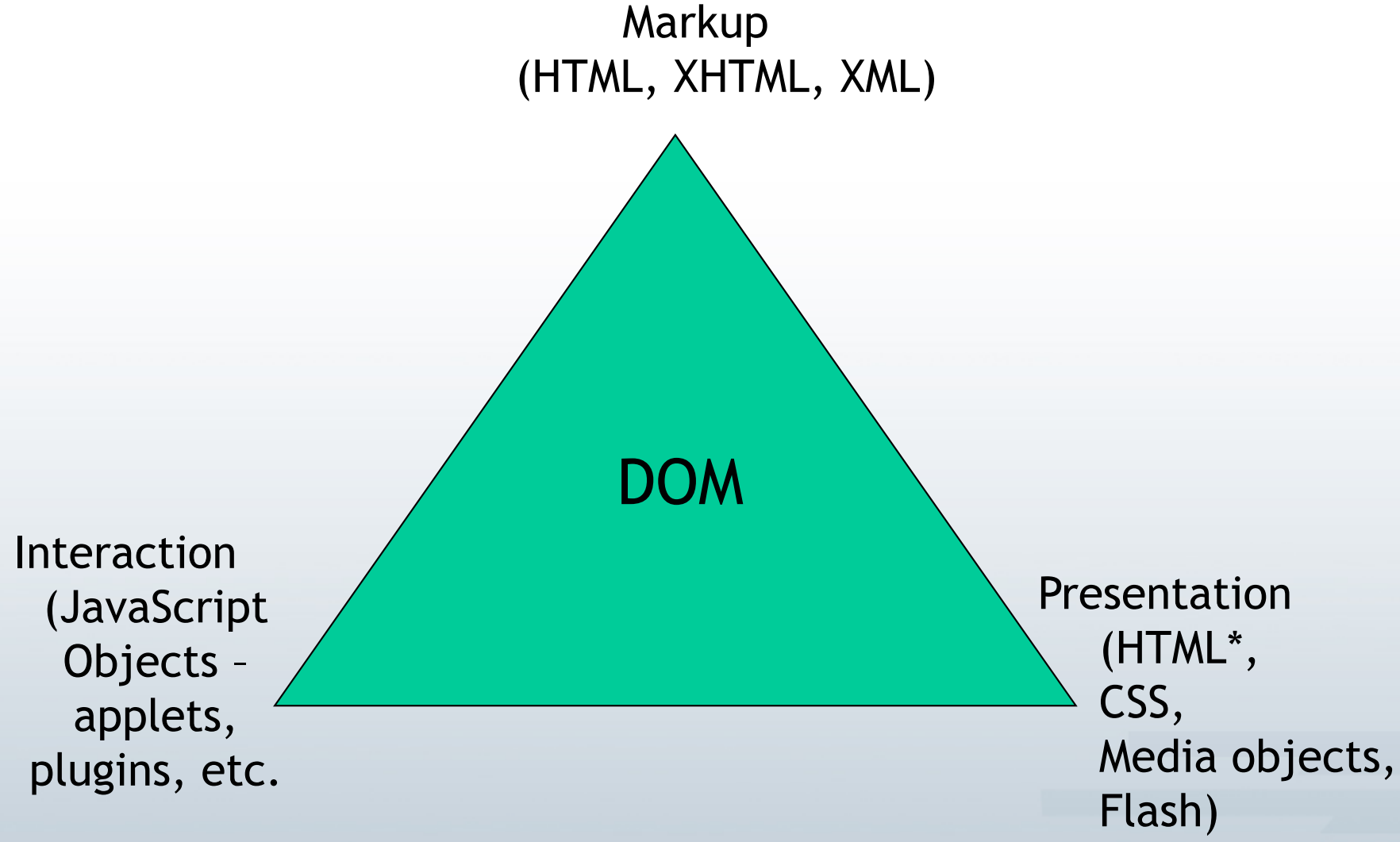
Web Programming Toolbox (Browser & Server)

Client-Side	Server-Side
Helper Applications	CGI Programs
Netscape Plugins (Deprecated)	Apache Modules
Google Native Client (NaCl and PNaCl)	Ngnix Module
Active X Controls (for Microsoft IE)	ISAPI Filters and Modules (for Microsoft IIS)
Java Applets	Java Servlets
JavaScript ~~VBScript~~	JSP ASP.NET PHP ColdFusion Classic ASP Ruby

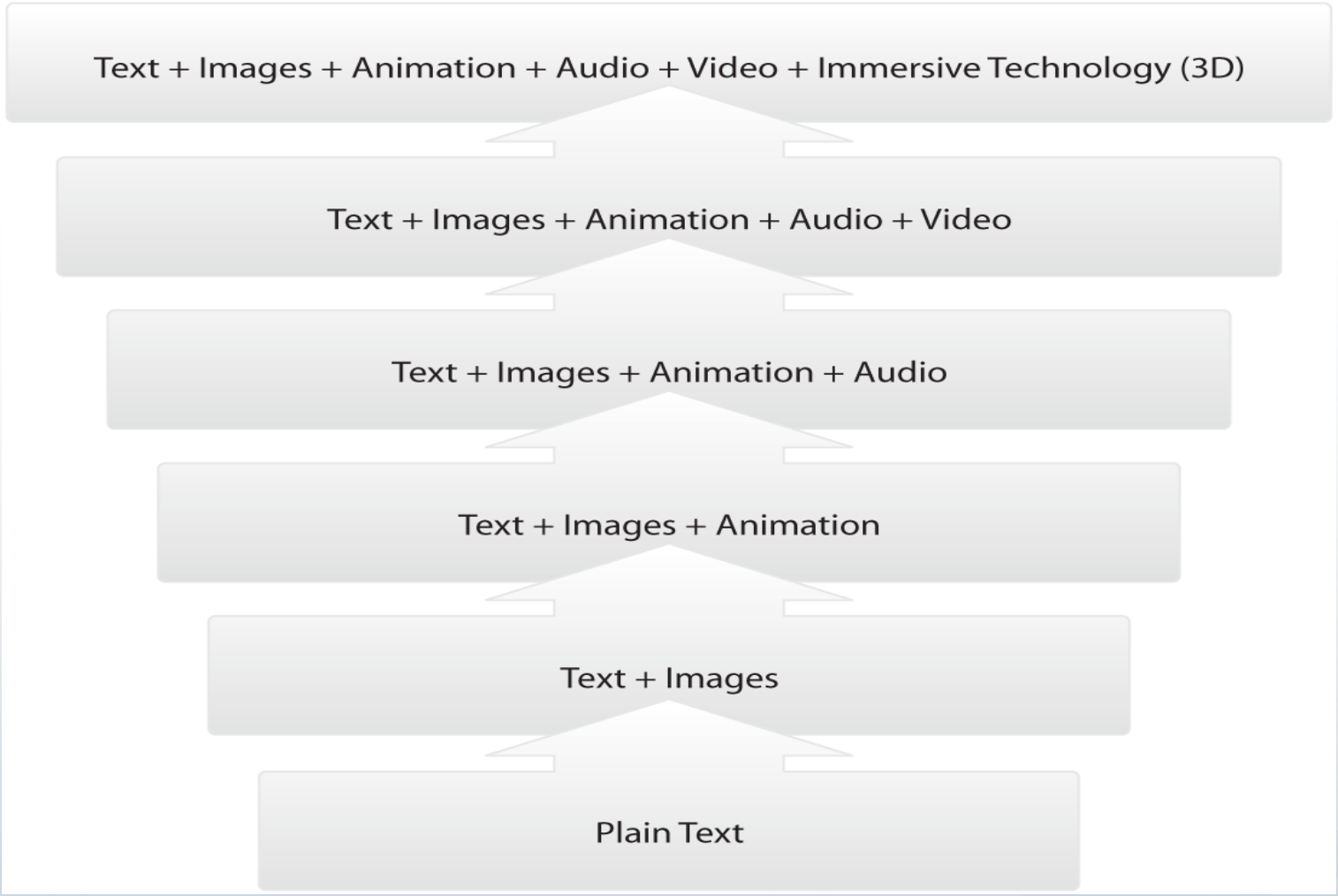
Recall the Class Lecture

- The idea of the spork should be clear
- All the boxes mean something and there are trade-offs within
  - Difficult to code / Ease vs. Speed of Execution
  - Coupling of code vs. Safety
  - Security vs Control
  - Etc. etc.
- The instances of tech in the boxes of the "toolbox" change over time the trade-offs and tensions are permeant. TL;DR – No single tool will do everything well

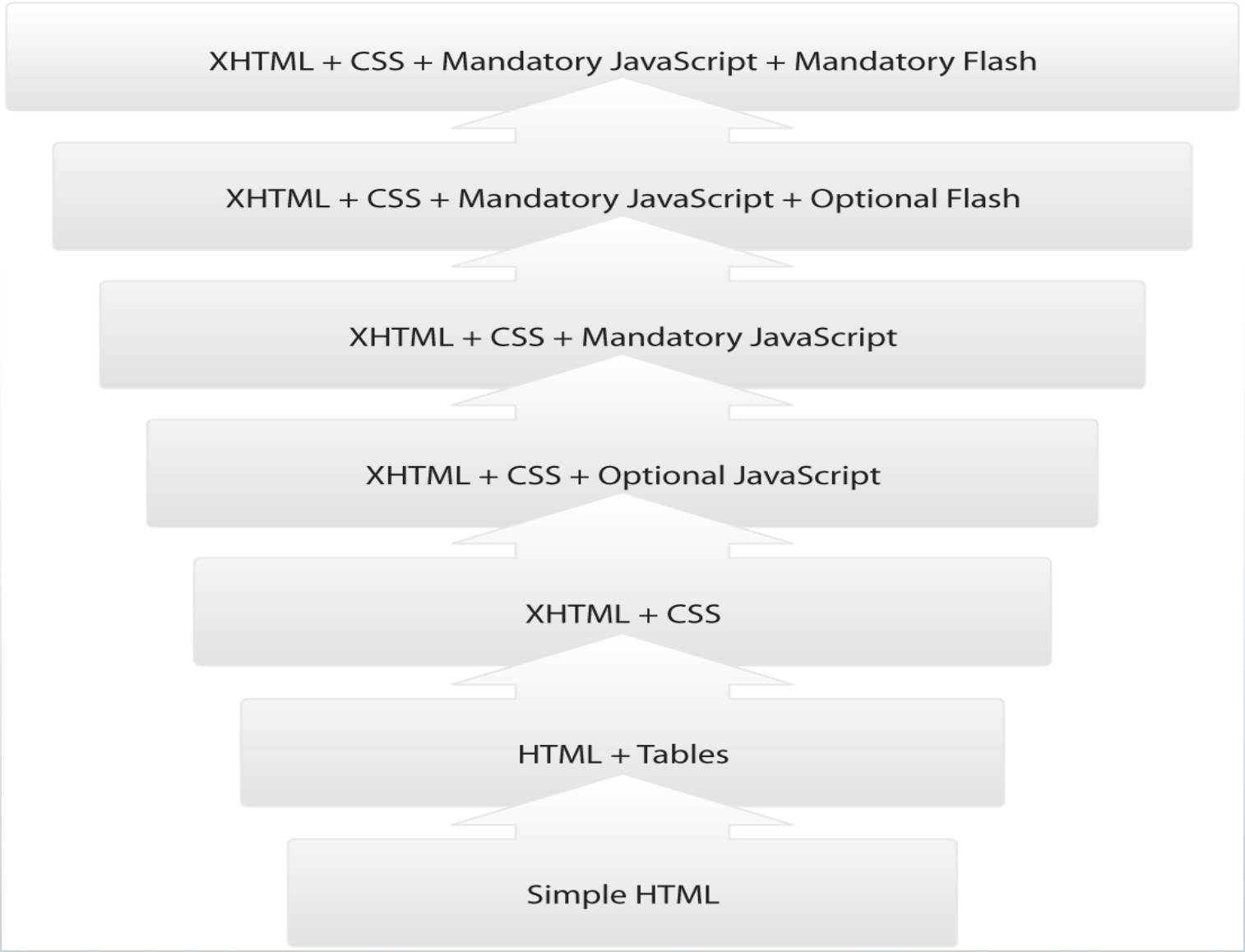
Simple View of Client Side Web Development



Range of Interface – Progressive Enhancement



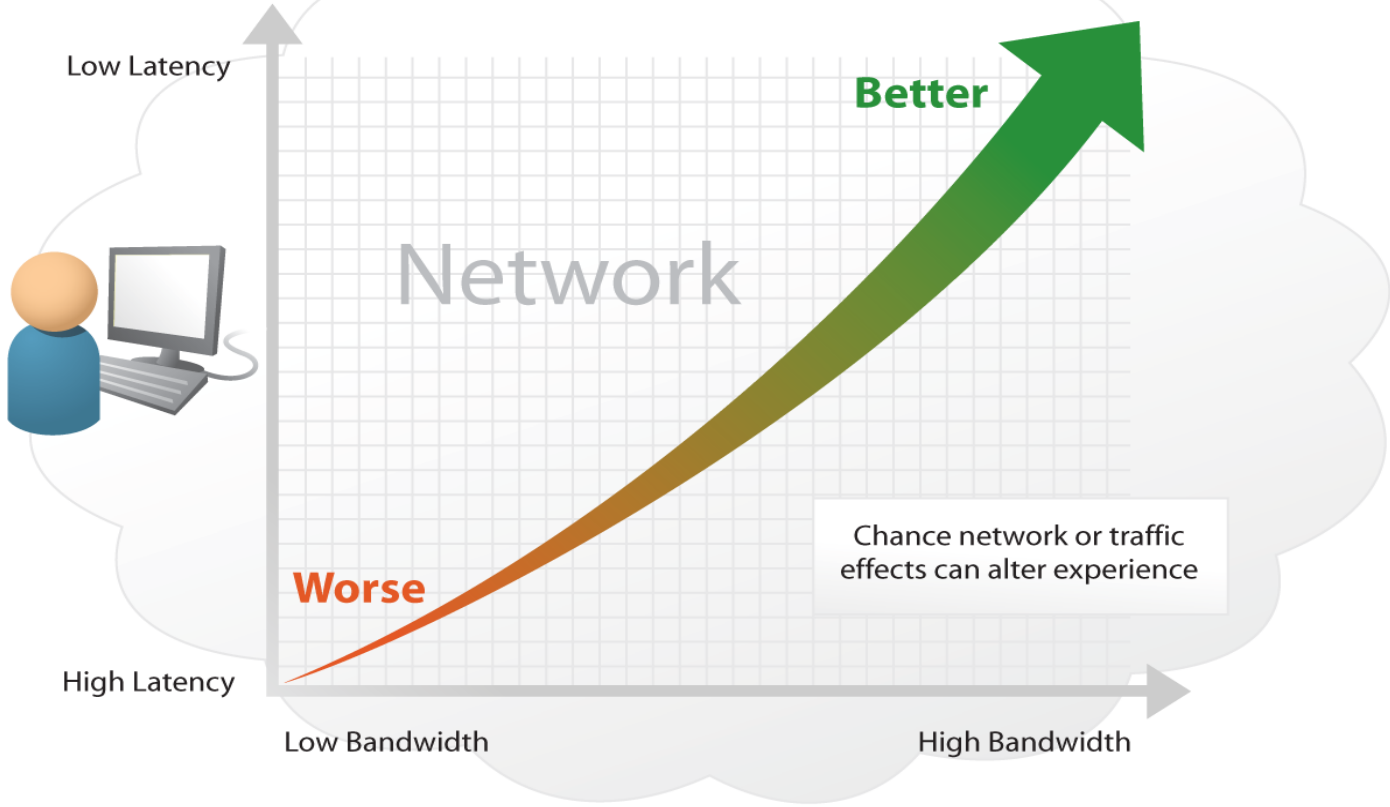
Range of Interface - Technology



Range of Connectivity

Offline
Disconnected

Online
Connected



Range of Interaction

Interface Style Choices

Traditional

Read, Understand, Click
(Simple links/press buttons)

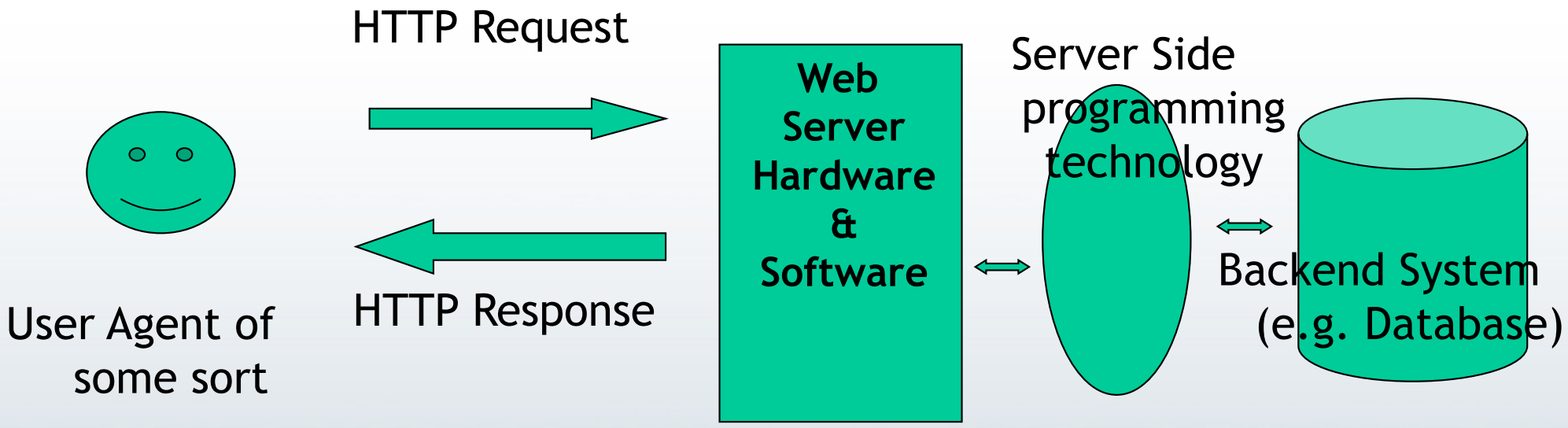
Direct Manipulation

Drag & Combine
(Select region, drag and drop, fill, etc..)

Architecture Generations Overview

- Gen 1 – Server Side Focused
- Gen 2 – Server Side + Optional Client Side JS
- Gen 3 – Client Side Focused "Ajax Generation"
- Gen 4 – "Mobile Generation" Part 1
  - Native apps (no offline)
  - Native apps with online vs JSON requests
  - Native apps with embedded Web views
  - Responsive Web apps
- Gen 5 – "Mobile Generation" Part 2
  - Progressive Web Applications (PWAs)
  - Streaming or more incremental native apps

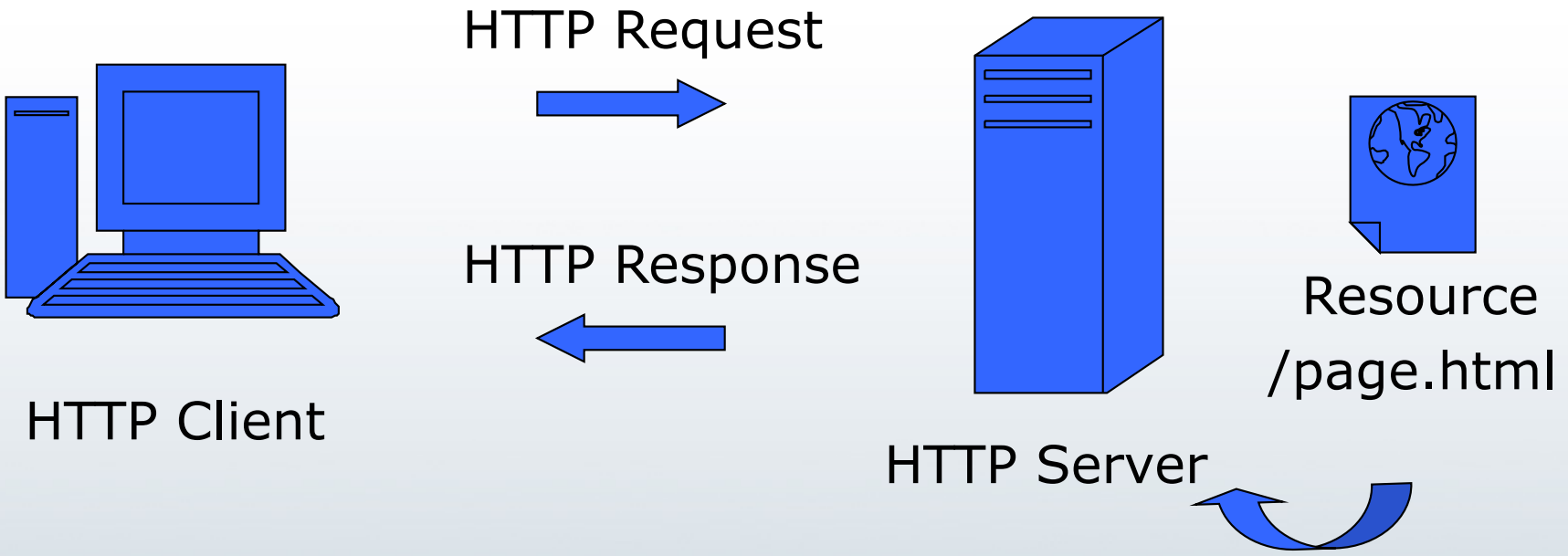
Simple View of Server-Side Web Development



Basic HTTP Request/Response Cycle

Asks for resource by its URL:

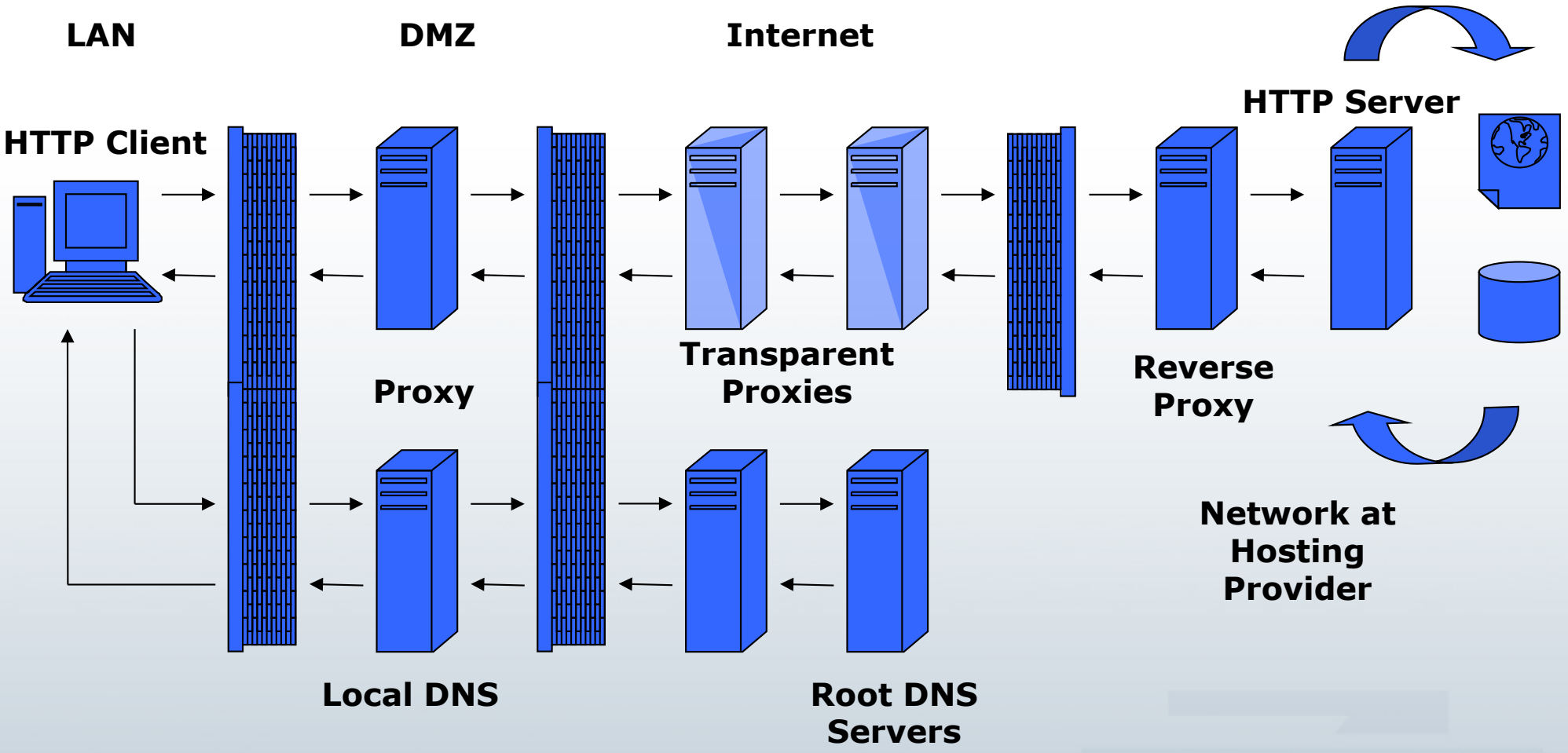
http://www.foo.com/page.html



Browser decodes MIME type and determines action

maps file extension .html to appropriate MIME type: text/html

BTW that was a gross over simplification



HTTP Request Example Shows Process

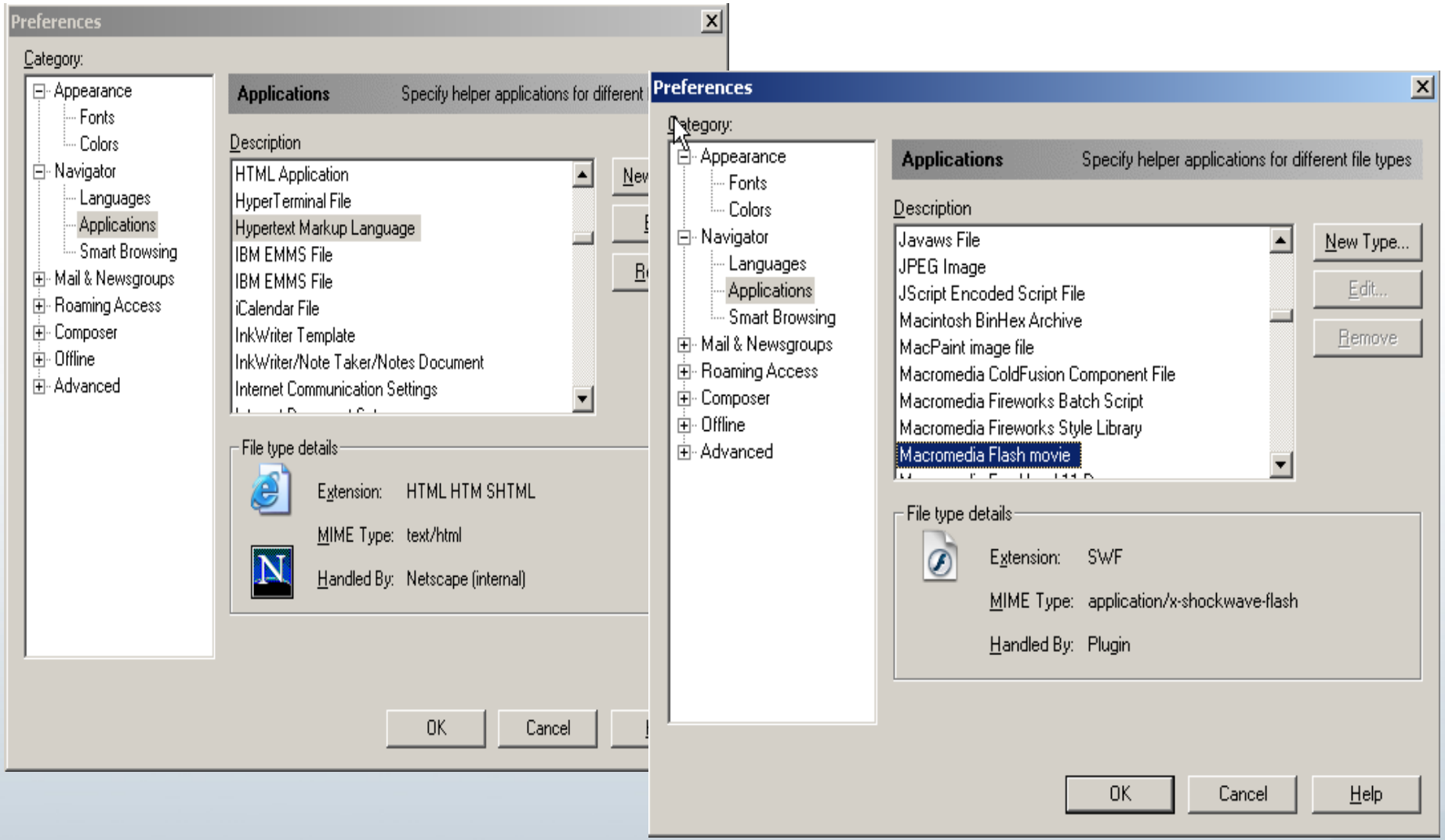
```
07/01/04 09:07:02 Browsing http://www.ucsd.edu
Fetching http://www.ucsd.edu/ ...
GET / HTTP/1.1
Host: www.ucsd.edu
Connection: close
User-Agent: Sam Spade 1.14

HTTP/1.1 200 OK
Date: Thu, 01 Jul 2004 16:07:00 GMT
Server: Apache/1.3.27 (Unix)
Last-Modified: Thu, 01 Jul 2004 16:01:00 GMT
ETag: "c992b-77df-40e4353c"
Accept-Ranges: bytes
Content-Length: 30687
Connection: close
Content-Type: text/html


<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="en">
<head>
<base href="http://www.ucsd.edu/">
<title>University of California, San Diego</title>
<meta name="generator" content="">
<meta name="author" content="UCSD Libraries, Information Technology Depar
<meta name="keywords" content="">
```

Request Headers

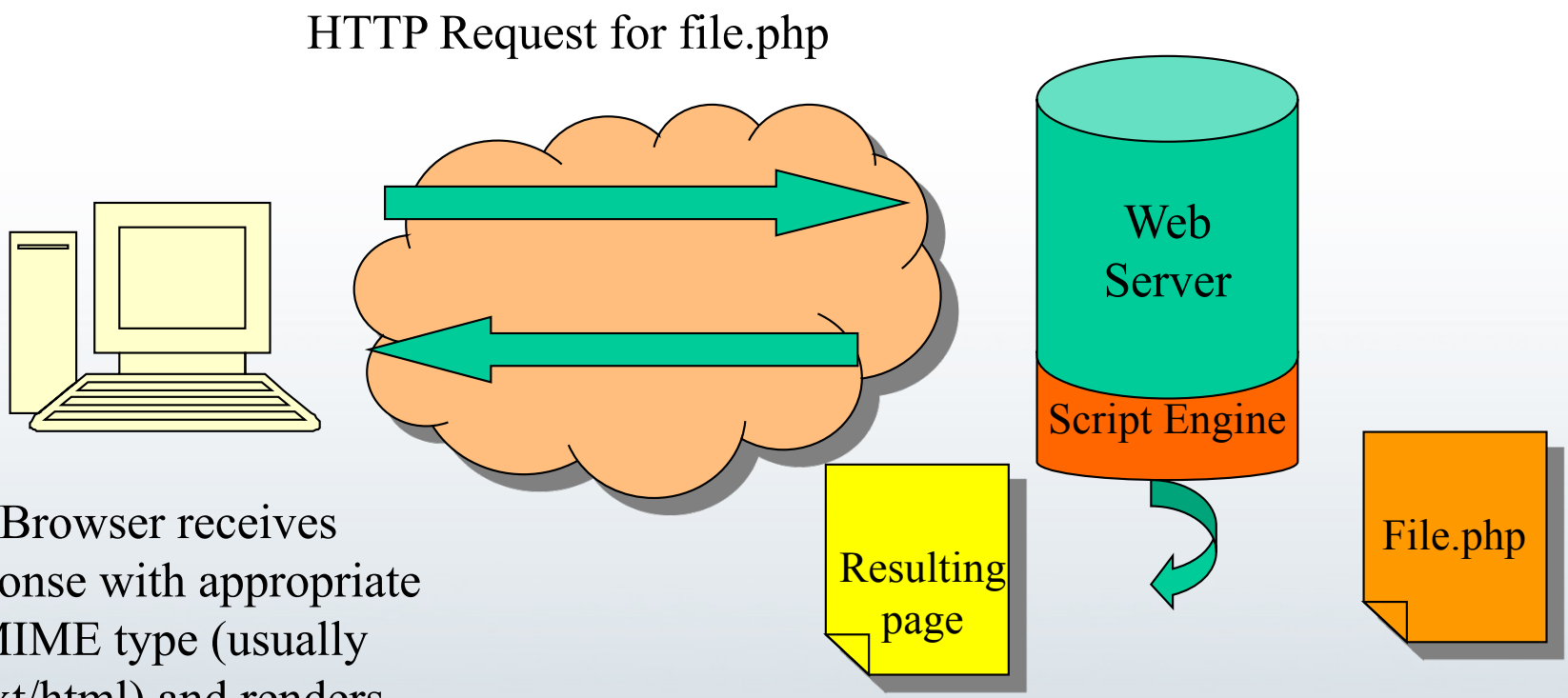
Response Headers

Response data

Browser Lookup for Action on Mime (or file type)



Example of file extensions and action



Q:If you view source what do you see?

A Web Design Definition

Web Design

"A multidisciplinary pursuit pertaining to the planning, and production of Web sites, including, but not limited to, technical development, information and its structure, visual design, and networked delivery."

A Web Development Definition

- The same thing but a programming emphasis?
- Q: What is the difference between Software Application and a Web Application
  - Container? Language? Delivery?
- Q: Now what is the difference between Software Engineering and Web Development

Form and Function

- Favorite Catchphrase: Form follows function!
- Rule: The visual form of a site should relate to its function
- Extreme examples to illustrate the point
  - Overly flash based site for your IRS tax form
  - All text driven move promotion site
- Interesting how design is not bottom-up today but top-down if this is true?

Execution: The Easy Part?

- Who does the user blame for mistakes?
  - Slow speed, no JavaScript, broken links, bad render, etc.
  - No limitations in most case for the user to leave – just click [no uninstall barrier]
- Reason for mistakes?
  - Misunderstanding the medium and its constraints
  - Lack of process
  - Lack of professional education
  - Lack of engineering style thinking

Conformity versus Innovation

- Users come to Web sites with history
  - 99% rule, big site effect, their operating system, etc.
- There is a rich history in computer interfaces
- There is an emerging history in Web design
- Rule: Appropriately respect GUI and Web interface conventions
- You may want to break the rules when you know the rules!

We Should Know the Details

- …but as we say we seem to get away with things
  - "If our app delivered to an end user crashes client-side do they curse out loud?" Did you hear them?
- The Web is different you see (or is it?)
- Browsers fix many of your problems
  - Markup, CSS, some network configuration with MIME types and even JavaScript!
  - Imagine a C compiler trying to guess what you mean!
- We really have to do our job right particularly as the distinction between software genres melts away, but what is our job…is there something more than knowing the tech that is maybe tougher?

User-Centered Design

User Focused Design

- UCD – the concept of designing something (in our case Web sites) always with the user and use in mind.

- Some important rules we cover:
  - Rule: YOU are NOT the USER
  - Rule: USERS are NOT DESIGNERS
  - Rule: Design for common, account for differences
  - Rule: Users are REAL PEOPLE

Yep it is all About end users

- Users declare us good or bad for better or worse
  - They often only see results and things above the water
  - We called this earlier "The Iceberg Model"

- Example: Speed is all important – you can never have it fast enough!
  - User don't care about bytes, they care about time so… implications?
    - Watch out for broadband fallacy
  - Is time and perception of slowness consistent?
  - Given the cycle of read, decide, click, wait, repeat can we play a trick? Yet to do so we have to have technical chops and an understanding of user and experience

Web Design Themes

- Generally the major themes behind modern Web design include:
  - Designer/Client needs versus user needs
  - The balance of form and function
  - The quality of execution
  - Interplay between convention and innovation

Usability

- **Definition: Usability is the extent to which a site can be used by a specified group of users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.**

- From Nielsen 5 ideas that determine the usability of a site
  - Learnability
  - Rememberability
  - Efficiency of use
  - Reliability in use
  - User satisfaction
- The fifth item may be the big one if you think carefully about the problem.

- Resource: www.useit.com

Usability

- **Rule: There is no absolute idea of what constitutes a usable site. Usability will vary as much as the users accessing the site.**

- **Rule: Usability depends on the medium of consumption.**
  Remember the onscreen, on paper, by audio, etc. problem.

- **Rule: Usability depends on the type of site as well as the user's familiarity with it.**

Usability

Rule: Usability and user satisfaction are directly related.

It seems obvious but user's tend to think that sites they are happy with are usable even when they might not be to someone who thinks the site isn't interesting. Conversely when a site is easy to use it tends to help users feeling about the site.

Web Users

Rule: Web browsers do not use sites, people do.

Be cautious of "Joe Enduser" with a cable modem. Even Yahoo! Shouldn't design for such an ambiguous individual.

- Truth: There are no generic people. Always try to envision a real person visiting your site.
- Let's meet some users – the video

User in Some Environment

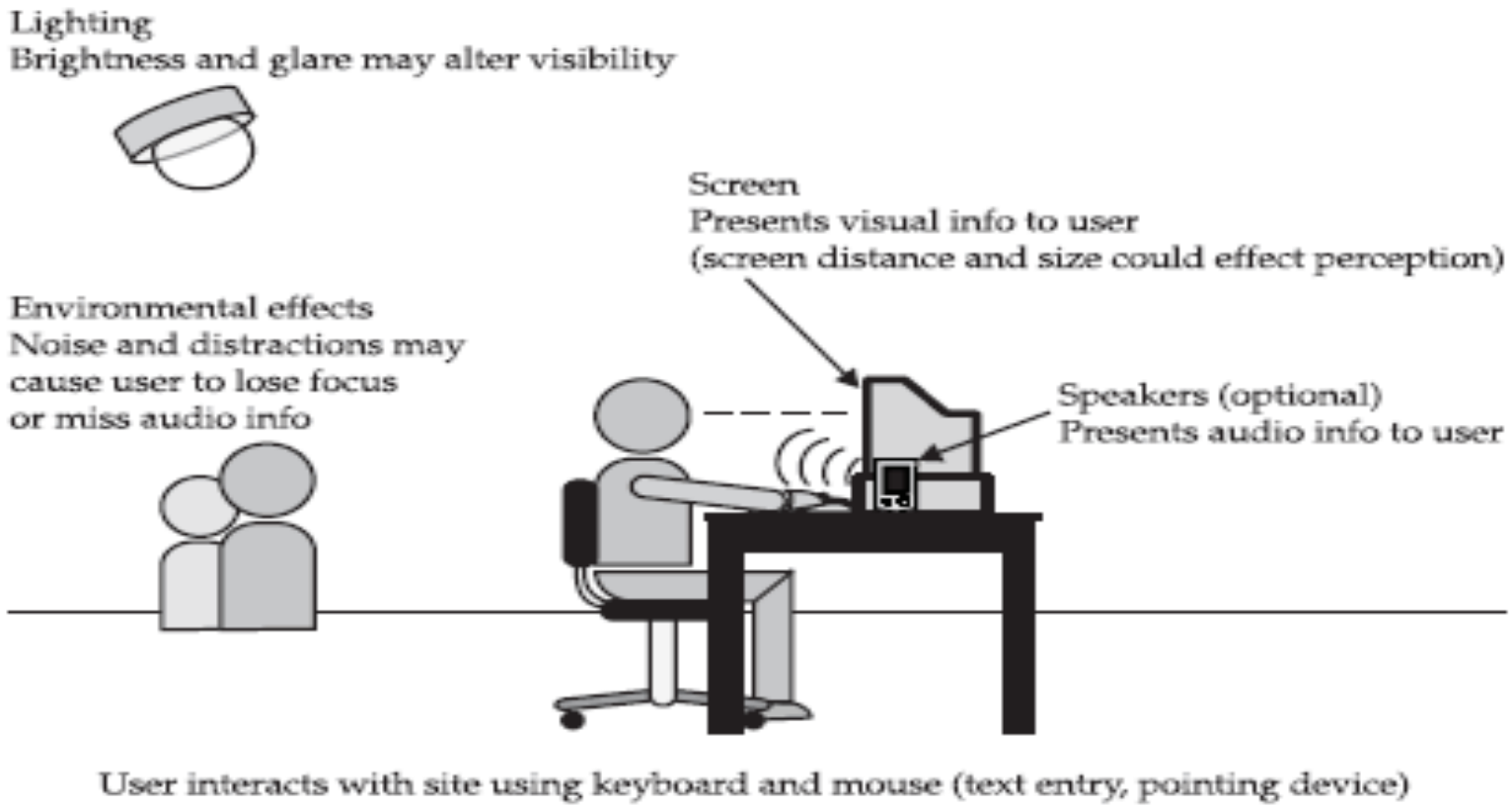
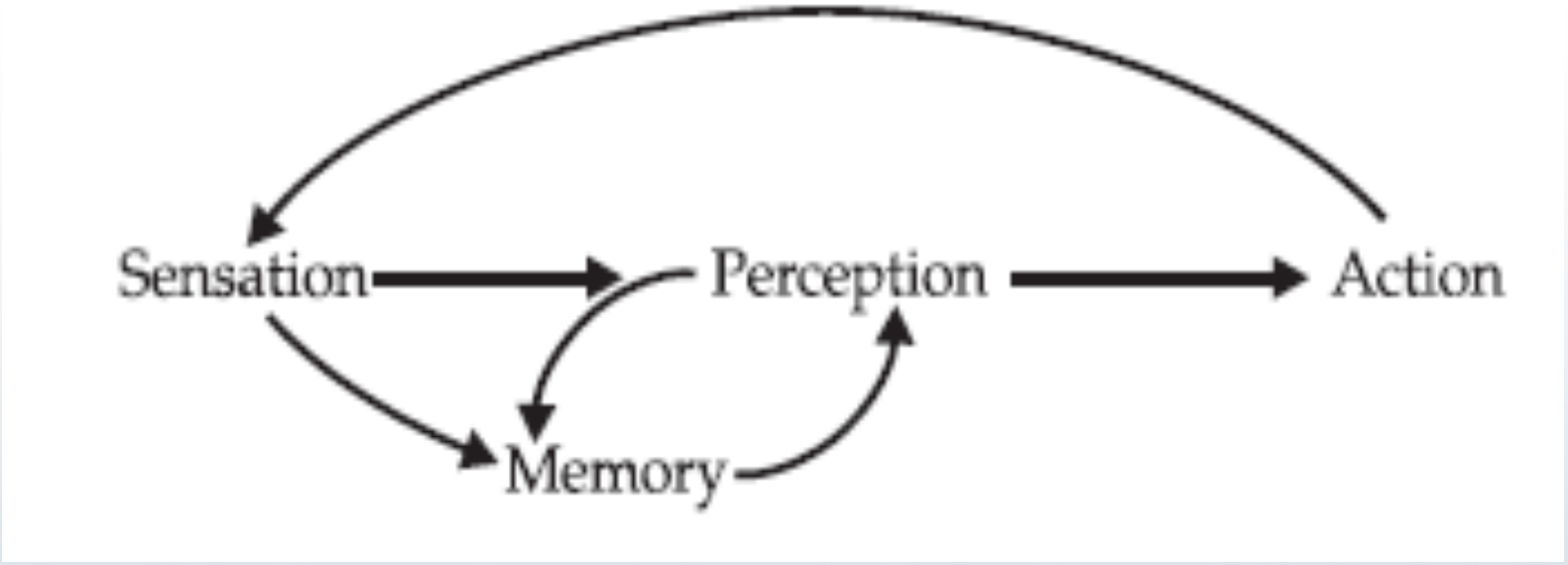


Figure 2-1. *Typical environment of user interacting with a site*

Simplified User Process



Common User Characteristics

- While there may not be a totally typical person people do have similar capabilities. Remember the main way that a user interacts with a Web site (keyboard, mouse, monitor).
  - Sight
  - Movement and Reaction time
  - Memory
  - Other possibilities may include hearing
- Be careful of assuming that even with similar capabilities that there is not a range.
  - Example: Vision varies from completely blind to excellent vision

Thinking About Human Capabilities First

- What's wrong with this idea?

- What's right about it?

Vision

- The main concern is can the user properly read the display
- Contrast
  - Dark on dark
  - Light on light
  - In short avoid any combinations of similar hue or lightness.

- Yellow and black provide greatest contrast, white and black is close.

  Notice that few huge sites really use complex background tiles or colors.

- Make sure that colors that are meant to distinguish items like links are significantly different in two ways, such as hue and lightness

Vision

- Size
  - Too small
  - Too big
  - Be careful sizing text is not consistent across systems
- Layout
  - Too close together
  - User has difficult time telling the difference between items
- Always remember to make things noticeably different. Slight differences will not pass our thresholds.
  - The “Fuzzy Eye Test”

Vision

- Color
  - Can the user even perceive it?
    - How it is perceived
    - Taste
    - Culture
  - Not absolute – color blindness, monitor age, gamma, etc.

Memory

- **Rule: Users try to maximize gain and minimize work. (The lazy person rule)**

  Memory requires work!

- **Rule: Recognition is easier than recall, so don't force users to memorize information.**

- This is pervasive in Web design. Even consider link color.
  - If you make visited links the same color or style as unvisited ones you are making the user memorize where they have been.
- Visual memory tends to be better than other forms of memory.
  - Consider how to make the home page visually different than others.

Memory

- **Short term memory wise, users are able to remember around 7+/- 2 items. This is important to consider for button clusters.**
  - Do not go crazy with this idea and limit your site to only this many buttons per page as this will result in excessive clicking. Users are not interested in more than 3 clicks to reach content in most cases.
  - Aim for memorization in the three range esp. for sequences

Reaction Time

- **0.1 second**
  - This is the limit for having the user feel that the system is reacting instantaneously, don't have to keep them informed in even the slightest way

- **1.0 second**
  - This is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.

Reaction Time

- **10 seconds**
  - This is limit for keeping the user's attention focused on the dialogue.
- **> 10 seconds**
  - Users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Think about status bars on installers
- Be careful on the Web, people may be patient more because it is new to them, over time this may not be the case.
- Consider that these are worst case times between responses or tasks. What does the user think a task is? A page load or an item being downloaded.

Response Reaction Time

- How long will people wait for sites?
  - 10 seconds, 100 seconds, etc.
- Is it consistent?
  - Users tend to be more patient with something they are unfamiliar with or that is a novelty
  - How about if people are very familiar with something?

- **Rule: The amount of time a user will wait is proportional to the expected payoff.**

Keeping Users Busy

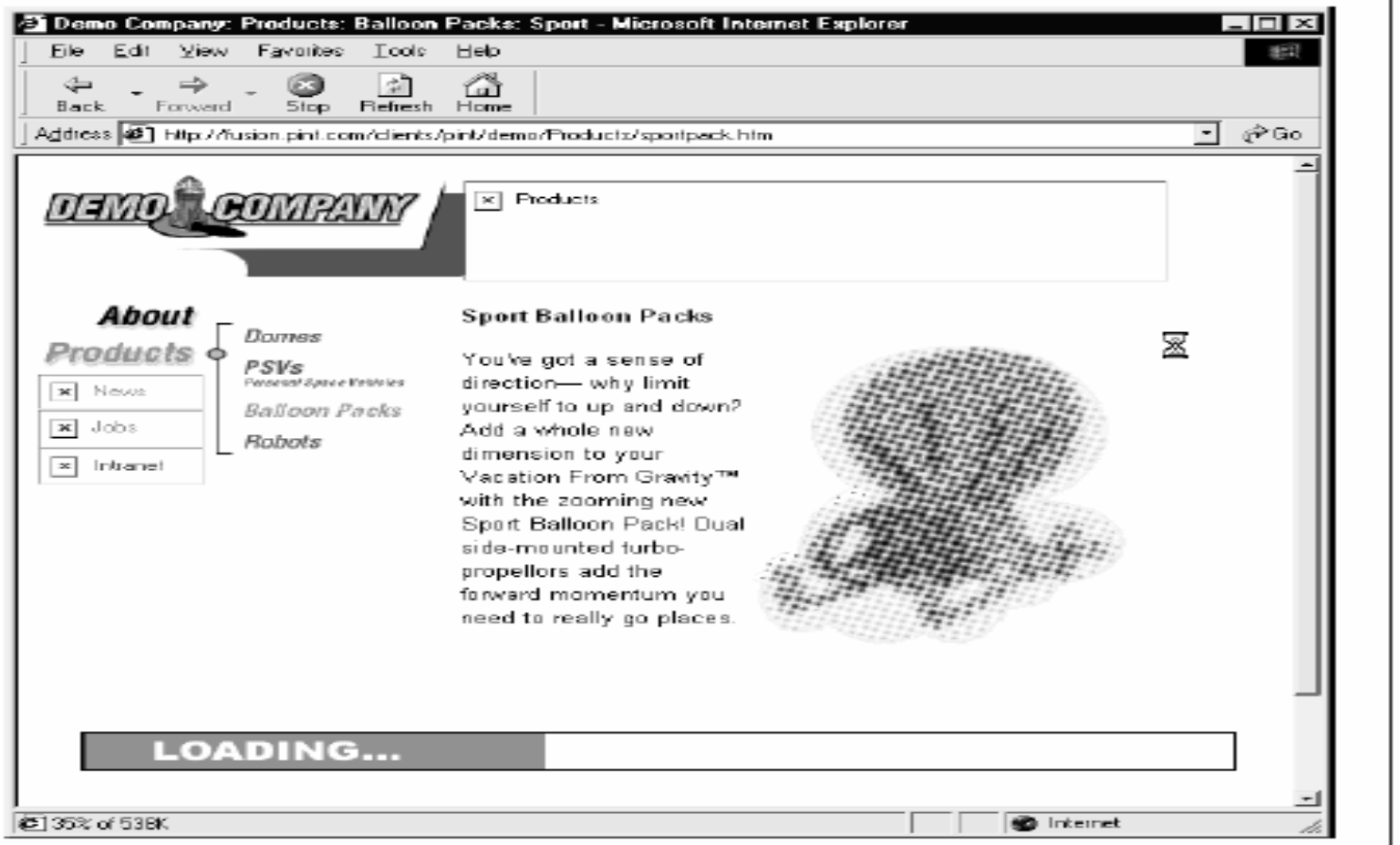


Figure 2-2. *Browser and site feedback shows progress of download.*

Time Matters

- Rule: Time matters more to a user than bytes delivered
  - Unless they pay by the byte!
  - Bytes delivered may be troubling to hosters
  - Number of requests may be the killer not byte count
- Take advantage of thinking time to load things
- Perception of load time is very important
- Beware of screen refreshes and loading perception

Movement Capabilities

- Consider mouse and keyboard travel always in sites.
  - Try to optimize keyboard access for all pages in a site.
    - Esp. form fill-outs (access keys, tabing, returns)
  - Minimize mouse travel distance between successive choices
  - Minimize the distance between the primary hover zone and the back button.
    - The top-left effect
- Consider Fitt's Law and make clickable regions large enough and close enough to be clicked accurately.

Dealing with Stimulus and Sensory Issues

- Be aware that users react to sensory stimulus in fairly similar ways
  - Thresholds
    - People's ability to notice things that are just barely different.
  - Cocktail party effect
    - The ability to focus on a single conversation within a large room of people might relate to a person's ability to focus on a single content item out of a page with many distracting elements.
  - Sensory adaptation
    - Users will start to ignore things which initially grab their attention.
    - Consider continuous animation of banners or other attention grabbers.
- *The idea of banner blindness probably results partially from these ideas*

The User's World

- Always remember that you need to bring a site into the user's world not the other way around

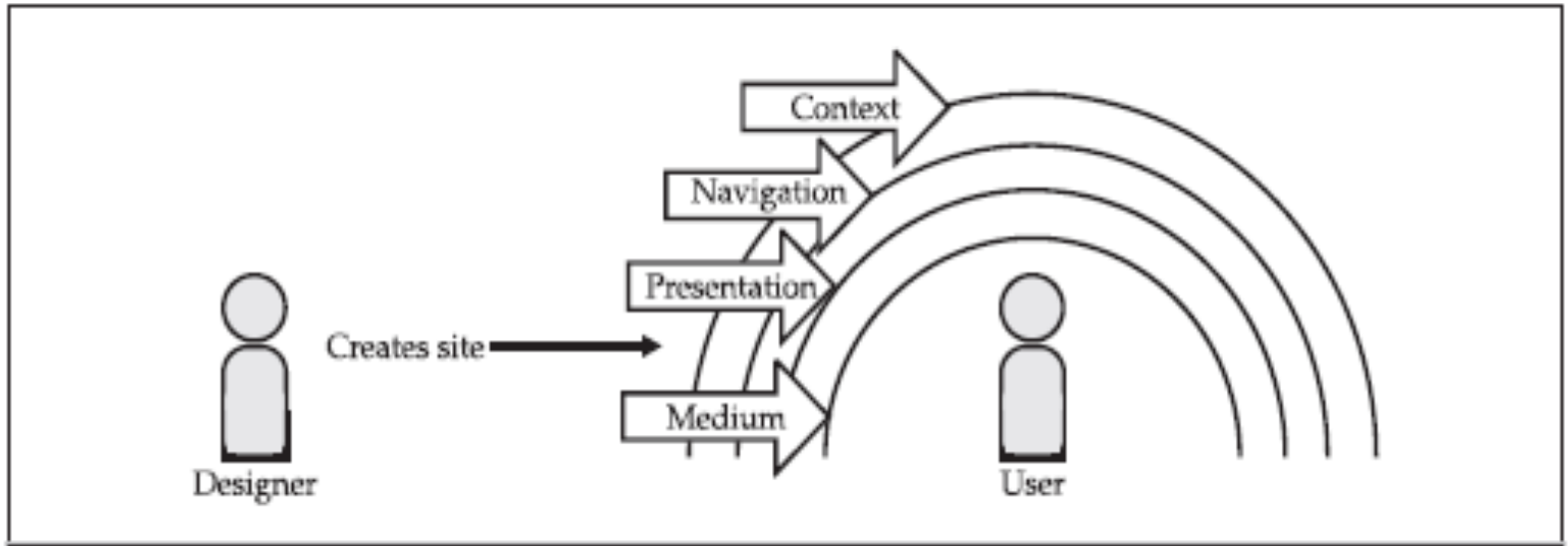


Figure 2-4. *The user's universe*

The User's Environment

- Home (living room, bedroom)
- Office (office, projection)
- Outside (café, car, mobile, etc.)

- Account for the environment of consumption if possible.

General User Types

- In general there are three types of users
  - Novices
    - May not understand Web conventions
    - Will not understand site organization
    - May want extra help like a tour, help system, or step-by-step wizard style interfaces
  - Infrequent intermediates
    - Know basic Web conventions well so they will know how to use various site features but may not know where things are
  - Power users
    - Like to use simple URLs, pull downs, search or site map for direct access
    - Want to customize things to suite their own tastes
      - Bandwidth, images, content shown, etc.
    - May know navigation very well

User Types

- Intermediate users are the most common for public Web sites.
  - These users will spend most of their time elsewhere but will understand basic Web conventions so you better not break them.
  - Other users may be important depending on your audience mix.
- Aim to build an adaptive Web sites that meets the requires of novices, intermediates and advanced user, otherwise aim for intermediate.
- Users are real people so beyond these basic ideas they may be very different.

- Tip: There will always be user's who don't like or get a site, no matter how good it is.

- Rule: User's bring past experiences with the world, software, and the Web to your site. Make sure you meet their expectations.

Real Users

- Understanding YOUR audience is key
- Audience varies from site to site
- Profiling users
  - Take a guess
  - Make up character names
  - Try to get in their shoes
- Interviewing real users
  - Interview Rule: Be quiet!
  - Figure out what kind of tasks they do.
  - Avoid focus groups initially and try one on one
  - Use focus groups to verify ideas
- Watch users
  - Look at the tasks they perform
  - Try to watch them without their knowledge