Web Architectural Philosophy

Outline

- History of the Web
- Alternative Web Architectures
- REST vs Web Services

Origins of Hypertext

- Vannevar Bush
- Coordinated WWII scientific effort

- Developed “social contract for science”
  - Federal government funds universities
  - Universities do basic research
  - Research helps bolster economy and national defense
- This became the basis for the National Science Foundation (NSF), EU Funding
  - Influenced DARPA too

Origins of Hypertext

- Famous 1945 article “As We May Think”
  - http://www.theatlantic.com/doc/194507/bush
  - Identified fundamental problem: too much information

Quotes from As We May Think

"[W]e are being bogged down today as specialization extends. The investigator is staggered by the findings and conclusions of thousands of other workers—conclusions which he cannot find time to grasp, much less to remember, as they appear."

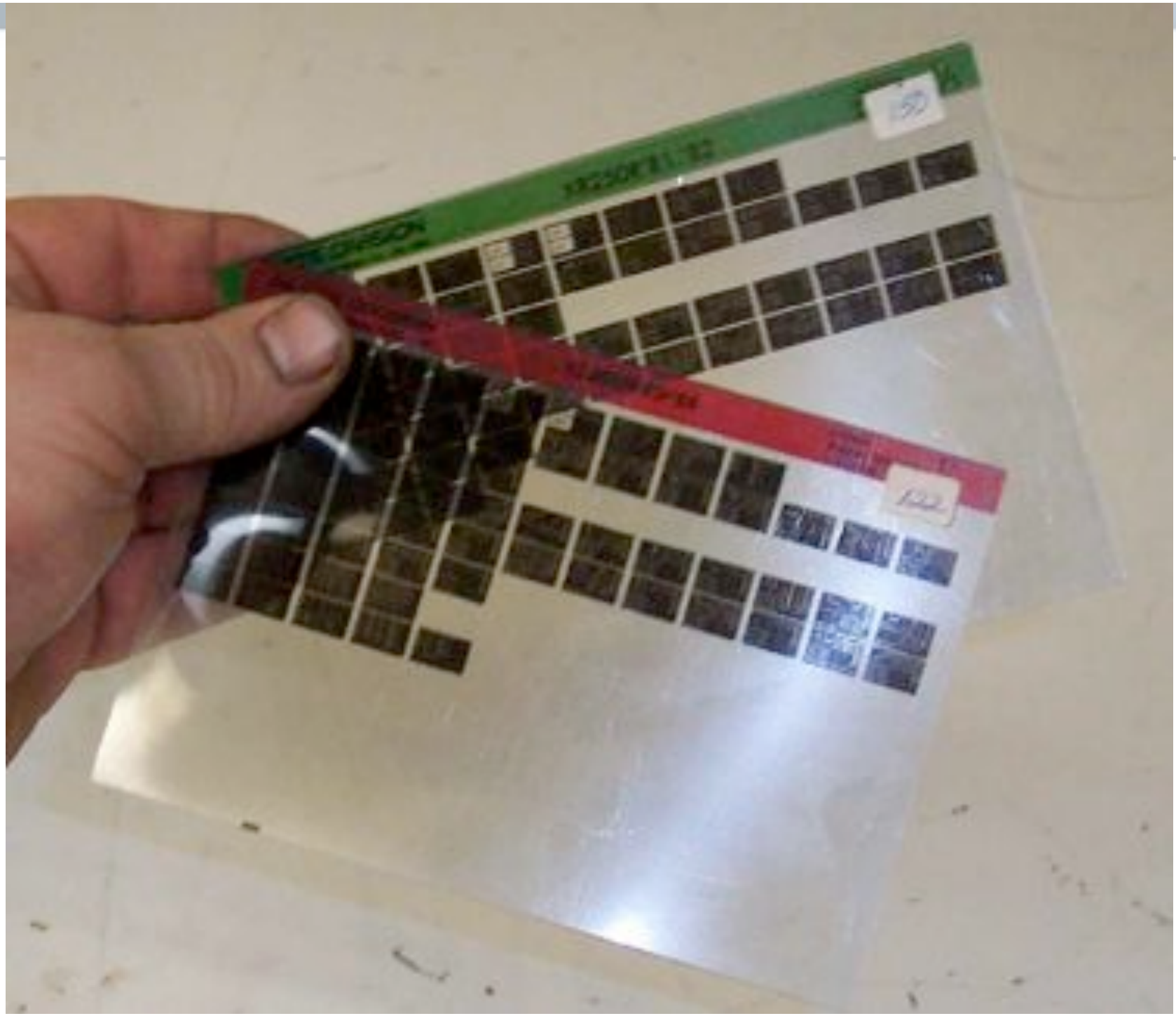
Quotes from As We May Think

"[P]ublication has been extended far beyond our present ability to make real use of the record. The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships."

Origins of Hypertext

- Lots of ideas for inventions
  - Wearable cameras for photographic records
  - Automatic transcripts of speech
  - Direct capture of nerve impulses

- Memex
  - First idea of hypertext




995
MICRO DESIGN

Memex Demo

Origins of Hypertext

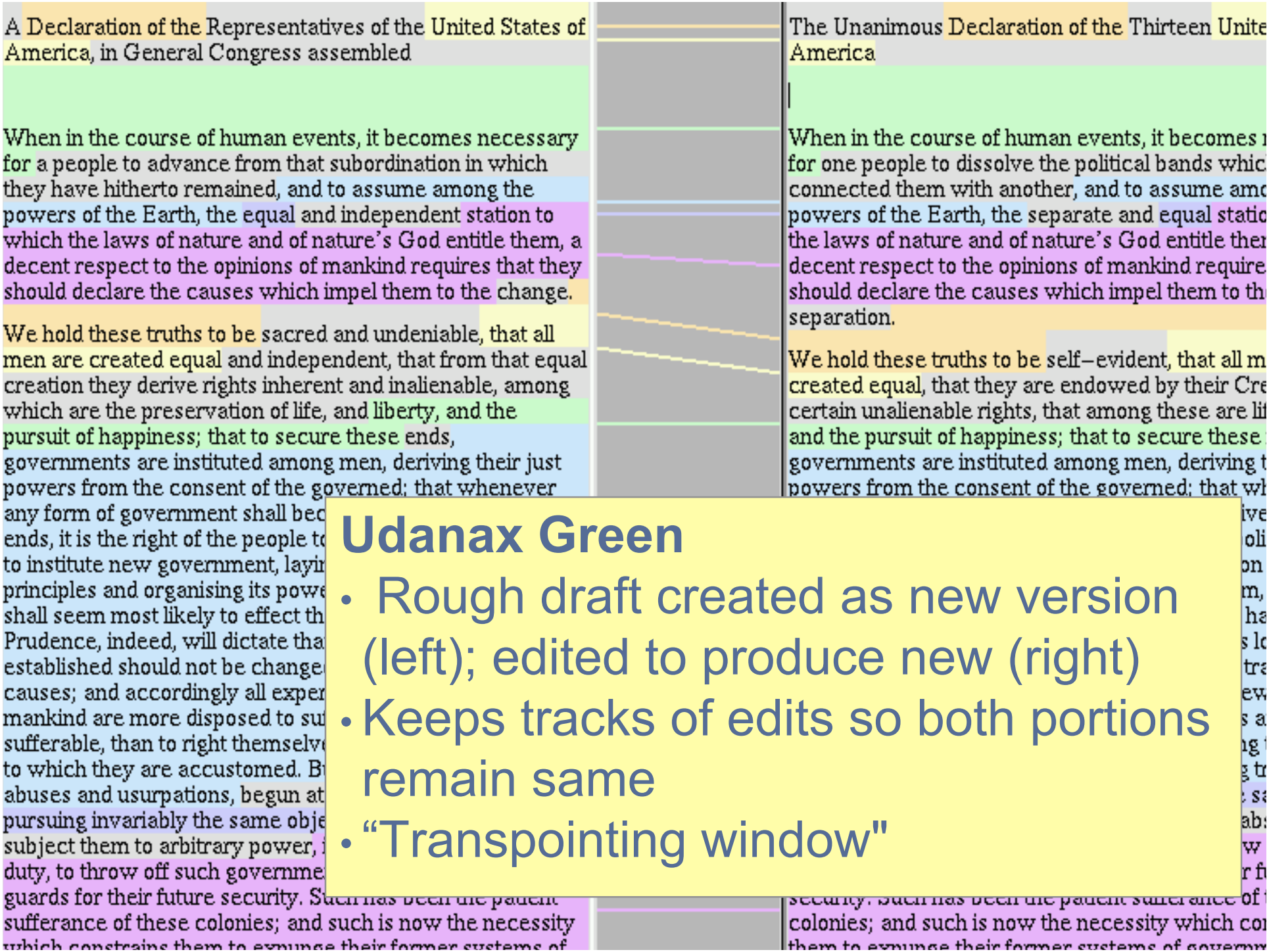
- Ted Nelson
  - Coined “hypertext” and “docuverse”
  - Xanadu, started in 1960s, end-all and be-all of hypertext
  - "longest-running vaporware story in the history of the computer industry" – Wired
  - Engelbart often credited with first real implementation

- Key Ideas:
  - Micropayments
  - Transclusion, combining multiple documents
  - StretchText

Stretchtext Example

In Stretchtext, text is stored as a stream. Extras text is coded to pop in and out at desired altitudes.

In Stretchtext, **a kind of hypertext,** text is stored as a stream. Extras text, **invisible at one level,** is coded to pop in and out at desired altitudes, **under user contro**l.


A Declaration of the Representatives of the United States of America, in General Congress assembled
When in the course of human events, it becomes necessary for a people to advance from that subordination in which they have hitherto remained, and to assume among the powers of the Earth, the equal and independent station to which the laws of nature and of nature's God entitle them, a decent respect to the opinions of mankind requires that they should declare the causes which impel them to the change.
We hold these truths to be sacred and undeniable, that all men are created equal and independent, that from that equal creation they derive rights inherent and inalienable, among which are the preservation of life, and liberty, and the pursuit of happiness; that to secure these ends, governments are instituted among men, deriving their just powers from the consent of the governed; that whenever
The Unanimous Declaration of the Thirteen Unite
America
When in the course of human events, it becomes
for one people to dissolve the political bands whic
connected them with another, and to assume amo
powers of the Earth, the separate and equal statio
the laws of nature and of nature's God entitle the
decent respect to the opinions of mankind require
should declare the causes which impel them to th
separation.
We hold these truths to be self–evident, that all m
created equal, that they are endowed by their Cre
certain unalienable rights, that among these are li
and the pursuit of happiness; that to secure these
governments are instituted among men, deriving t
powers from the consent of the governed; that wh
Udanax Green
• Rough draft created as new version (left); edited to produce new (right)
• Keeps tracks of edits so both portions remain same
• "Transpointing window"

Engelbart and Hypertext

Two Famous Firsts

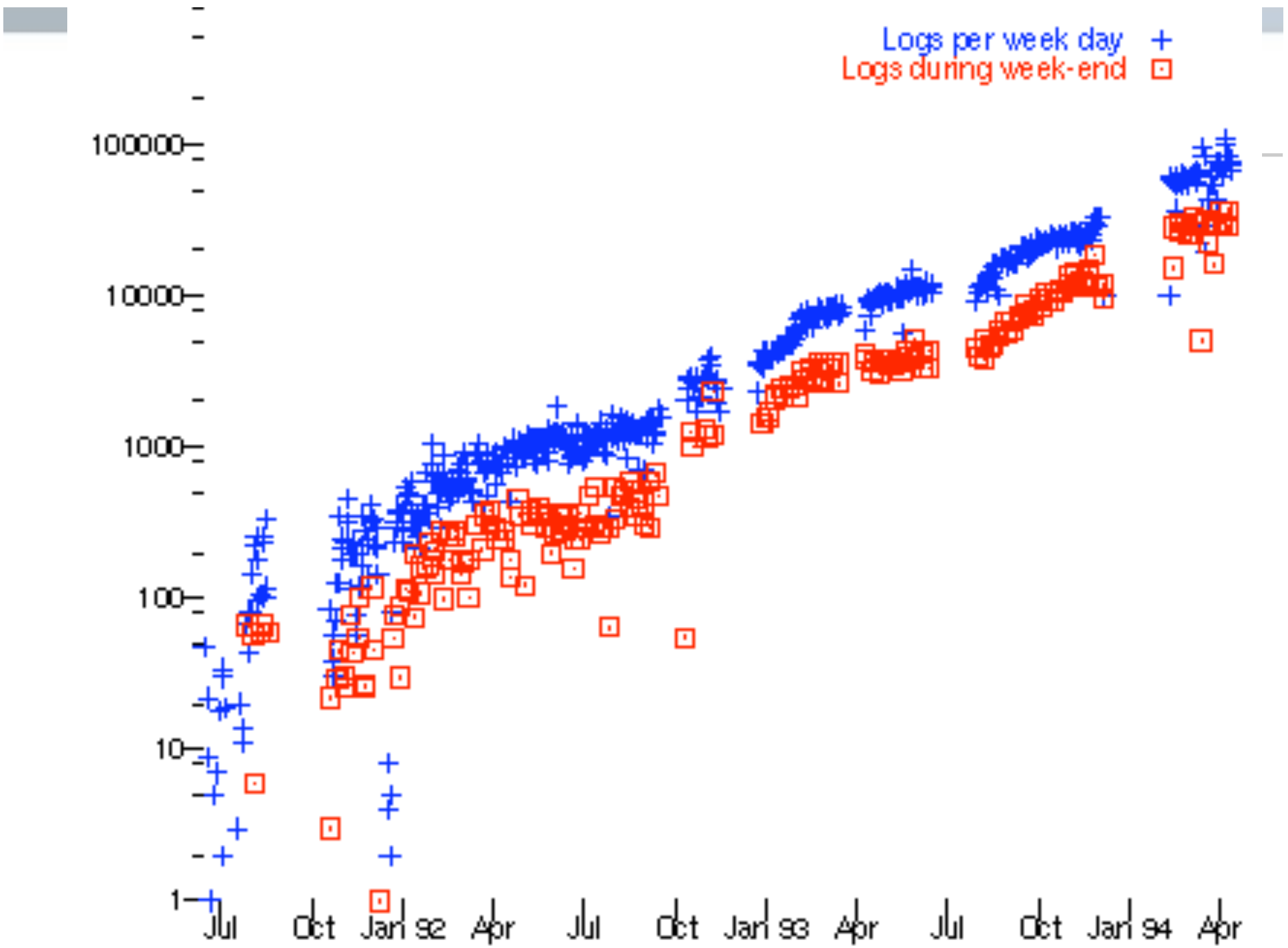
- **Edit text interactively**
- **Click on “links”**

State of Hypertext around Late 1980s

- Lots of research into hypertext but no real widely-deployed systems
  - Lots of small systems
  - Demonstrate understandability
  - Demonstrate authoring
- Most hypertext systems weren't aimed at network
  - Packaged for single computers
- Most assumed centralized control
  - No dead links!

The Web

- Invented by Tim Berners-Lee
  - ~1990
- Originally intended to make it easy to share documents at CERN
  - Lots of different computer systems
  - Lots of different incompatible formats
- Invented:
  - HTTP - transferring content
  - HTML - representing content
  - URLs - location of content


Logs per week day
Logs during week-end
100000
10000
1000
100
10
1
Jul
Oct
Jan 92
Apr
Jul
Oct
Jan 93
Apr
Jul
Oct
Jan 94
Apr

Design Rationale of the Web

- Early research paper to ACM Hypertext rejected
  - Images not yet supported (all text)
  - Broken hyperlinks bad, must not be allowed
  - URLs best way of representing hyperlinks?
  - Remote Procedure Calls (RPC) and distributed objects were “right way” to do distributed systems
  - Unaware of past work on distributed systems or hypertext
- Summary
  - Unclear research contribution
  - Pretty much ignored all previous research
  - It felt broken (though this helped it to succeed)
- Lesson? Ignore your professors? ☺

Alternative Architectures

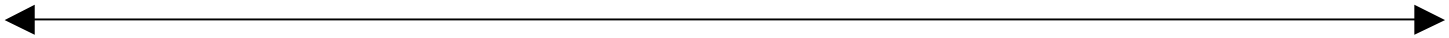
Alternatives

- CD ROMs
- Pure images
- Mobile code
- AOL / Compuserve
- Telnet
- Gopher
- Web Services

Core Questions

- How is content represented? (HTML)
- How is content transferred? (HTTP)
- How is content named? (URL)

How is Content Represented?



Images

Predefined Formats
Raw data
Metadata

Mobile Code

How is Content Represented?
Images

- Basic idea: Download a bitmap
- Pros:
  - Extremely simple, all rendering done elsewhere
  - Extremely simple client (just needs to render)
  - No security risks
- Cons:
  - No semantics, client can't do smart things
  - Other people can't build on your content
  - Server has to handle everything (lots of events)
- Examples: X-Windows, Remote Desktop, VNC

How is Content Represented?
Mobile Code

- Basic idea: Download mobile code + data
- Pros:
  - Richer interactions
  - Opaque, content is protected
  - Easier to scale up than images
- Cons:
  - Requires smarter clients
  - Huge security risk
  - Opaque, really hard to build on top of it
- Examples: Flash, Shockwave, Java

How is Content Represented?
Predefined Formats

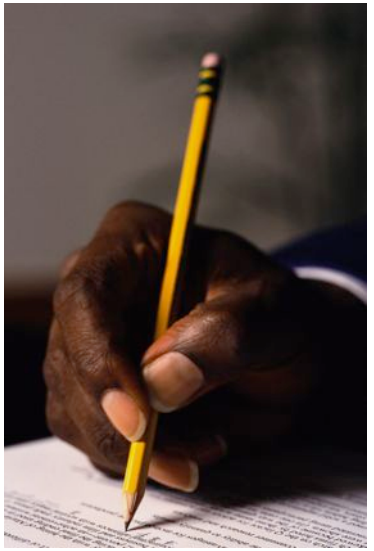
- Basic idea: Download raw data and metadata, client renders
- Pros:
  - Easier to scale up than images (less server load)
  - Can inspect content and build on it more easily
  - Fewer security risks
- Cons:
  - Hard to protect content
  - Everyone has to agree on standard (or de facto standard)
- Examples: HTML, PDF, LaTex

How is Content Represented?

- Worth comparing PDF / PostScript vs HTML
  - PDF meant to be pixel-perfect and printed
  - Implies poorer for small screens
  - Not clear if a print format makes sense for screens
  - Also, PDF is protected intellectual property
- Though in theory could have been done

How is Content Represented?

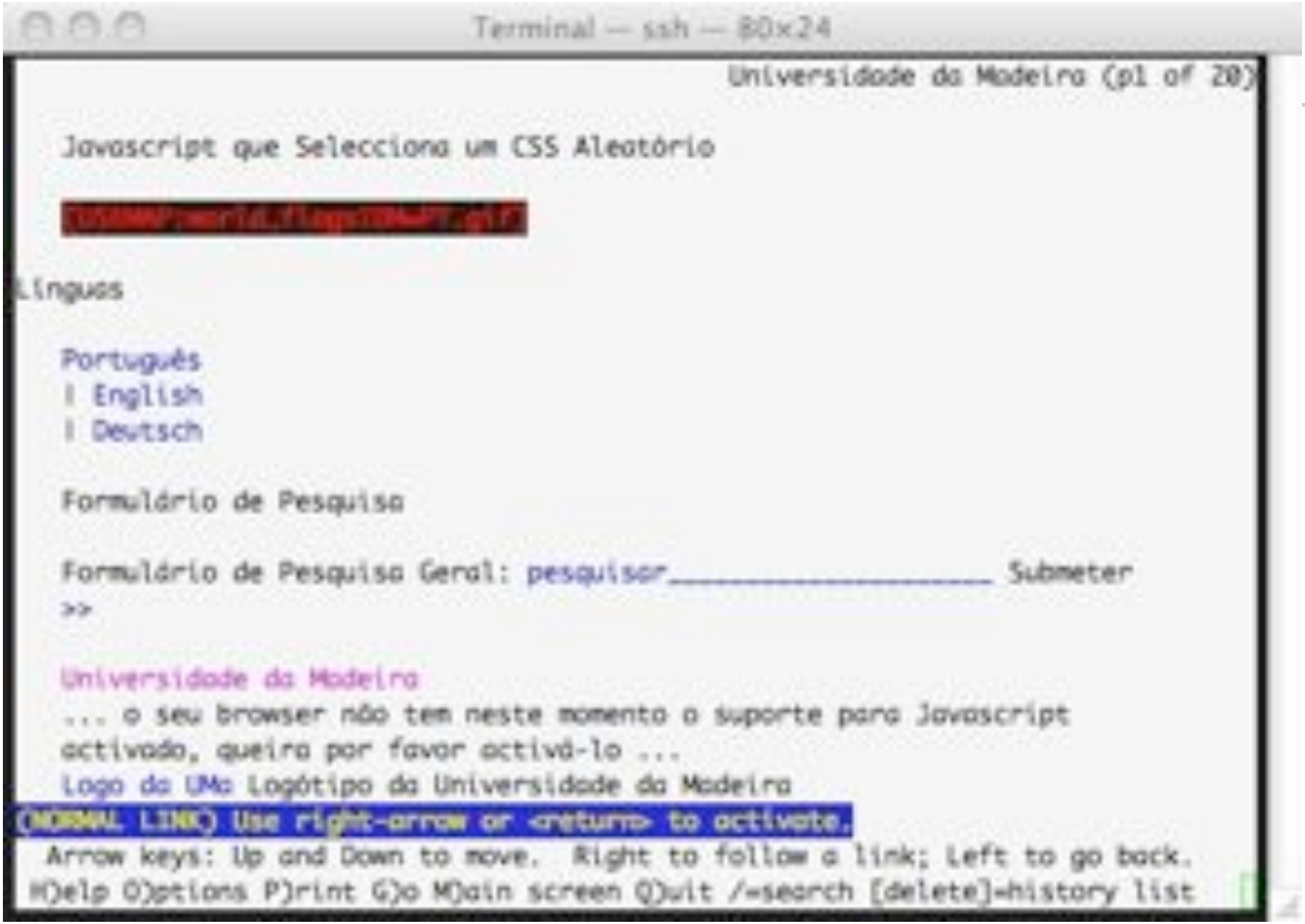
- Authoring also extremely important issue:
  - HTML open, everyone could see the standards
  - HTML text-based, low barrier to entry
  - Easy to see other people's content
  - Easy for anyone to add new content
  - Easy to create authoring tools on top



- Compare to AOL / CompuServe model
  - They own and control everything on their servers
  - Had proprietary formats
  - Limited innovation to what they could provide

How is Content Represented?

- HTML extensible for the future
  - Could add new tags if needed
  - Web browsers just ignore tags they don’t know
- In theory, could still use original Netscape web browser


Terminal — ssh — 80×24
Universidade da Madeira (p1 of 20)
Javascript que Selecciona um CSS Aleatório
Línguas
Português
| English
| Deutsch
Formulário de Pesquisa
Formulário de Pesquisa Geral: pesquisar Submeter
>>
Universidade da Madeira
... o seu browser não tem neste momento o suporte para Javascript
activado, queira por favor activá-lo ...
Logo da UMa Logótipo da Universidade da Madeira
(NORMAL LINK) Use right-arrow or <return> to activate.
Arrow keys: Up and Down to move. Right to follow a link; left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

Alternative Architectures

Alternatives

- CD ROMs
- Pure images
- Mobile code
- AOL / Compuserve
- Telnet
- Gopher
- Web Services

Core Questions

- How is content represented? (HTML)
- How is content transferred? (HTTP)
- How is content named? (URL)

How is Content Transferred?

- Worth re-iterating:
  - At this point, few people thinking of distributed hypertext
  - Wasn't clear at the time this was a need
- Lots of hypermedia transferred via CD-ROM
- Cost of distribution?
  - Relatively high
  - Burn new CDs, shipping costs
- How often content updated?
  - Can be quite often, but distribution costs prohibitive

How is Content Transferred?

- Binary vs Text Protocols
  - Binary protocols more compact
  - Binary protocols harder to implement and debug
  - Text protocol simpler to implement, cross-platform
  - Text easier for proxies
- Stateless vs Stateful
  - Stateless protocol means simpler server, easier to scale
  - Cookies sort of a hack

How is Content Transferred?

- Worth noting:

- HTTP somewhat extensible
  - Can add new headers easily
- HTTP open standard, anyone could implement server
- HTTP intentionally minimalist
  - Minimal set of functionality needed to make things work

Alternative Architectures

Alternatives

- CD ROMs
- Pure images
- Mobile code
- AOL / Compuserve
- Telnet
- Gopher
- Web Services

Core Questions

- How is content represented? (HTML)
- How is content transferred? (HTTP)
- How is content named? (URL)

How is Content Named?

- Worth re-iterating:
  - At this point, few people thinking of distributed hypertext
  - Wasn’t clear at the time this was a need

- URLs designed to be extensible
  - Could support multiple protocols
  - ftp:// http:// real://

How is Content Named?

- HTML designed to support one-way links only
  - Many research systems supported two-way

- Why one-way links?
  - Simpler to implement
  - No central control
  - Scaling much easier
  - But also led to broken hyperlinks

Some Design Themes

- Scaling
  - Make it work for thousands, millions of people
  - Part of Hypertext original vision
- Extensibility
  - Make it easy to add new things if needed
    - HTML tags, HTTP headers, URLs, browser helpers
  - But don’t break existing base
    - Forward-compatible (ignore unknown)

Some Design Themes

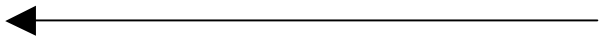
- Open Standards
  - Everyone could see (and learn by example)
- Inclusive
  - HTTP supports transfer of any object
  - URL supports naming of any object

Some Design Themes

- Simple, low barrier to entry
  - Text-based
  - Easy to implement on multiple platforms
  - Minimal set of functionality needed to make things work
- Good enough
  - Ugly syntax, but…
  - Addressed a need
  - Right mix of functionality & simplicity got us 80% way there
- For most part, was free (cost)
  - Free servers, free browsers, free content

Outline

- History of the Web
- Alternative Web Architectures
- REST vs Web Services



2 Minute Break

REST

- REpresentational State Transfer (REST)
  - Dissertation by Roy Fielding, an architect of HTTP 1.1
  - Explains a lot of design rationale behind HTTP
  - Sort of a post-hoc analysis of why HTTP works well

Design Rationale of the Web

- Architectural styles describe systems at high-level
  - Data flow (Unix pipes)
  - Blackboard (Speech recognition, AI)
  - Call-and-return (programming languages)
  - Event-based

Design Rationale of the Web

- Stateless client / server protocol
  - Easy to implement, load balance, restart
- Small set of well-defined ops for all resources
  - GET, POST, PUT, DELETE
  - Few universal verbs, applicable to lots of nouns
- Universal syntax for resource identification (URLs)
- Hypermedia for application information and state transitions
  - Transfers from one page to another

Design Rationale of the Web

- Anarchic Scalability
  - No single point of control, single point of failure
    - AOL, Compuserve had custom content at the time
    - Had to get permission, pay them money to deploy
  - Independent deployment
    - Can add new document, server, don't have to notify
  - No back-pointers (doesn't scale)

- Transparency
  - Easy to modify and debug HTTP in transit (text vs binary)
  - Easier support for proxies and caching

Comparison to Web Services?

- REST originally focused on HTTP and URLs
- A current debate is REST vs Web Services
  - Both are ways of accessing networked resources
  - Different underlying philosophies
- REST advocates also known as RESTafarians

REST-based Weather Service

- Rely exclusively on pre-defined methods
  - GET, POST, PUT, DELETE
- Define URLs
  - `http://weather.com/eu/portugal/funchal`
  - `http://weather.com/eu/uk/london`
- Return content in some form
  - HTML web page or XML-formatted
- To get the latest weather, just access the URL
  - Works for people, programs

Web Service-based Weather Service

- Define some methods in WSDL file
  - `getWeather(country, state, city)`
  - Define to be accessible via http on certain url
- Return XML-formatted content

- To get latest weather, create a program that POSTs the right SOAP call to the URL
  - Works for programs only

Comparison
REST

- Two ways of doing same thing
- REST, small set of well-defined ops for all resources
  - Few verbs (GET, POST), lots of nouns (resources)
  - Never have to update your API, universal compatibility
- With REST, your URLs are your API
  - Open API, makes it easy for others to “hack” and extend
  - Some semantics though (GET should have no side effects)

Comparison
Web Services

- Web services more like object-oriented programming
  - Create appropriate methods
  - Lots of verbs and lots of nouns

- Can support richer semantics
  - Ex. transactions, reliability
  - Also has better names than URLs can provide

Summary Comparison

- Web Services
  - Protocol independent
  - More easily machine processable
  - Reinventing everything REST already does
  - But somewhat complex
  - Lose network effects of web (islands of interoperability)

- REST
  - Very simple model
  - More open, easier for others to build on top of
  - But leaves more to client and developer (parsing)
  - Is GET, POST, DELETE, PUT sufficient?

Comparison

- Examples of REST services
  - A9, RSS feeds, MusicBrainz, AJAX apps (XMLHttpRequest)
- Examples of Web Services
  - Search: Amazon, EBay, Google

Summary

- History of the Web
  - Vannevar Bush, Memex
  - Early Web
- Alternative Web Architectures
  - Explain other possibilities, why they didn't really work
- REST architectural style