Web Application Development

Dr Sean Walton

@DrSeanWalton

General Expectations

Me from you

- Respect that other people in the room have paid to come here – so don't talk
- If you don't like the way I'm doing something come and tell me
- Engage with the module
- Come see me if you are struggling

You from me

- Deliver the course
- Support your learning
- Mark your work fairly

Where to get resources

- All lecture slides are already on blackboard
- Video tutorials on my YouTube channel
  - https://www.youtube.com/watch?v=mYwM7zjQvdg&list=PLcZQK8y_7_6nh6R3Qepbx-EPy3DewjM0S&index=1
- Lectures will be streamed to an unlisted YouTube channel
- The rest of the internet
- Recommended YouTuber
  - https://www.youtube.com/user/TechGuyWeb

Getting support

1. Google it
2. Ask on Slack
   https://join.slack.com/t/webappsswanseacompsci/signup
3. E-mail me: s.p.walton@swansea.ac.uk
4. If you email me a question that can be answered by reading notes or assignment sheets I will reply saying that
5. Twitter: @DrSeanWalton
6. Office hours: Monday 1pm-2pm and Thursday 11am-12pm in 120

- PSA - I work 8am-4pm Monday-Friday

Scope and Aim

To give a general introduction to the full stack of web development so that you can adapt to whatever framework you are told to use

This is done by supporting you to learn the web framework Laravel

More generally this will be the first time some of you work with a modern framework and modern software engineering tools

It will be a chance for you to develop a full (but small) application from scratch, you'll then be able to evidence crucial developer skills in job interviews

This means it’s going to be challenging and you’re going to have to do some independent learning

How this module will be taught

There will be some lectures to teach principles. All lecture notes and materials are already on blackboard – you should read ahead.

There will be some lectures and videos to teach the basics of Laravel. Each of these lectures will have tasks which you should complete at home and/or in labs to actually learn the stuff!

Then the videos and explicit tasks will stop as I introduce lots of more advanced features. This is me taking away the training wheels.

I won't be sharing the code source files, and most of my lecture slides are screenshots so you can't copy and paste - because you have to code and make mistakes to learn anything valuable

Assessment

This module is 100% object oriented programming. You'll also have to use command line tools.

Aims of the assessment

- Ensure you have met the learning outcomes for the module
- To provide evidence that you can work effectively with Laravel.
- To encourage you to independently research the concepts relating to web application development.
- To enable you to produce a web application you could use in a portfolio when applying for jobs.
- To give you the opportunity to stretch yourself and produce a high-quality web application to be proud of.

You will be given a project brief to design and implement a web application. There will then be three pieces of coursework to hand in based on this.

Project Brief

- Your task is to plan and create a simple web application, using Laravel, which has the following basic functionality:
- (1) Users can create posts and comment on their own and other users' posts.
- (2) (Masters only) Basic analytics is displayed for posts, for example number of unique views.
- For example, you could create a simple blog where the admin can post articles which users can comment on, or a simple version of blackboard where admin (or lecturers) can post lecture slides and students can comment on them.
- This is the basic functionality only, it is up to you to look at the mark scheme and add functionality to evidence various skills. Do this before you start.

Assessment Deadlines

- Coursework 1 – Code Review (30%)
  - Deadline 1 - 11th November - (handing in your code – worth 20%)
  - Deadline 2 - 18th November - (reviewing someone else's code – worth 10%)
- Coursework 2 – 20th December - Your Implementation and Justifications (70%)
- Missing a deadline gets you a mark of zero, any indication of academic misconduct will be reported

You will be required to hand in various pieces of coursework throughout the semester. This will require good time management to balance it with other modules, if this is something you struggle with watch the video I made about how I manage my own time.

Coursework 1 and 2 – Code Submission and Review (30% total weighting)

An important part of working in a software engineering organisation is code reviews. In this process engineers look at each other's code to spot bugs and ensure standards are being adhered to.

You must first submit all the source code relevant to setting up your database relationships and seeding. This should be submitted as a pdf which includes syntax colouring.

- You will then be assigned to review and mark another student's code (within one week)
- I will mark both your code and your review

Coursework 2 – Implementation (70% weighting)

This is the implementation of your web application using Laravel. There are several elements for you to submit. These are all done as answers to a blackboard test.

(1) You are required to submit a .zip file containing all your source files, the .zip file is required to be titled as your student number – failing to do this will be considered not meeting basic requirements. Ensure that your project works on multiple machines. **If your application does not meet these basic requirements it will be considered not working code.** Submitting code which works on multiple machines shows you understand the framework you are using and following guidelines *to the letter* is an essential skill for a web developer and software engineer. Source files alone will not be assessed, you must also submit (2) and (3).

(2) You are required to submit an up to **120 second video** which evidences the rubric rows “Progress”, “User Presence” and “Working with Data”. If aspects are not shown in the video, you will not receive marks for them.

(3) You are required to answer several short questions to show that you understand the framework. This will require you to identify where in your code you have evidenced good practice.

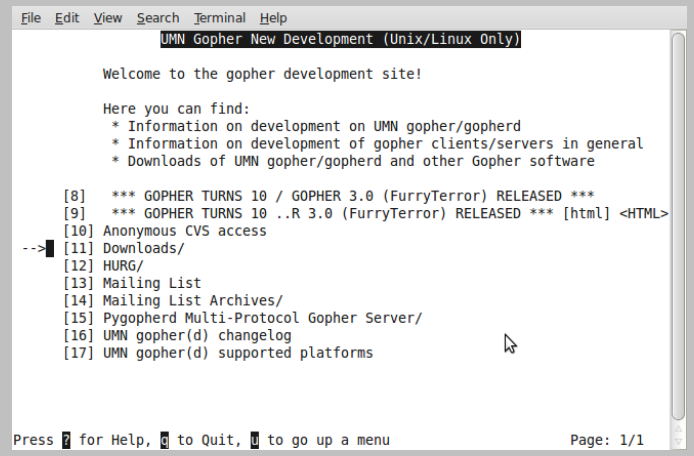
Timetable

- I will be giving some lectures in labs if your student number is less than or equal to 952883 then you are in Group A, otherwise you are in Group B
- This is a 15 credit module so you should spend a ¼ of your time on it this semester
  - In a 35 hour week that is around 9 hours – you work a 35 hour week right?
  - This means outside of lectures you should spend 6 hours working on the module – if you actually do this starting now you'll finish the coursework early

The evolution of the world wide web

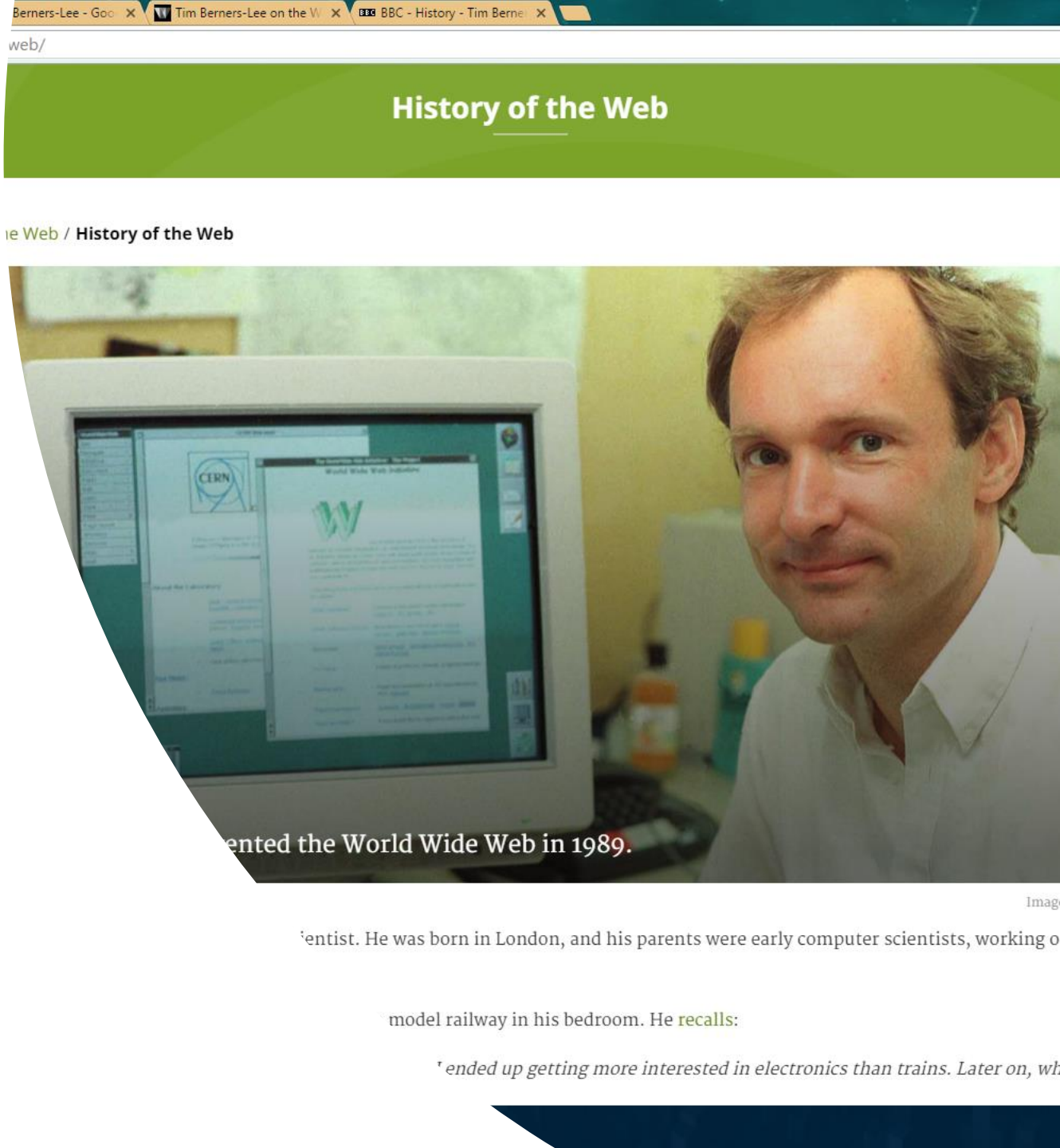
The Internet before the world wide web

- Entirely text based
  - Lots of manual command line stuff
  - Gopher was an ASCII-based user program which helped you using a menu to search through collections of files
- The domain name system (mid-80s)
  - Essentially replaced IP addresses by a string
  - A Distributed Name Service (DNS) maps domain names to IP addresses … made everything easier and was a massive breakthrough



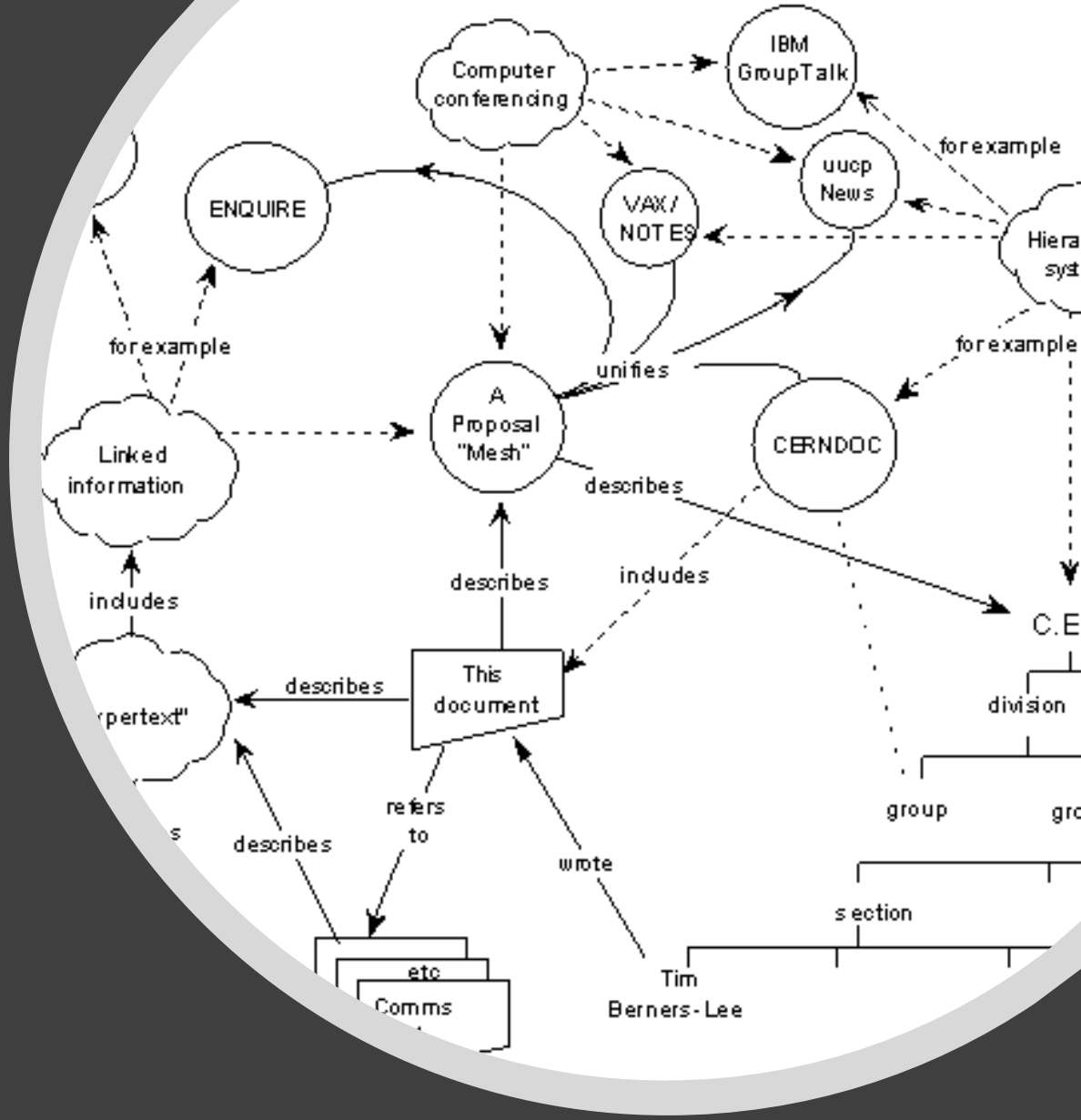
Birth of the Web – Sir Tim Berners-Lee (1989)

- Software engineer at CERN
- Noticed that scientists had problems sharing information
- You had to log into different computers to get different information and sometimes learn a different program on each computer
- In 1989 millions of computers were already connected together by the internet
- Berners-Lee saw a solution which would exploit an emerging technology - hypertext

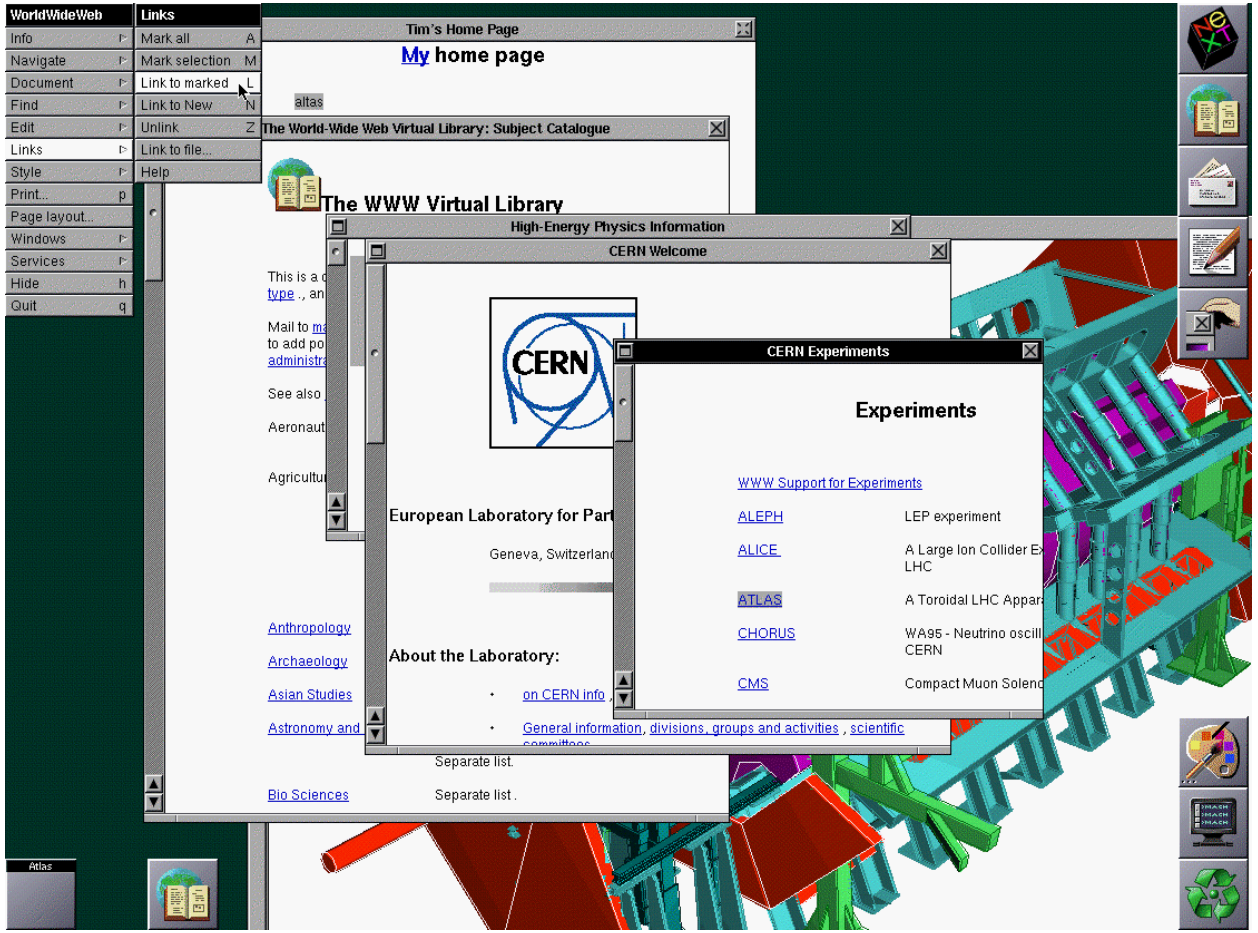


Information Management: A proposal - Berners-Lee

- Identified that the CERN experiment is constantly changing, so a printed instruction manual is useless
- He envisaged a ‘web’ of notes with links between them
- He identified the needs of CERN and a technology which could address them
- Feedback from management ‘Vague, but exciting’
- This went on to become the world wide web



- By October of 1990 Tim has written three fundamental technologies
  - HTML: HyperText Markup Language
  - URI: Uniform Resource Identifier
  - HTTP: Hypertext Transfer Protocol
- …and the first web browser, editor and server
- In April 1993 it was announced that the underling code for these technologies would be available on a royalty-free basis, **for ever**
- This sparked a global wave of creativity, collaboration and innovation like never seen before which is still moving today



The first website

World Wide Web

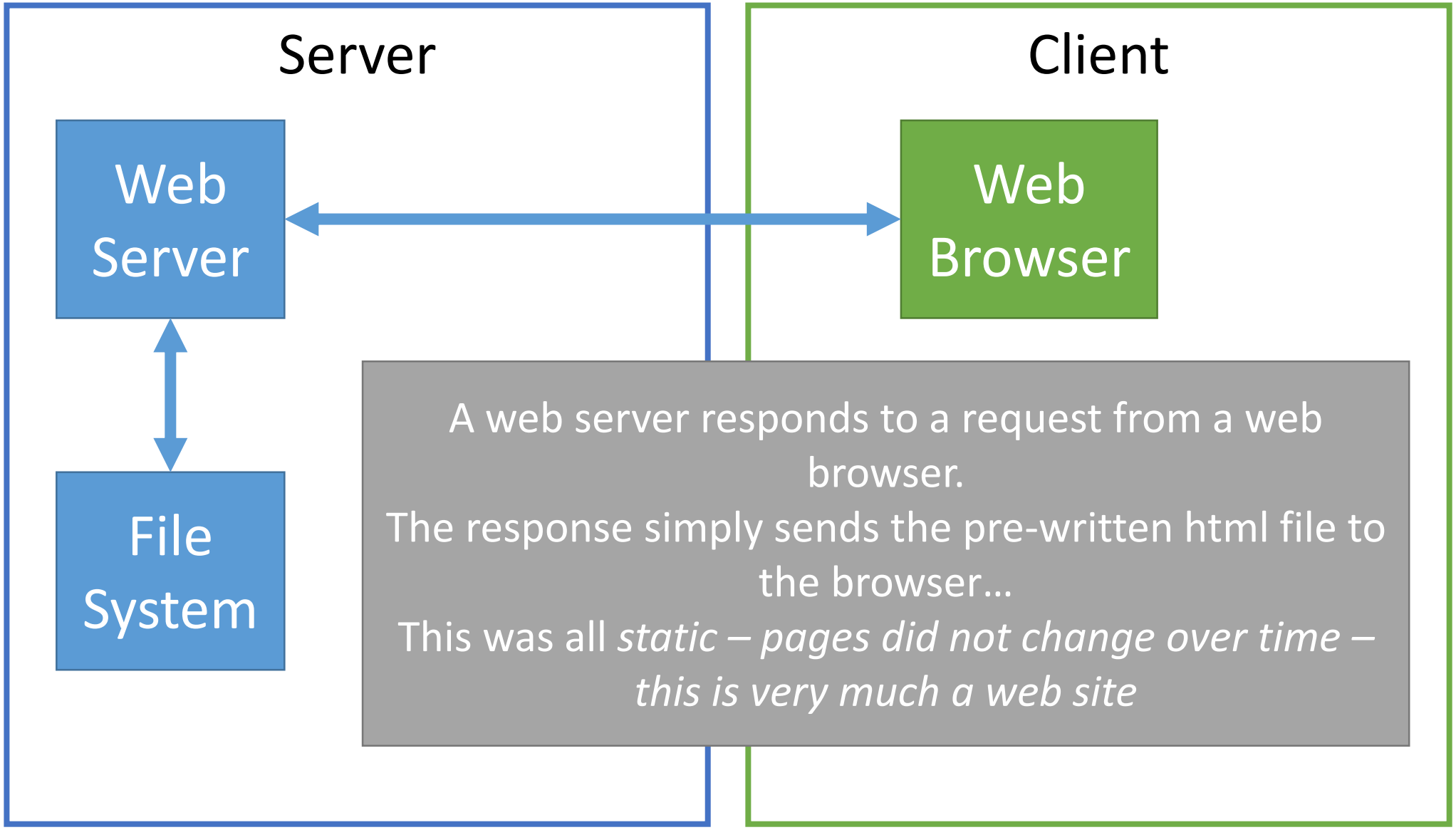
The WorldWideWeb (W3) is a wide-area hypermedia information retrieval i
universe of documents.

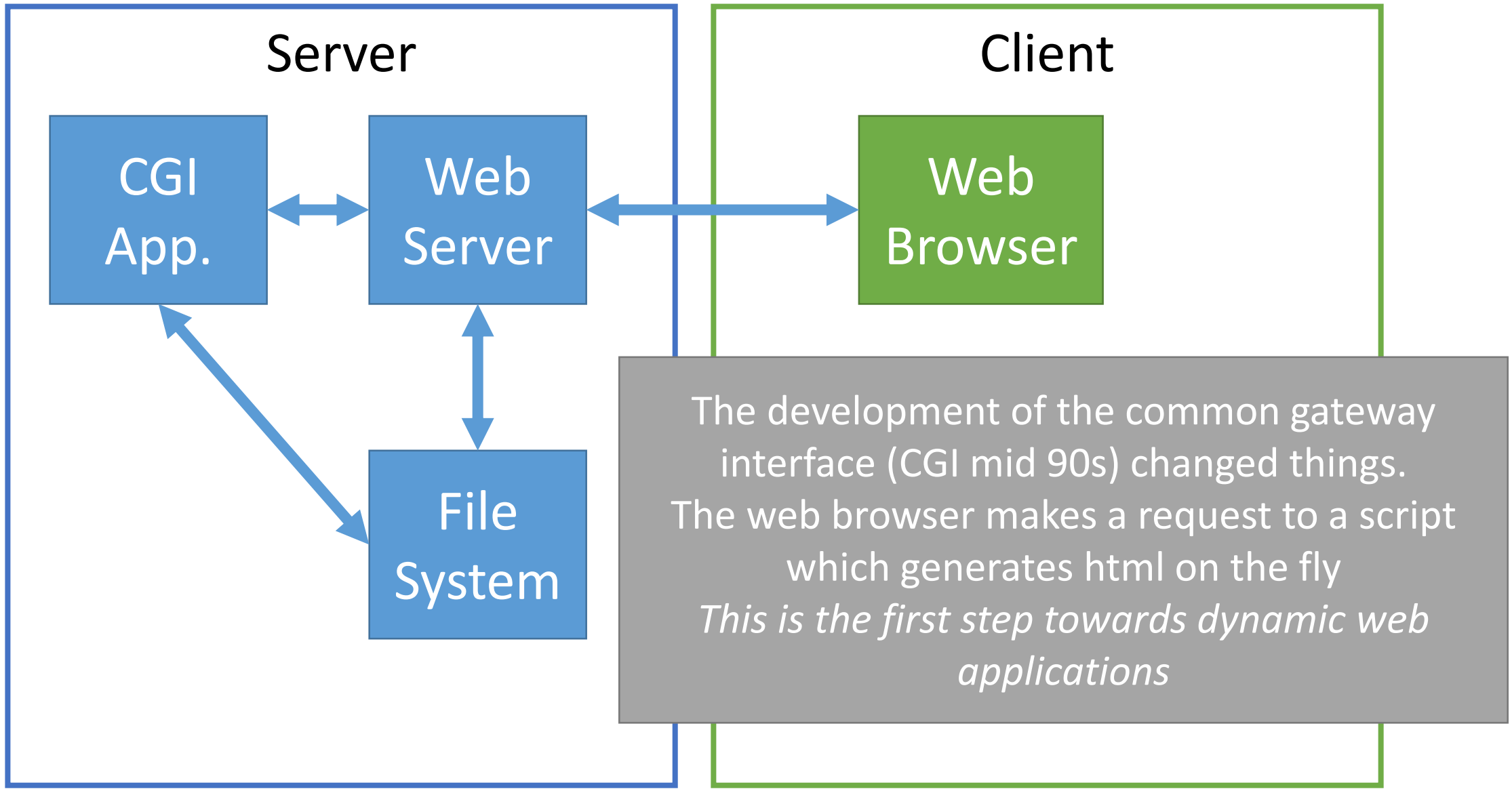
Everything there is online about W3 is linked directly or indirectly to this d
Mailing lists , Policy , November's W3 news , Frequently Asked Questions

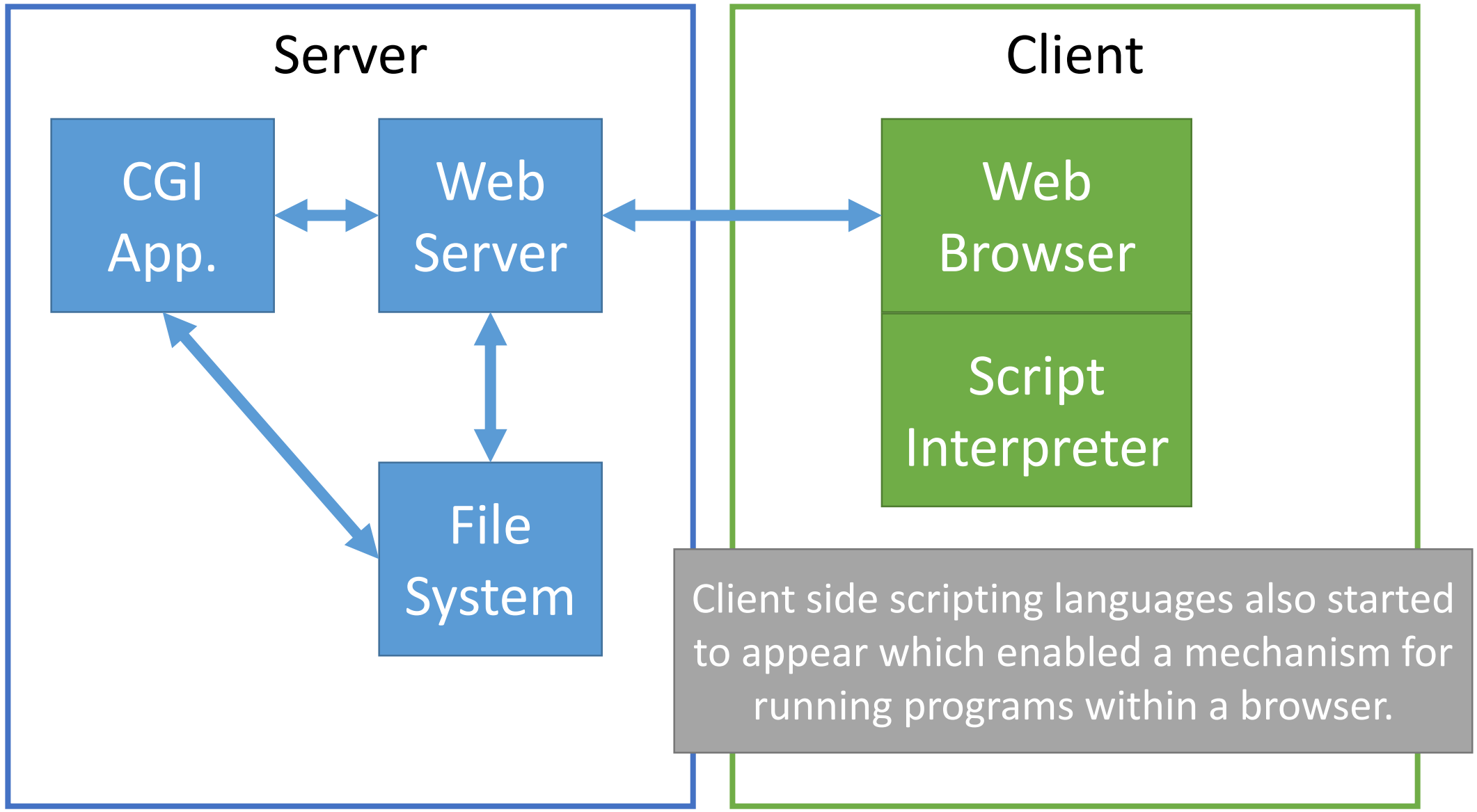
What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mo
Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

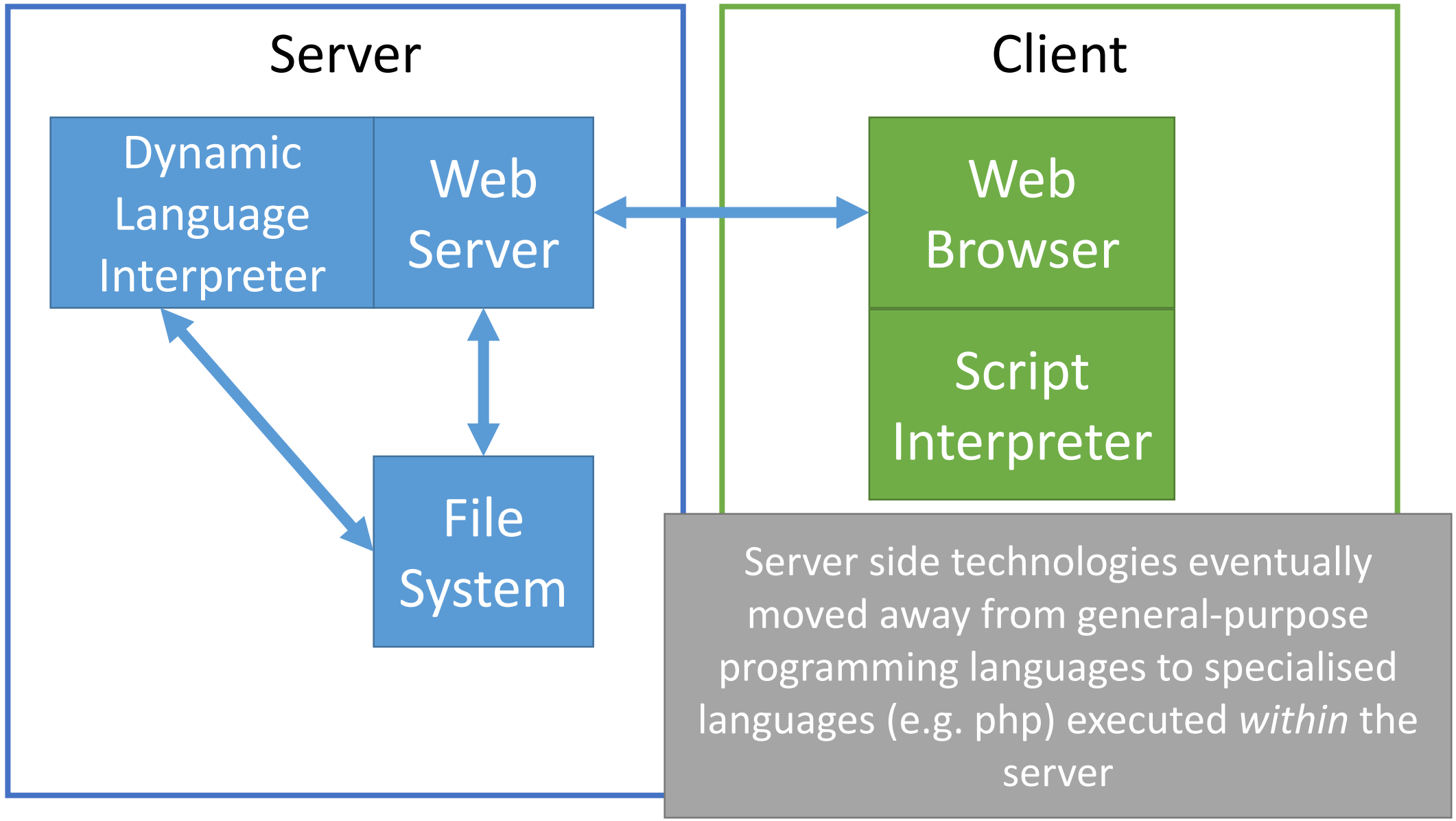
```
<HEADER>
<TITLE>The World Wide Web project</TITLE>
<NEXTID N="55">
</HEADER>
<BODY>
<H1>World Wide Web</H1>The WorldWideWeb (W3) is a wide-area<A
NAME=0 HREF="WhatIs.html">
hypermedia</A> information retrieval
initiative aiming to give universal
access to a large universe of documents.<P>
Everything there is online about
W3 is linked directly or indirectly
to this document, including an <A
NAME=24 HREF="Summary.html">executive
summary</A> of the project, <A
NAME=29 HREF="Administration/Mailing/Overview.html">Mailing lists</A>
, <A
NAME=30 HREF="Policy.html">Policy</A> , November's  <A
NAME=34 HREF="News/9211.html">W3  news</A> ,
<A
NAME=41 HREF="FAQ/List.html">Frequently Asked Questions</A> .
<DL>
<DT><A
NAME=44 HREF="../DataSources/Top.html">What's out there?</A>
<DD> Pointers to the
world's online information,<A
NAME=45 HREF="../DataSources/bySubject/Overview.html"> subjects</A>
, <A
NAME=z54 HREF="../DataSources/WWW/Servers.html">W3 servers</A>, etc.
<DT><A
NAME=46 HREF="Help.html">Help</A>
<DD> on the browser you are using
<DT><A
NAME=13 HREF="Status.html">Software Products</A>
<DD> A list of W3 project
components and their current state.
```

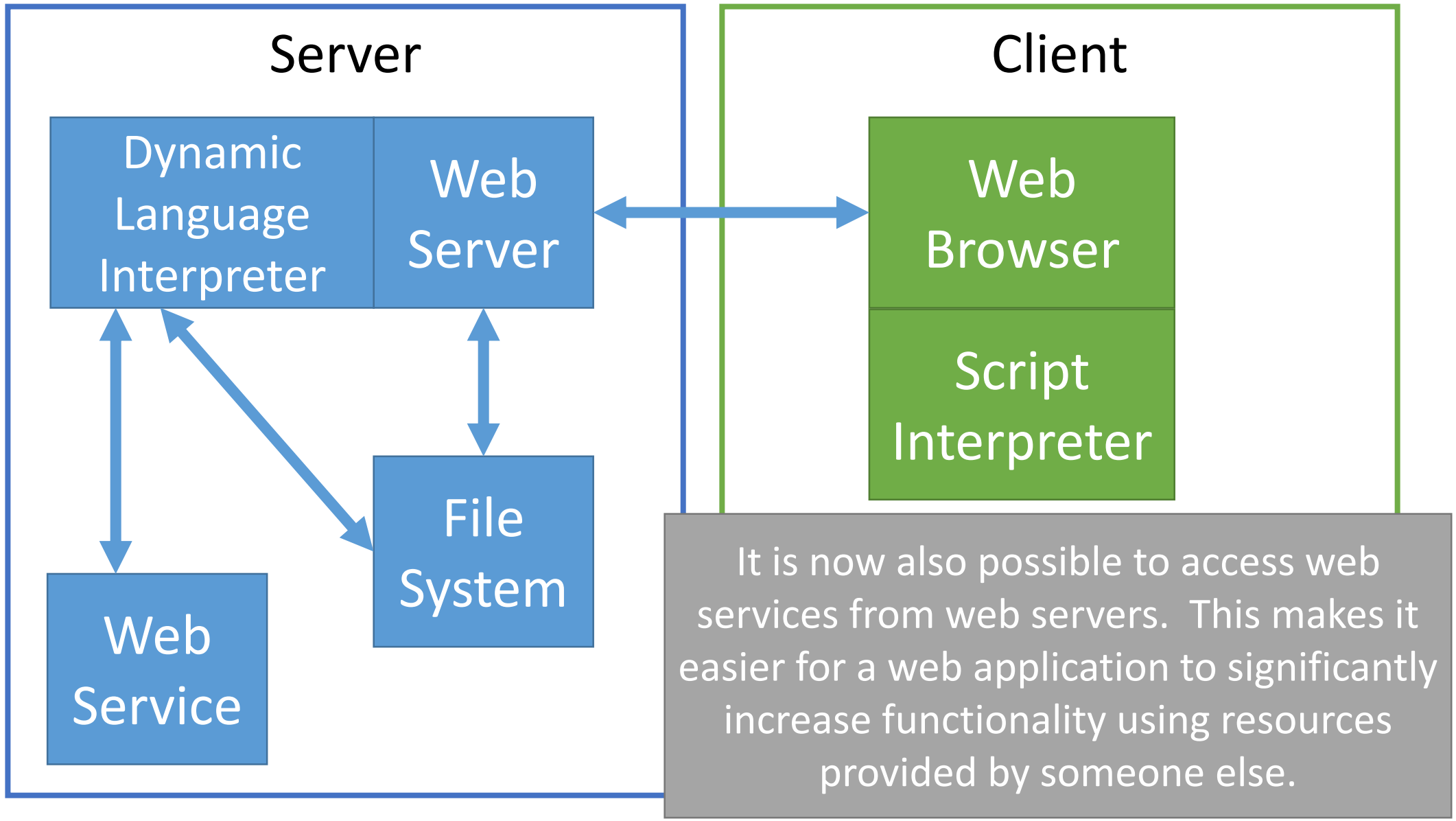
The Evolution of Web Sites into Web Applications


Server
Client
Web Server
Web Browser
File System
A web server responds to a request from a web browser.
The response simply sends the pre-written html file to the browser…
This was all *static – pages did not change over time – this is very much a web site*


Server
Client
CGI App.
Web Server
Web Browser
File System
The development of the common gateway interface (CGI mid 90s) changed things.
The web browser makes a request to a script which generates html on the fly
This is the first step towards dynamic web applications


Server
Client
CGI App.
Web Server
Web Browser
Script Interpreter
File System
Client side scripting languages also started to appear which enabled a mechanism for running programs within a browser.


Server
Client
Dynamic Language Interpreter
Web Server
Web Browser
Script Interpreter
File System
Server side technologies eventually moved away from general-purpose programming languages to specialised languages (e.g. php) executed *within* the server


Server
Client
Dynamic Language Interpreter
Web Server
Web Browser
Script Interpreter
File System
Web Service
It is now also possible to access web services from web servers. This makes it easier for a web application to significantly increase functionality using resources provided by someone else.