

WEB

DEVELOPMENT

What is the Internet?

- A global network of computers. Network of networks
- Researched by **DARPA** early 60s to survive **WWIII**
- **ARPANET**, a precursor, connected **military** and **academic** networks in **1980s**
- **Commercial** use expanded in **1990s** to what we have today

What is the Web?

- AKA The World Wide Web or **WWW** or just The Web
- A combination of technologies that **use the Internet** to **share formatted documents** connected to one another
- Invented by **Tim Berners-Lee** in '89 a physicist at **CERN**
- Documents are formatted in **HTML** (Hypertext Markup Language)

What is the Web?

- **Browsers** fetch HTML documents from **servers** connected by the internet
- Documents are found with **URLs** (Uniform Resource Locator)
- HTML documents format (hyper) text as **anchors** that refer or **link** to other documents

Some Milestones

1960s - The Internet

1970s - Client-server architecture

1980s - Personal computers

1990s - The World Wide Web

2000s - Web Applications

2010s - Mobile computing, big data

2020s - ??? Neuralink ???

Server Frameworks

1993 - Common Gateway Interface (Perl, C++, ...)

1995 - PHP (PHP)

1999 - Java 2 Enterprise Edition (J2EE)

2000 - .NET (ASPs, Visual Basic, C#, ...)

2005 - Ruby on Rails (Ruby)

2005 - Django (Python)

2009 - Node.js (JavaScript)

2010 - Flask (Python)

Client Frameworks

1995 - JavaScript (Netscape)

2010 - Backbone

1996 - Adobe Flash

2011 - Ember

2005 - Dojo

2012 - Sails

2006 - jQuery

2012 - Meteor

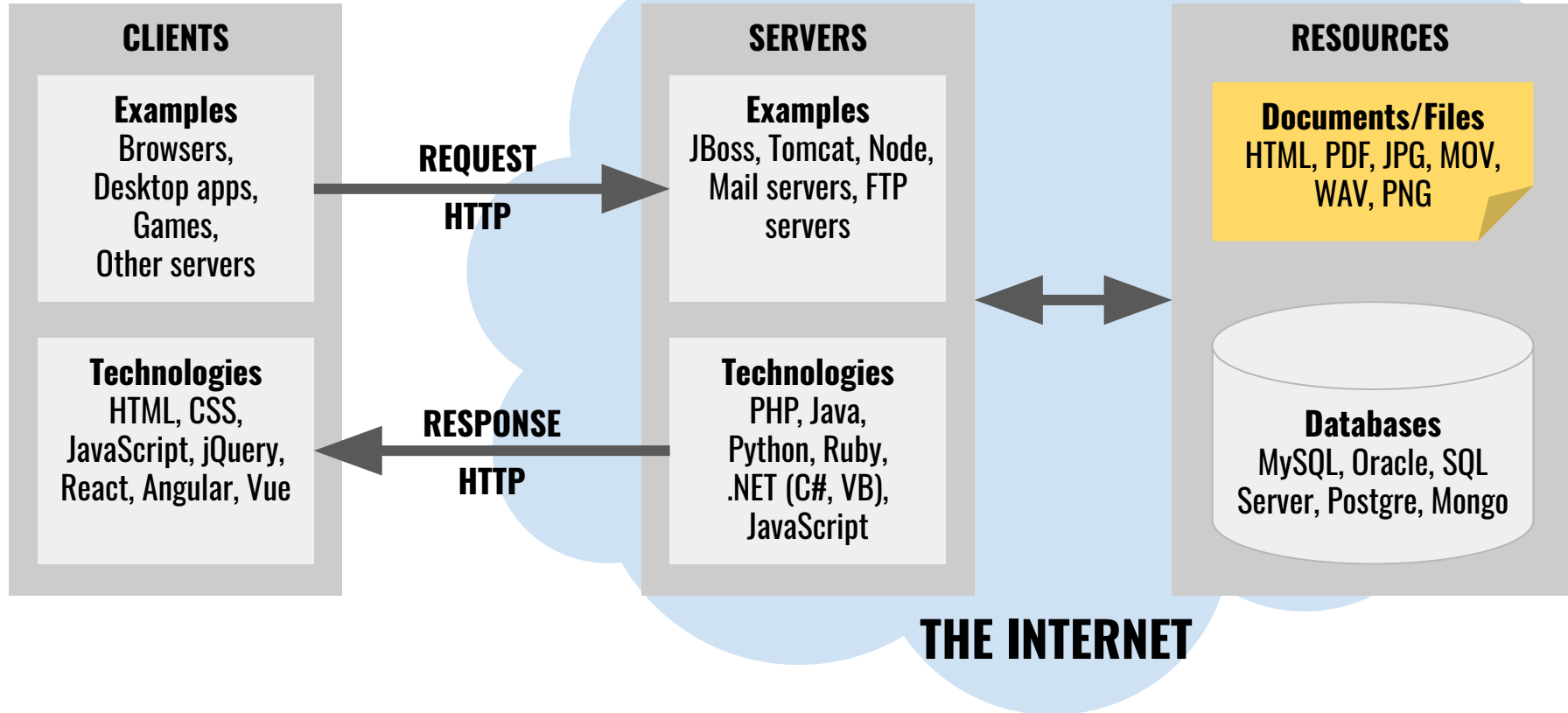
2010 - Knockout

2013 - React (Facebook)

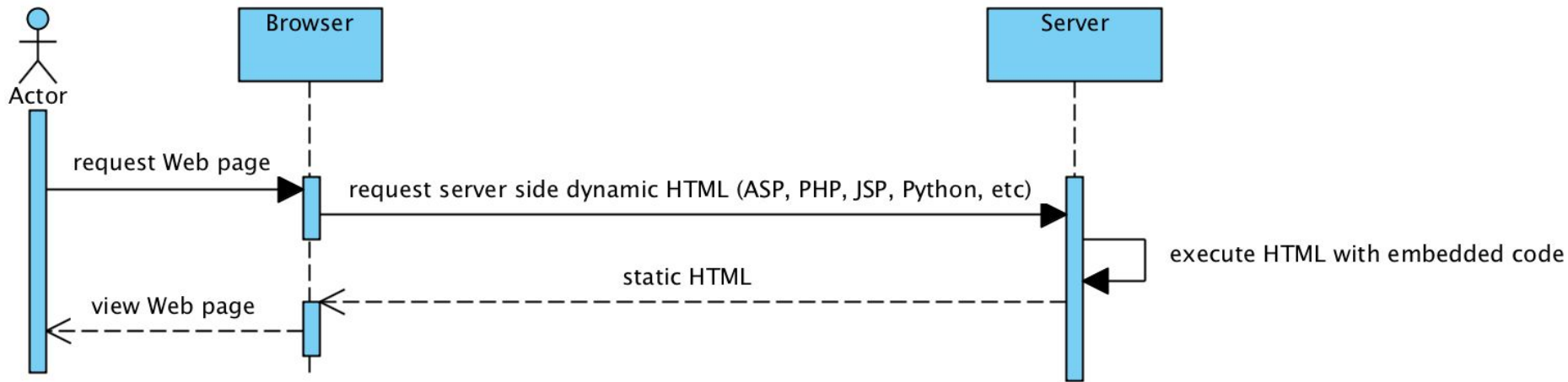
2010 - Angular (Google)

2014 - Vue

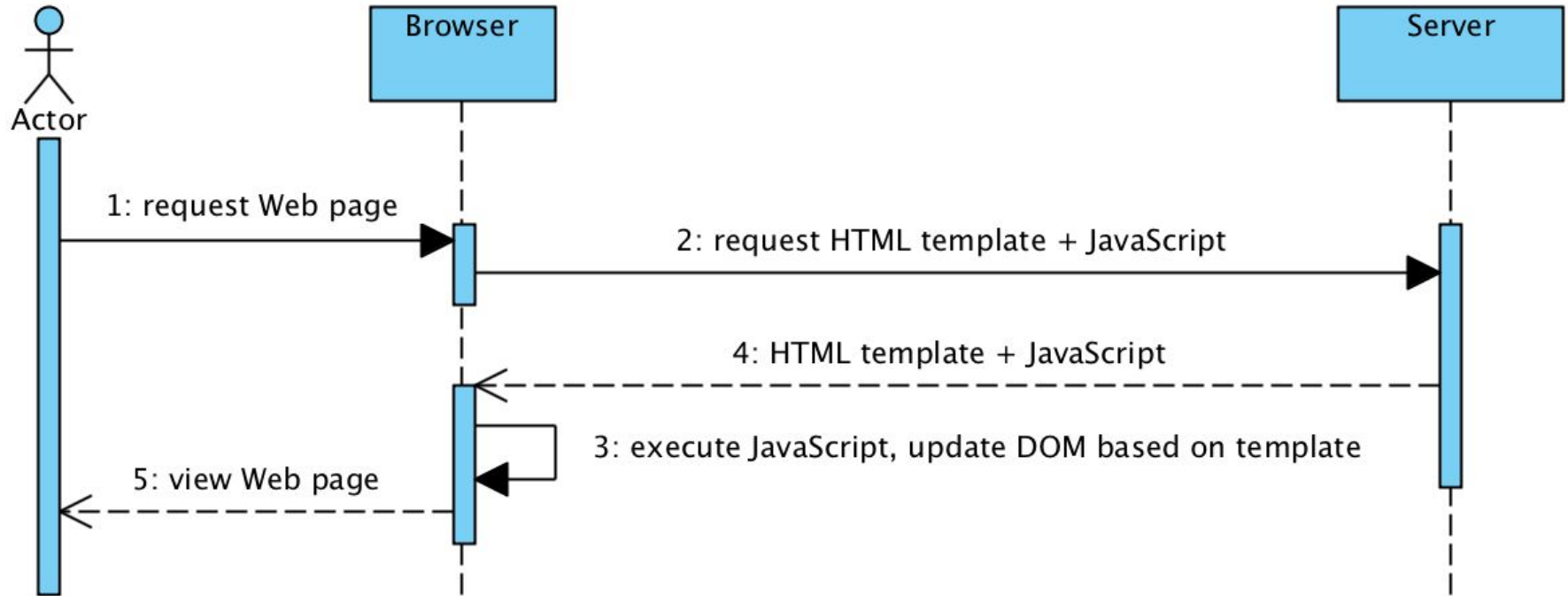
The Client Server Architecture



Server Side UI Rendering



Client Side UI Rendering



Web Application Development

Applying software engineering skills to build applications using the Web as a framework

- **Requirements**
- **Data modeling**
- **Interprocess communication**
- **Data storage/retrieval**
- **User interface design**

Working in Teams

The code belongs to the team

- **Best practices**
- **Source control, pull requests, code review**
- **Naming conventions, understandability**
- **Application structure**
- **Planification, communication**

Working in Large Projects

- **Scalability**
- **Maintainability**
- **Software development methodologies**
 - **requirements, design, implementation, feedback, testing**
 - **incremental software development**

Architecture

- Client-server architecture
- Multitier architecture
- Separation of concern (SoC)
 - Model View Controller (MVC)
 - Object-oriented programming
 - Aspect-oriented programming

Design Patterns

- Singleton
- Factory
- Service
- Model View Controller (MVC)
- Data access object (DAO)
- Inversion of control (IOC)