

# WEB Engineering



Press Space for next page →



# Plan

- Un peu d'histoire
- JavaScript : le langage d'action du Web (mais pas que ...)
- Outilage
- TypeScript
- Utilisation de frameworks : Angular
- Utilisation de frameworks : VueJS
- Utilisation de frameworks : React
- Conclusion

# Contexte historique

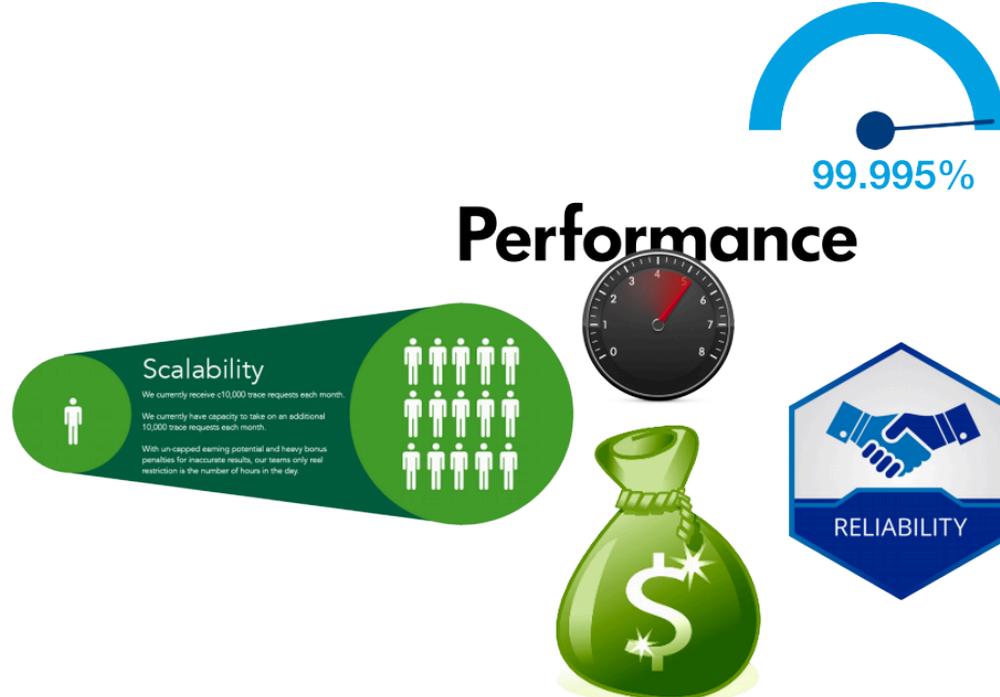
- Internet (~1980) et le Web (~1990)
- Repose sur Internet, TCP/IP, modèle OSI
- Un des services de l'Internet (port 80)
- 3 piliers : HTML, URI, HTTP
- Plusieurs évolutions
  - Séparation mise en forme (CSS), pages
  - Statiques vers interactions dynamiques
  - Cloud computing, Web sémantique, Web embarqué, systèmes pervasifs...
  - HTML5 (Conteneur d'applications complexes)

# Évolutions du Web

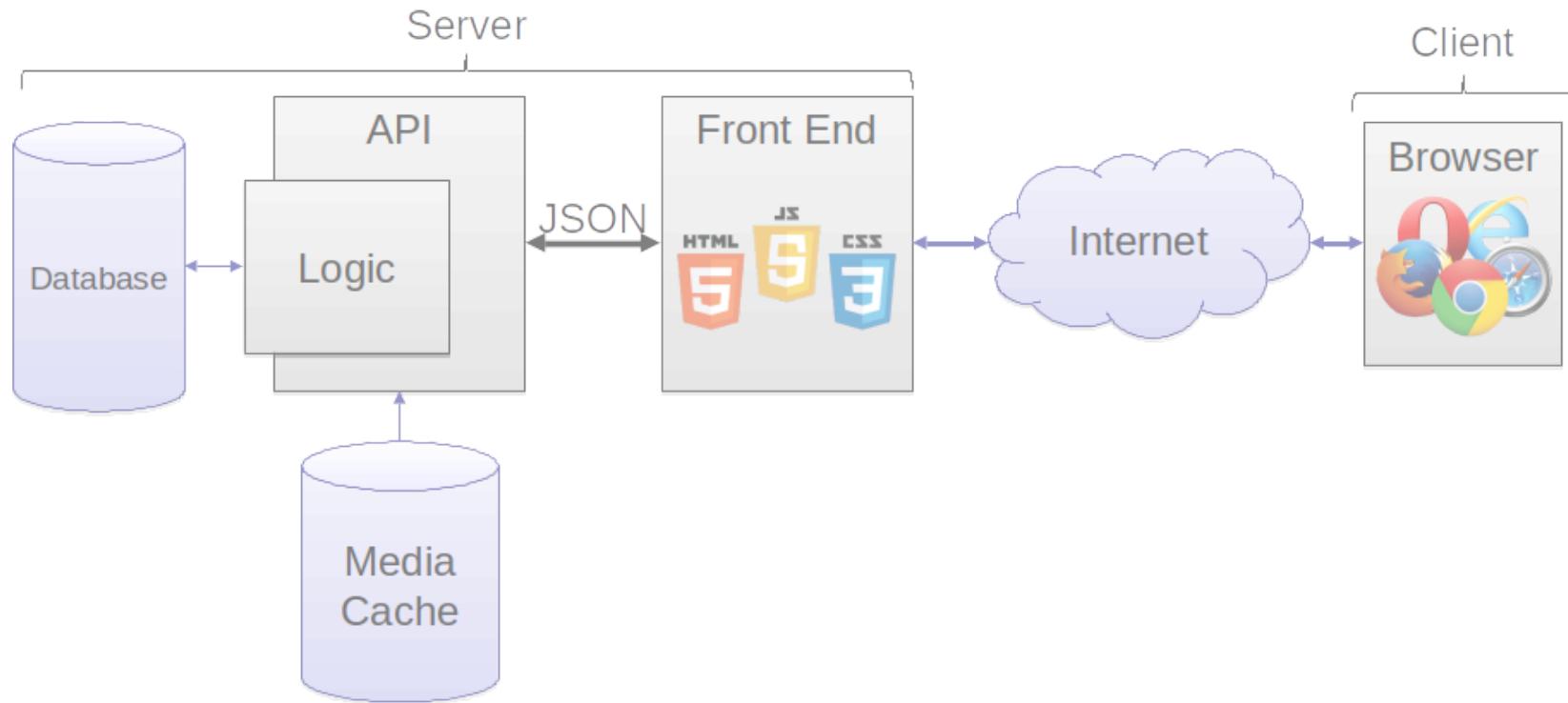
- Évolutions du Web => Web 2.0
  - Utilisateur proactif (wikipedia, blogs, etc.)
  - Interfaces riches, intégration d'applications
- Moyens techniques
  - Langage de scripts
  - côté serveur (PHP, ASP, C#...)
  - côté client (Javascript, flash...)
- Services Web (HTML => XML,JSON)
  - Échange de données entre applications

# Ingénierie Web

- Availability
- Performance
- Reliability
- Scalability
- Manageability
- Cost



# Element au coeur d'une application Web



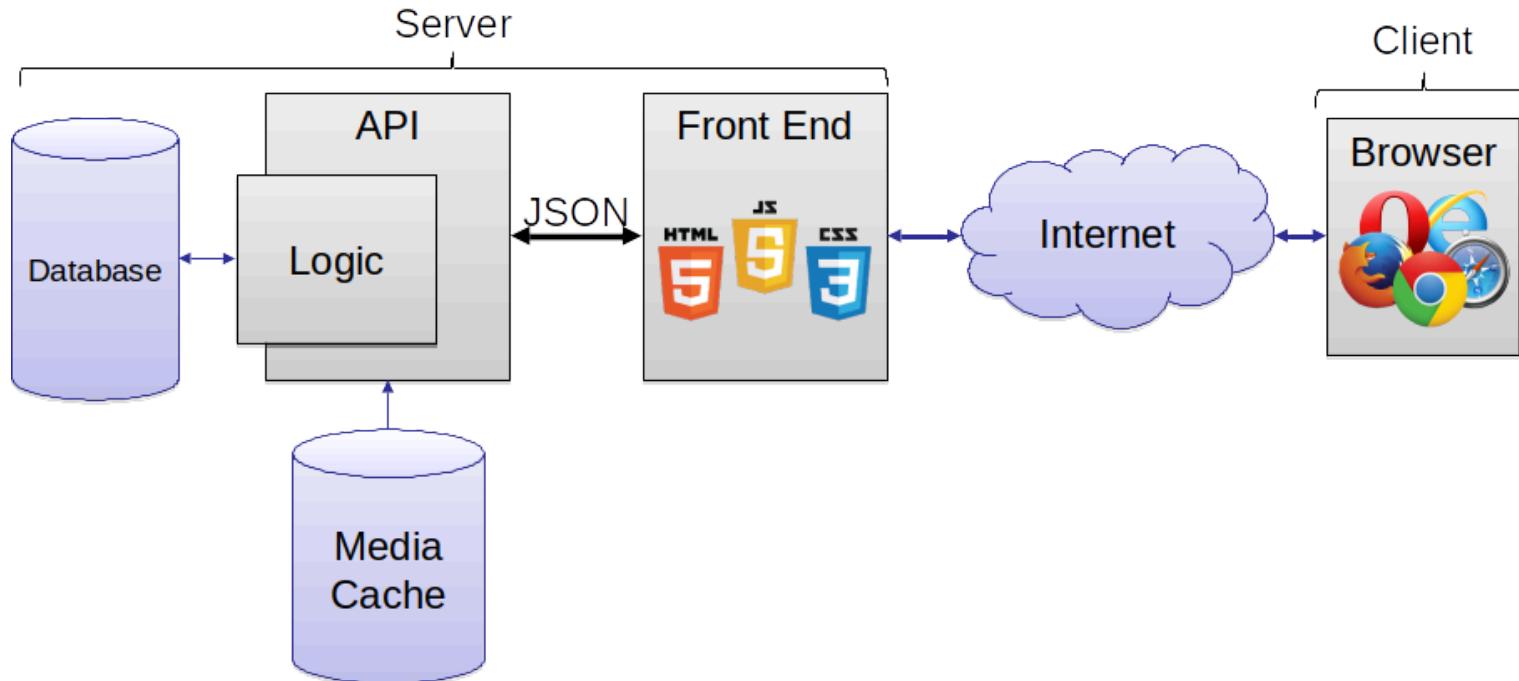
# Use of Frameworks?

# What is a Framework?

- Designed to reduce overhead in web development
- Types of Framework Architectures
  - Model-View-Controller (MVC)
  - **Push vs Pull** Based
    - Most MVC Frameworks user push-based architecture “action based” (Django, Ruby on Rails, Symfony)
    - Pull-based or “component based” (Vue, Angular2, React)
  - Three Tier Organization
    - Client (Browser running HTML/Javascript/CSS)
    - Application (Running the Business Logic) Database (Data Storage)

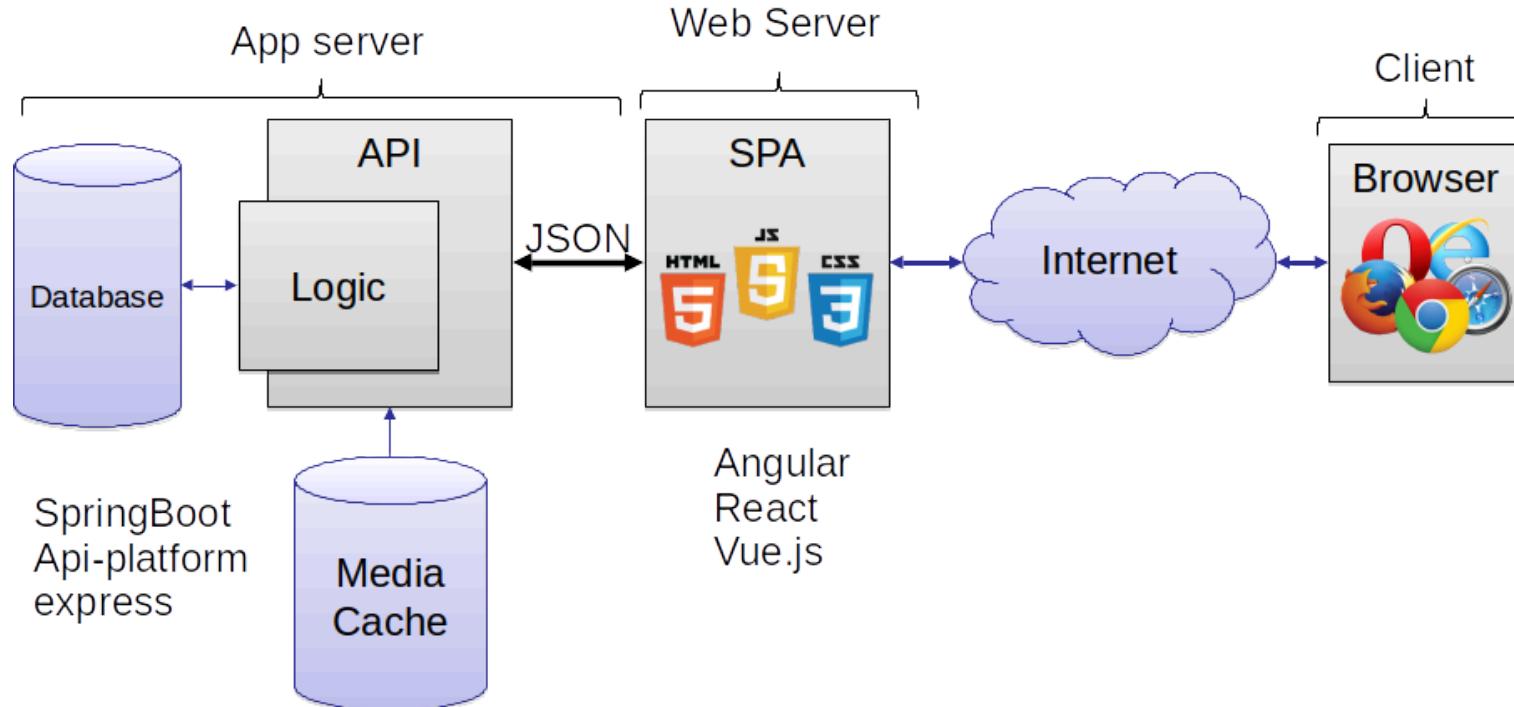
# Types of Frameworks

- Server Side: Django, symfony, Ruby on Rails



# Types of Frameworks and Single Page Application

- Server Side: Spring Boot, Express, api-platform
- Client Side: Angular, React, Vue



# Frontend development

# Front End Runtime Languages

- HTML/CSS
- Javascript

Javascript/HTML/CSS is the only real option for front-end native languages and is basically the standard. But there are many variations on JavaScript that are used.



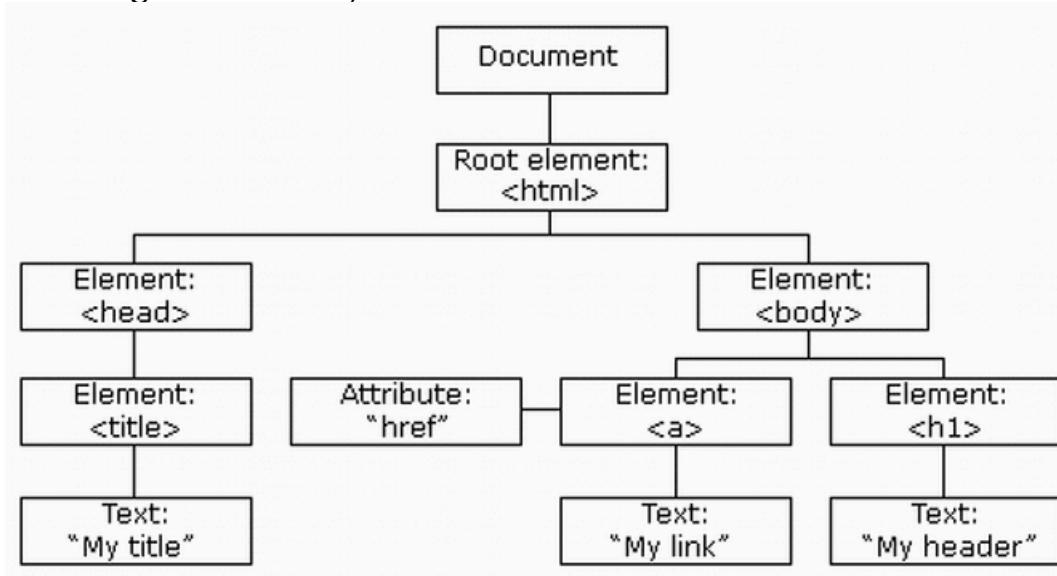
## Emerging solutions

WebAssembly or wasm is a new portable, size- and load-time-efficient format suitable for compilation to the web.

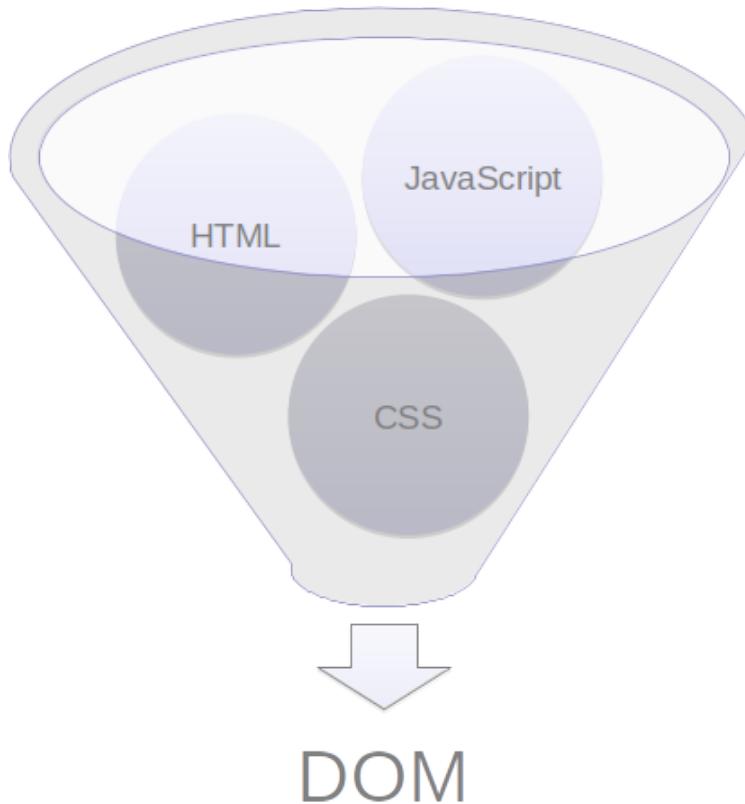
WebAssembly is currently being designed as an open standard by a W3C Community Group that includes representatives from all major browsers.

# Runtime platform

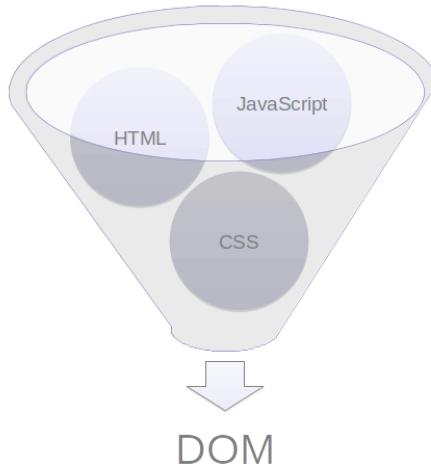
- DOM (Document Object Model)
- Document Object Model makes every addressable item in a web application an Object that can be manipulated for color, transparency, position, sound and behaviors.
  - Every HTML Tag is a DOM object



# DOM (Document Object Model)

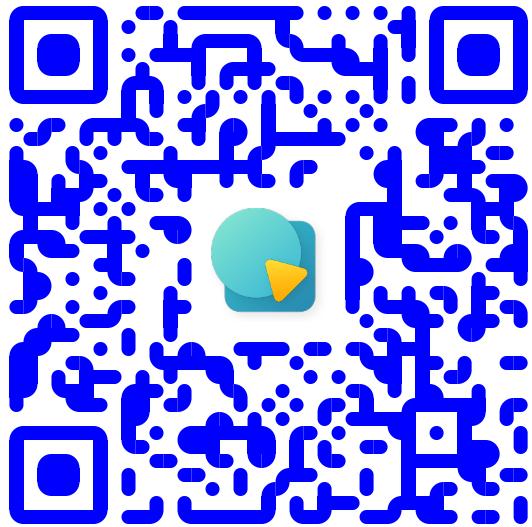


# How to create Web Applications



Est il toujours nécessaire de développer votre application  
Web à l'aide de JS/CSS/Html

The poll is not open



[http://localhost:12445/web.intro/print?](http://localhost:12445/web.intro/print?print=true)  
print=true