

DAR 2018 - Cours 1

Développements d'Applications Web

Romain Demangeon

APR, LIP6, UPMC

17/09/2018

1. Client/Serveur
2. Applications Web
3. Histoire du Web
4. Architecture des Applications

Modèle Client-Serveur

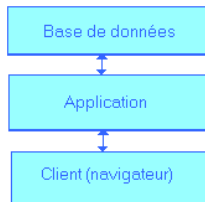
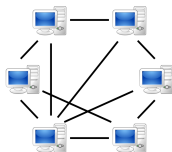
- ▶ Modèle de **communication** entre **programmes** à travers un **réseau**.
- ▶ Modèle asymétrique:
 - ▶ les **Clients** (applications, browsers, programmes ou serveurs) envoient des requêtes,
 - ▶ les **Serveurs** (puissance de calcul) traitent les requêtes et répondent.



- ▶ **Avantages:**
 - ▶ centralisation des données,
 - ▶ centralisation de la puissance de calcul (clients légers),
- ▶ **Inconvénients:**
 - ▶ centralisation des connexions,
 - ▶ peu robuste.

Architecture Client-Serveur

- ▶ **Mainframe**: machine dédiée au centre du réseau.
- ▶ **Peer-to-peer**: chaque agent joue le rôle de client ou de serveur.
- ▶ **Architecture 2-niveau**: client-serveur classique à travers le Internet.
- ▶ **Architecture 3-niveaux**: division entre serveur de **données** et serveur d'application.
- ▶ **Architecture n -niveaux**.



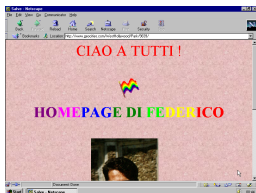
Définition

Application **client/serveur** utilisant un **navigateur** comme programme client, proposant un service **interactif** à travers une **connection** avec des serveurs sur le **Internet**.

- ▶ Site Web: propose du contenu à partir de données **statiques**
- ▶ Application Web: propose du contenu *à la carte* basée sur des **requêtes paramétrées**
 - ▶ gère les utilisateurs,
 - ▶ gère la sécurité.

Age d'or des Applications Web

- ▶ **Raison**: omniprésence des navigateurs (développement du Web sur le Internet)
- ▶ un Web de plus en plus **dynamique**
 - ▶ pages **statiques**,
 - ▶ puis interactions **dynamiques** possibles (formulaires),
 - ▶ langages de **scripts** clients (JS)
 - ▶ **interactivité** (AJAX, HTML5)



Name	Value
Name	
Sex	<input type="radio"/> Male <input checked="" type="radio"/> Female
Eye color	green
Check all that apply	<input type="checkbox"/> Over 6 feet tall <input type="checkbox"/> Over 200 pounds
Describe your athletic ability:	
<input type="button" value="Enter my information"/>	



Exemples d'applications



The screenshot shows the Wikipedia article for "Web application". The article text explains that web applications are executed client-side and are accessed through a web browser. It also includes a section on "Web-based application" which notes that such applications use a browser as a client and are often written in JavaScript. A callout box highlights the need for additional citations for verification. The interface includes a search bar, navigation tabs (Article, Talk, Read, Edit, View history), and a sidebar with various Wikipedia navigation options.

Web application

From Wikipedia, the free encyclopedia

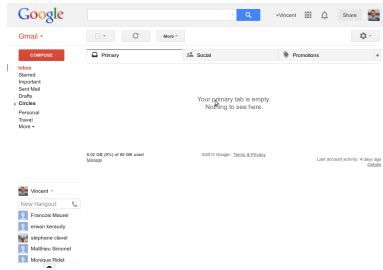
For applications accessed through the web that are executed client-side, see [Rich Internet application](#).

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unourced material may be challenged and removed. (January 2010)

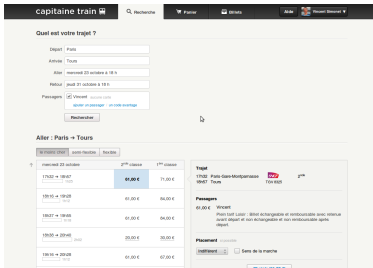
A **web-based application** is any application that uses a web browser as a client.^[citation needed] The term may also mean a computer software application that is coded in a browser-supported programming language (such as JavaScript), combined with a browser-rendered markup language like HTML and reliant on a common web browser to render the application executable.

Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a

Google Calendar is a contact- and time-management web application offered by Google.



The screenshot shows a Gmail inbox interface. At the top, there is a search bar and navigation icons. The main area displays a list of emails, with the primary tab selected. The inbox is currently empty, with a message stating "Your primary tab is empty. Nothing to see here." The interface includes a sidebar with navigation options like "Compose", "Inbox", "Starred", "Important", "Sent Mail", "Drafts", "Circles", "Personal", "Trash", and "More". The bottom of the page shows the user's name "Vincent" and some account activity information.



The screenshot shows the "capitaine train" website, which is a travel agency. The main heading is "Quel est votre trajet ?" (What is your journey?). Below this, there are several input fields for "Depart", "Arrive", "Date", and "Hours". There is also a "Messages" section with a "Rechercher" button. The main content area is titled "Aller : Paris -> Tours" and displays a table of train schedules. The table has columns for "Train", "Départ", "Arrivée", "Durée", "Prix", and "Statut". The "Train" column lists "TGV" and "TGV*". The "Départ" column shows times like "17:52" and "18:02". The "Arrivée" column shows times like "01:00" and "04:00". The "Prix" column shows prices like "61,00 €" and "71,00 €". The "Statut" column shows "OK" and "OK*".

Train	Départ	Arrivée	Durée	Prix	Statut
TGV	17:52	01:00	05:08	61,00 €	OK
TGV*	18:02	04:00	05:58	71,00 €	OK*



The screenshot shows a Facebook post from the user "Vincent". The post is titled "Facebook in a portrait (album by Boston of New York)" and includes a photo of a woman with tattoos. The post text reads: "Yesterday, we had a special visit at Facebook HQ from Brandon Stanton, the photographer behind Humans of New York." Below the photo, there is a caption: "Photo from Lorian - Boston photographer et photographe avec internet avant d'être un photographe et son redouteable agent support." The post also includes a "Partager" button and a "Rechercher" button. The interface shows the user's profile picture, name, and location (Boston, MA).

- ▶ **Interface** ergonomique: éditeur de texte, *drag n'drop*, raccourcis claviers.
- ▶ **Multimedia**: audio, vidéo, jeux.
- ▶ **Avantages**:
 - ▶ facile à déployer, mettre à jour,
 - ▶ interopérabilité client,
 - ▶ charge de travail, espace mémoire réduits pour le client,
 - ▶ multiplateforme (téléphones, tablettes, consoles, télévisions),
- ▶ **Inconvénients**:
 - ▶ interface limitée (HTML5 ?),
 - ▶ dépendant des navigateurs,
 - ▶ dépendant d'une connexion (HTML 5?),
 - ▶ déplacement du rapport de force vers les entreprises:
 - ▶ collection de données,
 - ▶ vers un monde informatique propriétaire.

Histoire du Web

- ▶ Internet:

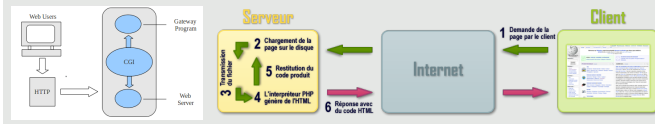
- ▶ Web:

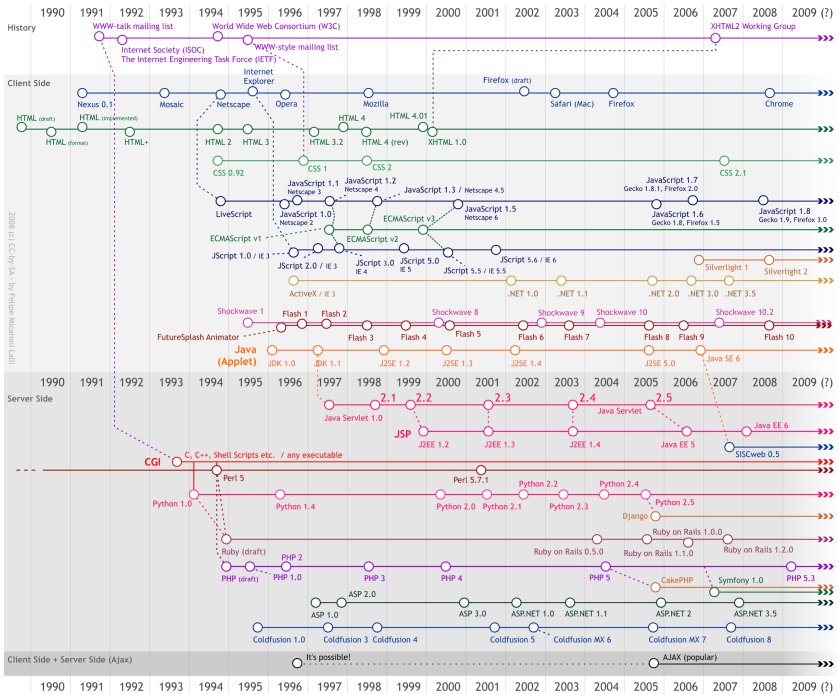
Histoire du Web

- ▶ **Internet**: réseau mondial
 - ▶ medium du web, de l'email, du chat, du FTP, de SSH ...
- ▶ **Web**: système hypertexte public
 - ▶ application de Internet - port 80

Evolutions du Web

- 1993 CGI (génération de contenu par un programme)
- 1995 PHP 1.0 (pages web dynamiques)
- 1995 JavaScript (langage de script pour client)
- 1999 Servlet Java (CGI-like de haut niveau)
- 2005 AJAX (page dynamiques asynchrones)
- 2008 HTML5 - draft



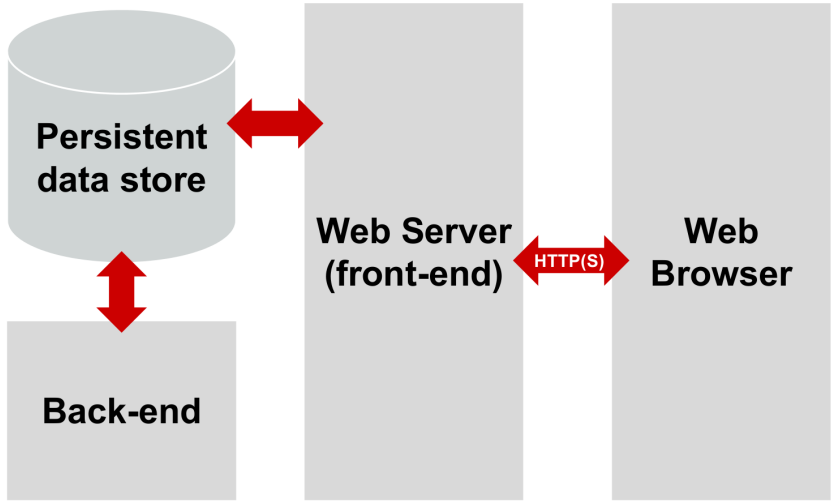


2008 (c) CC-by-SA - by Felipe Micaroni Lalli

Transfert de charge

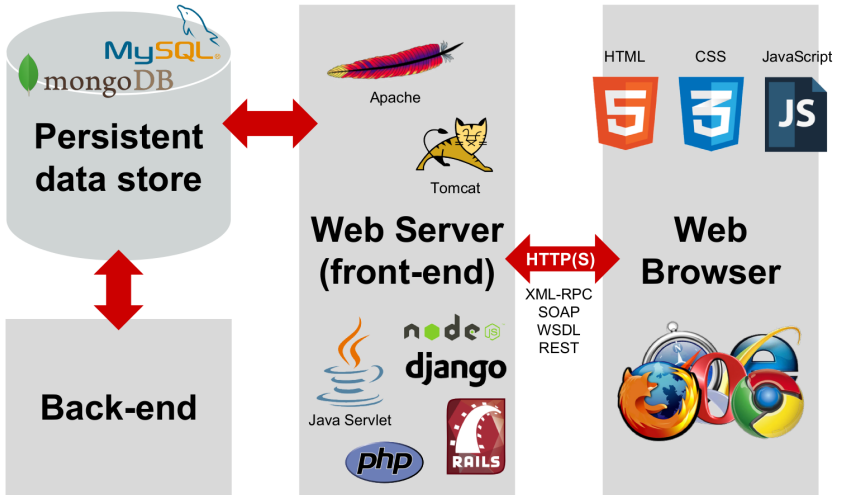
- 1970 Terminaux légers, tout est fait sur le serveur,
- 1980 Ordinateurs personnels (programmes), tout est fait sur le client,
- 1990 Clients légers (navigateurs), logique dans le serveur.
- 2000 Retour de la logique dans les clients (“Web 2.0”).
- 2010 Applications mobiles.

Architecture classique d'une application web



- ▶ **Navigateur**
 - ▶ interface utilisateur,
 - ▶ **état** à court terme,
 - ▶ peut implémenter de la **logique** (confiance ?),
 - ▶ communique avec le serveur web via HTTP(S),
 - ▶ exécute du code HTML, CSS, JS.
- ▶ **Serveur Web**
 - ▶ répond aux requêtes,
 - ▶ sans **état**,
 - ▶ lit et écrit dans le serveur de données,
 - ▶ responsable de la **logique**,
 - ▶ comporte un serveur/*container* et un système pour la **logique**.
- ▶ **Serveur de Données**
 - ▶ **état** de l'application web,
 - ▶ point de synchronisation.
- ▶ **Back-end**
 - ▶ **logique** du serveur indépendant du client.

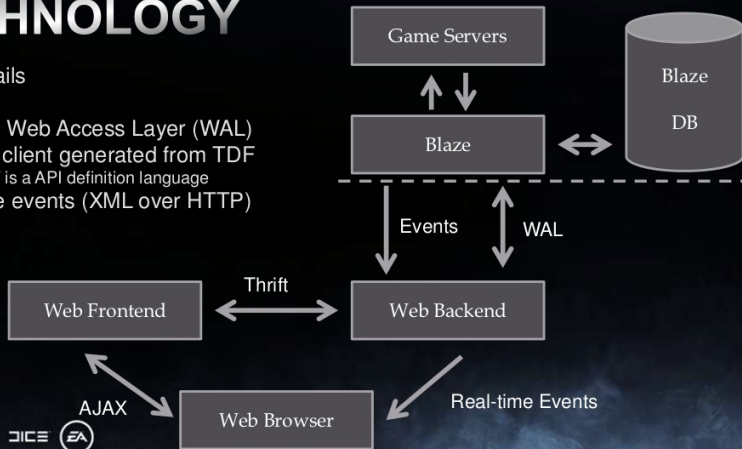
Technologies



TECHNOLOGY

Blaze Details

- › Uses Web Access Layer (WAL)
- › WAL client generated from TDF
 - › TDF is a API definition language
- › Blaze events (XML over HTTP)



Exemple d'architecture de l'application web associée à un jeu vidéo.

Tendances actuelles de la **recherche**:

- ▶ **Frontière Client/Serveur**: Ocsigen, Hop, node JS, ...
- ▶ **Omniprésence de JS**: compilateurs **vers** JS, ...
- ▶ **Meta-données**: collecte, stockage, traitement, apprentissage, ...
- ▶ **Sécurité**: identification, usurpation, ...