

HTTPSLOW v.4.1

- ✓ **01.06.22** TH Première rédaction (01/06/2022)
- ✓ **03.06.22** C2D Relecture et V.2
- ✓ **03/06/2022** C2D + TH Relecture et repérage des modifications et ajouts à faire

=> Répartition des tâches pour finaliser si possible le 12/06/2022

- ✓ **06.06.22** C2D modifications et ajouts jusqu'à **Maquettage** publication en v.3
- ✓ **07.06.22** TH ajout de bouts de texte suggérés.
- ✓ **08.06.22** TH modification sur le mandataire
- ✓ **09.06.22** TH ajout d'un service passerelle confié aux CHATONS ; compléments pour les commentaires de l'état de l'art.
- ✓ **10.06.22** C2D prochaine étape modifications et ajouts
- ✓ **11.06.22** TH relectures et quelques modifications cosmétiques
- ✓ **11.06.22** C2D relecture, ajouts de liens pour les sources, corrections orthographiques et lexicales, bibliographies et crédits
- ✓ **12.06.22** TH relecture
- ✓ **13.06.22** C2D Ok pour moi

- HTTPSLOW v.4.1
 - Low-technicisation et numérique, une formation Libre Cours / UPLOAD
 - Libre Cours
 - Les outils de communication
 - Sujet proposé
 - Résumé
 - Piste complémentaire
 - HTTPSLOW - Le Pitch by Audrey
 - Contexte technique
 - L'enjeu HTTPS - LOW
 - Idéation lowtechnicisante
 - Identification des acteurs
 - Idéation à partir des interactions humaines (utilisateur/éditeur)
 - Idéation à partir des interactions techniques (navigateur/serveur web)
 - Résultat de l'idéation
 - Ajouter des méthodes HTTP
 - Compléter les en-têtes HTTP
 - Première itération: Maquettage par des récits d'utilisateurs
 - Niveau Utilisateur
 - Niveau Éditeur
 - Niveau Services
 - Première évaluation: le risque de complexification
 - Boucle C->K->C : maquettage <-> idéation
 - Les hébergeurs
 - Seconde itération: représentations fonctionnelles
 - Les fonctionnalités d'un mandataire HTTP
 - Les fonctionnalités d'une passerelle
 - Des formes d'automatisation
 - Seconde évaluation: repérer les fantômes
 - Quelle.s empreinte.s fantôme.s pour HTTPSLOW ?
 - Vers une matrice SWOT HTTPSLOW ?
 - Un code éthique d'hébergeurs pour lutter contre les 7 péchés capitaux du GreenWashing ?
 - Publication et suites possibles
 - Ressources




- Les bases documentaires
- Crédits, licence et... merci !

Low-technicisation et numérique, une formation Libre Cours / UPLOAD








Libre Cours (<https://librecours.net/parcours/upload/lownum/index.html>)

Les outils de communication

-  MatterMost (<https://team.picasoft.net/librecours/channels/lownum>) canaux *Lownum* et *LOWNUM-projet-httpsloweb*
-  Picasoft (<https://picasoft.net/>)
-  Mastodon (<https://framapiaf.org/web/timelines/tag/lownum>) avec les mots-clefs *#lownum*, *#librecours* et *#upload*

L'enjeu pour les participants à la formation : mobiliser une réflexion autour de projets numériques low-technicisants à partir des ressources transmises, dans un contexte ouvert, sur une période courte (6 semaines) et rythmée par une conférence hebdomadaire :

-  The Big Conf : sensibilisation aux enjeux climatiques (<https://tube.picasoft.net/w/3ce4fisgLzYrpxxezYnhRt>)
-  Introduction à la low-technicisation (<https://tube.picasoft.net/w/acjF1dPwgmeEczhJuH9kfU>)
-  Présentation de la démarche et des outils lownum (<https://tube.picasoft.net/w/75NSDpPYuoVrbLpENQfQX4>)
-  le concept d'empreinte fantôme (<https://tube.picasoft.net/w/cQ4vgXfRRUUnxDgReKdEVB>)
-  Informatique et soutenabilité (<https://tube.picasoft.net/w/tRWCxsP2T566KTVzQTdq8r>)

Le projet sélectionné par les participants a servi de fil rouge à une démarche de réflexion comprenant la constitution d'un état de l'art, une phase d'idéation dans l'objectif d'une conception low-technicisante, une phase maquettage, une évaluation pour prendre un recul critique, et finalement la publication d'un document de synthèse.

Ce document est le rendu polié de la démarche suivie pour le projet HTTPSLOW. La première phase d'évaluation ayant suscité de nouvelles interrogations, nous avons formalisé une seconde boucle idéation - maquettage, avec une amorce d'évaluation pour le passage de relais vers les prochaines sessions #LowNum (voire toute personne intéressée par le sujet).

Sujet proposé

Résumé

Proposer un protocole permettant aux sites et aux navigateurs de communiquer afin de minimiser le poids des informations échangées.

L'enjeu est de permettre aux terminaux des utilisateurs de réclamer des ressources web plus simples, plus légères, plus facilement appropriables. Exemples : pas de pub, pas de vidéos, version audio à la place de vidéo, images minimisées, code HTML standard sans CSS ni JavaScript, qualité du code HTML, version résumée des pages longues...

Piste complémentaire

Intégrer des protocoles de rémunération de type micro-tipping pour remplacer la pub (cf Xanadu de Ted Nelson...)

Notes:

1. La piste complémentaire n'a pas été abordée;
2. Pour simplifier un travail déjà complexe, choix a été fait de faire des ajouts à HTML plutôt que de bâtir un nouveau protocole.

HTTPSLOW - Le Pitch by Audrey

 Version audio (https://hubcontent.ecovalence.fr/pod/PGen/media/2022-06-06_lownum-httpslow_le-pitch-by-audrey.mp3)

L'idée : proposer un protocole permettant aux sites et aux navigateurs de communiquer afin de minimiser les informations échangées

Contexte technique

- Qu'est-ce qu'un protocole ?
- Qu'est-ce que HTTP ?

Lorsqu'on navigue sur le web, l'affichage des pages que l'on consulte est le fruit d'un échange d'informations qui a lieu entre le navigateur (firefox chrome par exemple) et les serveurs qui hébergent les pages web en question. Et pour être en mesure de consulter le contenu de ces pages web, il est nécessaire que ce contenu nous parvienne dans un format donné et selon des règles précises que l'on appelle donc des protocoles.

HTML est le format dans lequel le contenu d'une page web est codé puis transmis au

navigateur et HTTP quant à lui, c'est justement le protocole de transfert qui est utilisé pour transmettre des pages web codées HTML depuis le serveur où elles sont hébergées vers le navigateur qui les consulte.

On y ajoute le s de “*secure*” - sécurisé - lorsque l'identité du site que l'on consulte a été vérifiée ou lorsque le transfert d'informations est chiffré.

C'est ainsi qu'on obtient alors le protocole HTTPS

L'enjeu HTTPS - LOW

C'est de permettre à des utilisateurs de réclamer des ressources web plus simples, plus légères, plus facilement appropriables. C'est-à-dire, par exemple, minimiser les informations échangées : pas de pub, pas de vidéo, des versions audio à la place des vidéos, des images minimisées, un code HTML standard qui soit sans CSS ni javascript, c'est-à-dire sans ce qui concerne la mise en forme graphique, les interactions, etc... sur une page web.

“Simplifiez... simplifiez ! Il en restera toujours quelque chose...”

HTTPSLOW est un projet mené par une équipe de quatre personnes : David, Céline, Thierry et Sylvain, qui ont choisi de porter un regard low-technicisant sur le protocole HTTP

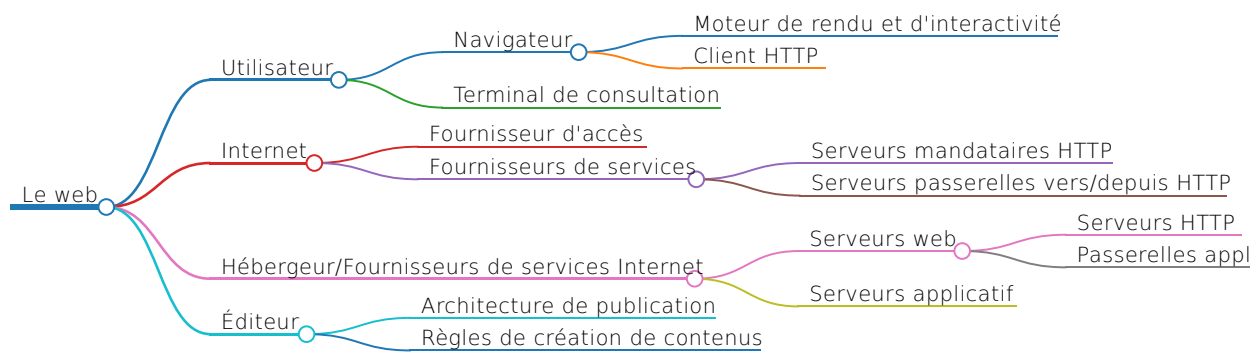
Idéation lowtechnicisante

Le sujet se focalisant sur les échanges protocolaires, les interactions entre machines, l'exercice d'idéation a porté sur les interactions HTTP entre la machine (client) et la machine (serveur).

Cependant **ce sont des choix humains de programmation qui définissent les règles d'interactions entre machines**, il nous est donc apparu crucial de tenir compte dans un premier temps de cet aspect du sujet.

Identification des acteurs

La première étape a consisté à identifier des acteurs (humains, organisations, machines) intervenant dans Web ainsi que leur place dans les échanges d'informations en jeu (communication tant sociale que technique).

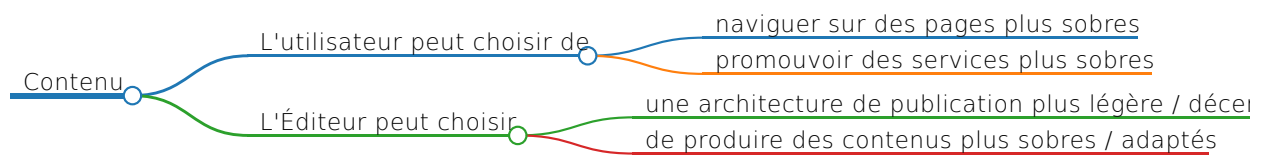


Synthétiser est toujours une opération à la fois complexe et frustrante. Dans la réflexion qui nous occupe, nous avons tenté d’aborder la question du protocole sous l’angle qui nous a semblé le plus accessible : différencier les interactions humaines et les interactions machines

- L'utilisateur (comme lecteur de contenu) interagit avec l'éditeur (comme producteur de contenu)
- Le navigateur (qui agit pour le compte de l'utilisateur) interagit avec le serveur web (qui met à disposition le contenu mis en ligne par l'éditeur)

Idéation à partir des interactions humaines (utilisateur/éditeur)

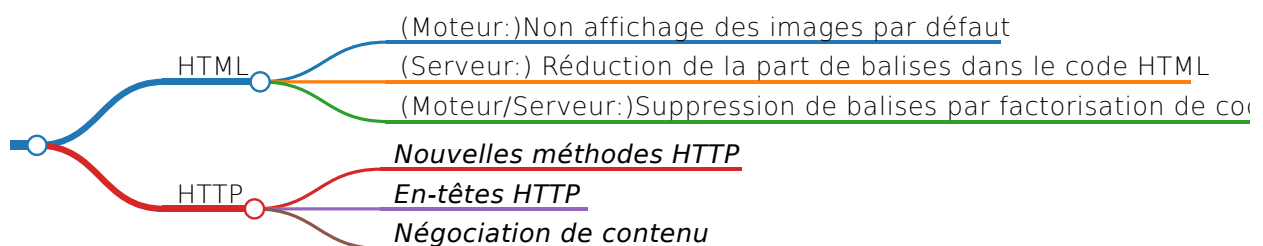
Les acteurs humains sont à la fois demandeurs et prescripteurs: quelle que soit leur place dans la chaîne de l'information, ils peuvent agir dans le sens d'un web plus sobre. On peut parler ici de notion d'engagement, tout comme d'action politique au sens large, correspondant à la sphère **"Alerter"** proposée dans UPLOAD/LowNum/Libre Cours (<https://tube.picasoft.net/w/acjF1dPwgmeEczhJuH9kfU>) par Stéphane Crozat (voir aussi 🚩 "Tous des Amish ?" (<https://aswemay.fr/co/040012.html>)).



Idéation à partir des interactions techniques (navigateur/serveur web)

Les acteurs humains agissent au travers d'artefacts techniques (navigateurs, serveurs,...).

Ceux-ci reflètent dans leur choix, voire l'augmente par les fonctionnalités proposées. Ce niveau d'interaction technique correspond à la sphère “**optimiser**” proposée par Stéphane Crozat (Voir [Ingénierie, low-technicisation et numérique - Séminaire CRED](https://stph.scenari-community.org/pres/20220307-cred-lownum/co/meso.html?mode=html) (<https://stph.scenari-community.org/pres/20220307-cred-lownum/co/meso.html?mode=html>)).



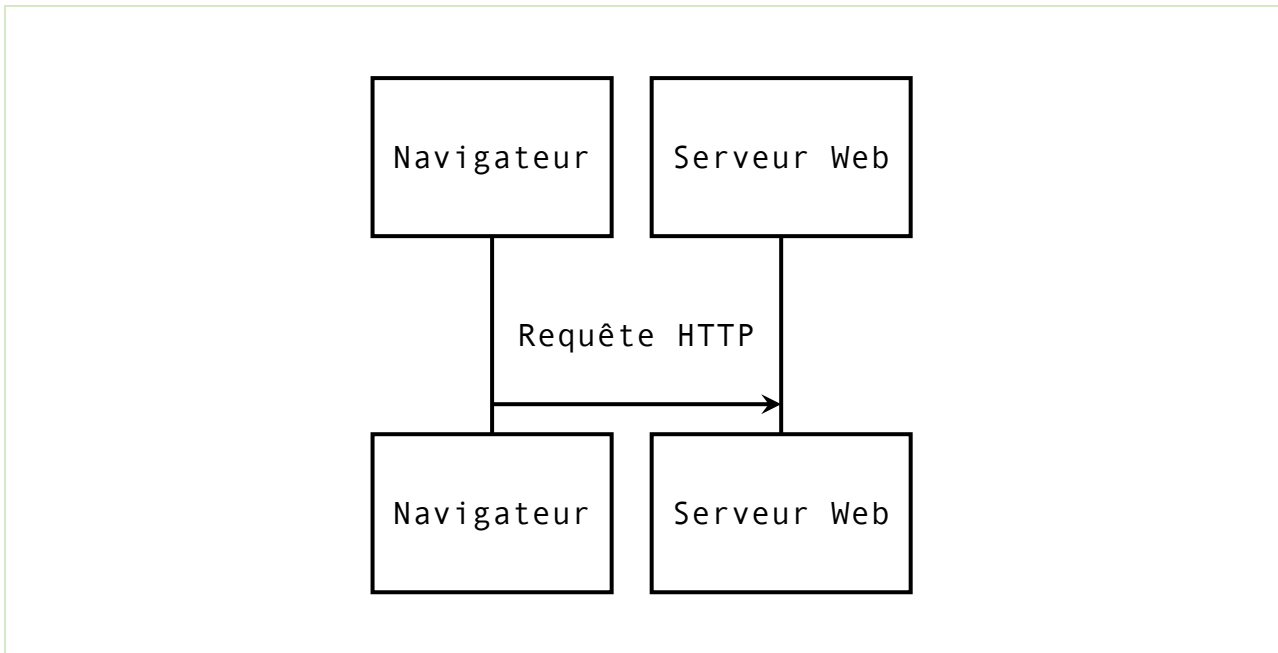
Résultat de l'idéation

Pour aboutir au résultat recherché, on peut **imaginer** ajouter au protocole HTTP des

méthodes supplémentaires , mais aussi compléter les en-têtes pour permettre à l'utilisateur de préciser "la finesse" du document qu'il souhaite obtenir selon ses choix ou les capacités de sa machine (lowtech en amont et/ou en aval)

Remarque liminaire

Quand un navigateur demande une page web à un serveur, il le fait au moyen d'une **requête HTTP**,



Une requête HTTP est précise l'action demandée (ici demande d'une page) par une **méthode HTTP**, et des précisions techniques diverses sur le document demandé au travers des **en-têtes HTTP**.

Par exemple, quand un navigateur demande la page **<https://librecours.net/parcours/upload/lownum/>** (<https://librecours.net/parcours/upload/lownum/>), celui-ci transmet au serveur web "librecours.net (<http://librecours.net>)" la requête suivante

```
GET /parcours/upload/lownum/ HTTP/2
Host: librecours.net
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
           Gecko/20100101
           Firefox/91.0
Accept: text/html,
       application/xhtml+xml,
       application/xml;q=0.9,
       image/webp,
       */*;q=0.8
Accept-Language: fr-FR,fr;q=0.5
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Pragma: no-cache
Cache-Control: no-cache
TE: trailers
```

Sans rentrer dans les détails, il est demandé (méthode **GET**), la ressource `parcours/upload/lownum/` sur le serveur “librecours.net (<http://librecours.net>)”

```
GET /parcours/upload/lownum/ HTTP/2
Host: librecours.net (http://librecours.net)
```

La ressource doit si possible être en français

```
Accept-Language: fr-FR,fr;q=0.5
```

et en priorité un document HTML

```
Accept: text/html,
       application/xhtml+xml,
       application/xml;q=0.9,
       image/webp,
       */*;q=0.8
```

Ajouter des méthodes HTTP

- **ASK (*)** : le navigateur demande de récupérer la page Web par mél (fourni dans l'entête `From:`). La consultation des pages Web est donc asynchrone.
- **ABSTRACT (*)** : le navigateur demande un résumé de la page (c'est en quelque sorte

le pendant non technique de la méthode HEAD)

Compléter les en-têtes HTTP

- `User-Agent` : Compléter le nom du navigateur par les ressources matérielles disponibles localement (taille RAM, fréquence CPU, présence d'un GPU)
- `Accept` : Ajouter des types MIMES supplémentaires spécifiant plus finement le type de document attendu par l'utilisateur. Par exemples
 - `text/html ; layer=N` : limiter le nombre d'enchâssement de balises
 - `text/html+noscript`: page n'utilisant pas de scripts
 - `text/html+textonly`: page ne contenant que du contenu textuel
 - `text/html ; image_sizemax=N`: n'inclure que les images de taille choisie
 - `text/html ; image_volumemax=N`: le volume total des images est à limiter
 - `text/html+ditherimages`: appliquer un filtre de tramage aux images
 - `text/html+noaudio`: supprimer les flux audio
 - `text/html+novideo`: supprimer les vidéos
- `Require` : ce nouvel en-tête a le même rôle que `Accept` mais **oblige impérativement** le serveur à retourner une réponse du type attendu.

Première itération: Maquettage par des récits d'utilisateurs

Sont présentés ici sous forme de récits d'usages des scénarii possibles reposant sur HTTPSLOW

Niveau Utilisateur

Scénario 1 (Merci à Steph)

Bob a juste besoin d'accéder aux services administratifs en ligne obligatoire et l'état fournit à présent une version simplifiée de tous ses formulaires. Le vieil ordinateur de Bob, malgré ses 20 ans, tourne sous le système OSlow qui implémente httpslow et est donc suffisant pour ces tâches.

Scénario 2

Anselme n'est pas un professionnel des ordinateurs. Il sait cependant que l'administration suggère d'utiliser les sites web pour effectuer ces démarches pour plus de confort et de simplicité (de toute façon les guichets physiques sont toujours fermés, pour plus de confort et de simplicité à Bercy). Il utilise donc le navigateur WaterGoose qu'il sait demander des pages web simplifiées pour être consultées et manipulées sur des machines avec peu de ressources. Il doit cependant passer par un mandataire situé chez le Chatons de sa Communauté de Communes, la plupart des sites étant conçu pour les dernières versions des navigateurs classiques.

Niveau Éditeur

Scénario 1

Adalbert participe au développement d'un système de gestion de contenu en code source ouvert. Il vérifie que chaque fois qu'une balise est insérée dans le code HTML des pages produites par ses modifications celle-ci est réellement utile. Il s'appuie en partie sur sa connaissance fine d'HTML, sur des logiciels de test de pertinence de balise. Cela est cependant chronophage, ce qui est peu apprécié de sa hiérarchie.

Scénario 2 (Merci à Steph)

La chaîne éditoriale de document structuré que Charlie utilise produit deux versions des contenus web, une version légère et une version plus complète. Le protocole httpslow permettra de ne demander que la version légère. La SCOOP dans laquelle travaille Charlie a décidé que par défaut, c'était la version slow qui était renvoyée, ceux qui veulent la version complète doivent implémenter le protocole httpslow et la demander explicitement, ça permet au protocole de se diffuser... surtout depuis que Wikipédia a décidé de faire pareil !

Niveau Services

Hébergeur

Conscient des enjeux énergétique, l'hébergeur «Obligations Vertes pour l'Hébergement» décide d'ajouter par défaut des machines analysant le code HTML produit par les serveurs Web de ses clients et fait remonter pendant un an les remarques à ceux-ci. Au bout d'un an les machines remontent non seulement les remarques mais également nettoient le code HTML produit ; le client peut bien évidemment éviter le nettoyage de ses pages web en changeant d'hébergeur.

Serveur

Euphrasie participe activement à la nouvelle version du serveur web Sioux. Elle et son équipe sont très attentifs à suivre les dernières RFC du protocole HTML9999 qui implémente la méthode ASK. Celle-ci permet aux internautes de demander des documents Web qui seront générés de façon asynchrone par les passerelles applicatives Web, puis retournés par mél à l'adresse du demandeur. Cette méthode, qui aurait été curieuse aux usagers du XXIème siècle, sera obligatoire dans les prochaines années. En effet, il deviendra nécessaire de réutiliser des machines des années 1980 puisque l'exploitation des terres rares, en voie d'épuisement, est réservée aux applications médicales.

Première évaluation: le risque de complexification

Ajouter des fonctionnalités techniques, qui complexifient les interactions clients et serveurs semble antinomique avec le but recherché.

Cependant la **complexification** ajoutée nous semble **minime côté navigateur** et peut aller dans le sens d'une plus grande convivialité (donner aux usagers le moyen de contrôler les en-têtes que réclame son navigateur) et permet de supprimer certaines fonctionnalités (trivialement, si l'utilisateur est aveugle, il n'a aucune raison d'utiliser un navigateur qui réclamera et affichera des images).

Si la complexification est plus grande côté serveur, il est cependant imaginable que lors d'une phase de négociation de ressource, le traitement de la demande de l'utilisateur soit déléguée si besoin à un serveur gérant **uniquement** les nouvelles fonctionnalités.

Boucle C->K->C : maquettage <-> idéation

Note: La théorie C-K

Il s'agit une théorie de la conception qui :

- offre une formalisation complète de la conception indépendamment de tout domaine de conception.
- explique l'invention, la création, et la découverte dans un même cadre comme processus de conception.

Le nom de la théorie est basé sur son principe central : la distinction entre deux espaces :

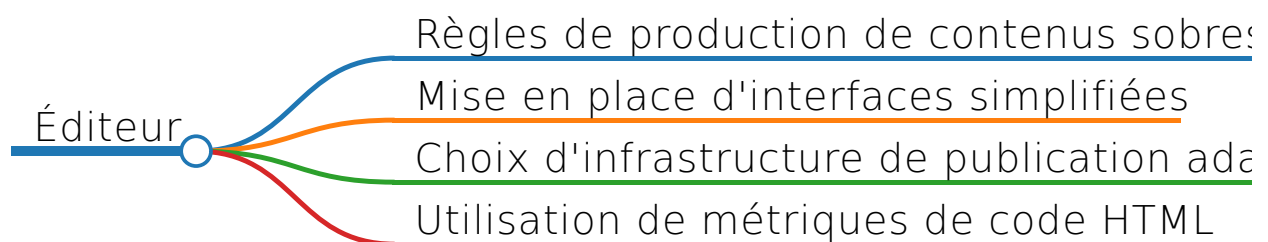
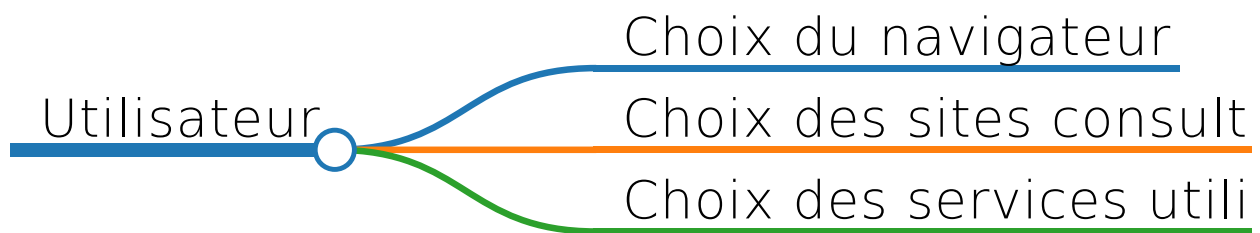
- un espace de concepts C (Concept)
- un espace de connaissances K (Knowledge)

Le processus de conception est défini comme une double expansion des espaces de C et K par l'application de quatre types d'opérateurs : $C \rightarrow C$, $C \rightarrow K$, $K \rightarrow C$, $K \rightarrow K$.

source : https://fr.wikipedia.org/wiki/Théorie_C-K (https://fr.wikipedia.org/wiki/Th%C3%A9orie_C-K)

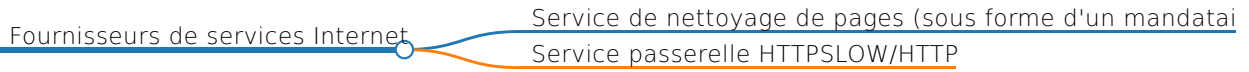
On peut voir, au travers des histoires précédentes que, lorsque des réponses low-technicisantes sont imaginées, elles peuvent comporter un risque de complexification. Pour clarifier les processus en jeu et repérer les acteurs opérants, nous avons identifié deux typologies de processus d'actions "humaines" :

- Les **actions positives**, issues de la volonté des groupes d'acteurs "*Utilisateur*" et "*Editeur*".
- Les **actions passives**, issues des "*interactions machines*", qui peuvent être transparentes au regard des groupes d'acteurs précédents.



Apparaît alors un troisième groupe d'acteurs : **les hébergeurs** et qui correspond à l'échelon intermédiaire des actions positives des groupes *Utilisateur* et *Éditeur*. En tant qu'hébergeurs, ils opèrent non seulement des choix dans le catalogue de prestations de services qu'ils proposent en "back-office" et en "front-end", mais aussi dans la mise en place des spécificités techniques des serveurs, par exemple.

Les hébergeurs



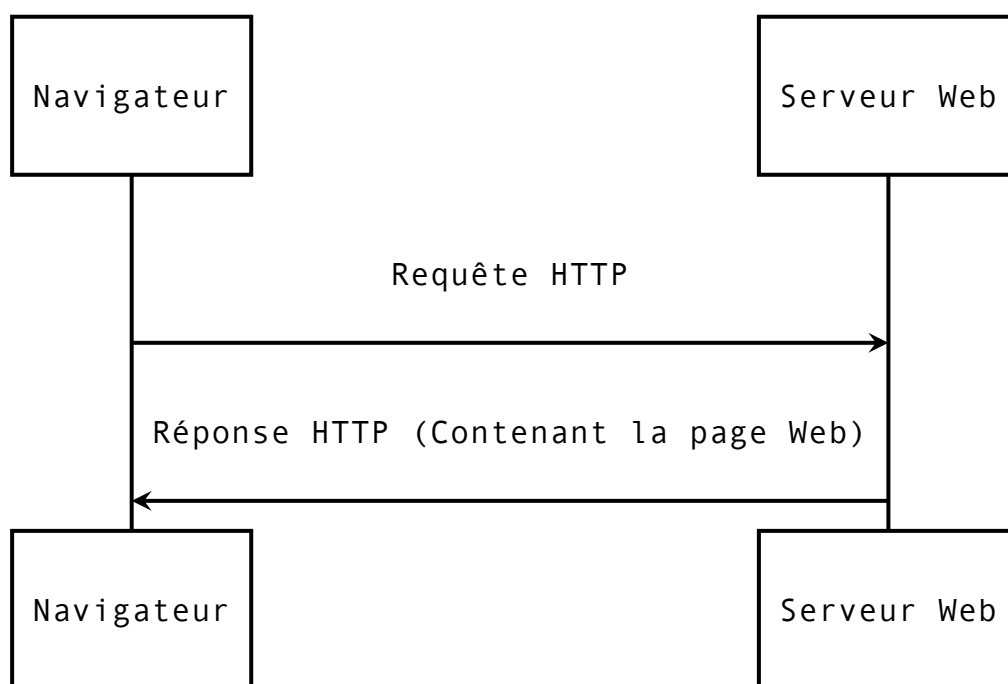
Seconde itération: représentations fonctionnelles

Les fonctionnalités d'un mandataire HTTP

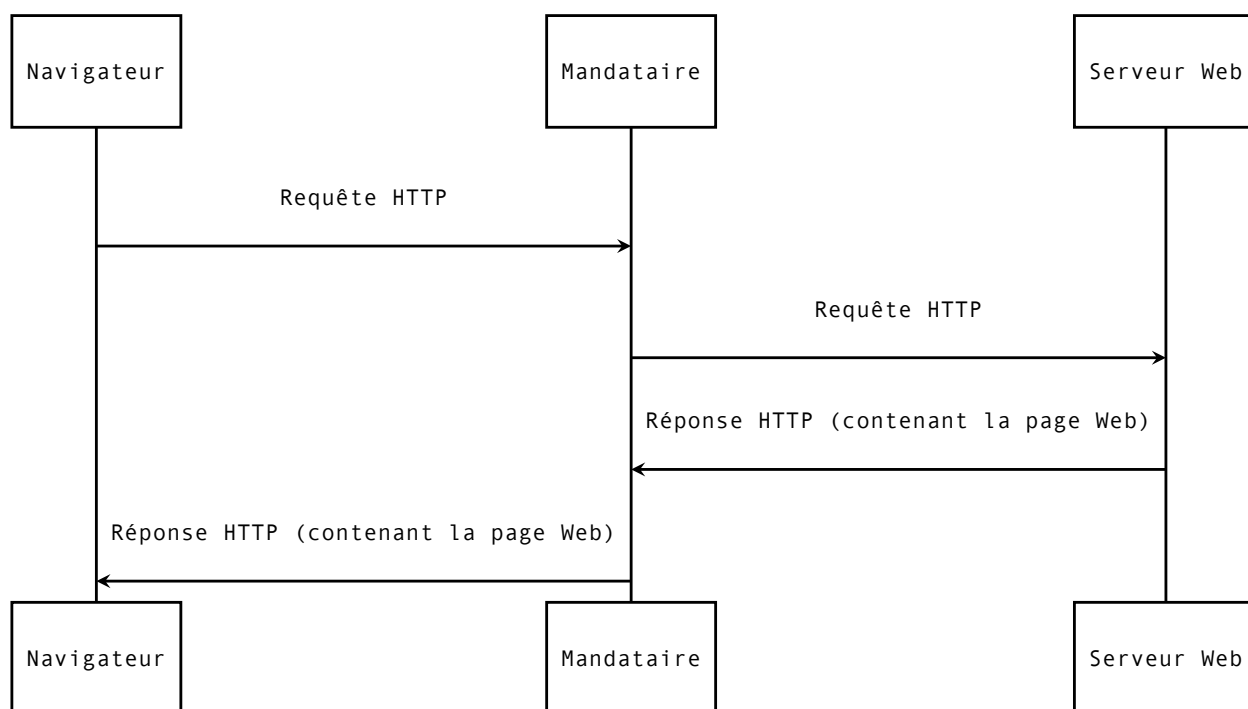
Vocabulaire:

Un **mandataire HTTP**, aussi appelé *proxy HTTP* est un serveur HTTP qui s'interpose entre un navigateur-client HTTP et un serveur HTTP et sert d'intermédiaire dans les transactions client/serveurs.

Sans mandataire, l'échange HTTP est

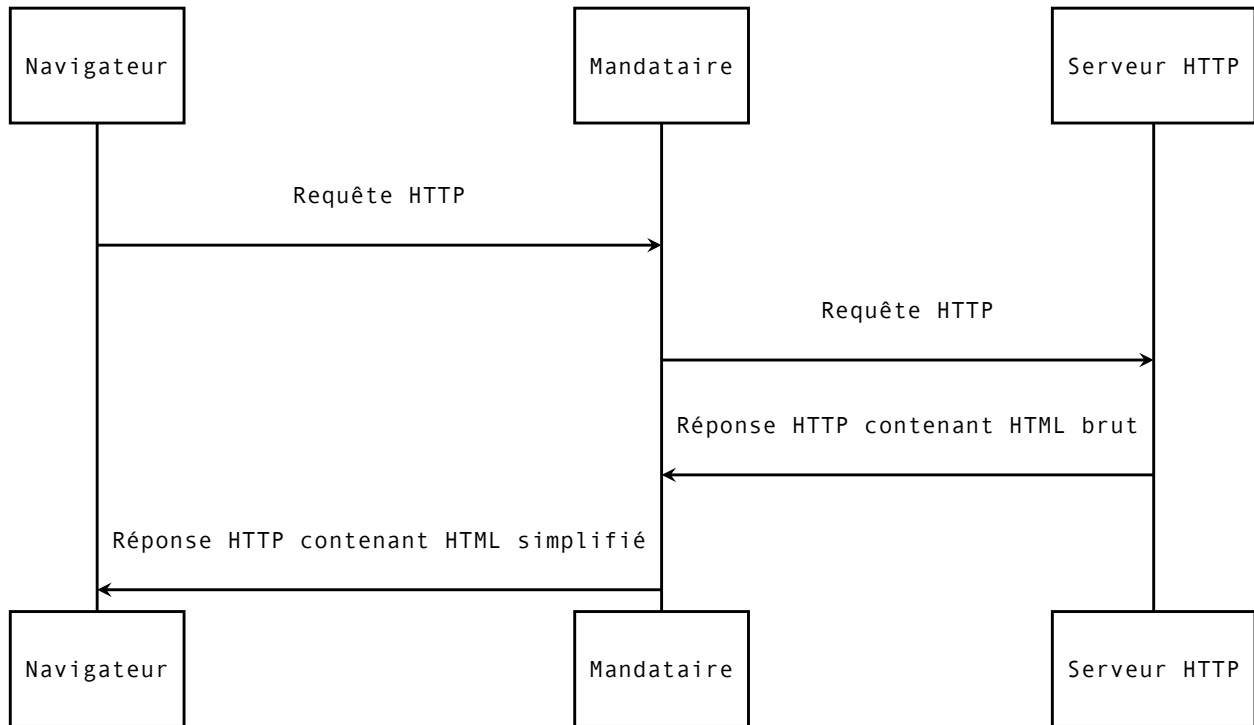


avec un mandataire il devient



L'avantage est que la requête comme la réponse sont toutes les deux accessibles par le mandataire. Il est donc en capacité de l'analyser. Si les mandataires de sécurité (recherchant des virus dans des pages HTTP) peuvent être **transparents** (il ne font qu'inspecter les contenus à la recherche de "virus" éventuels), certains mandataires peuvent intervenir en modifiant la page HTTP. Dans notre cas, le mandataire pourrait

intervenir sur le code HTML présent dans les réponses HTTP pour le simplifier avant de le retourner au navigateur ayant fait la demande.

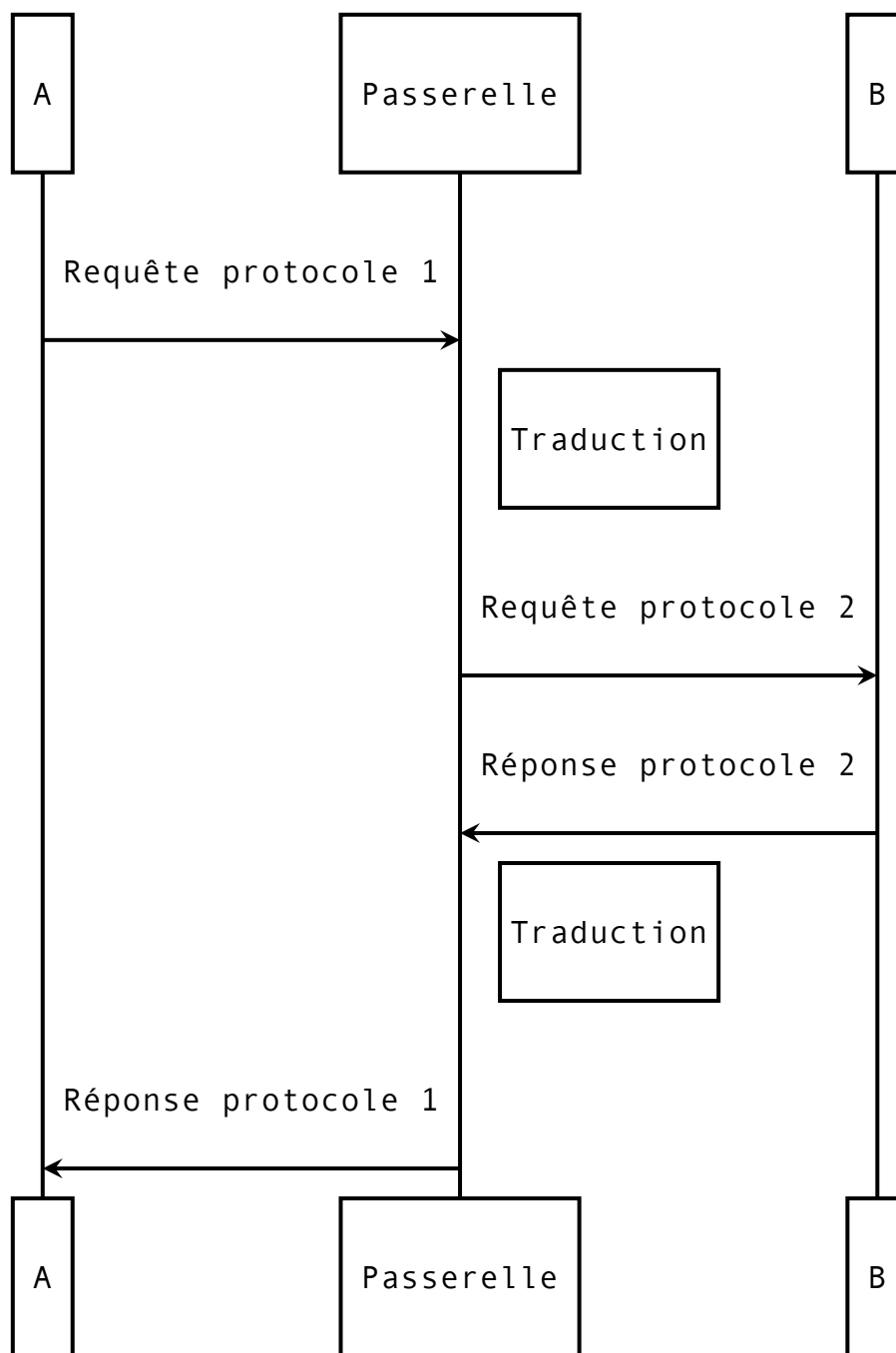


Cette approche permet aux utilisateurs (par choix ou par contrainte) de naviguer sur le web avec une machine peu puissante (dotée de peu de ressources RAM/CPU). En effet, passer par un mandataire est une solution facile à mettre en place pour accéder à des contenus présents sur des pages web trop lourdes/impactantes en ressources.

Les fonctionnalités d'une passerelle

Vocabulaire:

Une **passerelle**, est un serveur qui comprend deux protocoles et traduit les messages d'un protocole à l'autre, permettant ainsi à deux machines "ne parlant pas la même langue" de communiquer.



Dans notre cas, une passerelle entre HTTPSLOW et HTTP, pour gérer la transition entre ces deux protocoles, permettrait aux navigateurs HTTPSLOW de consulter le web historique - en conservant des serveurs web HTTP - et permettrait également aux navigateurs classiques de bénéficier des ressources plus simples de HTTPSLOW.

Des formes d'automatisation

Serveurs web/serveurs applicatifs  méthode et gestion automatisée des en-têtes HTI
Module automatisé de nettoyage de page



Deux approches sont possibles pour le module de nettoyage automatisé de pages HTML:

- une première piste possible est de s’inspirer de la fonction “Mode lecture (<https://support.mozilla.org/fr/kb/retirer-tout-le-superflu-des-pages-web-grace-au-mode-lecture>)” présent sur certains navigateurs : le code HTML brut serait purgé des balises “inutiles” (`<div>` imbriqués par exemple) avant de générer la réponse définitive retournée. Des solutions sont actuellement à l’étude (voir les références bibliographiques)
- une autre piste est celle suivie par Woob ([https://fr.wikipedia.org/wiki/Woob_\(logiciel\)](https://fr.wikipedia.org/wiki/Woob_(logiciel))): extraire les éléments pertinents d’une page web (technique appelée **moissonage** aussi appelé *scraping*) pour permettre leur affichage dans une interface textuelle. Si cette approche est radicale, il est cependant envisageable de recomposer une page HTML à partir de éléments extraits.

Seconde évaluation: repérer les fantômes

- On a vu que la mise en place de certaines solutions nécessite l’implication conjointe des différents acteurs autour de la question HTTPSLOW :
 - usagers/développeurs logiciels (navigateurs/serveurs)
 - développeurs logiciels(cms/serveurs applicatifs)/éditeurs
- Certaines propositions ne semblent pas nécessiter de développements lourds voire sont déjà disponibles mais simplement non mises en oeuvre (eg: gestion de MIME text/plain). (voir par exemple un exemple de configuration du serveur web Nginx (<https://www.nginx.com/resources/wiki/start/topics/examples/full/>))




Quelle.s empreinte.s fantôme.s pour HTTPSLOW ?

- Repérer des pistes de réflexion à partir de l'émission UPLOAD/LowNum Saison 1 Episode 4
 -  La conférence de Guillaume Carnino (<https://tube.picasoft.net/w/cQ4vgXfRRUUnxDgReKdEVB>)
 -  Définition express de l'empreinte fantôme par Guillaume Carnino (https://hubcontent.ecovalence.fr/pod/PGen/media/2022-06-10_lownum-httpslow_empreintefantome.mp3)

Vers une matrice SWOT HTTPSLOW ?

- S'inspirer de l'exemple du projet LowNum  EcoZilla (<https://md.picasoft.net/s/8TZLMVQVQ#>)
 -  Extrait audio (https://hubcontent.ecovalence.fr/pod/PGen/media/2022-06-10_lownum-httpslow_le-swot-ecozilla.mp3)

Un code éthique d'hébergeurs pour lutter contre les 7 péchés capitaux du GreenWashing ?

- Construire une grille d'analyse à partir des 7 péchés capitaux du GreenWashing ? ->  Article de GreenIt (<https://www.greenit.fr/2010/02/16/les-7-peches-du-greenwashing-appliques-aux-technologies-de-l-information/>)
- Formaliser un code éthique des hébergeurs ? ->  La Charte des Chatons (<https://www.chatons.org/charte>)
- Certaines propositions sont simples à mettre en oeuvre (eg: gestion de MIME text/plain) : -> **Critère d'efficience ?**  Les 6 critères du Comité d'Aide au Développement de l'OCDE (<https://www.eval.fr/quest-ce-que-levaluation/les-criteres-devaluation/>)

Publication et suites possibles

Arrivé au terme des 6 semaines de cette saison 1 #LowNum, il nous faut passer le relais pour la prochaine saison. Bien que nous n'ayons pas eu le temps d'explorer toutes les pistes d'évaluation indiquées plus haut, nous souhaitons les noter pour indiquer les différentes orientations possibles qui pourront servir lors de la seconde saison.

Pour rester dans la continuité de la méthode Lownum / Libre Cours / UPLOAD, nous proposons une dernière idée opérationnelle : formaliser les 4 pistes proposées ci-dessous en fiches techniques #WorkInProgress, accessibles en ligne, avec pour objectif de lancer une consultation #masto sous forme de sondage, afin de recueillir une "première impression" des utilisateurs / développeurs / concepteurs sur ces questions

-  Un modèle de fiche technique présentant les fonctionnalités attendues de :

1. Client web Gemini capable d'afficher des pages HTML (META text/html)
2. Client web HTTP capable d'afficher des pages HTML sans CSS ni JS (par défaut ; les greffons servent à gérer CSS et JS)
3. Développement de prototypes de passerelles HTTPSLOW/HTTP
4. Développement de prototype de serveurs HTTPSLOW
5. ~~Navigateur multiprotocole gérant simultanément les schémas HTTP:, HTTPSLOW:, GEMINI: et GOPHER: (là c'est pas LowNum)~~

Ressources

Les bases documentaires

- Pour un aperçu rapide de la base de connaissances : https://md.picasoft.net/gGMb_soXShGUdMI2ZNemdg#Ressources--Base-de-connaissances
(https://md.picasoft.net/gGMb_soXShGUdMI2ZNemdg#Ressources--Base-de-connaissances)
- La bibliothèque zotero du groupe LowNum : <https://www.zotero.org/groups/4417353/lownum/library> (<https://www.zotero.org/groups/4417353/lownum/library>)

Crédits, licence et... merci !

La colonne vertébrale de cette publication est le Libre Cours : UPLOAD/LowNum (<https://librecours.net/parcours/upload/lownum/index.html>) auquel vous pouvez librement accéder, sans inscription sur le site.

Libre...

Libre ça veut dire que je peux copier tout ce qui est dans les modules, et même les utiliser ailleurs avec d'autres gens, dans mon lycée ou dans mon entreprise ?

Oui. Tout ce que vous voulez. Même pas besoin de demander.

La licence utilisée est la plupart du temps une CC BY-SA (<https://creativecommons.org/licenses/by-sa/4.0/>), tout ce que vous avez à faire est de citer les auteurs et la source (en plus, c'est sympa ça fait un clin d'œil aux auteurs) et si vous faites des modifications vous devrez aussi déposer vos modules améliorés sous la même licence (comme ça tout le monde continue d'en profiter).

Vous pouvez aussi informer les auteurs, ils seront contents de voir que ce qu'ils font est utile, et même leur proposer vos améliorations.

Merci

- Framasoft (<https://framasoftware.org/>) et Picasoft (<https://picasoft.net/>) 🐱
- L'Université Technologique de Compiègne (<https://www.utc.fr/>) et le doctorat Fedithese (<https://fedithese.fr/>). On espère avoir été des *sujets d'expérimentation* productifs. 😊
- La radio <https://grafhit.net/> (<https://grafhit.net/>) et l'ensemble des personnes qui ont pris le temps de compiler, formaliser et transmettre les connaissances partagées durant ces 6 semaines 🙏
- Les autres participants pour les échanges très constructifs et inspirants 👍 100
- ... et nos mastopotes sur les réseaux fédérés... parce qu'ils ont vu des tags #lownum, #librecours et #UPLOAD défiler sur leurs flux pendant 6 semaines... et même certains ont participé 🙌