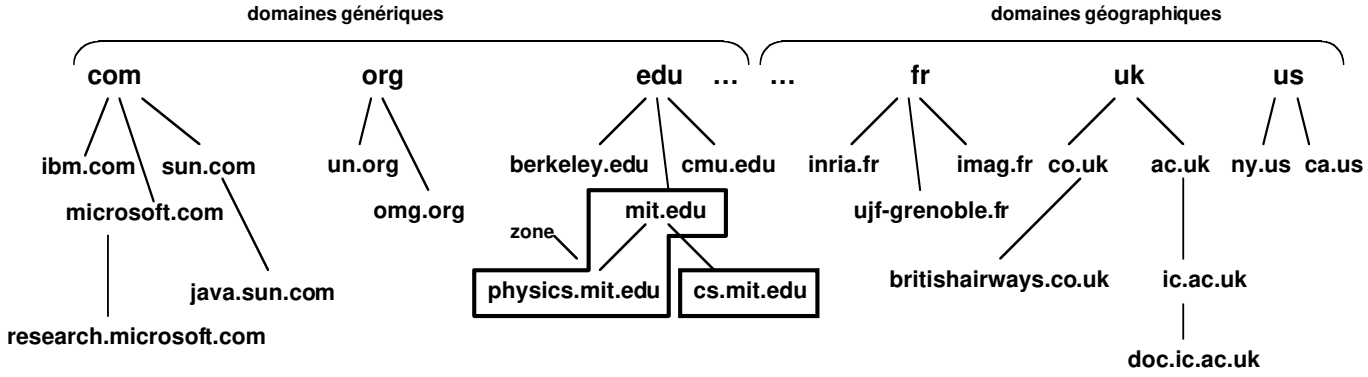
Les services usuels de l'Internet

- **Services principaux (applications) disponibles sur l'Internet**
  - Courrier électronique (`mail`) - protocole SMTP (*Simple Mail Transfer Protocol*)
    - inclut maintenant tous types de documents, grâce au format MIME (*Multipurpose Internet Mail Extensions*)
  - Forums de discussion (`news`) - protocole NNTP (*Network News Transfer Protocol*)
  - Transfert de fichiers (`ftp`) - protocole FTP (*File Transfer Protocol*)
  - Accès à une machine distante (`telnet`) - protocole *telnet* (terminal virtuel)
  - X-Window, service de fenêtres pour client distant
  - Accès au World Wide Web - protocole HTTP, formats HTML, XML, …
  - Services divers utilisant le Web (bibliothèques virtuelles, commerce électronique, installation de logiciel, …)
  - Extensions diverses pour la sécurité (SSL, SET, PGP, …)
- **Ces services utilisent les protocoles de transport (TCP, UDP), ainsi que le service de noms DNS (*Domain Name Service*)**
- **Plan de la suite**
  - Fonctionnement de DNS
  - Fonctionnement et utilisation du World Wide Web

DNS : noms de domaines (1)

- **La désignation sur l'Internet utilise un schéma hiérarchique (noms de domaines)**
  - Exemples : thales.ujf-grenoble.fr, cmu1.acs.cmu.edu, www.w3.org, firewall.ma02.bull.com
  - Un domaine est une unité pour la construction et la gestion des noms (analogue à un répertoire dans un système de fichiers)
- **La hiérarchie des noms a une racine multiple : les domaines du premier niveau**
  - domaines "géographiques" : fr (France), uk (Royaume Uni), de (Allemagne), jp (Japon), …
    - USA est souvent compris "par défaut", bien qu'il existe un domaine us
  - domaines "d'activité", génériques : com (commercial), org (organisations), edu (universités, aux USA), net (l'Internet), …
- **Dans chaque domaine, les noms sont attribués par une autorité responsable du domaine**
  - Pour établir la liste des domaines du premier niveau : l'Internet Society (ISOC) via un groupe technique ad hoc
  - Pour les domaines "publics" du premier niveau (com, org, net, …) : une autorité centrale, l'ICANN (www.icann.org), avec des autorités déléguées
  - Pour les domaines géographiques : une autorité nationale par pays - en France, l'AFNIC (www.nic.fr) - Association Française pour le Nommage sur l'Internet en Coopération)
  - Pour les domaines inclus : autorités locales (entreprise, administration, etc.)

DNS : noms de domaines (2)



- **Domaines et zones**
  - Domaine = unité de désignation (espace de noms)
  - Zone = unité de gestion administrative (serveur de noms propre à la zone)
  - Le plus souvent, un domaine est aussi une zone, mais une zone peut grouper plusieurs domaines, administrés en commun

Service de noms sur l'Internet : problème et principes

- **Le problème**
  - Trouver l'adresse IP d'un site sur l'Internet, connaissant son nom symbolique
    - le DNS actuel résout aussi le problème inverse
- **Bref historique**
  - Premières années (jusqu'à quelques milliers de sites) : un serveur de noms unique pour tout l'Internet, gérant une table de correspondance nom symbolique - adresse IP
    - Inconvénients
      - mauvaise résistance aux pannes
      - goulot d'étranglement, pas de capacité de croissance
  - À partir de 1985 : introduction d'un service réparti, DNS, qui a bien résisté depuis à une croissance exponentielle
- **Principes**
  - Algorithme de recherche décentralisé (pas de point de décision unique)
  - Hiérarchie de serveurs calquée sur la hiérarchie des zones (voisines des domaines)
  - Usage intensif de caches (informations dupliquées) et d'indicateurs (informations probablement valides permettant un accès rapide la plupart du temps)
    - toute information peut être obtenue par plusieurs voies
    - la validité de toute information peut être confirmée si nécessaire
  - Ces règles favorisent aussi la tolérance aux fautes (service assuré en cas de pannes)

Fonctionnement de DNS (1)

- **Toute zone (unité de gestion) comporte au moins deux serveurs de noms**
  - Motivation de la duplication : pour performances, et surtout tolérance aux fautes
  - Chaque serveur maintient une table de correspondance Nom de domaine-Adresse IP
  - Plus précisément, une table est une collection d'enregistrements de la forme :

Nom, *Valeur*, *Type*, *Classe*, *Durée de vie*

- **Principe de la recherche**
  - Ou bien le serveur contient l'adresse recherchée, ou bien il contient l'adresse d'un autre serveur qui a davantage de chances que lui de la connaître
- **Contenu des tables du serveur de noms**
  - *Type* : indique comment le champ *Valeur* doit être interprété
    - *Type* = A : la *Valeur* est une adresse IP
    - *Type* = NS : la *Valeur* est le nom de domaine d'un serveur de noms
    - *Type* = CNAME : la *Valeur* est le nom "canonique" d'un hôte (le nom de référence en cas d'alias)
    - autres : non traités ici (aide à la gestion du *mail*, etc.)
  - *Classe* : IN (l'Internet), quasiment toujours (utilisation de DNS comme annuaire pour d'autres services de noms)
  - *Durée de vie* (TTL ou *Time To Live*) : durée de validité de l'enregistrement
    - Application d'un principe général : toute information doit être périodiquement validée
      - cf les tables de routage

Fonctionnement de DNS (2)

- **Le serveur racine contient les noms des serveurs de noms des domaines du second niveau et les adresses IP de ces serveurs**

  <inrialpes.fr, if.inrialpes.fr, NS, IN, …> -- if est un serveur de noms du domaine inrialpes.fr

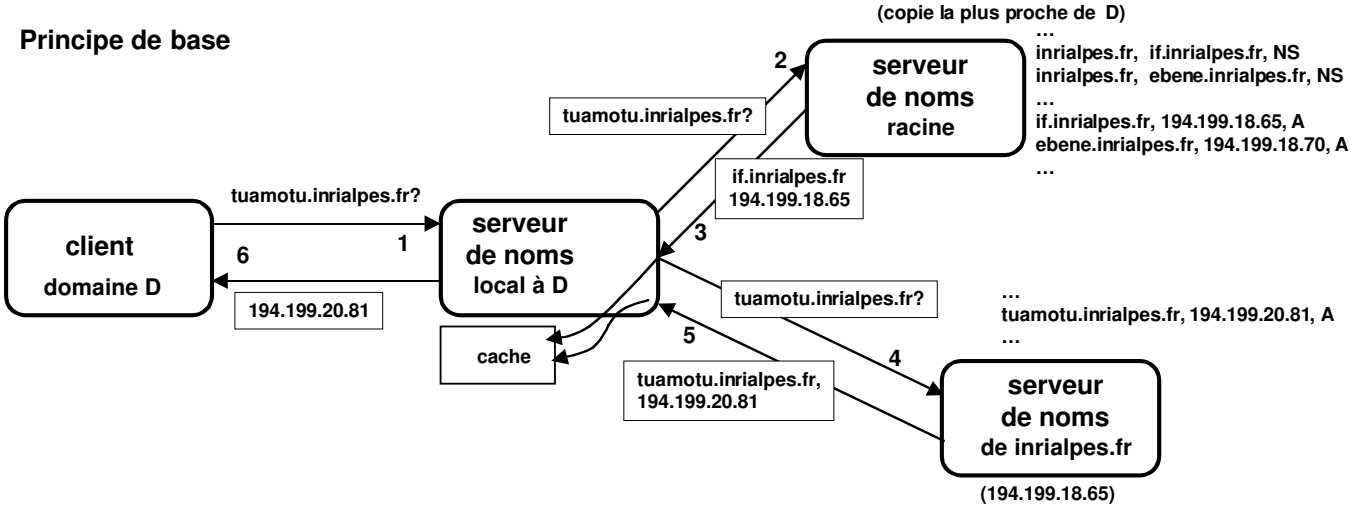
  <if.inrialpes.fr, 194.199.18.65, A, IN, …> -- l'adresse IP de if est 194.199.18.65

  - Le serveur racine est très largement dupliqué
    - Plus un serveur est "haut", plus souvent on en a besoin, plus il a de copies, et moins souvent il change (long TTL)
- **De même, chaque serveur de zone contient**
  - Des couples (noms, adresses) d'hôtes appartenant aux domaines contenus dans la zone
  - Éventuellement des couples (noms, adresses) de serveurs de noms de sous-domaines inclus
  - Par exemple, if.inrialpes.fr contient

    <tuamotu.inrialpes.fr, 194.199.20.81, A, IN, …>
- **Pour amorcer la résolution**
  - Tout hôte doit connaître l'adresse IP d'un serveur de noms local (dans sa zone) - il est recommandé de connaître deux ou trois adresses, pour la tolérance aux fautes
  - Ces adresses sont fournies aux utilisateurs de l'hôte par l'administrateur de sa zone (ou son fournisseur d'accès à l'Internet)
  - Elles sont inscrites "à la main" dans les tables de configuration pour l'accès à l'Internet

Fonctionnement de DNS (3)

Principe de base



Mécanismes d'accélération

Tout serveur conserve dans un cache les couples (nom, adresse IP) récemments résolus
Dans la pratique, il est rare de consulter plus de deux serveurs

Autorités

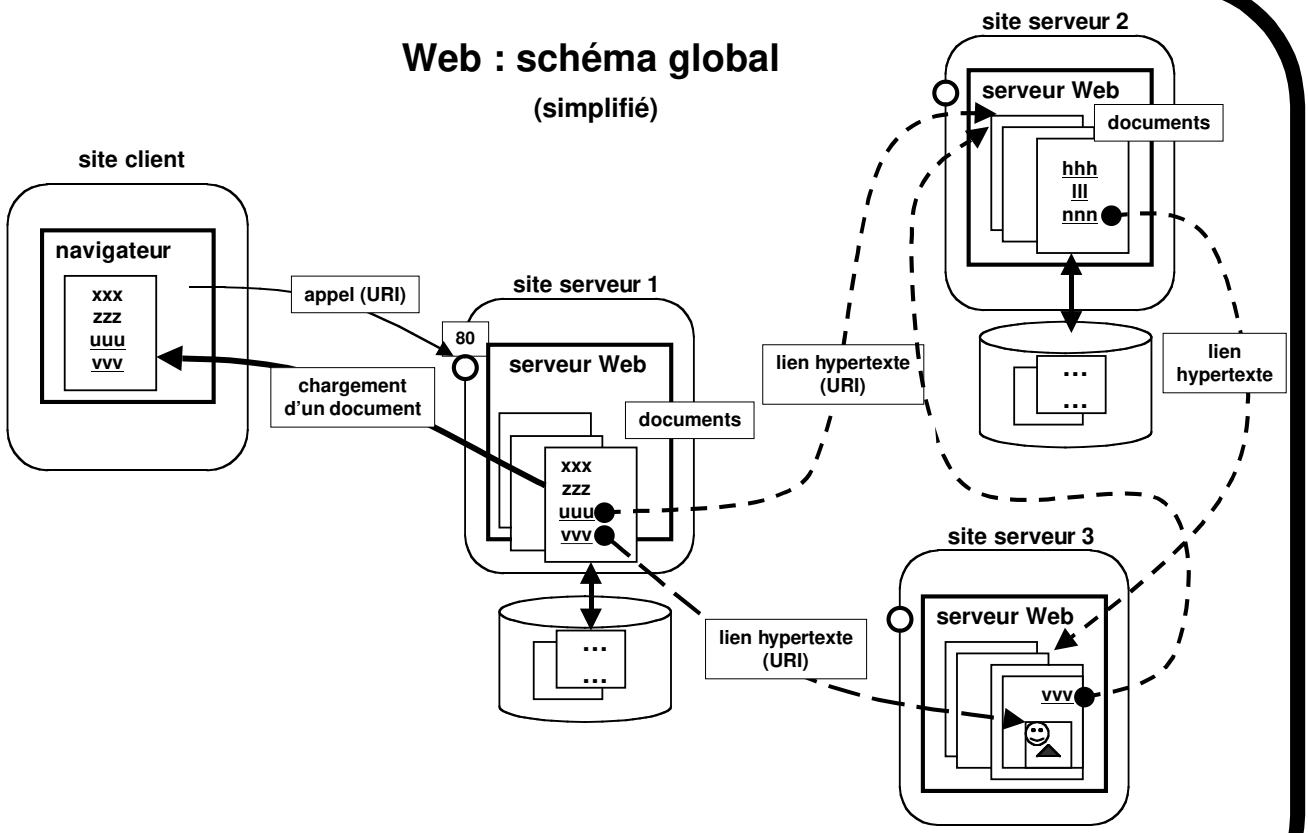
Tous les serveurs ne sont pas mis à jour en permanence de toutes les modifications. Certains seulement le sont, et font autorité
d'ou les réponses "*authoritative answer*" ou "*non authoritative answer*"

World Wide Web : principes et composants

- **Bref historique**
  - **Idée de base : ensemble de documents répartis reliés entre eux par des liens hypertexte. Objectif initial (Tim Berners-Lee, CERN, 1989-90) : créer un outil pour le travail en collaboration, sur des données communes, pour une communauté répartie de physiciens**
    - **en fin 1993, 250 serveurs, 1% du trafic de l'Internet (10 fois plus qu'en début 1993)**
  - **Le vrai démarrage (1994)**
    - **les premiers navigateurs : Mosaic (NCSA), puis Netscape**
    - **les premiers moteurs de recherche : AltaVista, Yahoo!**
    - **création du World Wide Web Consortium (W3C) <www.w3c.org>**
    - **en fin 1994, environ 10 000 serveurs**
  - **Depuis, croissance explosive (l'application la plus utilisée de l'Internet) ~10^9 pages web**
- **Éléments de base du Web**
  - **Un espace de noms global pour la désignation des ressources (URL, puis URI)**
  - **Un protocole (client-serveur) pour le transfert d'information : HTTP**
  - **Un langage de balisage (*markup*) pour la description de documents hypertextes (HTML)**
- **Extensions**
  - **Utilisation de langages de script (activation chez le client - *applets*, ou le serveur - *servlets*)**
  - **Utilisation de types de données multiples ; descriptions génériques (XML), outils associés**

Web : schéma global

(simplifié)



Web : désignation

Format d'un URI (Uniform Resource Identifier)

<protocole>:<chemin d'accès>

Le format du chemin d'accès dépend du protocole. Par exemple (les parties entre crochets sont facultatives) :

- Protocole `http` : <chemin d'accès> ::= //<identité de serveur> [:<numéro de porte>] [<chemin d'accès local>] [?requête] [#étiquette]
- Protocole `ftp` : idem, sans requête ni étiquette
- Protocole `file` : <chemin d'accès> ::= <chemin d'accès à un fichier local>
- Protocole `mailto` : <chemin d'accès> ::= <adresse *email*>
- Protocole `news` : <chemin d'accès> :: = <nom de *newsgroup*>

- Exemples
  - http://sirac.imag.fr/Infos/Personnes/Sacha.Krakowiak/
  - ftp://ftp.imag.fr
  - mailto:president@whitehouse.gov
  - news:imag.38
- Valeurs par défaut : http pour le protocole, 80 pour le numéro de porte, index.html pour le nom de fichier dans un répertoire, etc.

Utilisation des URI

- Ce sont les voies d'accès à toutes les ressources (documents, serveurs, programmes, etc.)
- Ce sont les constituants des liens hypertexte

Web : HTTP (*HyperText Transfer Protocol*)

- **HTTP : le protocole standard du World Wide Web**
  - Protocole client-serveur, construit au-dessus de TCP
  - Utilisation principale : entre navigateur et serveur Web, mais peut être utilisé de manière autonome par toute application
- **Principales commandes du protocole**
  - GET <URI> : demande au serveur indiqué dans l'URI d'envoyer la page désignée par l'URI.
    - option : n'envoyer la page que si elle a changé depuis une date spécifiée
  - HEAD <URI> : demande au serveur d'envoyer l'en-tête de la page (contenant des informations diverses : titre, date, etc.)
  - PUT <URI> <page> : envoie une page au serveur spécifié pour la rendre disponible sur ce serveur à l'URI indiquée ; replace le contenu courant de cet URI s'il existe
  - POST <URI> <page> : comme PUT, mais intègre les nouvelles données à celles existant déjà à l'URI (dépend de la nature des données)
  - DELETE <URI> : supprime la page figurant à l'URI indiqué
  - Toutes ces commandes sont soumises à autorisation, en fonction des droits du client demandeur et des protections associées aux ressources sur le serveur
  - La réponse à une commande comporte un code (OK ou type d'erreur) et éventuellement un résultat (contenu de page pour GET, etc.)
  - Les commandes d'envoi de données utilisent une convention standard (MIME) pour les données non textuelles

Web : HTML (*HyperText Markup Language*)

- **HTML est un langage de "balisage" (*markup*)**
  - Un tel langage comporte des marques (balises) insérées dans le texte et destinées à donner des indications de formatage (présentation, interprétation du texte). Exemples plus loin
  - Un langage de balisage très général, utilisé dans l'édition de documents, est SGML (*Standard Generalized Markup Language*) ; HTML en est inspiré.
  - Intérêt du balisage : permet de séparer le contenu de la présentation ou de l'interprétation, et donc permettre des interprétations différentes selon (par exemple) les capacités d'affichage d'une station de travail
  - HTML est en évolution constante (version 4.0) - normalisé par le W3C <www.w3c.org>
- **Comment sont produits les documents HTML ?**
  - "À la main". Pas recommandé, il est préférable d'utiliser un des outils qui suivent
  - Par un éditeur de documents (pour l'écriture de pages Web)
    - directement (frappe du texte, insertion d'images, etc.)
    - par traduction depuis un autre format de document (LaTeX, Word, ou autre)
  - Par un générateur spécialisé, à partir (par exemple) du résultat d'une requête sur une base de données. Chaque application peut construire son générateur (exemples séance suivante)

HTML - quelques exemples (1)

- **Principe du balisage**
  - Les balises vont en général par paires, encadrant un texte à interpréter
    balise début : <xxx paramètres éventuels> - balise fin : </xxx>
- **Structure d'un document HTML** (indicatif)

```
<HTML>
<HEAD> en-tête </HEAD>      -- contient le titre, la date, d'autres méta-informations.
<BODY> corps </BODY>        -- contient le document proprement dit
</HTML>
```

- **Quelques balises de présentation (exemples)**
  - Présentation de caractères
    - <B> texte </B> : caractères gras (*bold*) ; <I> texte </I> : caractères italiques
    - Caractères accentués. Exemples : é = é à = à Ê = Ê etc.
  - Titres
    - <H1> texte du titre </H1> : titre de 1-er niveau (idem pour H2, H3, …)
  - Paragraphage
    - <P> texte </P> : paragraphe ;
 retour à la ligne
    - <HR> coupure du texte (trait horizontal)
  - En fait, l'interprétation précise des balises de présentation peut être définie séparément (feuilles de style). En modifiant la feuille de style, on modifie la présentation sans changer le document

HTML - quelques exemples (2)

- **Autres informations de présentation**
  - Tables (nombreux attributs possibles : disposition relative des cases, couleur du fond, épaisseur des traits, etc.)
  - Listes (numérotées ou non)
- **Inclusion d'images**
  - <IMG SRC = "le fichier ou l'URL contenant l'image" - autres paramètres (échelle, alignement par rapport au texte, affichage de texte alternatif si l'image ne peut être affichée, etc.) >
- **Liens hypertexte**
  - <A HREF="l'URI associée au lien" NAME="le nom" autres paramètres (affichage dans une fenêtre autonome, etc.) > le texte ou l'image qui constitue l'hyperlien </A>
  - Ce lien est affiché de manière particulière par le navigateur (par exemple souligné en bleu)
  - Un "clic" de souris sur ce lien est interprété par le navigateur comme : demander le chargement (GET) du document désigné par l'URI du paramètre HREF

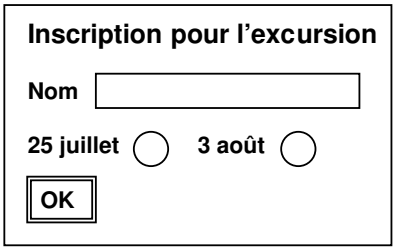
HTML - quelques exemples (3)

- **Un premier exemple d'interaction**
  - Jusque là ont été décrites des caractéristiques uniquement liées à l'affichage
  - On souhaite aussi permettre l'interaction entre client et serveur
  - Exemple : remplir un formulaire simple

le texte HTML

```
<HTML> <HEAD> <TITLE> Inscription </TITLE></HEAD>
<BODY>
<H1>Inscription pour l'excursion</H1>
<FORM ACTION = "http://sejourvacances.com/cgi-bin/choix" METHOD = POST>
Nom <INPUT NAME ="client" SIZE = 20><BR>
Indiquez la date choisie et cliquez sur OK<BR>
25 juillet <INPUT NAME="date" TYPE=RADIO VALUE="2507"
3 ao&ucirc;t  <INPUT NAME="date" TYPE=RADIO VALUE="0308" <BR>
<INPUT TYPE=SUBMIT VALUE = "OK">
</FORM></BODY></HTML>
```

ce qui est affiché



ce qui est envoyé (par exemple)

client=Dupont&date=2507

c'est le programme (script) indiqué dans le paramètre ACTION qui traitera cette entrée

Résumé de la séance 11

- **Quelques protocoles usuels de l'Internet**
  - Utilisation du schéma client-serveur
  - Service de noms de l'Internet : protocole DNS
    - désignation symbolique sur l'Internet : domaines
    - exemple de gestion décentralisée, capacité de croissance
  - Le World Wide Web : principe, fonctions de base
- **Plan de la suite**
  - World Wide Web : suite
    - exécution de programmes répartis
    - applications sur le Web
    - aspects système : serveurs et caches