

Systemes d'Information Distribués

GI-4-SID-S1

mathieu.loiseau@insa-lyon.fr

GI-4-SID

- Dispensé l'an dernier par

Céline Robardet

→ créditée sur certaines
diapos ©®

- ≠ lecture du programme
 - + orienté application

UE / GI-4-INFO1-S1 Informatique

GI-4-SID-S1 Systèmes d'informations distribués

Les connaissances :

- Architecture d'applications basée sur les services Web ;
- Modèle-Vue-Contrôleur, protocole HTTP, serveur d'application, service Web REST, format d'échange de données.

et les capacités à :

- Concevoir et développer une application distribuée permettant le stockage, l'accès à distance et la visualisation de données associées à la réalisation de processus (A5, C10, C15) ;
- Traiter des données (C1) ;
- Analyser les performances des processus à partir des traces générées par le système d'information (A6, C2) ;
- Réaliser un travail en équipe (C13, C15, B2,B3).

PRE-REQUIS : ALGORITHMIQUE ET PROGRAMMATION - BASE DE DONNEES ET SQL
-XML

Modalités de contrôle des connaissances

- Cours-TD-TP
- Évaluation
 - Devoir sur table
 - Individuel
 - Tous documents autorisés
 - Projet collectif
 - Contour à définir en fonction de ...
 - Bonus sur exposé individuel ?



<https://moodle.insa-lyon.fr/mod/questionnaire/view.php?id=111468>

Information

Données, Information et Connaissance

- « Data are symbols that represent properties of objects, events and their environments. They are products of *observation*. To observe is to sense. »
(Ackoff, 1989)

DONNÉES

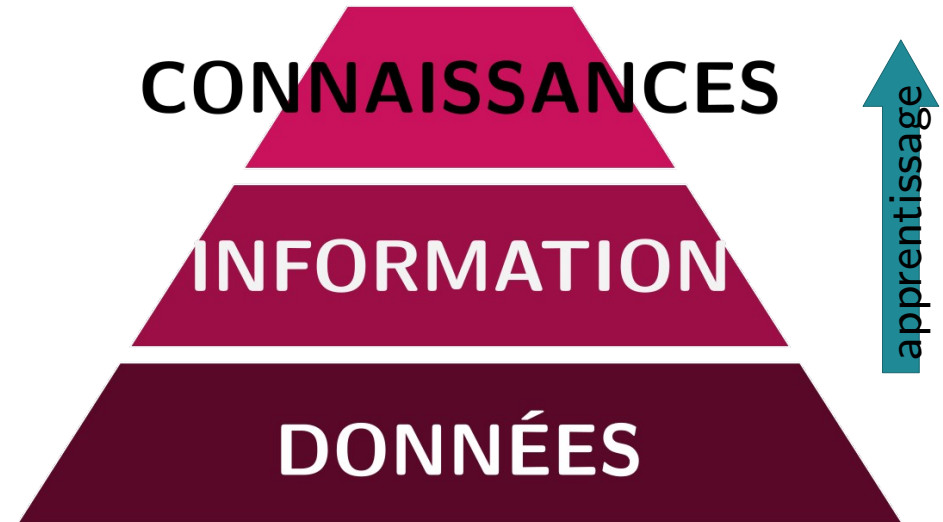
Données, **Information** et Connaissance

- « Information is contained in *descriptions*, answers to questions that begin with such words as *who, what, where, when, and how many* »
(Ackoff, 1989)



Données, Information et **Connaissance**

- « [Knowledge] is what makes possible the transformation of information into instructions. »
(Ackoff, 1989)



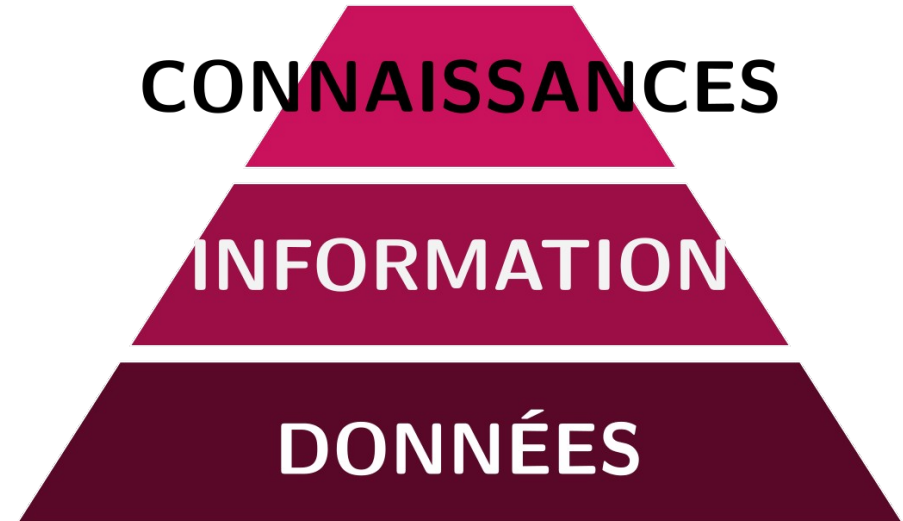
Données, Information et Connaissance

- « Intelligence is the ability to increase efficiency ; wisdom is the ability to increase effectiveness. »
(Ackoff, 1989)



Domaine universel vs domaine subjectif (Zins, 2007)

- Connaissances
subjectives : intimes
d'un individu
- Connaissances
Universelles : partagées
avec d'autres individus



Domaine universel vs domaine subjectif (Zins, 2007)

Subjectif

$$E = MC^2$$

$$\underline{E} = MC^2$$

$$E = M \times C \times C$$

Universel

Artéfacts humains

Signes représentant des
pensées justifiables

Signes représentant des
connaissances empiriques

Ensemble de signes
représentant des stimuli ou
perceptions

Pensée d'un individu
caractérisée par la croyance
justifiable qu'elle est vraie

Connaissance empirique

Stimuli sensoriels ou leur
signification

CONNAISSANCES

INFORMATION

DONNÉES

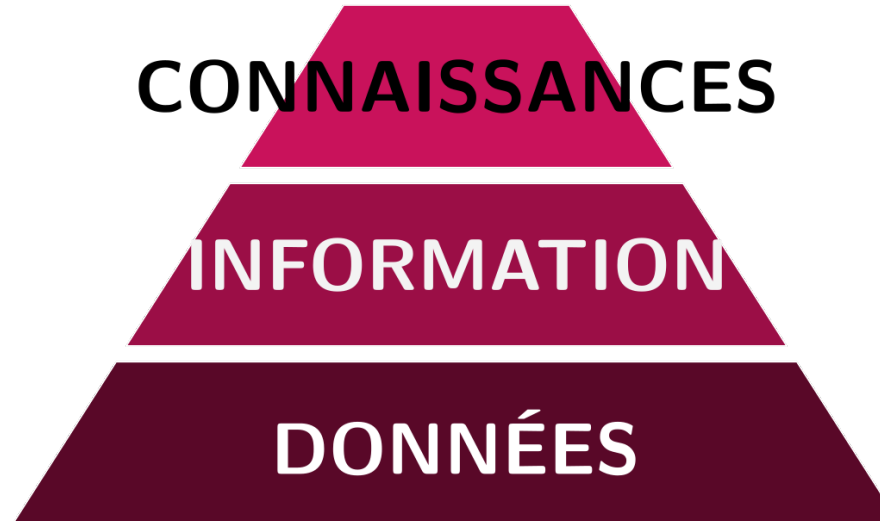
Domaine universel vs domaine subjectif (Zins, 2007)

Tenopir

Phénomènes internes

Phénomènes externes

Informations internalisées
ou comprises utilisables
prendre des décisions.



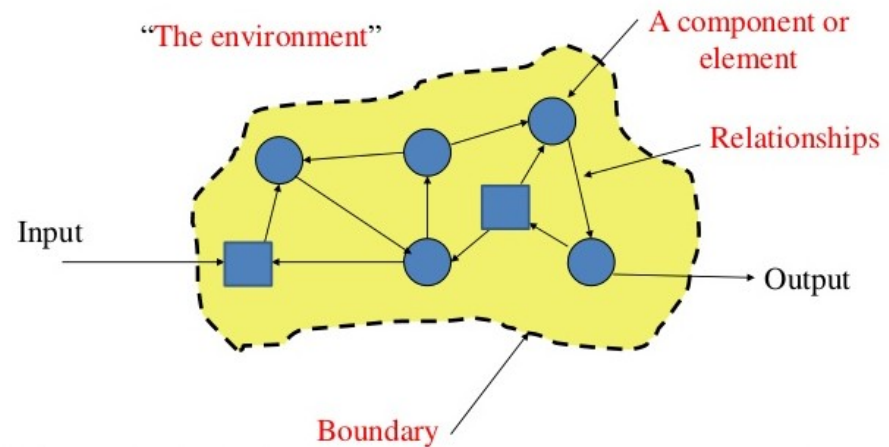
Données traitées pour
fournir du sens.

Faits résultants de mesures
ou d'observations.

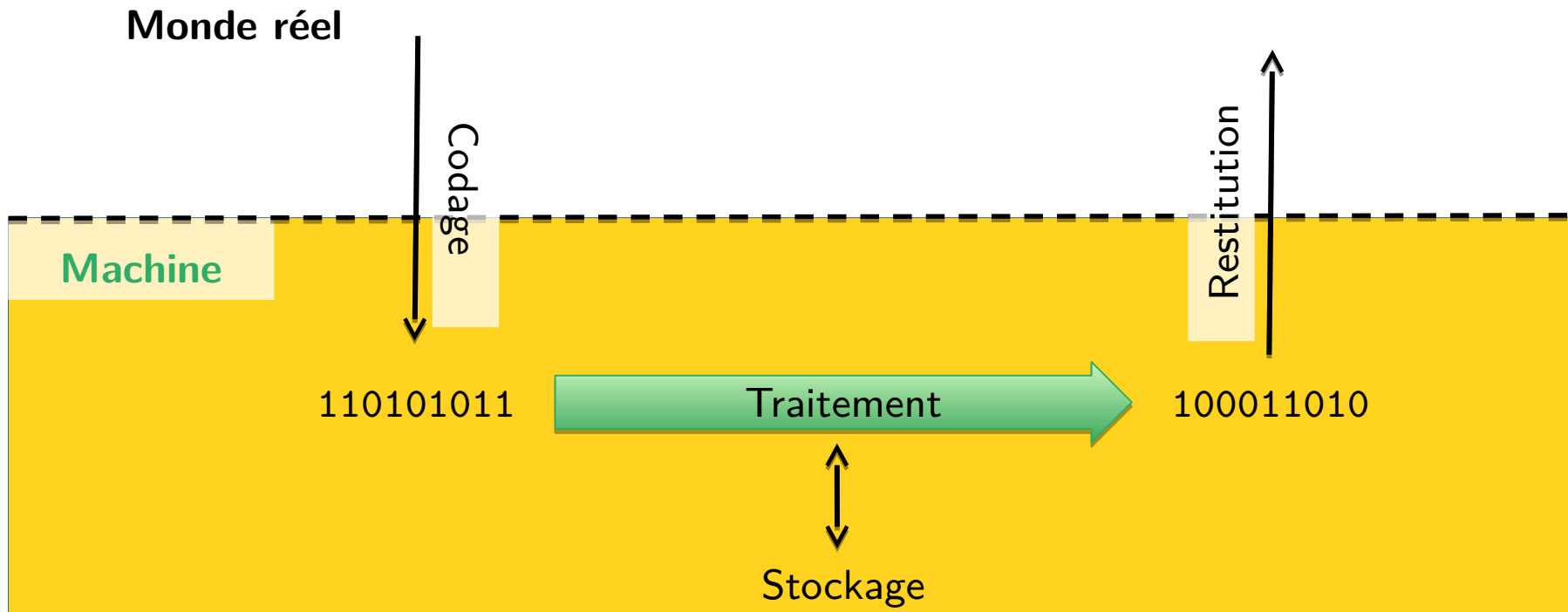
Systemes

Systeme

- Composants inter-reliés qui coopèrent pour atteindre un but.
- Interactions d'un système avec son environnement se font à travers
 - Entrées (Input) : un flux d'objets matériels, d'énergie ou d'information de l'environnement vers le système ;
 - Sorties (Output) : éléments produits par le système vers son environnement ;
- Les inputs subissent certaines transformations spécifiques au niveau de la frontière du système réalisée par des composants appelés interfaces.



Traitement Numérique de l'Information



Systeme distribue

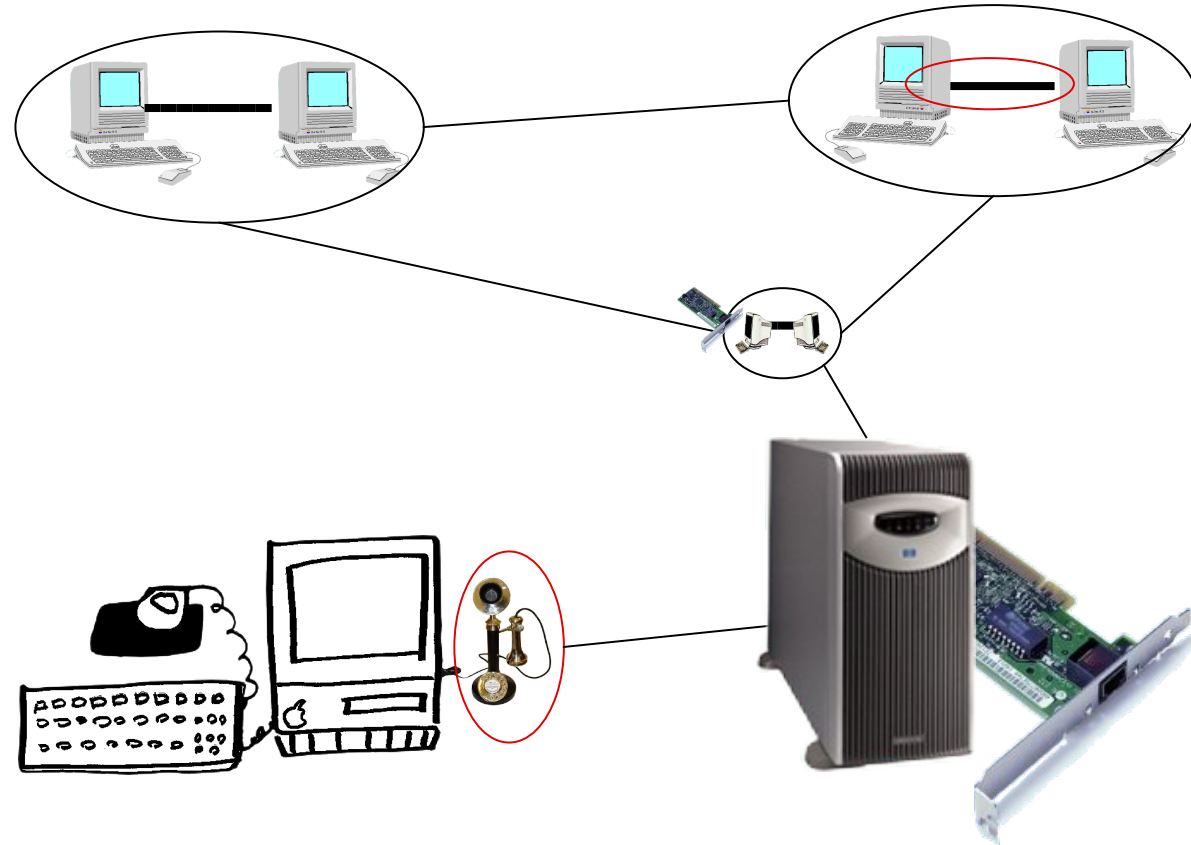
- Un systeme distribue est une collection d'ordinateurs autonomes mis en reseau qui apparaissent a l'utilisateur comme un unique ordinateur.
- Les ordinateurs communiquent et coordonnent leurs actions en echangeant des messages.
- Exemple de systeme distribue...
 - Internet

« Focus » du cours

- « [Knowing] how a system works is what makes possible the transformation of information into instructions. It makes control of a system possible. » (Ackoff, 1989)
- « [The technology in IT is] the infrastructure that makes it possible to store, search, retrieve, copy, filter, manipulate, view, transmit, and receive information. » (Shapiro & Varian, 1999)

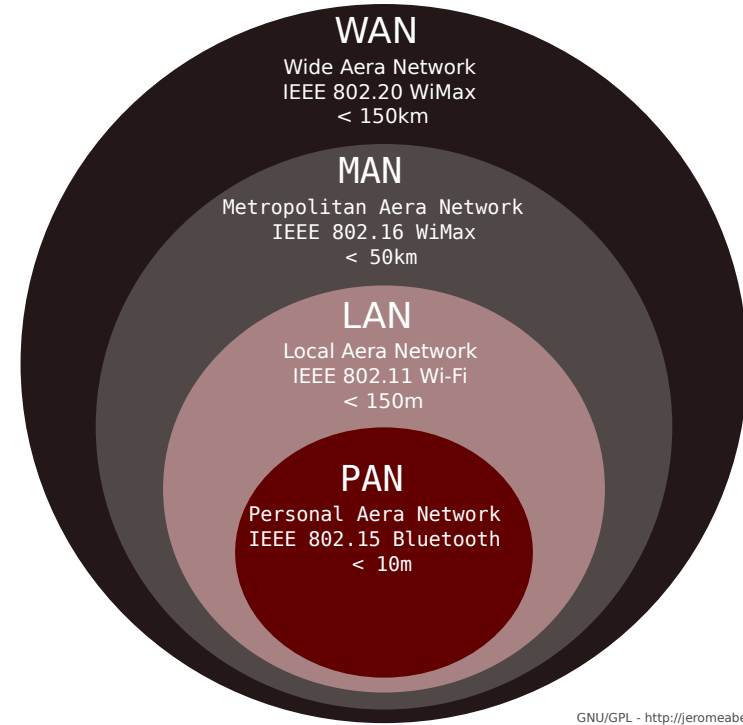
Réseaux

Internet



Échelle de réseaux

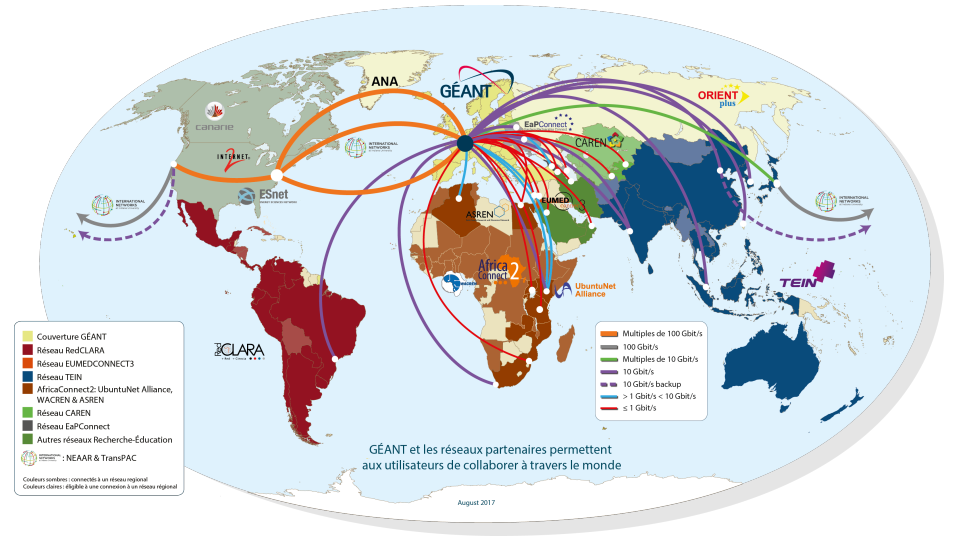
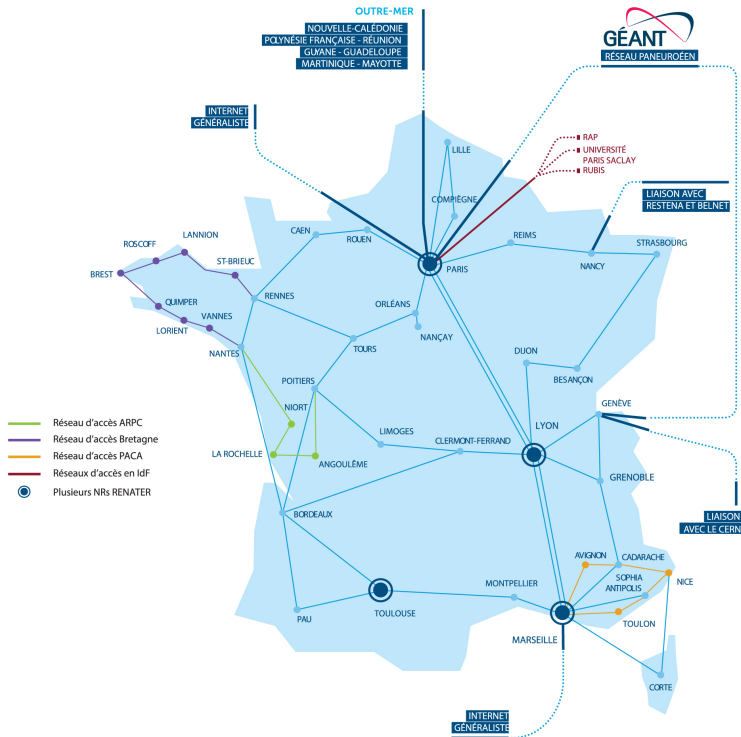
- WAN : Wide Area Networks
 - Réseau des opérateurs Télécom
 - National ou international
- MAN : Metropolitan Area Network
 - Boucle local
 - Transport de données sur "le dernier km"
- LAN : Local Area Networks
 - Réseau d'entreprises
 - Wireless LAN : WiFi
 - Réseau domestiques
- PAN : Personal Area Networks
 - Interconnexion de dispositifs intelligents



GNU/GPL - <http://jeromeabel.net>

Exemples de WAN

<https://www.renater.fr/fr/reseau>



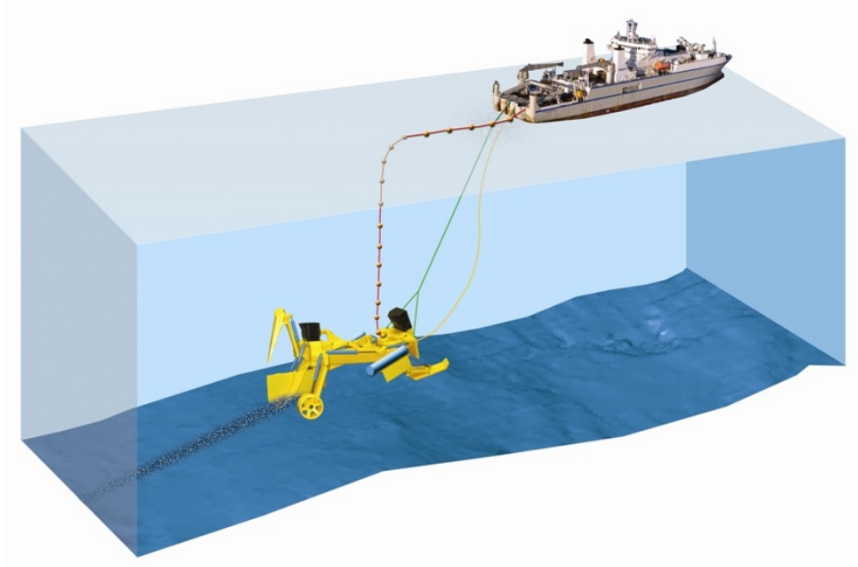
<https://www.renater.fr/fr/geant>

Couches TCP/IP

5. Applications	HTTP	FTP	DNS	SMTP
4. Transport	TCP	UDP		
3. Réseau	IP	ICMP	IGMP	
2. Liaison	Ethernet	Token ring	PPP	FDDI
1. Physique	Cuivre	Satellite	Téléphonie sans fil	Fibre optique

Couche Physique

Déploiement de la fibre

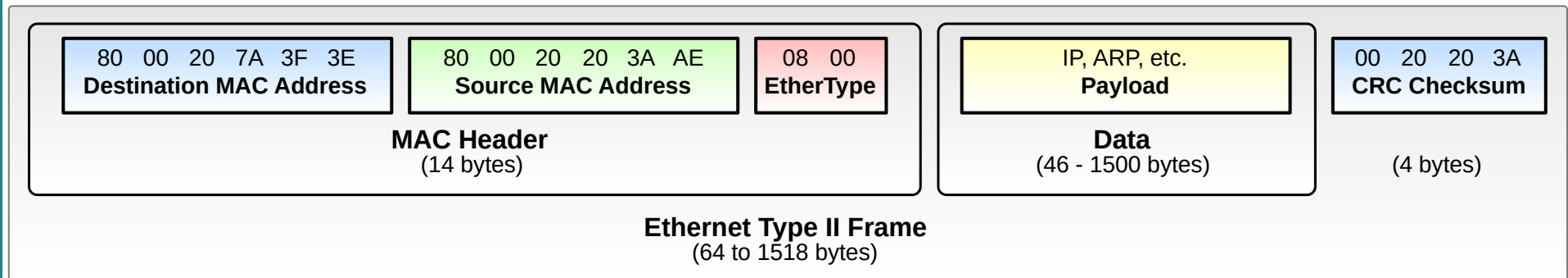
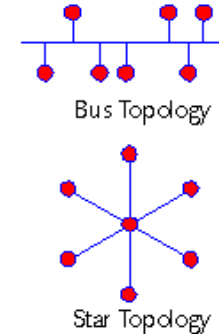


Couches TCP/IP

5. Applications	HTTP	FTP	DNS	SMTP
4. Transport	TCP	UDP		
3. Réseau	IP	ICMP	IGMP	
2. Liaison	Ethernet	Token ring	PPP	FDDI
1. Physique	Cuivre	Satellite	Téléphonie sans fil	Fibre optique

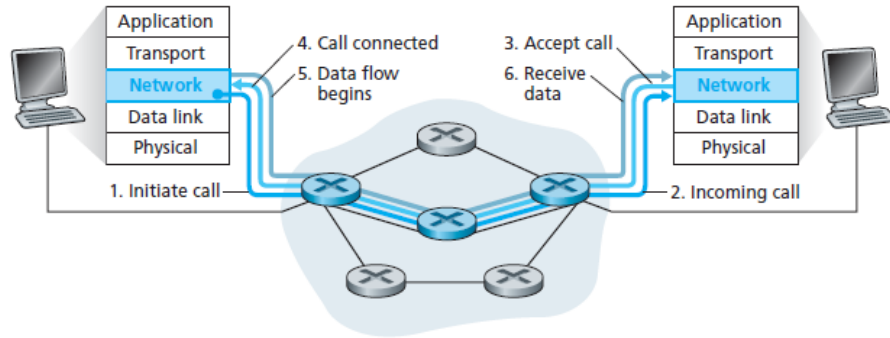
Couche liaison — exemple d'ethernet

- Ether
- Standard en évolution
 - Topologie Bus
 - Puis topologie étoile
- Chaque pair est identifié par une adresse MAC

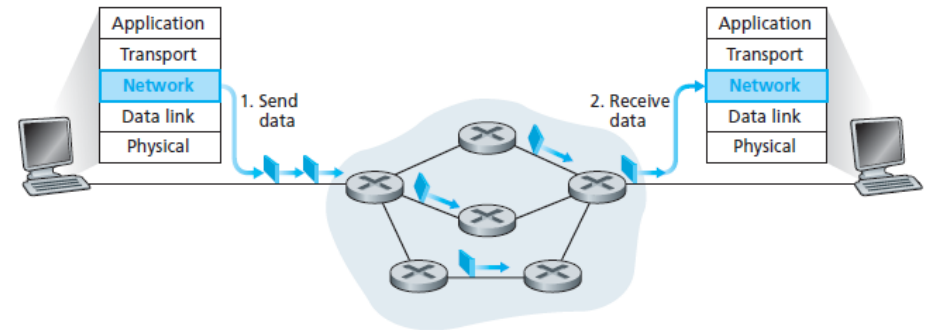


Mode connecté vs mode sans connexion

Virtual circuit



Datagram network



Datagrammes : paquets dans un réseau en mode sans connexion

<https://electronicspost.com/virtual-circuit-and-datagram-networks/>

Couches TCP/IP

5. Applications	HTTP	FTP	DNS	SMTP
4. Transport	TCP	UDP		
3. Réseau	IP	ICMP	IGMP	
2. Liaison	Ethernet	Token ring	PPP	FDDI
1. Physique	Cuivre	Satellite	Téléphonie sans fil	Fibre optique

Couche réseau — IP

- Mission : livrer des datagrammes de la source à la destination en se basant uniquement sur l'adresse IP.
- Le routage est effectué par tous les routeurs (passerelles/*gateways*) à partir de sous-réseaux

Structure d'une adresse IP

- IPv4 : 22.4.197.255
- IPv6 : 2001:db8::8a2e:370:7334
- À l'inverse d'une adresse postale, du plus large au plus spécifique
 - Préfixe du sous-réseau
 - Identifiant de la machine

Masque de sous-réseau

- Les nœuds du réseau ont des tables qui associent un ensemble d'adresses à un nœud spécifique qui routera les paquets de cet ensemble d'adresses
 - Adresse d'une passerelle + préfixe associé
- 124.15.44.21/23
 - 01111100.00001111.00101100.00010101
 - masque de sous-réseau : 255.255.254.0
 - 11111111.11111111.11111110.00000000

Allocation des adresses IP

- IANA alloue aux RIR des blocs d'adresses IP de taille $/24$ ou plus ;
- Le RIR (RIPE NCC pour l'Europe) alloue aux LIR des blocs de taille $> /8$;
- Les LIR (ex : Renater, FAI) affectent aux utilisateurs finaux des blocs, dont la taille correspond aux besoins immédiats et concrètement anticipés.
- FAI et autres utilisateurs finaux sont propriétaires d'un ensemble d'adresses et les affectent comme ils veulent

Couches TCP/IP

5. Applications	HTTP	FTP	DNS	SMTP
4. Transport	TCP	UDP		
3. Réseau	IP	ICMP	IGMP	
2. Liaison	Ethernet	Token ring	PPP	FDDI
1. Physique	Cuivre	Satellite	Téléphonie sans fil	Fibre optique

Ports

- Chaque ordinateur (client ou serveur) peut avoir plusieurs logiciels destinataires/émetteurs de messages (ex : Client mail vs navigateur vs VoIP vs jeu)
- Chaque logiciel a un port associé

UDP vs TCP

- UDP

User Datagram Protocol

- Sans connexion (datagrammes)
- Multicast / Broadcast
- Léger mais non fiable

- TCP

Transmission Control Protocol

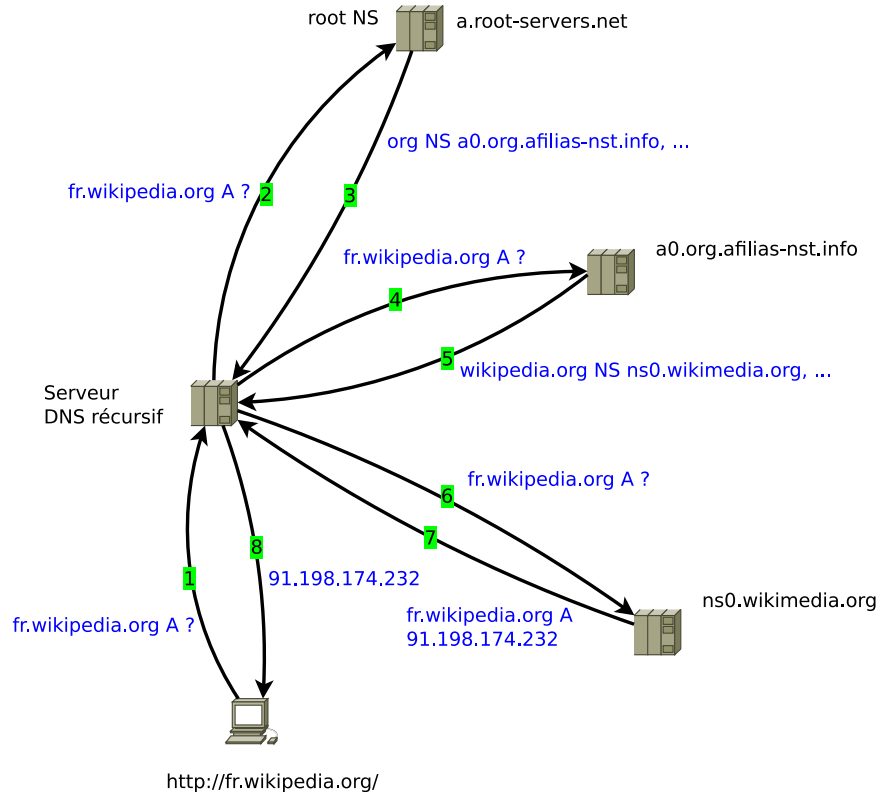
- Connecté + Segmentation
- Ordonné
- Fiable mais lourd
 - Handshake à l'établissement et la coupure de la connexion
 - Plusieurs types de correction d'erreur

Couches TCP/IP

5. Applications	HTTP	FTP	DNS	SMTP
4. Transport	TCP	UDP		
3. Réseau	IP	ICMP	IGMP	
2. Liaison	Ethernet	Token ring	PPP	FDDI
1. Physique	Cuivre	Satellite	Téléphonie sans fil	Fibre optique

Domain Name System

- Associe une IP à un nom de domaine
- Bureau d'enregistrement



Couches TCP/IP

5. Applications	HTTP	FTP	DNS	SMTP
4. Transport	TCP	UDP		
3. Réseau	IP	ICMP	IGMP	
2. Liaison	Ethernet	Token ring	PPP	FDDI
1. Physique	Cuivre	Satellite	Téléphonie sans fil	Fibre optique

HTTP

...et l'architecture client-serveur

HTTP

- HTTP + Url + HTML → WWW
- Plusieurs commandes pour spécifier une requête à un serveur
- Par défaut, HTTP est sans état (*stateless*)
→ guidé par l'état des ressources

En-têtes généraux

- URL
- Version de http
- Méthode utilisée (commande)
- Statut

Unified Resource Locator

- Permet d'identifier une ressource de manière unique

<https://serveur.uns.it:443/~johndoe/blog/bla/bla.php?a=bla&b=bla>

Unified Resource Locator

- Permet d'identifier une ressource de manière unique

`https://serveur.uns.it:443/~johndoe/blog/bla/bla.php?a=bla&b=bla`

- Protocole
- Nom de domaine (et sous-domaine) → Machine
- Port → Application
- Chemin d'accès → Ressource
- Variables

Rôle d'un serveur

- Url

/home/johnn/Public_Html/personnel/blog/bla/bla.php

https://serveur.uns.it:443/~johndoe/blog/bla/bla.php?a=bla&b=bla

- Identifier la ressource concernée
- Agir dessus selon la requête de l'utilisateur

```
1 Alias /~johndoe "/home/johnn/Public_Html/personnel"
2 ~
3 <Directory "/home/johnn/Public_Html/personnel">
4   » Options Indexes FollowSymLinks
5   » Options FollowSymLinks
6   » AllowOverride AuthConfig
7   » Options +Indexes
8   » Order allow,deny
9   » Allow from all
10  » Require all granted
11 </Directory>
```

Statuts

- Chaque réponse du serveur à une requête HTTP aura un statut qui permettra au client d'interpréter la réponse
- 200 : OK
- 301 : déplacée (permanent)
- 304 : non modifiée (utiliser le cache)
- 404 : non-trouvée
- [liste](#)

Méthodes

- GET : récupérer une représentation de la ressource
- HEAD : récupérer les métadonnées de la ressource
- POST : créer une ressource
- DELETE : supprimer la ressource
- PUT : remplacer l'état de la ressource d'après une représentation
- PATCH : modifier la ressource d'après sa représentation

Réponse et Type de média

- Certaines requêtes demandent du serveur de fournir une représentation de la ressource
- Cette représentation a un type (MIME / content-type)
 - `text/html`
 - `text/css`
 - `application/javascript`
 - `application/json`
 - `application/xml`

Échanger des données

Codage numérique de l'information
textuelle

Le code ASCII

American Standard Code for Information Interchange

USASCII code chart

Bits					Column	0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b ₇	b ₆	b ₅	b ₄	b ₃	Row	0	1	2	3	4	5	6	7
0	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
0	1	0	0	0	8	BS	CAN	(8	H	X	h	x
0	1	0	0	1	9	HT	EM)	9	I	Y	i	y
0	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
0	1	0	1	1	11	VT	ESC	+	;	K	[k	{
0	1	1	0	0	12	FF	FS	,	<	L	\	l	
0	1	1	0	1	13	CR	GS	-	=	M]	m	}
0	1	1	1	0	14	SO	RS	.	>	N	^	n	~
0	1	1	1	1	15	SI	US	/	?	O	_	o	DEL

文字化け

- Moji : Caractère
- Baké : Changé



Mais qui est donc ÁñéóôïôÝëçò ?

il est grec

ISO/IEC 8859-1

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_0																
1_16																
2_32	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3_48	ø	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
4_64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_128																
9_144																
A_160	NBSP	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
B_176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C_192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D_208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E_224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F_240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

ISO/IEC 8859-7

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_																
1_																
2_	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3_	ø	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_																
9_																
A_	NBSP	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
B_	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Mais qui est donc **ΆñέóôîôÝεçò** ?

il est grec

ISO/IEC 8859-1

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_0																
1_16																
2_32	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3_48	ø	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
4_64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_128																
9_144																
A_160	NBSP	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
B_176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C_192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D_208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E_224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F_240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

- C1 A
- F1 ρ
- E9 ι
- F3 σ
- F4 τ
- EF ο
- F4 τ
- DD ε
- EB λ
- E7 η
- F2 ς

ISO/IEC 8859-7

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_																
1_																
2_46	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3_	ø	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_																
9_																
A_	NBSP	ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
B_	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C_	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D_	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E_	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F_	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Mais qui est donc Ἀριστοτέλης ?

Aristote
Ἀριστοτέλης



Portrait d'après un original en bronze de Lysippe.

il est grec

C1 Α

F1 ρ

E9 ι

F3 σ

F4 τ

EF ο

F4 τ

DD έ

EB λ

E7 η

F2 ς

Une solution au
problème du Mojibaké :
Unicode

application/xml

- eXtensive Mark-up Language
 - Non-vide
 - Un seul élément racine
 - Structure arborescente
 - Ne fait appel qu'à des documents bien formés

- Élément :
`<conteneur attr1="val1" ... attrn="valn">Contenu de l'élément</conteneur>`
 - Contenu de l'élément est un sous-arbre.
- Élément vide :
`<marqueur attr1="val1" ... attrn="valn"/>`
- Commentaire
`<!-- ceci <est> un commentaire, il peut contenir une balise, mais ne peut pas être inclus dedans... -->`

application/xml

- **eXtensive Mark-up Language**
 - Noms de balises non prédéfinis...
 - ...dépendant d'un type de document

- Déclaration

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
```
- Type de document

```
<!DOCTYPE Root SYSTEM "URL">
```

 - Root : élément racine
 - URL : adresse de la ressource

application/json

- JavaScript Object Notation
- Notation
 - { } → objet
 - [] → tableau
 - "attribut" : valeur → attribut
- Possibilité de spécifier un type de document avec json/schema

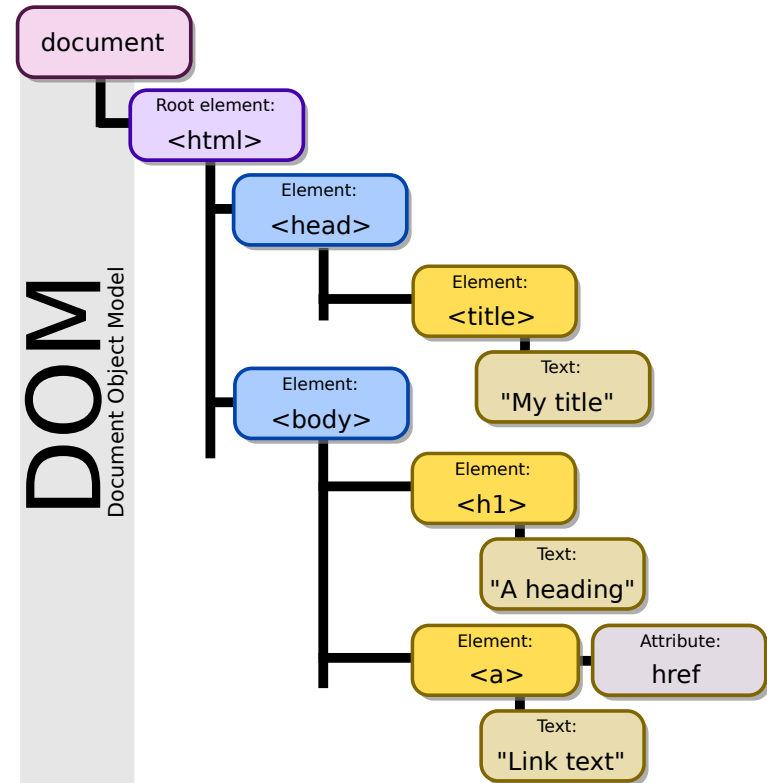
```
var monObjetComposite = {  
  > "attribut1": "valeur",  
  > "attribut2": "valeur",  
  > "etc": "et cætera"  
};  
  
var personne = {  
  > "prenom" : "John",  
  > "nom" : "Doe",  
  > "age" : 66,  
  > "adressesMail" : [  
    >> {"type": "perso",  
    >>   "adresse": "jdoe@mail.com"},  
    >> {"type": "pro",  
    >>   "adresse": "jd@anonymous.net"}  
  ]  
};
```

Manipuler des données

...en Python TD2

XML et DOM

- Document Object Model
- API normalisée par le w3c pour manipuler des documents HTML et XML (cf. navigateurs)



JSON en Python

decode / load

JSON → Python	
object	dict
array	list
string	str
number (int)	int
number (real)	float
true	True
false	False
null	None

encode / dump

Python → JSON	
dict	object
list, tuple	array
str	string
int, float, int- & float-derived Enums	number
True	true
False	false
None	null

Pour le TD n°2

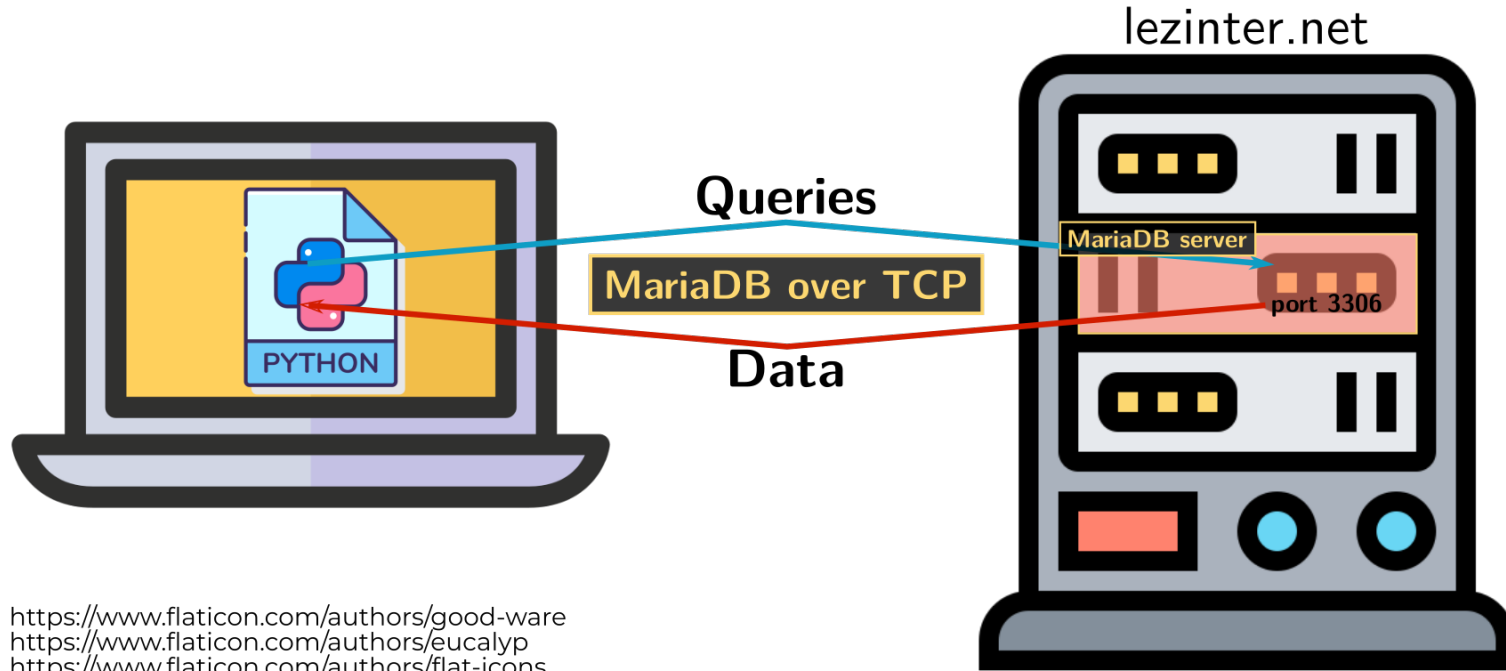
Formule de Haversine

$$d = 2r \arcsin \left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)} \right)$$
$$= 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Rappels SQL

Des bases de données aux données
structurées et *vice-versa*

Architecture TD n°3



<https://www.flaticon.com/authors/good-ware>
<https://www.flaticon.com/authors/eucalyp>
<https://www.flaticon.com/authors/flat-icons>

Python-mysql.connector

```
try:
    #connection
    db = mysql.connector.connect(host="url.fr", user="aUser", password="&A1;e", database="mydb")
    cursor = db.cursor() #to query the db
    cursor.execute("INSERT INTO `users` (`name`,`pass`) VALUES ('u1','zer'), ('u2','%1o')")
    insert_id = cursor.lastrowid #id of the last inserted record
    cursor.execute("SELECT `id`, `name` FROM `users`")
    results = cursor.fetchall() #create a list of tuples with all results
    foreach res in results:
        if (res[0]==insert_id): print(res,"← last inserted user")
        else: print(res)
    db.commit()#validate the alterations of the tables
except mysql.connector.Error as e:
    print(f"Error: {e}") #execute in case of error in the above
```

SGBD et langage de requête

- Un **Système de Gestion de Bases de Données** est un système informatique permettant de stocker, organiser, modifier, rechercher des données
 - MySQL, MariaDB, OpenOffice Base, etc. sont des SGBD relationnelles
- L'interaction avec ces systèmes se fait en général par l'usage d'un langage.
- Pour les BD relationnelles **SQL** est le langage le plus fréquemment utilisé
- *Phpmyadmin* **n'est pas un SGBD**, ni un langage, c'est une interface pour interagir avec le SGBD d'un ordinateur.

Organisation des données

- En général, on va créer une **base de données** différente pour chaque projet
 - Parfois, chaque utilisateur d'un système n'a qu'une base de données (utilisation de préfixes)
- Dans une base de données, les informations sont stockées dans des **tables**
- Les tables spécifient un ensemble de **colonnes**
(≈liste des champs à remplir)
- Chaque donnée est un **enregistrement** (une ligne) dans une table (spécifiant une valeur — ou non) pour chaque colonne
- L'une des fonctions de SQL est de donner un moyen de spécifier toutes ces structures.

4 opérations de base

- La manipulation des données (sous-ensemble de SQL) se fonde sur 4 opérations principales :
 - Sélectionner des données `SELECT`
 - Ajouter des données `INSERT`
 - Mettre à jour des données `UPDATE`
 - Supprimer des données `DELETE`

SELECT

- Pour récupérer un ensemble d'informations dans *une* table
`SELECT `col1`, `col2` FROM `table` WHERE condition ;`
 - On indique dans quelle table on cherche
 - On sélectionne les colonnes qui nous intéressent (* pour toutes les colonnes)
- Les conditions permettent un ensemble d'opération sur les colonnes ou des valeurs. Par exemple
 - `SELECT * FROM `users` WHERE `users`.`prenom` = 'Jean'`
 - `SELECT `nom`, `prenom` FROM `users` ;`
`WHERE `users`.`age` > 40 AND `users`.`age` < 51 ;`

INSERT

- Pour ajouter un ou plusieurs enregistrement dans une table
`INSERT INTO `table` (`col1`, `col2`) VALUES ('val1', 'val2'), ('val1', 'val4') ;`
 - Les colonnes non indiquées dans la première parenthèse prendront la valeur par défaut ;
 - On peut ajouter plusieurs enregistrements en une requête en séparant les enregistrements entre parenthèses par des virgules
- Exemple
 - `INSERT INTO `users` (`nom`, `prenom`, `age`) VALUES ('Jean', 'Michel', 46), ('Jean', 'Micheline', 24), ('Jean', 'Killian', 1), ('Jean', 'Zinedine', 20) ;`

UPDATE

- Modifier un ou plusieurs enregistrement dans une table

```
UPDATE `table` SET `col1` = 'val1' WHERE `col2` =  
'val2' ;
```

- On modifie tous les enregistrements qui respectent la condition

- Exemple

```
UPDATE `users` SET `age` = `age` + 1  
WHERE `nom` = 'Jean' ;
```

DELETE

- Pour supprimer 1 ou plusieurs enregistrements
`DELETE FROM `table` WHERE condition`
- On indique quelle table est concernée par la suppression
- On indique sur quel critère on supprime les enregistrements
 - `DELETE FROM `users`
WHERE `users`.`nom` = 'Dupont' ;`
 - `DELETE FROM `users`
WHERE `users`.`age` > 18 AND `users`.`age` < 30 ;`

Clés étrangères

- **FOREIGN KEY** (clé étrangère)
 - Identifie les valeurs d'une colonne `t1`.`c1` comme liées à celles d'une colonne `t2`.`c2` (clé primaire ou unique de la table t2) → rôle logique
 - Permettent de
 - vérifier/maintenir l'intégrité de la table
 - Déclencher des actions dans `t1` en cas de certaines opérations dans `t2`

Clés / indexes

- Pour trouver un enregistrement en fonction d'une colonne, le seul moyen est de parcourir toutes les lignes une à une et de regarder la valeur de la colonne...
- Pour gagner du temps, on peut trier la table selon cette colonne...
- Mais que se passe-t-il si on veut pouvoir rechercher dans une table en fonction de plusieurs colonnes selon les cas...
 - Ex : chercher dans un annuaire par nom ou par prénom...
- On peut définir chaque colonne comme un index (ou clé) pour que le système crée ce tri.

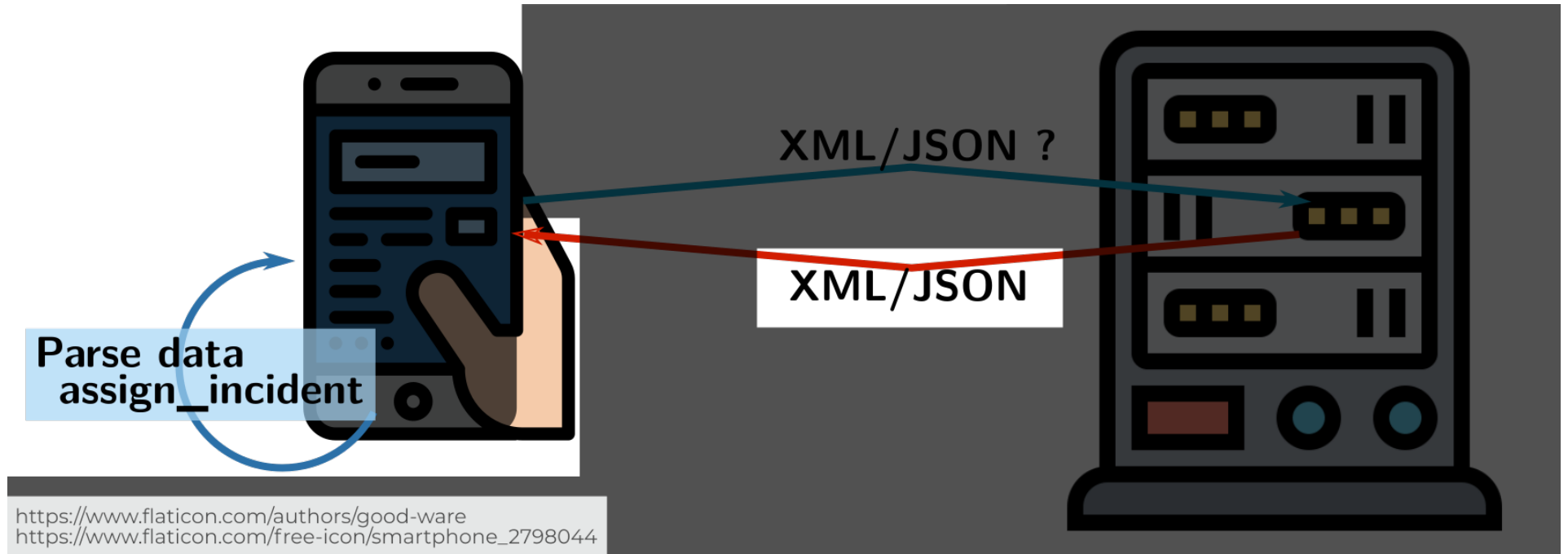
Types d'index

- **PRIMARY KEY** (`cle primaire`)
 - Chaque valeur de la colonne est unique
 - Tous les enregistrements ont une valeur dans cette colonne
 - Une seule clé primaire par table
 - Cette colonne permet d'identifier de manière unique chaque enregistrement
- **UNIQUE** (`colonnes` , `a` , `valeur` , `unique`)
 - Chaque valeur de la colonne (ou du n-uplet de colonnes) est unique
 - Cette colonne (ou ensemble de colonnes) permet d'identifier de manière unique les enregistrements où elle est renseignée
- **INDEX** (`index`)
 - N'est pas nécessairement unique

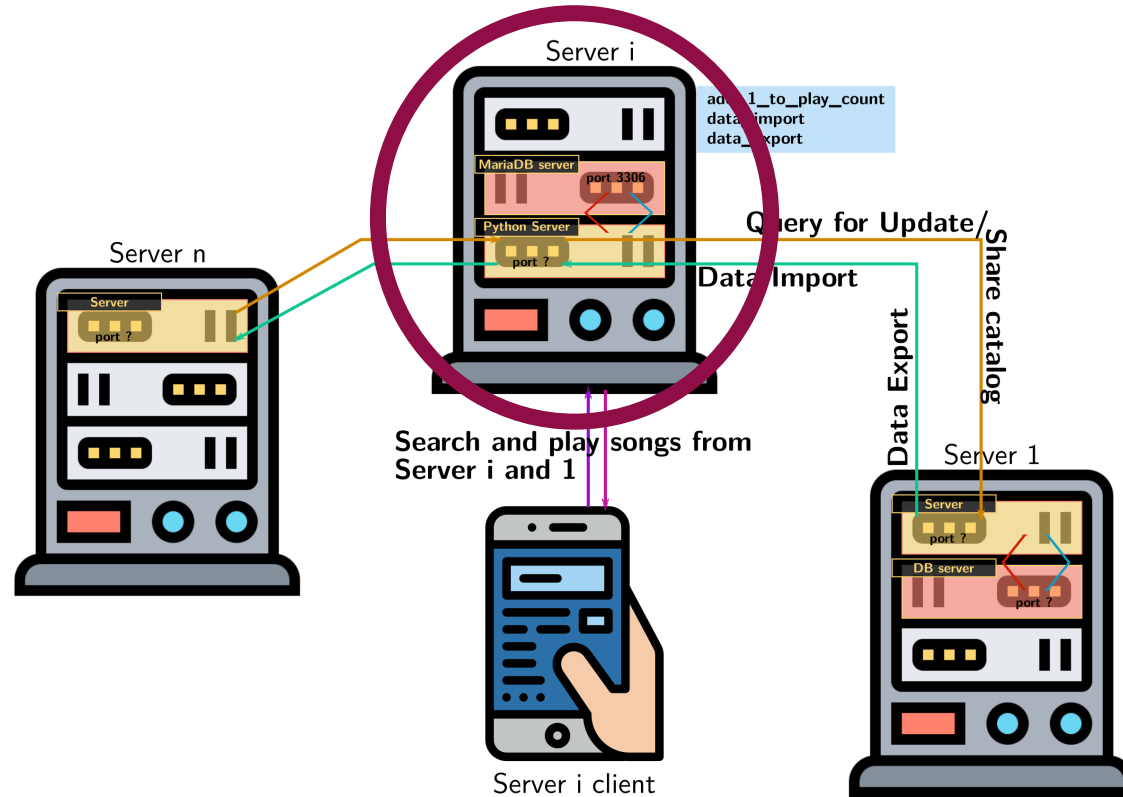
Patrons de conception d'application

Architectures communes de systèmes
informatiques

Architecture du système du TD2



Architecture du système du TD3



Projet

Clients et serveurs

...en « agile » avec git

Gestion de projet

Git + Gitlab

Position du problème

- La version 1 d'un logiciel fonctionne
 - Pour arriver à la version 2, une dizaine de bugs mineurs doivent être corrigés
 - Chaque bug est corrigé, mais à un moment dans la chaîne de production, quelque chose s'est cassé...
 - Il faudrait revenir en arrière, ce serait plus simple de ne pas ajouter l'erreur que de la corriger...
- 2 codeurs (A et B)
 - A travaille sur le module 1
 - B travaille sur le module 2
 - → pas de problème...
 - ...sauf que les deux modules dépendent d'une librairie commune développée pour le projet
 - A et B ont chacun besoin d'une fonction spécifique qu'ils ajoutent à la librairie
 - A et B mutualisent leur travail : rien ne marche
 - A se rend compte qu'il n'a pas ajouté sa modification de la librairie, espoir...
 - Il remplace le fichier lib.js par le sien... ça ne marche toujours pas
 - B se rend compte que la librairie doit aussi contenir ses modifications...
 - Galère...

Qu'est-ce que "git" ?

- Créé par Linus Torvalds
- Open source (<http://git-scm.com/downloads>)
- En ligne de commande (par défaut) :

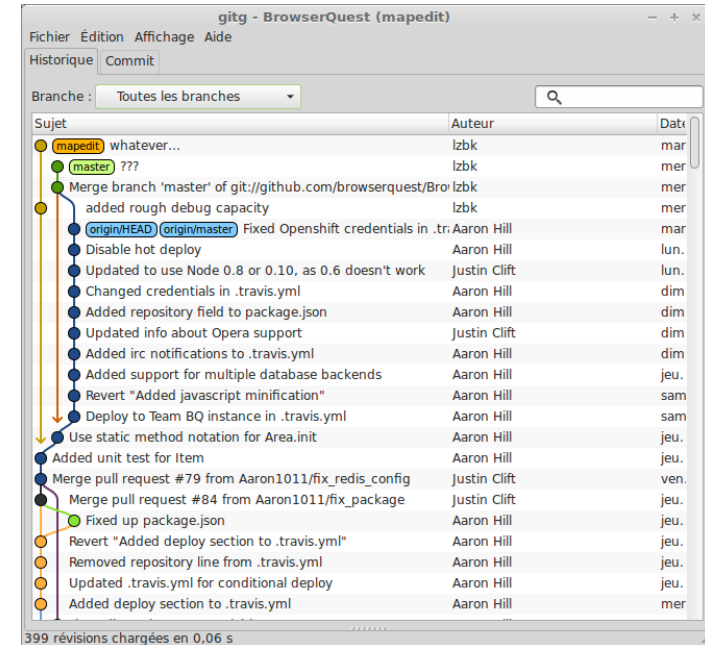
```
loizbek@bajaja: ~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest
Fichier Édition Affichage Rechercher Terminal Aide
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin browserquest.conf client launchServer.sh LICENSE node_modules package.json README.md server shared tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
* janus
  mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git checkout mapedit
Basculement sur la branche 'mapedit'
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
  janus
* mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin          client      epi1.html  histart1.html  index.html    LICENSE      package.json  quizz      server  TODO.md
browserquest.conf  dump.rdb  epi2.html  histart2.html  launchServer.sh  node_modules  print        README.md  shared  tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$
```

- Pour gérer des révisions...

Qu'est-ce que "git" ?

- Créé par Linus Torvalds
- Open source
<http://git-scm.com/downloads>
- Plusieurs points de vue
 - En ligne de commande...
 - Avec une interface graphique...


```
loizbek@bajaja: ~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest
Fichier Édition Affichage Rechercher Terminal Aide
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin browserquest.conf client launchServer.sh LICENSE node_modules package.json README.md server shared tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
* janus
  mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git checkout mapedit
Basculement sur la branche 'mapedit'
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
* janus
* mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin client epi1.html histart1.html index.html LICENSE package.json quizz server TODO.md
browserquest.conf dump.rdb epi2.html histart2.html launchServer.sh node_modules print README.md shared tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$
```



~~Des~~ La commande_s “git” (sous-)

- `git config` : Configurer l'outil git
- `git init` : Créer un dépôt git
- `git clone` : Créer un dépôt à partir d'un dépôt existant
- `git add` : Sélectionner des changements pour le prochain commit
- `git reset` : Désélectionner des changements pour le prochain commit
- `git commit` : Créer une nouvelle version
- `git checkout` : Changer de branche
- `git merge` : Fusionner deux branches
- ...

Cycle de vie (de base) d'un projet git

- Dépôt local
 - Initialisation
 - `git init`
 - Travail local
 - Modifications
 - Préparation du commit :
 - `git add fichier`
 - `git add .`
 - Commit
 - `git commit -m "<explication commit>"`
 - Dépôt distant
 - Initialisation
 - `git clone <adresse>`
 - Travail
 - `git pull <dépôt> <branche>`
 - Travail local
 - `git push <dépôt> <branche>`
- 

Scrum en “1” diapositive

Backlog produit

User story 1

User story 2

User story 3

User story 4

User story 5

User story 6

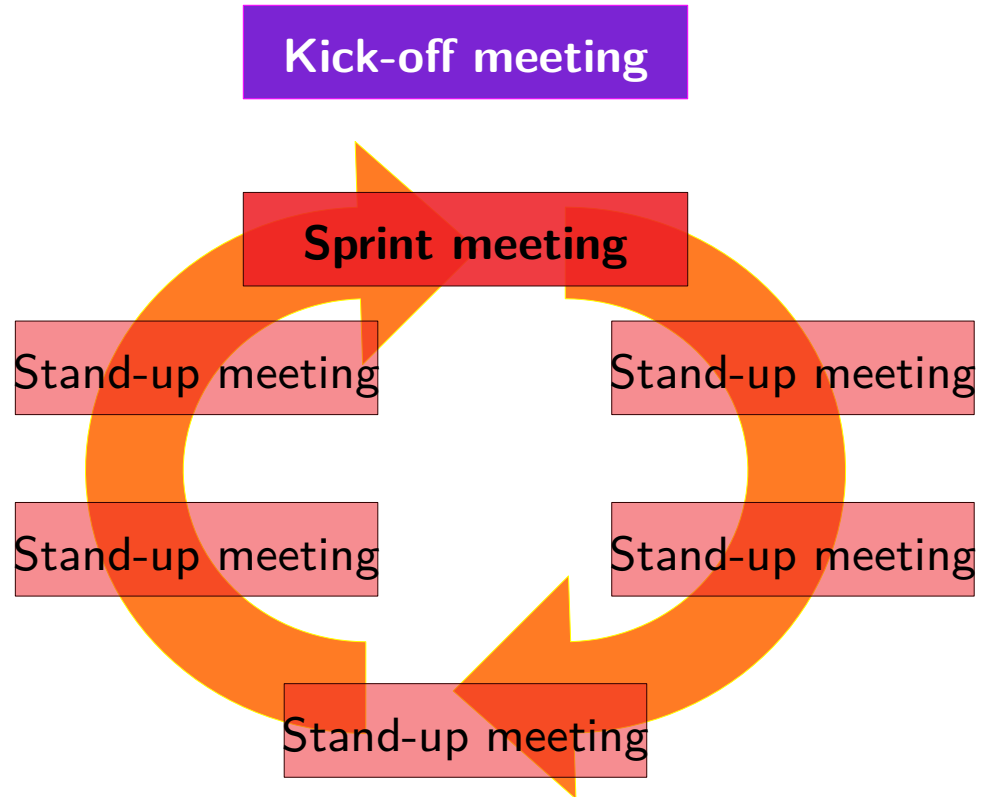
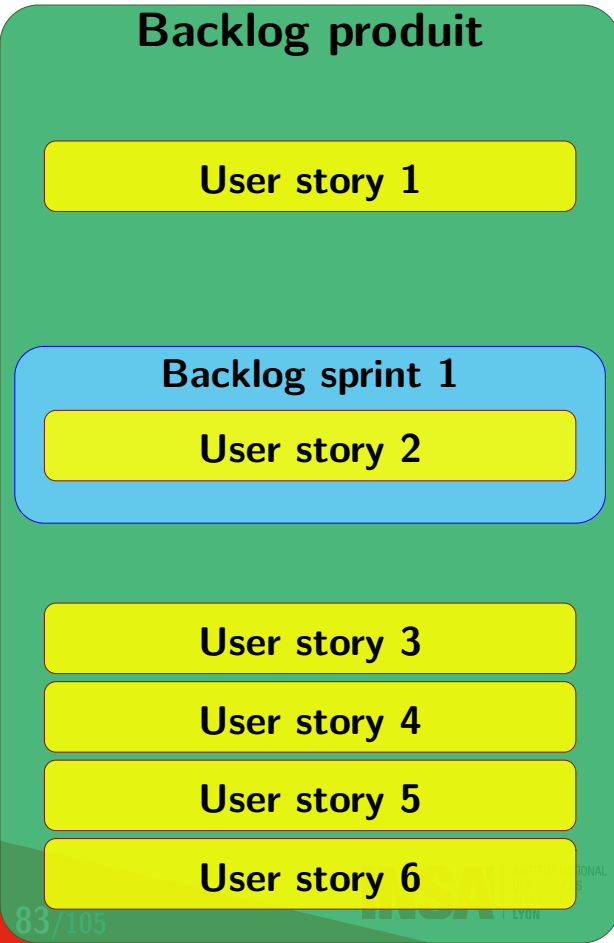
Récit utilisateur type :

« En tant que <qui>, je
veux <quoi> afin de
<pourquoi> »

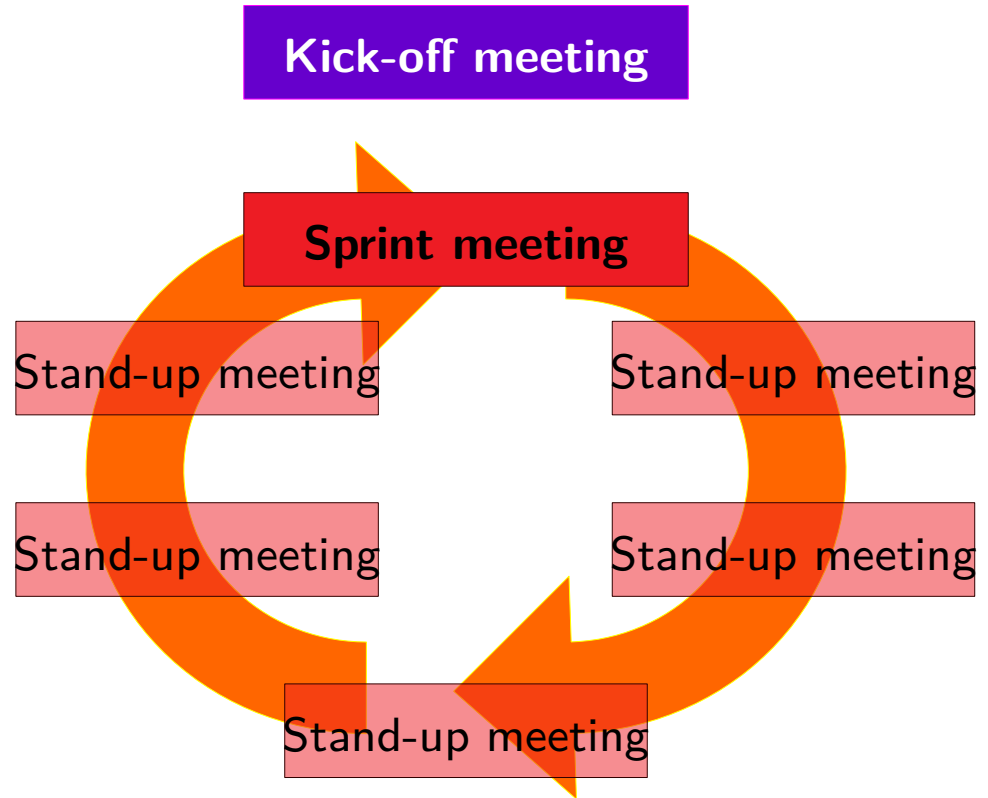
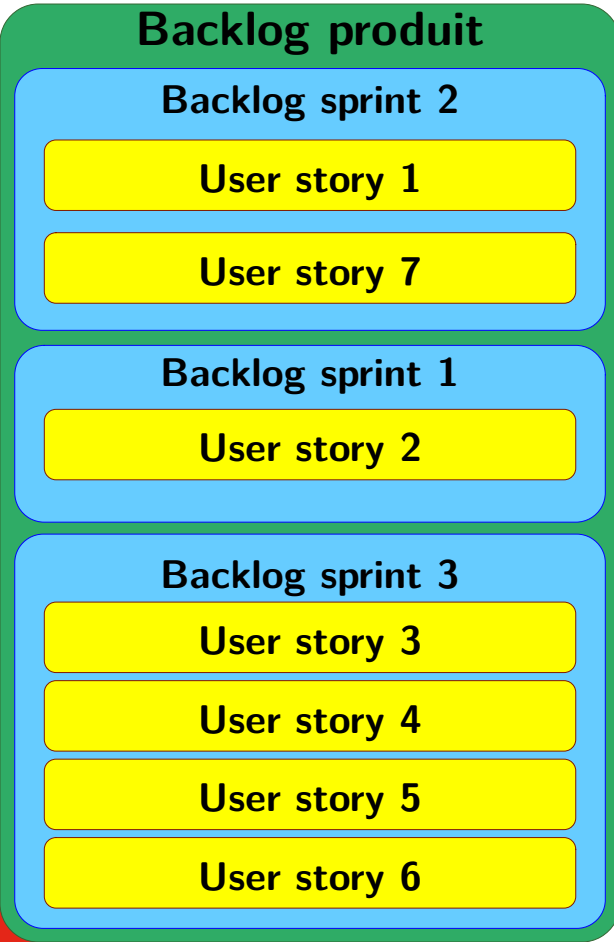
Kick-off meeting

- Valider le backlog produit
- Déterminer
 - La durée des sprints
 - Le contenu du premier sprint
 - (Backlog de sprint)

Scrum en “1” diapositive



Scrum en “1” diapositive



La syntaxe Markdown

- Titres

Titre 1

==

Title 2

-

Title 3

Title 4

- Marques de paragraphe

→ ¶¶

- Sauts de ligne

→ 2 espaces en fin de ligne

La syntaxe Markdown

Listes

- Listes à puces

- * e^t de liste
 - * e^t de sous-liste
 - * e^t de sous-liste
- * e^t de liste
- * e^t de liste

- Liste numérotée

1. e^t de liste numérotée
2. e^t de liste numérotée
 1. e^t de liste numérotée
 3. e^t de liste numérotée
7. e^t de liste numérotée

Issues gitlab

- Chaque « récit utilisateur », besoin, fonctionnalité, bug est un ticket
 - Identifiant
 - Intitulé
 - Description
 - Tags
 - Deadlines
 - Responsable

Milestones gitlab

- On peut assigner des tickets à des jalons, qui peuvent correspondre à des sprints (ou des objectifs à plus long terme)
 - Identifiant
 - Intitulé
 - Description
 - Deadline
 - Liste de *issues* associés

Checklists dans markdown

- [] Un élément de *checklist*
- [] Un autre élément de *checklist*
- [] Un élément de *checklist*



Mathieu Loiseau @lzbk commented 30 seconds ago

Owner



- Un élément de checklist
- Un autre élément de checklist
- Un élément de checklist

Edited by Mathieu Loiseau 30 seconds ago

Liens

- Avec markdown :
 - [texte affiché]
(destination)
 - #22 → lien vers le ticket d'id 22
 - %4 → lien vers le jalon d'id 4
 - \$2 → lien vers le « snippet » d'id 2
- On peut faire un lien vers une page de code, même vers une ligne (quand on clique sur le numéro de ligne on a une url vers cette ligne)

Liens

Url d'une ligne de code précise pour créer un lien

```
118 initialize = function (instance_url, user, pass){
119     tryCatch({
120         self$url <- paste(instance_url, "/", "api.php", sep="")
121         self$user_name <- user
122         self$auth_token <- self$connect(pass)
123     },
124     error = function(e){
125         print(e$message)
126         self <- FALSE
127     })
128 },
129
130 #####
131 # setPage
132 #####
133 setPage = function(pageName){
134     self$page_name <- pageName
135 },
136
137 #####
138 # resetContent
139 #####
```

Lien depuis les commits

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
commit 33c5b67612a5b55a69a770bfd97bc2fef303138
Author: Mathieu Loiseau <mathieu.loiseau@univ-grenoble-alpes.fr>
Date:   Wed Dec 12 09:00:28 2018 +0100

    améliorer doc 2

commit e1e2d708fc698a9f9888fb7246ecb14255209b1b
Author: Mathieu Loiseau <mathieu.loiseau@univ-grenoble-alpes.fr>
Date:   Wed Dec 12 08:59:10 2018 +0100

    Améliorer la doc #2
```

mediawikir

Mathieu Loiseau > mediawikir > Commits

master mediawikir

12 déc., 2018 2 commits

- améliorer doc 2** Mathieu Loiseau a créé il y a 2 minutes 33c5b676
- Améliorer la doc #2** Mathieu Loiseau a créé il y a 3 minutes e1e2d708

10 déc., 2018 2 commits

- Now works with just one field** Mathieu Loiseau a créé il y a une journée f17a8da3


92/10

Lien depuis les commits

- ```
git commit -m "résoudre #2"
```

Mathieu Loiseau >  mediawiki > Issues > #2

Closed

Opened il y a 5 minutes by  Mathieu Loiseau

Reopen ticket

New issue

## Améliorer la doc



0



0



Afficher toute l'activité ▾



Mathieu Loiseau @lzbk mentioned in commit [e1e2d708](#) 3 minutes ago



Mathieu Loiseau @lzbk closed 2 minutes ago

# Lien vers les commits



Write Preview

B I

Corrigé dans le commit `f17a8da3859f001856704cc873dcb6859dea68dd`

Markdown is supported

Attach a file

Save comment

Cancel

## Commits

Branches

Étiquettes

Contributeurs

Graphique

Comparer

Statistiques



**Améliorer la doc #2**

Mathieu Loiseau a créé il y a 3 minutes

Améliorer la doc

10 déc., 2018 2 commits



**Now works with just one field**

Mathieu Loiseau a créé il y a une journée



**stringr imports + link err correction**

Mathieu Loiseau a créé il y a 2 jours

09 déc., 2018 1 commit

Filtrer par message de commit

33c5b676



e1e2d708



f17a8da3



Copier le condensat SHA du commit

976c4089



# Lien vers les commits

The image shows a GitHub interface with a commit message editor at the top and a list of commits below. In the editor, the text "Corrigé dans le commit" is followed by the full commit hash "f17a8da3859f001856704cc873dcb6859dea68dd", which is enclosed in a black box. In the commit list, the commit "Corrigé dans le commit" by Mathieu Loiseau is highlighted with a red box around the truncated hash "f17a8da3". A red arrow points from this box to the full hash in the editor. A green arrow points from the "f17a8da3" box in the list to a tooltip that says "Copier le condensat SHA du commit".

**Write** Preview

Corrigé dans le commit `f17a8da3859f001856704cc873dcb6859dea68dd`

Mathieu Loiseau @lzbk commenté 5 minutes ago

Corrigé dans le commit `f17a8da3`

Edited by Mathieu Loiseau 30 seconds ago

Améliorer la doc #2  
Mathieu Loiseau 3 minutes

10 déc., 2018 2 commits

Now works with just one field  
Mathieu Loiseau a créé il y a une journée

stringr imports + link err correction  
Mathieu Loiseau a créé il y a 2 jours

09 déc. 2018 1 commit

Filtrer par message de commit

33c5b676


e1e2d708



f17a8da3

Copier le condensat SHA du commit

# Lien vers les commits

- Vue du commit concerné...

Mathieu Loiseau >  mediawiki > Commits > **f17a8da3**

Commit **f17a8da3**  authored il y a une journée par  **Mathieu Loiseau**


 1

Parcourir les fichiers

Options ▾

## Now works with just one field

 parent [976c4080](#)  master

 Aucune demande de fusion associée trouvée

Showing **1 changed file** ▾ with **29 additions** and **11 deletions**

Hide whitespace changes

Inline

Side-by-side

▶  **R/class.R** 



Voir le fichier @ f17a8da3

This diff is collapsed. [Click to expand it.](#)



Mathieu Loiseau @lzbk mentioned in issue [#1 \(closed\)](#) · il y a 14 minutes

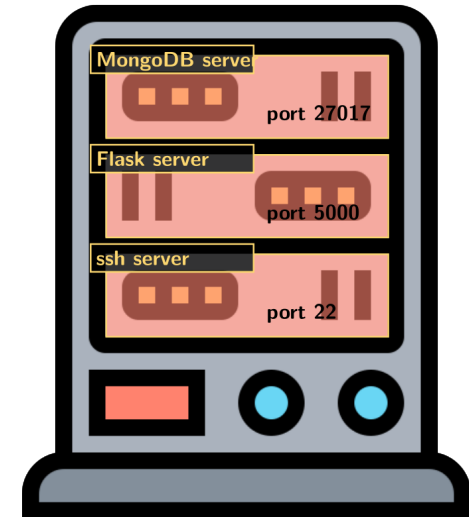
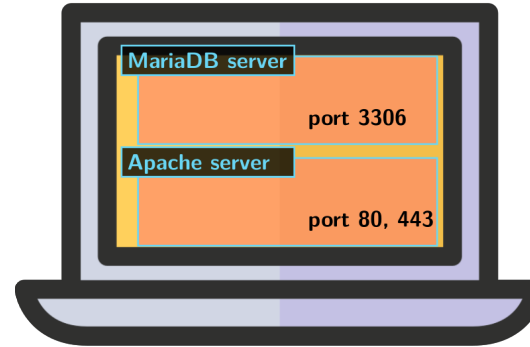


# Client, Serveur et API

Point de vue pratique en python avec  
Flask

# Serveur : Rappels

- Un **serveur** est un logiciel capable de répondre à des **requêtes** d'autres logiciels (**clients**). Un serveur écoute en général les messages sur un **port** donné.



# Serveur : Rappels

- Client : le programme qui envoie une requête
- Serveur : le programme qui écoute la requête et répond au client
- Souvent client sur un ordinateur et serveur sur un autre, mais ce n'est même pas nécessaire
  - Ex : serveur Web interrogeant un serveur de base de données (le serveur Web joue alors le rôle de client du serveur de BD)

# Projet : Liste des messages échangés

- En tant que particulier

- I say
- I hear
- Déclarer le covid et informer « l'hôpital »
- Demander la liste they said à l'hôpital

- En tant qu'hôpital

- Recevoir une liste de message I said
- Envoyer une liste de message they said

Client

Serveur

# RESTful API

- Architecture REST

- REpresentational State Transfer
- 6 contraintes architecturales pour améliorer performance, simplicité, évolutivité, visibilité, portabilité et fiabilité

1) Client-serveur

2) Sans état

3) Interface uniforme

- **ressources** identifiées dans la requête
- manipulation de représentations
- **message auto-descriptif**
- hypermédia comme moteur d'état

4) Mise en cache

5) En couches

6) Code à la demande (facultatif)

# Projet : Liste des messages échangés

- En tant que particulier

- I say
  - I hear
  - Déclarer le covid et informer « l'hôpital »
  - Demander la liste they said à l'hôpital
- Route 1 / POST
- POST
- Route 2 / they-said
- GET
- 

- En tant qu'hôpital

- Recevoir une liste de message I said
- Envoyer une liste de message they said

Client

Serveur

# Construire des routes avec Flask

```
@app.route('/', methods=['GET'])
def maFonction():
```

- `Response(msg)`

- Classe pour créer une réponse de serveur
- Paramètres optionnels du constructeur
  - `mimetype`
  - `status`

- `@` : decorateur
  - Identifie
    - le chemin de la ressource pris en charge
    - les méthodes autorisées
  - Associé à une fonction déclenchée quand la ressource est interrogée

# Construire des routes avec Flask 2

```
@app.route('/chemin/<param>',
methods=['POST'])
def maFonction(param):
```

- Objet request

- `request.method` : méthode utilisée par le client (GET, POST, etc.)
- `request.is_json` : données json envoyées par le client ? (booléen)
- `request.get_json()` les données json envoyées par le client (`str`)

- `<>` permet d'inclure des données dans l'url
  - Peuvent être rappelées comme paramètre de la fonction suivante



# Envoyer des requêtes avec `requests`

- `pip install requests`
- Envoyer une requête
  - `r = requests.get(url)`
  - `r = requests.post(url)`
  - `r = requests.post(url, json=data)`
- Recevoir la réponse du serveur
  - `r.status_code`
  - `r.text`
  - `r.json()`