CENTRALE

RESEAUX

Amphi Révision de SI

Par Stanislas Plessia, Julien Raspaud, Edmond de Roffignac et Nayef Ghattas

CENTRALE

RESEAUX

I – Bases de Données

II – Architecture d’un Ordinateur

III – Réseaux

IV – Sécurité

V – Télécom

CENTRALE

RESEAUX

Base de Données

Modélisation et SQL

Par Stanislas ‘Over’ Plessia
stanislas.plessia@student.ecp.fr

D’après un très légitime plagiat de ‘Wiwi’

Une Base de donnée ? C'est quoi?

- **Ensemble de Tables de Valeurs**
- **Objectifs**
  - Stocker un grand nombre de données
  - Hiérarchiser et Structurer ces données
  - Faciliter l'accès et le traitement

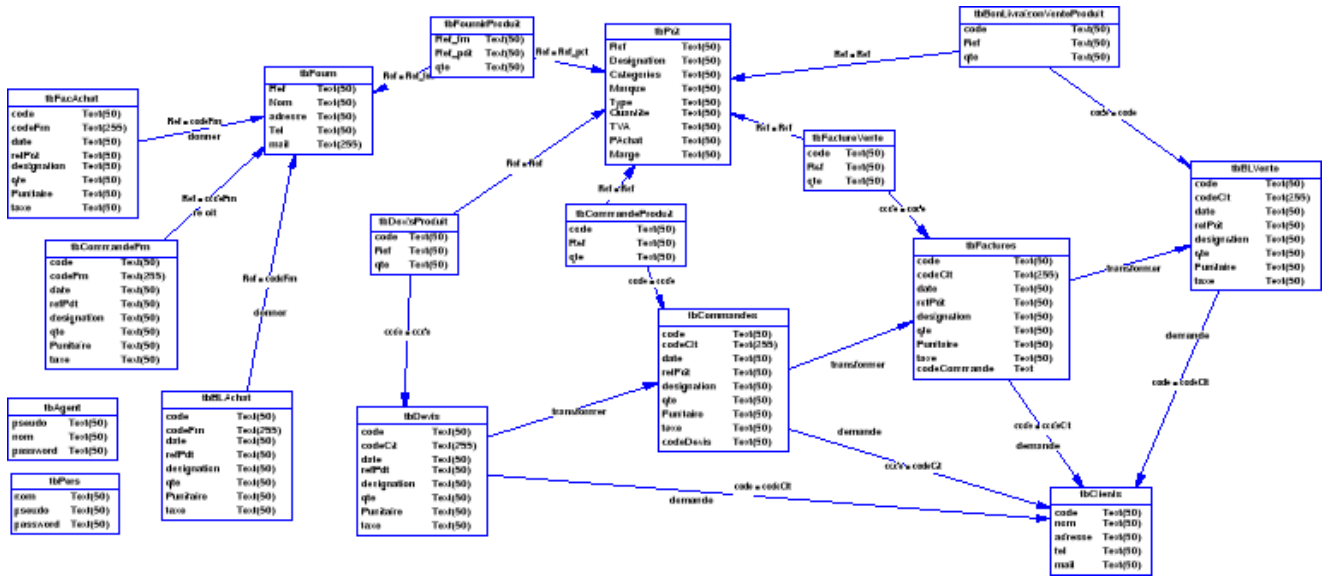
Exemple

Numéro Nicolas gpa : 0682828282 - Chambre Pierre : 13C – Num chambre Camille : F211 – Nico Henri N Chambre : 202G – Nico G Num : +33739393939 _ Nico Giro : F222 -\ Nico 2A : 0681818181 ; Camille Martin num : 612311232 – Nico gpa Dupont num : 118D – Vincent Pierre : 0712121212 – Grégoire Deschamps : 215H

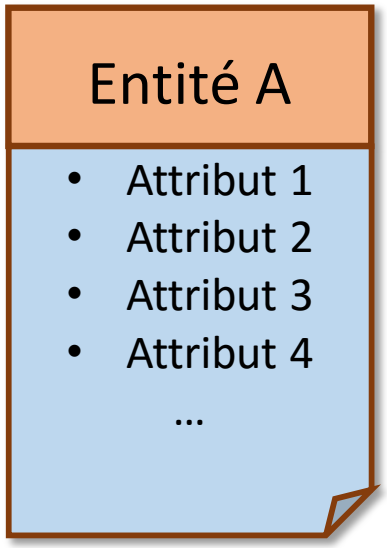
Organisation d'un ensemble de données chaotique

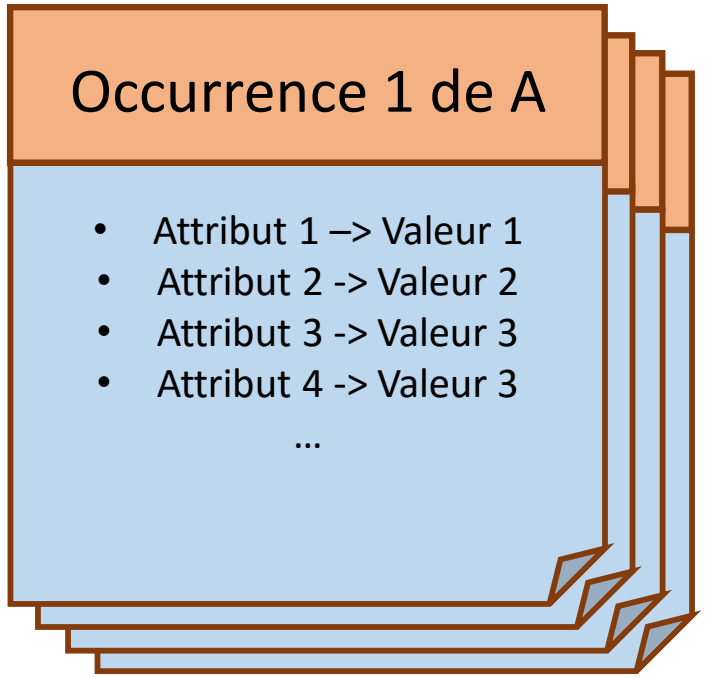
Amis			
Prénom	Nom	Chambre	Téléphone
Camille	Martin	F211	0612311232
Nicolas	Giro	F222	0739393939
Grégoire	Deschamps	H215	
Nicolas	Dupont	D118	0682828282
Nicolas	Henri	G202	0681818181
Pierre	Vincent	C013	0712121212

Modélisation
Logique/Physique



Entité/Occurence





Entité = boite ou tiroir (nom commun)

- Exemple : étudiant, voiture

Attribut = caractéristique de l'entité

- Exemple : âge, rôle en entreprise

Occurrence = Instance ou élément

- Exemple : un étudiant, trois voitures

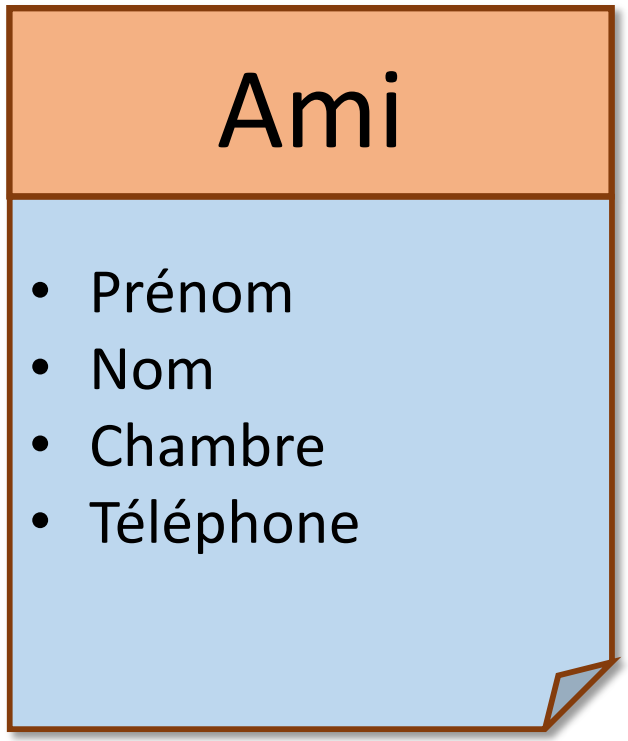
Valeur= une valeur de l'attribut

- Exemple : 20 ans, trésorier

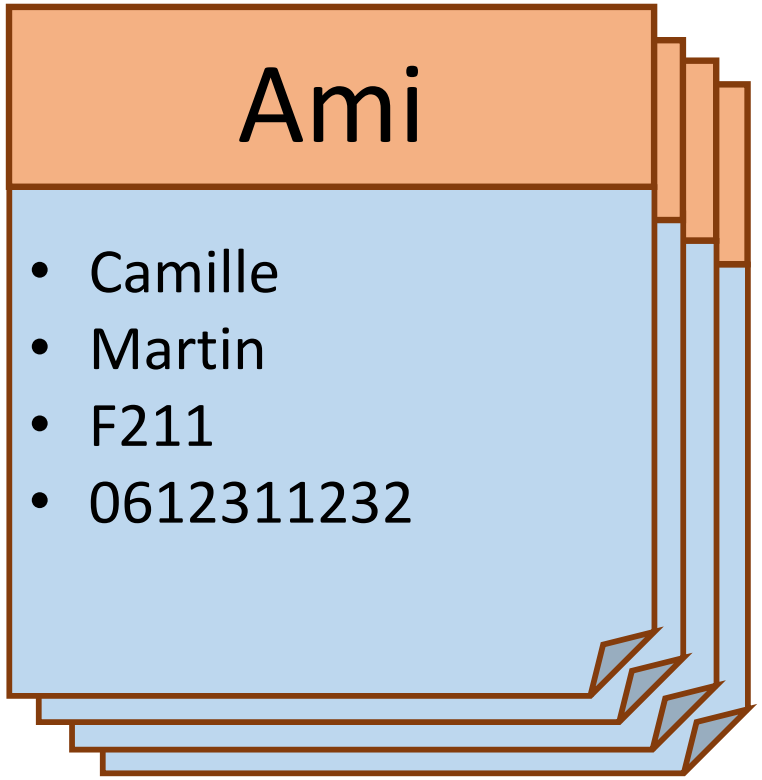
Exemples

Fiches

Entité

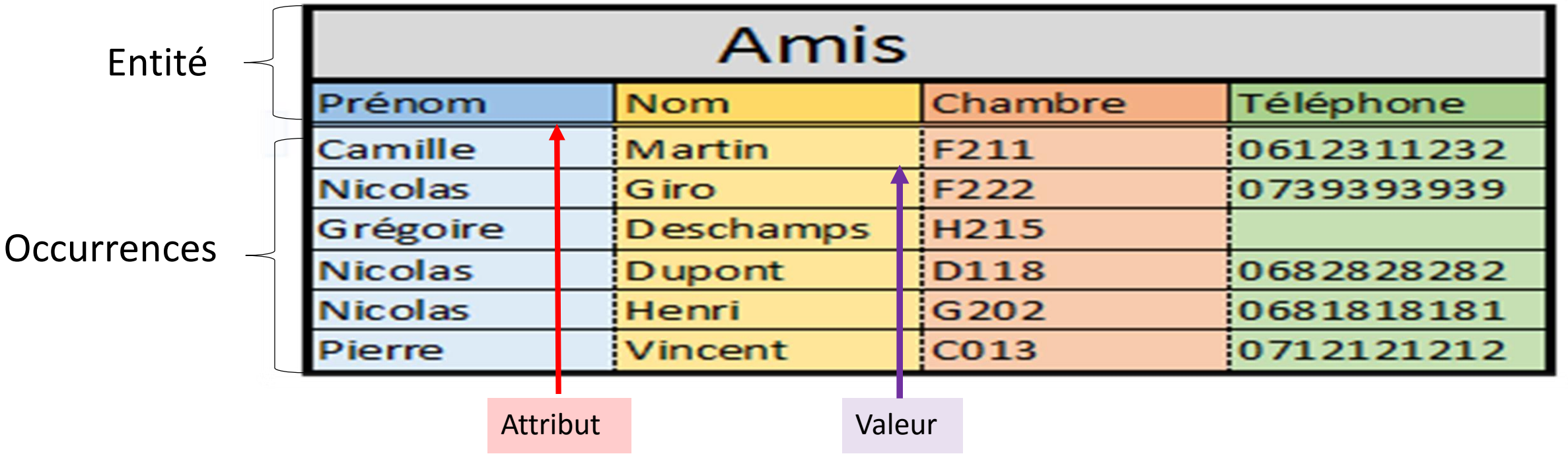


Occurrences



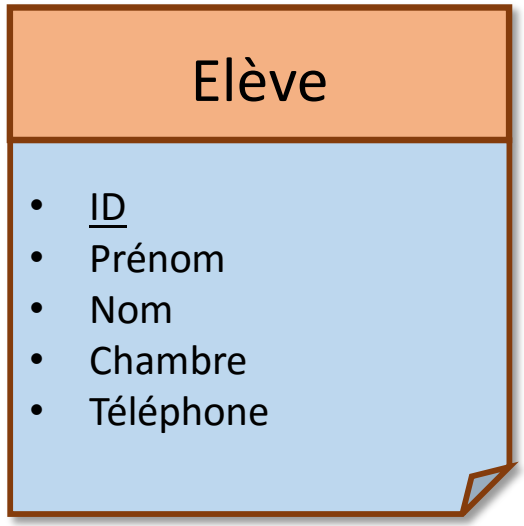
Exemples

Tableau



Identifiant
Descripteur

- Ce sont des attributs.
- L'identifiant décrit une occurrence, il est unique, caractéristique (souvent un nombre) et souligné. Choisi au pif (N° d'inscription) ou bien avec un code (sécu sociale).
- Le descripteur est un attribut non identifiant (pas unique).
- Les attributs ont des conditions d'entrée (entier, date …) pour éviter les erreurs et protéger la base de donnée (empêcher de rentrer des ordres supplémentaires)

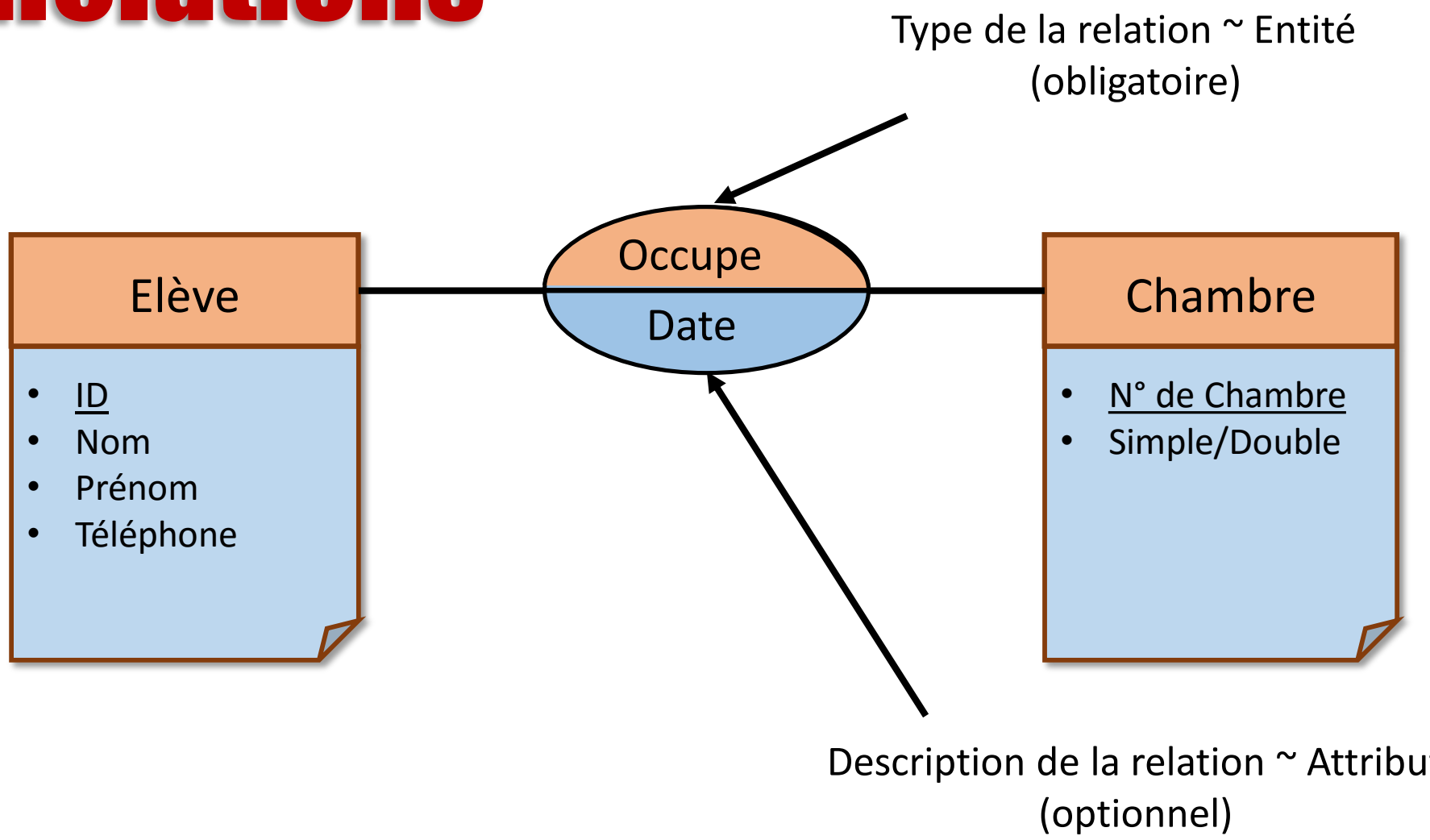


Modélisation Logique

Relations

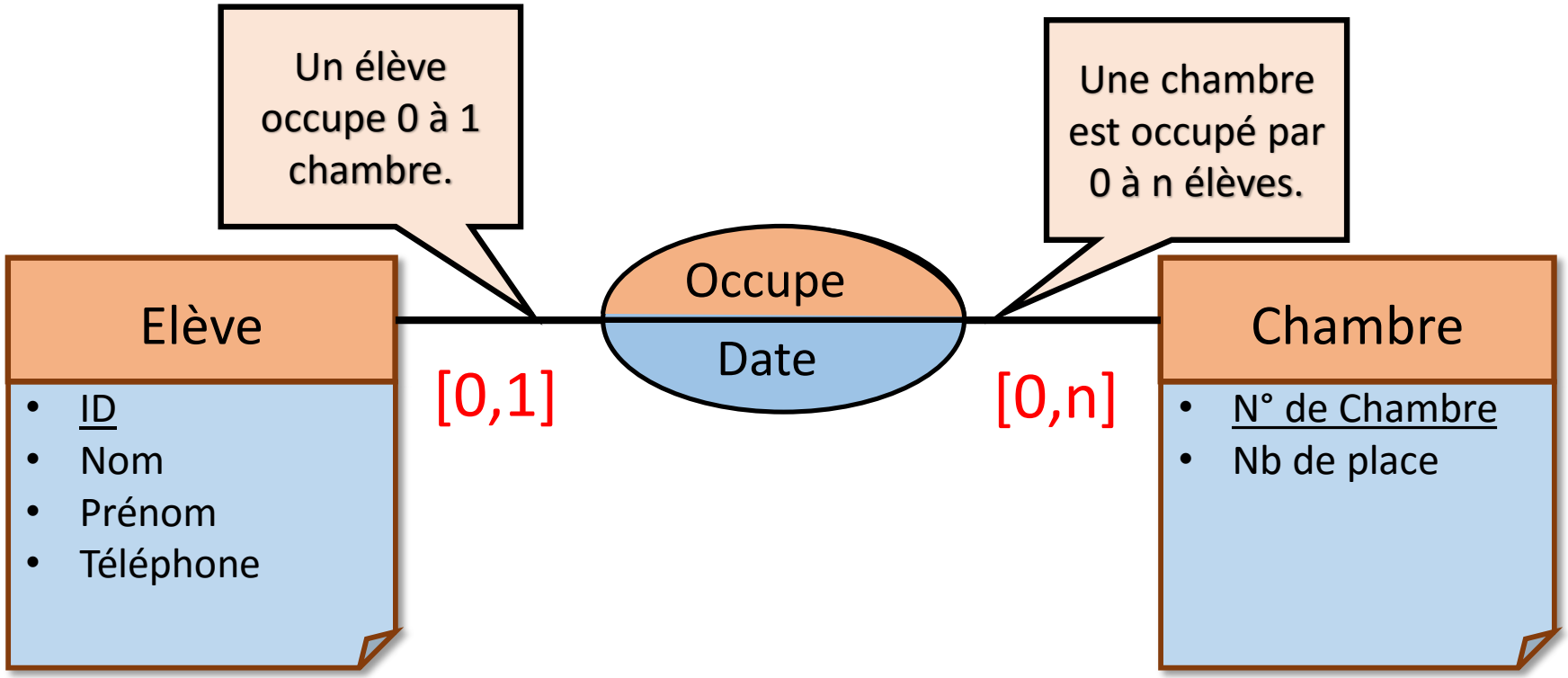
Cardinalité

Relations


Type de la relation ~ Entité
(obligatoire)
Occupe
Date
Elève
ID
Nom
Prénom
Téléphone
Chambre
N° de Chambre
Simple/Double
Description de la relation ~ Attribut
(optionnel)

Cardinalité

- Intervalle indiquant le nombre de relation possible entre une occurrence et les autres pour cette relation.
- De la forme [m,M] où :
  - m minimum atteignable : 0 ou 1
  - M maximum atteignable : 1 ou n
- Pris en prenant l'occurrence concernée en temps que sujet de la relation.



Modélisation Physique

Relations [. , 1] / [. , .]

Dépendances Fonctionnelles

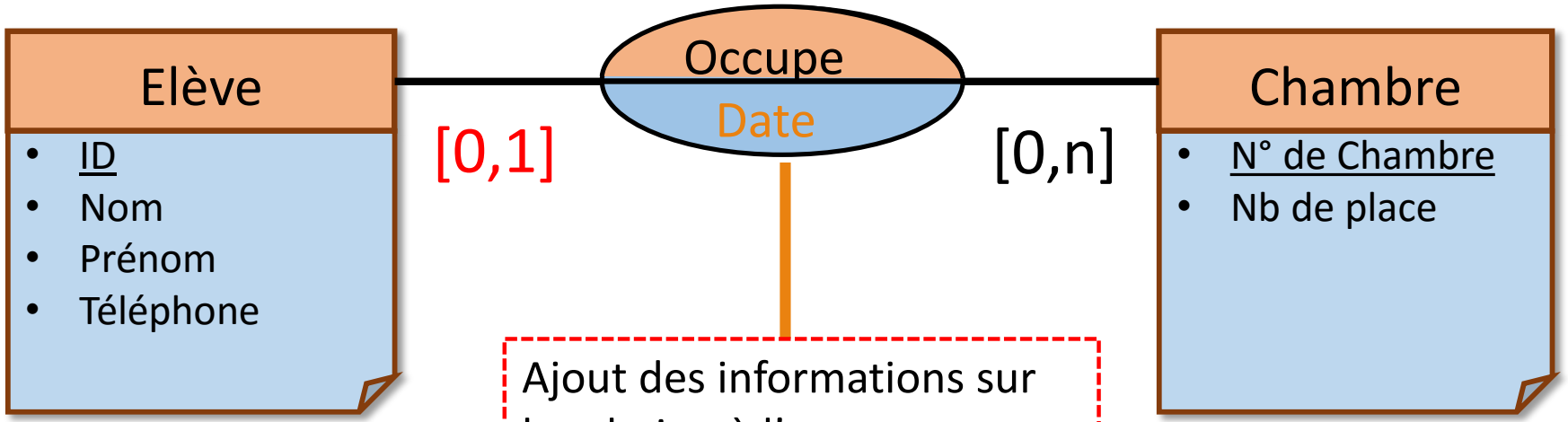
Relations [. , n] / [. , n]

Dépendances n/n

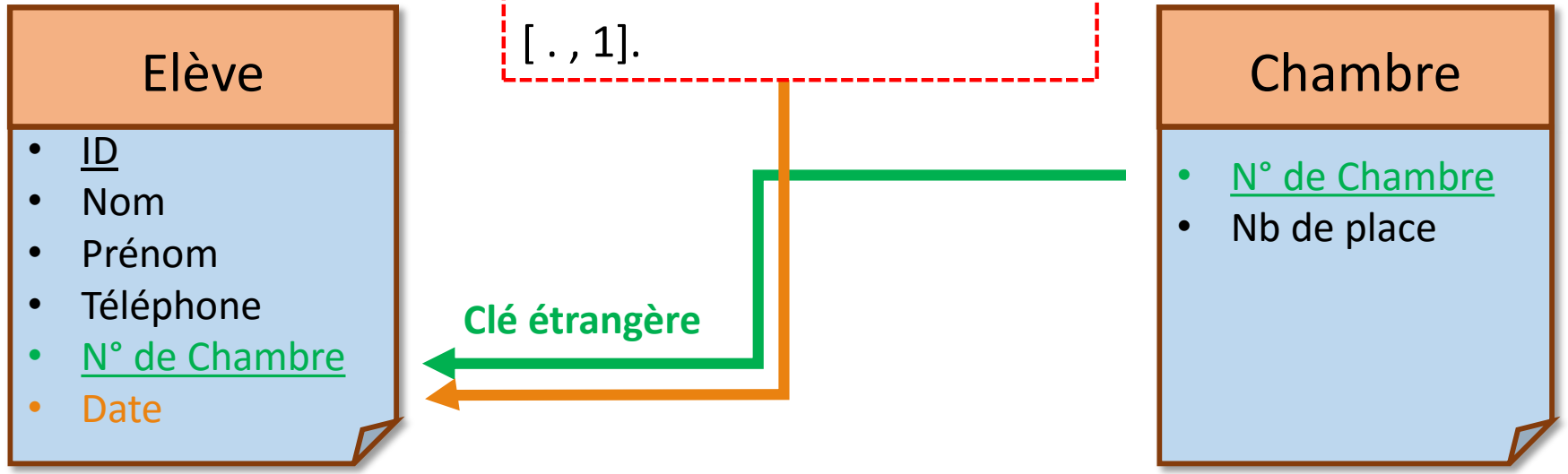
Relations

Pour une cardinalité [. , 1] / [. , .]

Modèle logique



Modèle physique

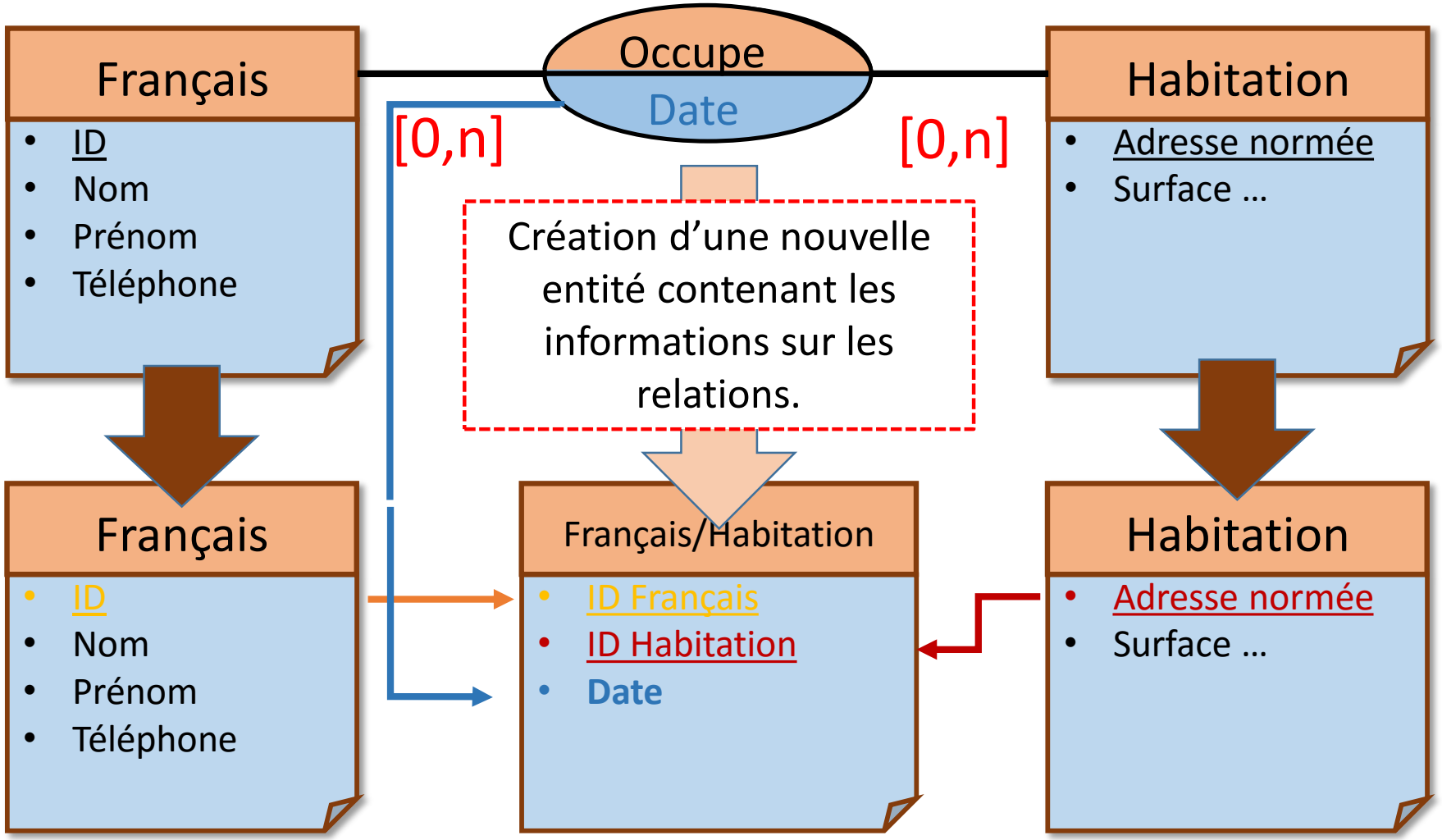


Relations

Pour une cardinalité [..,n] / [..,n]

Modèle logique

Modèle physique

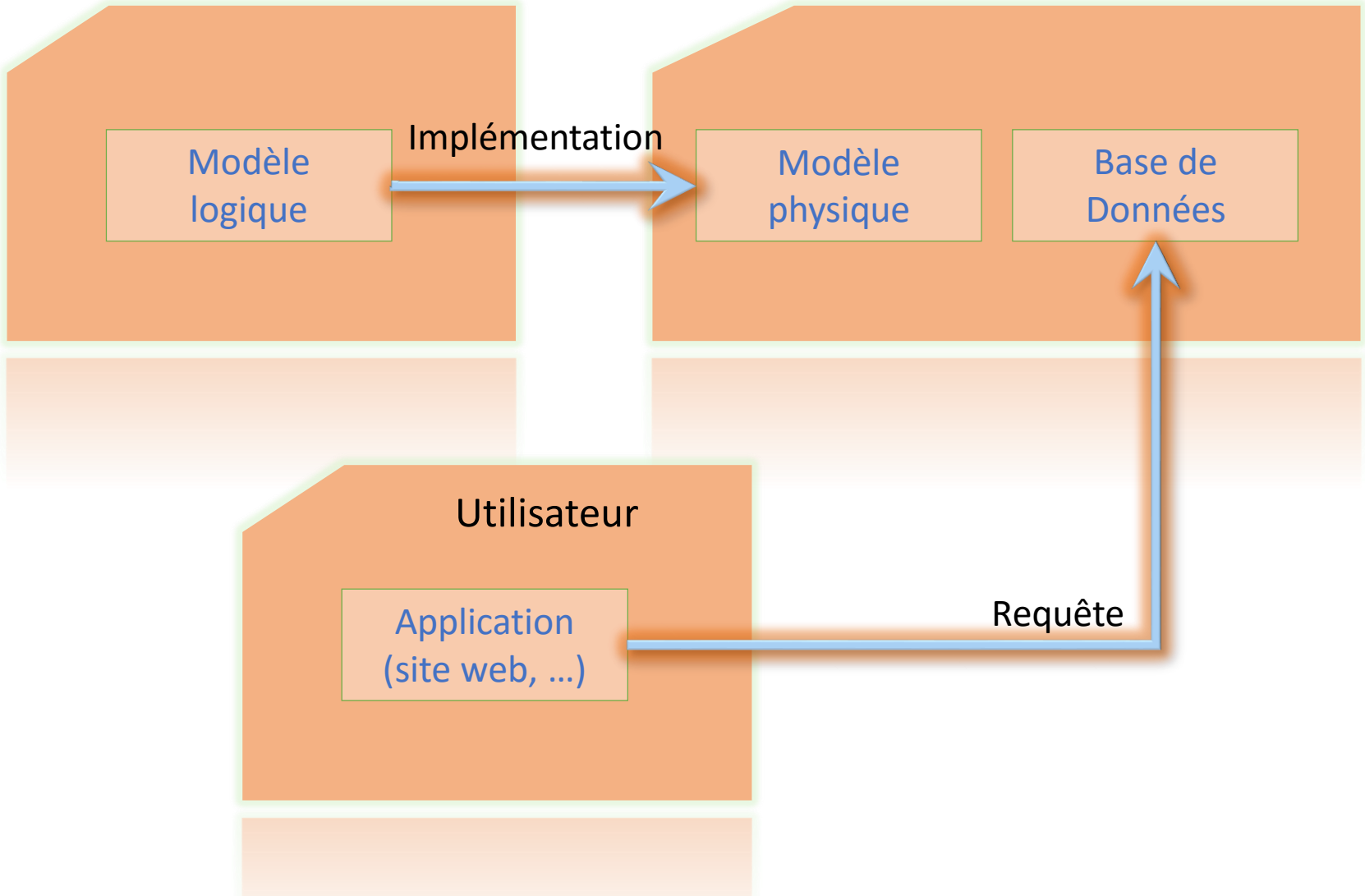


Les Requêtes SQL

Ordres

Syntaxes

Requêtes


Modèle logique
Implémentation
Modèle physique
Base de Données
Utilisateur
Application (site web, …)
Requête

SQL

Ordres

- CREATE : crée une entité
- DROP : Supprime une entité
- INSERT : ajoute des données (occurrence)
- DELETE : supprime des données
- UPDATE : modifie des données
- SELECT : demande des données

Sur les entités/tables

Sur les occurrences/ données

A retenir

SQL

Syntaxe

```
SELECT      attribut_1, … , attribut_n  OU fonct° d'agrégation(attribut_i)
FROM        table_1, … , table_m
WHERE       attribut_1 =  ' valeur_1 '
            AND attribut_2 >= ' valeur_2'
            AND attribut_3 = attribut_2018
INNER JOIN  latable_faible
            ON attribut_42 = attribut_69
ORDER BY    attribut_2 DESC, attribut_3
```

Conditions

Jointure de 2 tables

Classement des Valeurs par attribut (le premier prioritaire)

Optionnel

SQL

Fonctions d'agrégation

- COUNT : Compte le nombre d'entité ayant une même valeur d'attribut
- MAX : pour un attribut à valeur numérique, donne la plus grande valeur
- MIN : idem , la plus petite
- AVG : idem, donne la valeur moyenne

Exemple:

SELECT COUNT(Name) FROM Elèves WHERE Prénom="Thomas"

SELECT AVG(Note),MIN(Note),Max(Note) FROM P2018 WHERE Matière="Systèmes d'informations"

Exemples de requêtes

Résident

ID	Nom	Prenom	Batiment	Etage	Chambre
210	Baillard	Valentin	F	2	22
123	Grasset	Valentin	F	2	24
213	Petibon	Valentin	F	2	24
812	Trolon	Valentine	F	2	20
952	Valentin	Antoine	F	2	21
...					

- SELECT * FROM Résident WHERE Batiment=''F''
- SELECT DISTINCT Prénom FROM Résident AS Rés WHERE Bâtiment=''F'' AND Rés.Etage=''2''

Remarques Importantes :

- « SELECT * » : Sélectionne tout les attributs de la table
- AS : Crée un alias sur un nom de table ou d'attribut quand celui-ci est trop long
- DISTINCT : Sélectionne les occurrence dont les attributs sont différents (évite les doublons)
- « Rés.Etage » : Si deux attributs ont le même noms dans deux tables différentes, on précise l'appartenance à la table sous cette forme : Table.Attribut
- Certains additifs de requête peuvent être ajoutés , comme BETWEEN ou NOT

Requêtes sur plusieurs tables

Etudiant
- ID
- Nom
- Prénom
- Téléphone

Elève-Asso
- ID_Etudiant
- ID_Asso
- Fonction

Asso
- ID
- Nom
- Classe

Objectif : Trouver le nom, le prénom et le numéro de téléphone du président de l'association « Centrale Chapeaux »

```
SELECT Nom, Prénom, Téléphone FROM Etudiant
INNER JOIN Elève-Asso AS EA ON Etudiant.ID = EE.ID_Etudiant
INNER JOIN Asso ON Asso.ID = EA.ID_Asso
WHERE Asso.Nom = ''Centrale Chapeaux''
AND Fonction = Président
```

```
SELECT Nom, Prénom, Téléphone FROM Etudiant, Elève-Asso AS EA, Asso
WHERE Etudiant.ID = EA.ID_Etudiant
AND Asso.ID = EA.ID_Asso
AND Asso.Nom = ''Centrale Chapeaux''
AND Fonction = Président
```

Les deux syntaxes sont équivalentes
Au CF ayez une préférence pour le INNER JOIN qui sera valorisé

Exemple de requête beaucoup plus complexe (HP)

```
myecp-# (SELECT date_of_birth FROM (SELECT DISTINCT login, date_of_birth FR
OM member m INNER JOIN membership ms ON ms.member_id = m.id AND ms.gang_id
= 324 AND ms.year = 2015) as result GROUP BY date_of_birth HAVING count(*)
> 1)
myecp-# ORDER BY date_of_birth;
     login     | first_name |         last_name          | date_of_birth
---------------+------------+----------------------------+---------------
 2015mekkis    | Seddik     | Mekki                      | 1994-10-21
 2015morinierg | Gabriel    | Le Lièvre De La Moriniere  | 1994-10-21
 2015bnioukila | Anass      | Bnioukil                   | 1995-06-15
 2015noelv     | Valentin   | Noel                       | 1995-06-15
 2014altanal   | Ludovico   | Altana                     | 1995-08-07
 2015sakhio    | Otmane     | Sakhi                      | 1995-08-07
 2015lahloua2  | Ahmed      | Lahlou Mimi                | 1995-10-01
 2015meynardr  | Romain     | Meynard                    | 1995-10-01
(8 lignes)
```

A RETENIR

- Créer un modèle logique à partir de données
- Savoir passer de modèle logique à modèle physique
- Savoir faire des requêtes en SQL pour recueillir des informations d'une BDD existante

CENTRALE

RESEAUX

Architecture de l'ordinateur

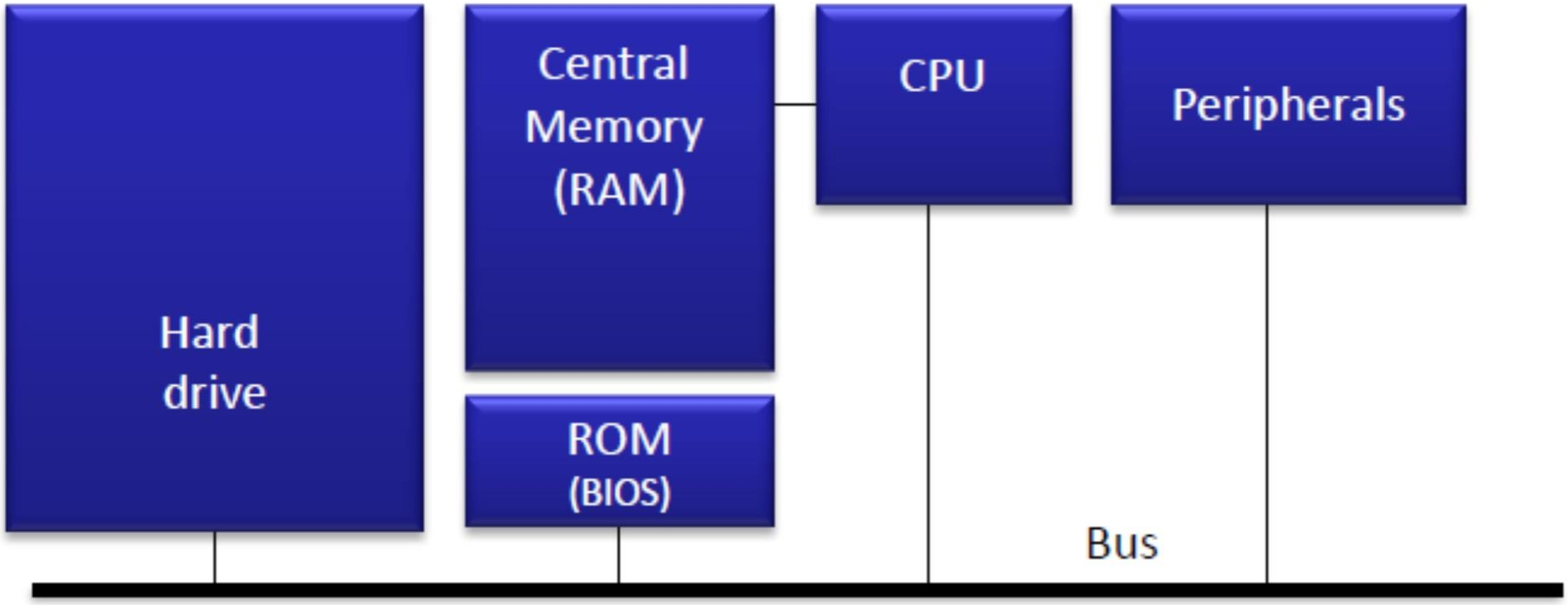
Par Julien '*Djou*' Raspaud

Avec honteux plagiat de '*Kapi*'

Plan

- **Composition et fonctionnement**
  - Composants
  - Système d'exploitation (OS)
  - Fonctionnement d'un programme
- **Le CPU**
  - CPU
  - Pagination
- **Gestion de la mémoire**
  - Disque dur
  - Fragmentation

Structure d'un ordinateur



CPU = Central Processing Unit = Processor
ROM = Read Only Memory
RAM = Random Access Memory

Les composants sont tous branchés sur la carte mère ou motherboard.

Différentes Mémoires

- Registres du processeur : ultra rapide.
- Cache du processeur : très rapide.
- Mémoire vive (RAM) : rapide.
- Mémoire morte non volatile : lente.

RAM

HDD (Disque Dur)

Flash

! La « mémoire » d'un pc ne désigne pas uniquement son disque dur !

Mise au point !

Bits ≠ Bytes

1 Byte = 1 Octet = 8 Bits

Le Processeur (CPU)

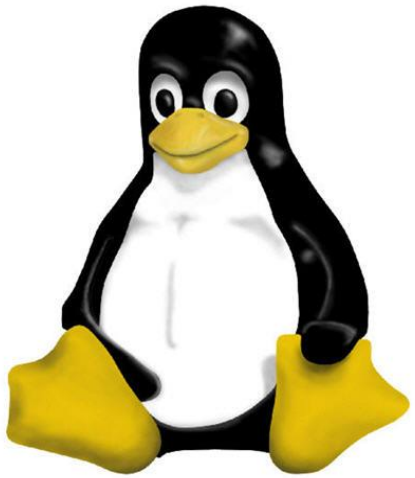
- CPU : Central Processing Unit

- Récupère mes instructions et les données dans la mémoire puis exécute les instructions demandées.

L’OS (Système d’exploitation)

L’OS est composé :

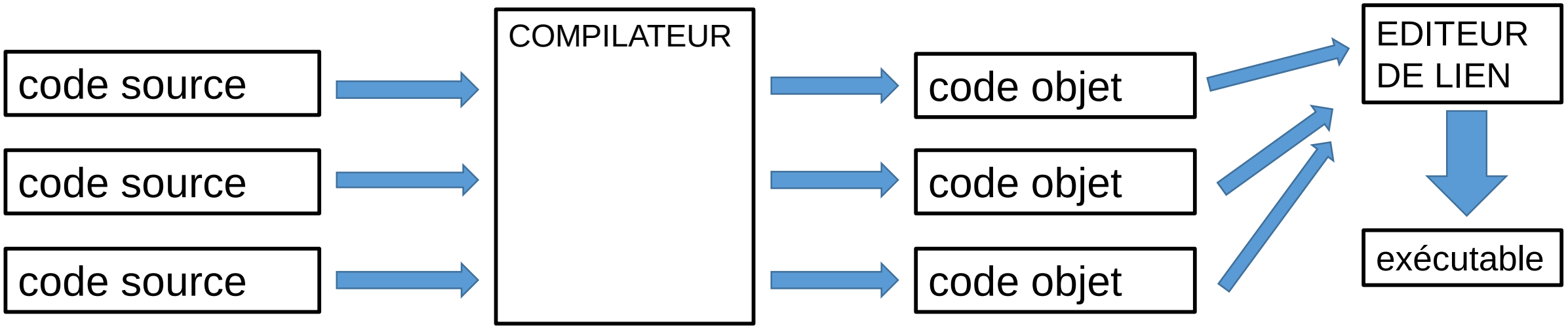
- De programmes (ex : explorateur de fichiers)
- D’une interface entre le système et l’utilisateur.
- D’un noyau qui relie les programmes et le matériel.



Compiler pour executer

L'ordinateur ne parle pas notre langue, il faut compiler !

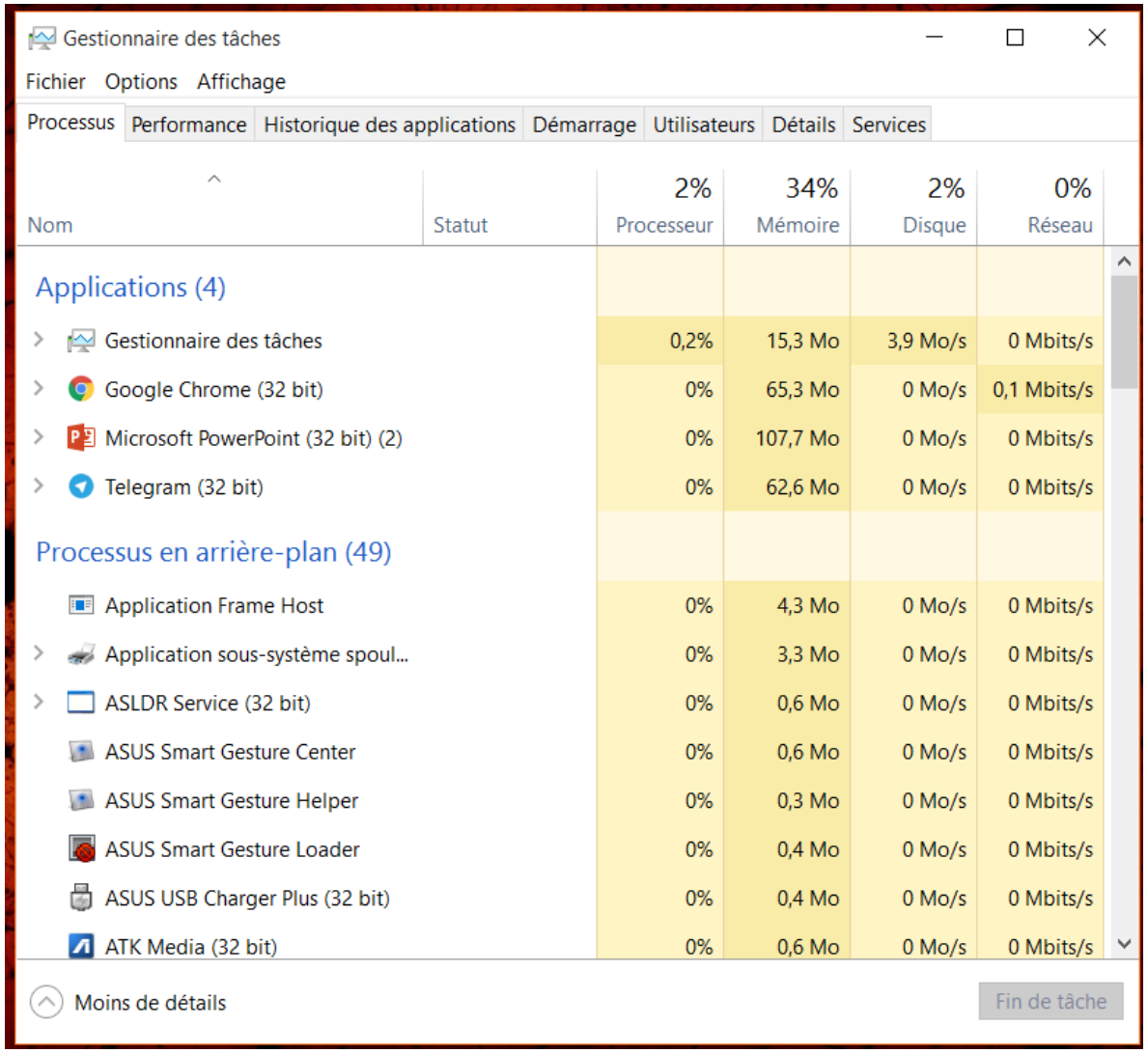
- On part d'un code source écrit dans un langage informatique : Java, Python, C++, …
- On le compile, puis on créé des liens pour appeler des bibliothèques extérieures nécessaires à son fonctionnement.



Le multi-tâche

La plupart des OS actuels sont multi-tâche :
Ils effectuent plusieurs processus simultanément…

Ou pas !



Le multi-tâche

Un CPU, à moins qu'il n'ait plusieurs cœurs, ne peut exécuter qu'une instruction à la fois, et on ne peut lire ou écrire qu'une information à la fois sur le disque dur.

Voyons-donc comment l'OS gère cela …

Préemption et Commutation de contexte

Préemption :

- Un processus est en cours de traitement
- Un autre processus veut prendre la main. Il est prioritaire.
- Mais le processus précédent n'est pas terminé. Le processeur doit donc l'arrêter.

Commutation de contexte :

- Le processus arrêté est stocké en vue d'un traitement ultérieur.
- Le nouveau processus est lancé.

Le multi-tâche fait des tours

Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !

Le multi-tâche fait des tours

Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !

Le multi-tâche fait des tours

Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !

Le multi-tâche fait des tours

Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !

Le multi-tâche fait des tours

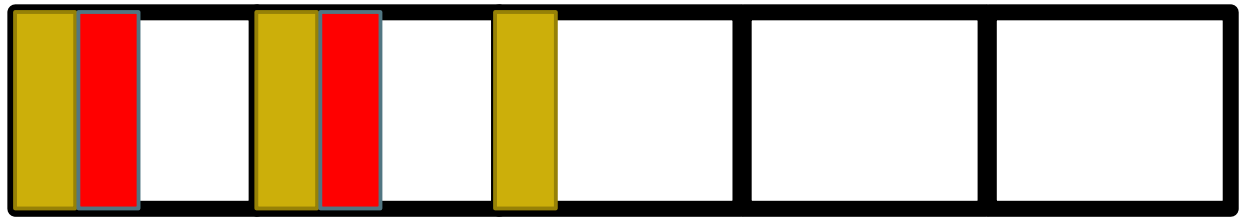
Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !



Le multi-tâche fait des tours

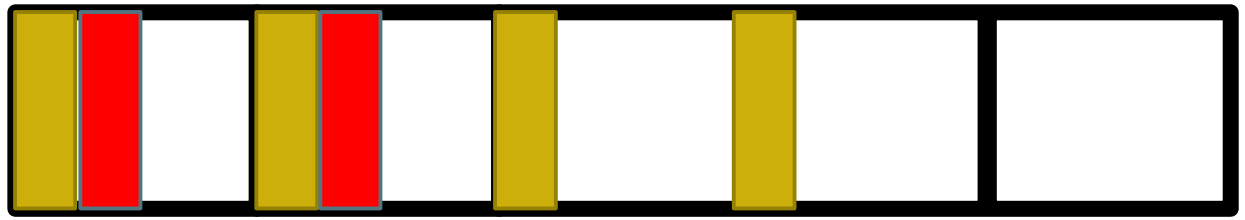
Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !



Le multi-tâche fait des tours

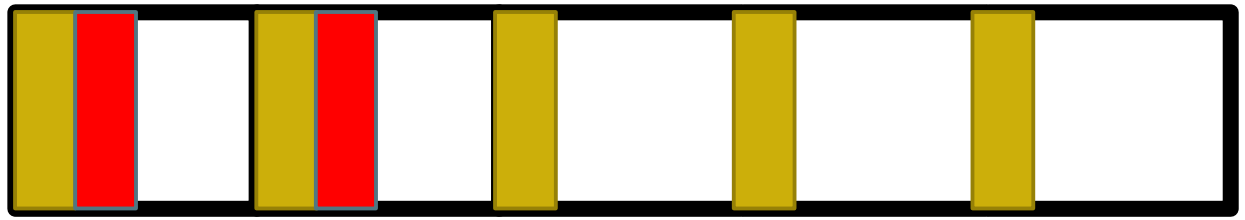
Comment nous trompe le multi-tâche ?

>> Il fait de la commutation de contexte en permanence !

Il traite un petit bout d'un processus, puis d'un autre, puis d'un autre…

Puis revient au premier processus, et recommence…

Et tous les processus semblent avancer simultanément !



Choisir l'ordre de traitement des processus

Le CPU est une ressource critique

→Possible engorgement à l'entrée.

→Certains processus sont plus gourmands que d'autres

Il faut donc choisir quel processus traiter en premier.

Différents Alogirthmes

Quelques Exemples courants :

- FIFO (First in First out)
- SPN (Shortest Process Next)
- Round Robin

Différents Alogirthmes

Quelques Exemples courants :

- FIFO (First in First out)
- SPN (Shortest Process Next)
- Round Robin

Un algorithme est bon s'il n'y a pas **famine**, si le **temps moyen de traitement** d'un processus n'est pas trop long, et si le CPU ne passe pas trop de temps à **ne rien faire**.

Différents Alogirthmes

Si on vous demande de commenter la performance d'un algorithme :

Pas d'algorithme miracle, il faut faire des compromis entre **3 exigences incompatibles !**

Et tout dépend de ce que l'on veut privilégier !

Ce qui semble être une amélioration dans un cas peut être un défaut dans un autre cas.

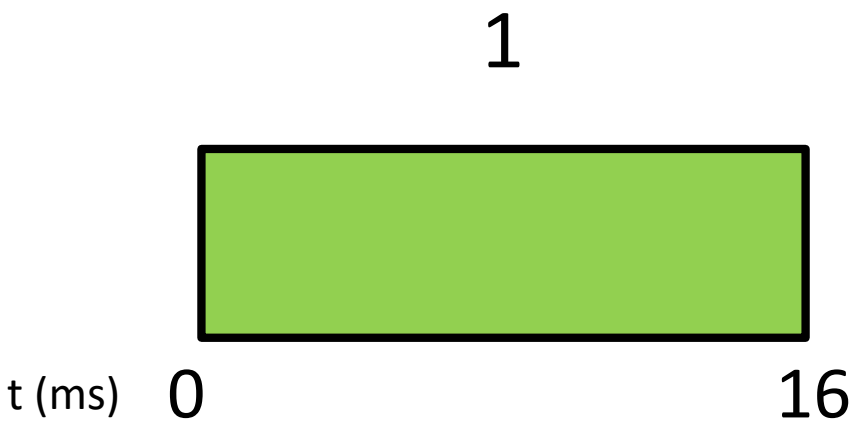
Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

- Traitement dans l'ordre d'arrivée
- Pas de préemption

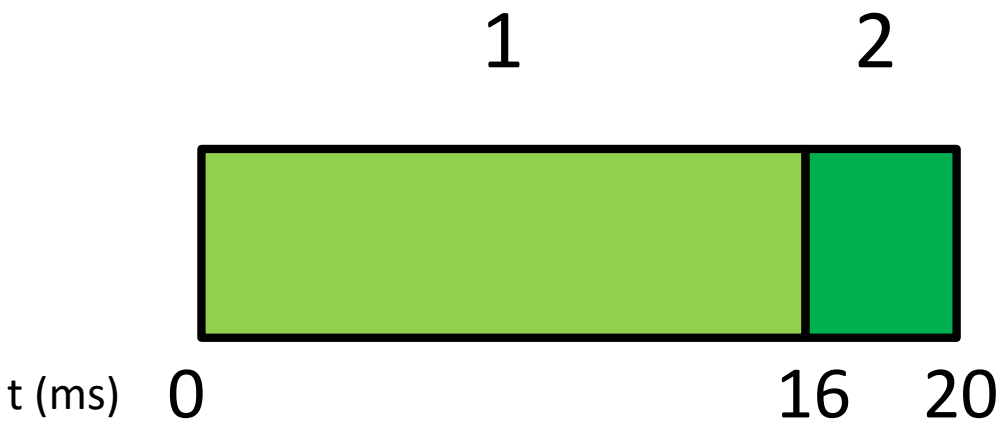
Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



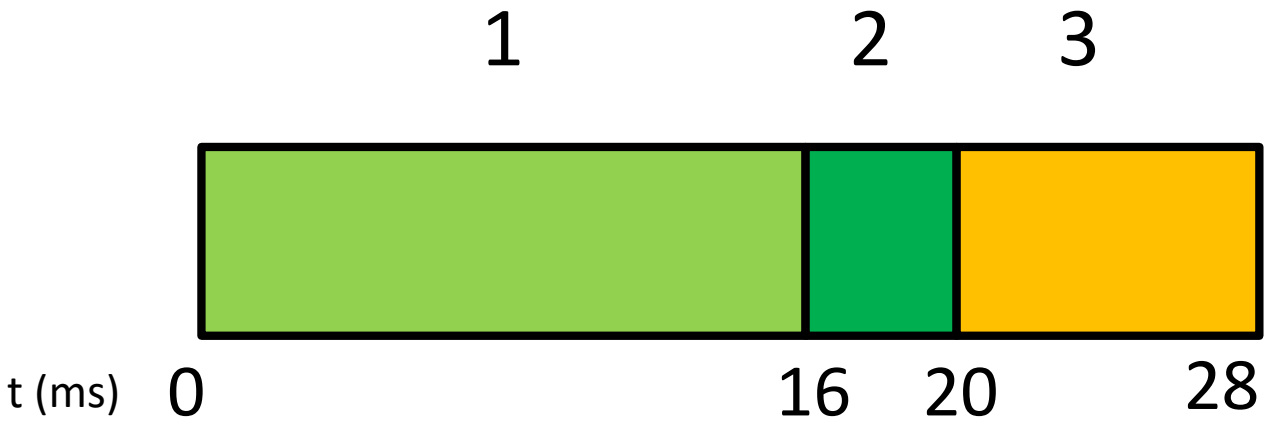
Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



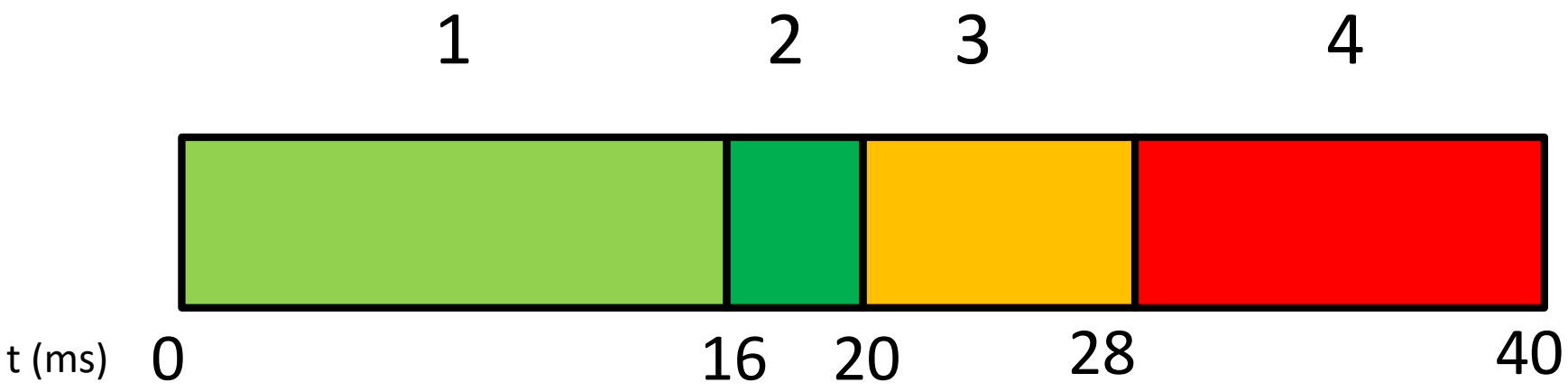
Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



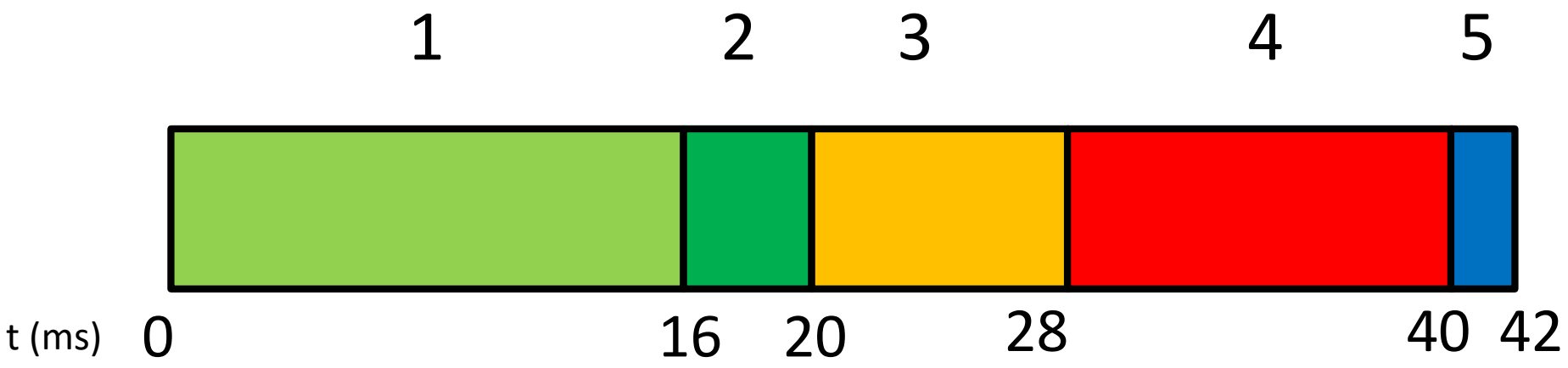
Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



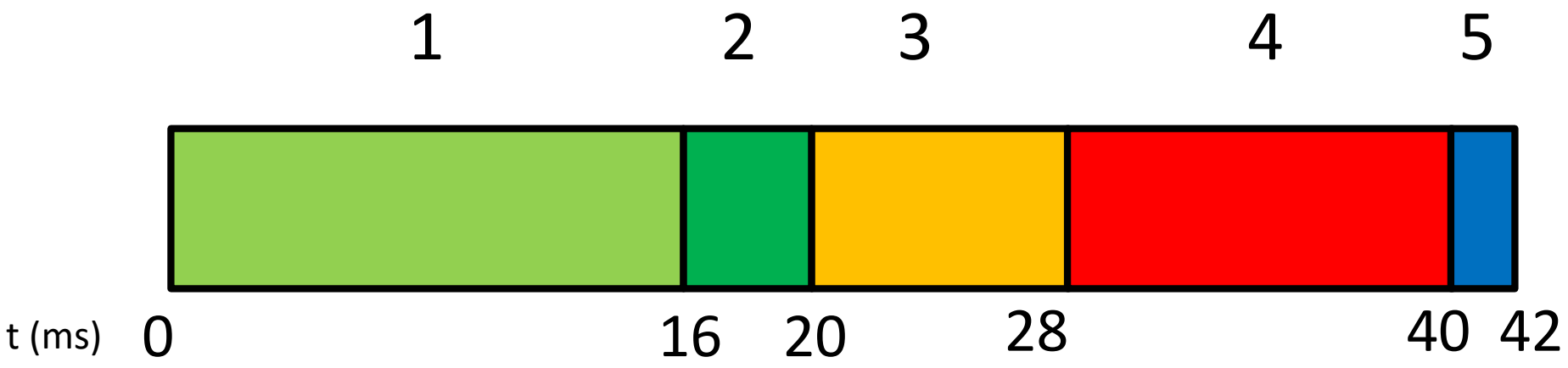
Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



Exemple : FIFO

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



Temps moyen de traitement : **(16 + 20 + 28 + 40 +42) / 5 = 29,2 ms**

Pas de famine, mais pas optimisé du tout !

Autre exemple : SPN

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

1 2 3 4 5

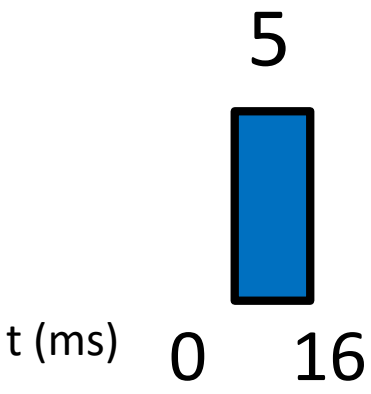
Pas de préemption

Le processus le plus court passe en premier

Autre exemple : SPN

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

1 2 3 4 5



Autre exemple : SPN

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

1 2 3 4 5

5 2

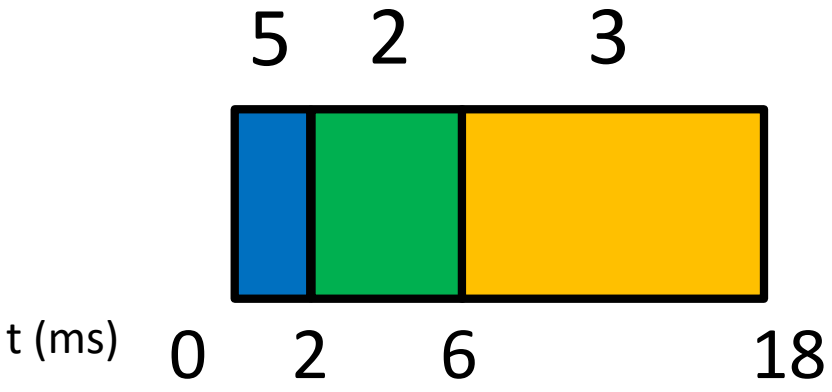
t (ms)

0 2 6

Autre exemple : SPN

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

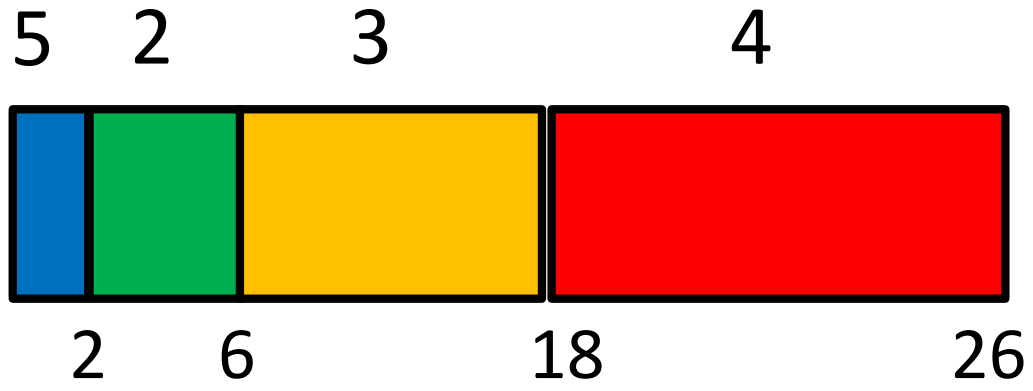
1 2 3 4 5



Autre exemple : SPN

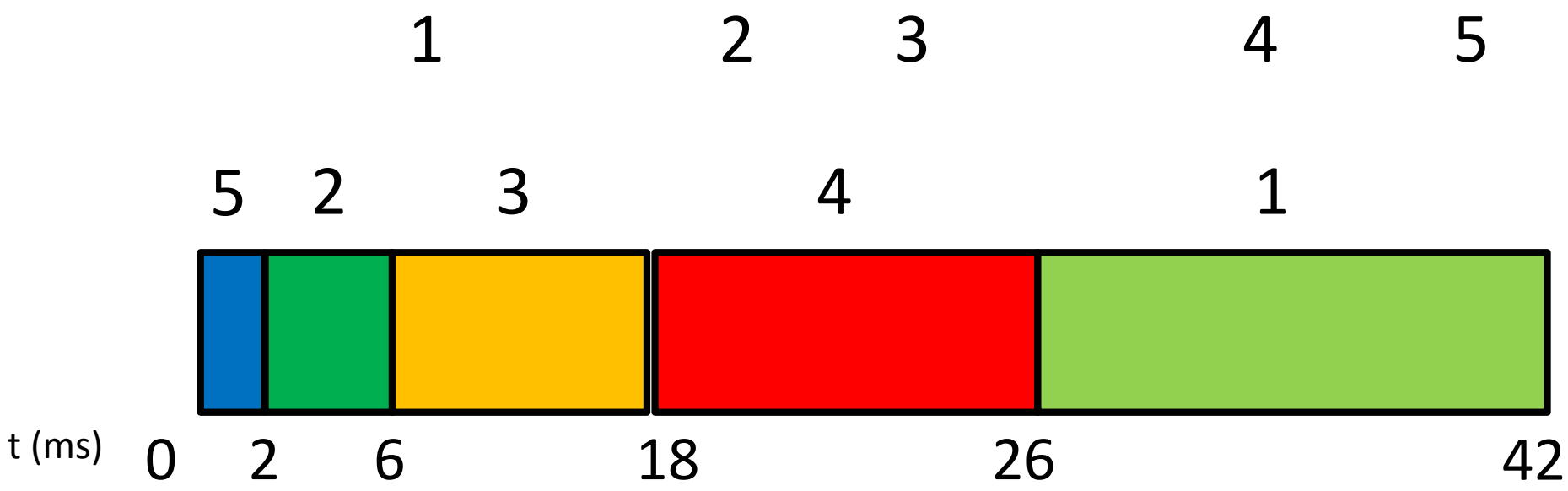
Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

1 2 3 4 5



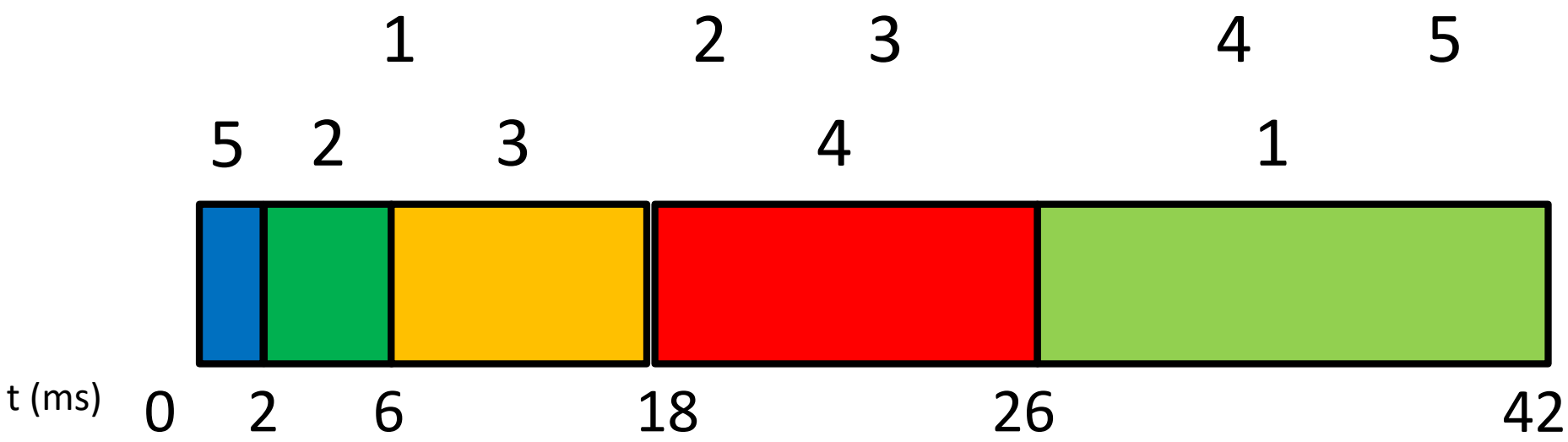
Autre exemple : SPN

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



Autre exemple : SPN

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



Temps moyen de traitement : (2 + 6 + 18 + 26 +42) / 5 = 18 ms

Temps optimisé, mais risque de famine…

Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

1 2 3 4 5

Un certain quantum de temps est alloué aux processus pour s'effectuer

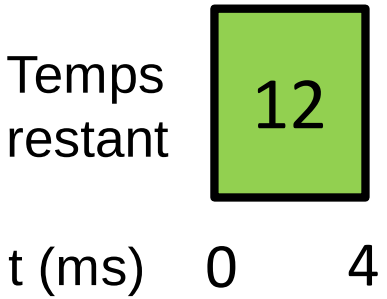
S'il ne s'est pas terminé avant la fin du temps imparti : changement de contexte

L'algorithme est préemptif

Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

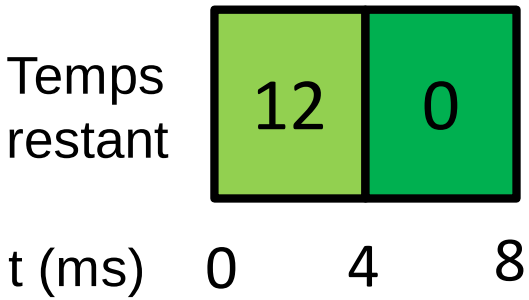
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

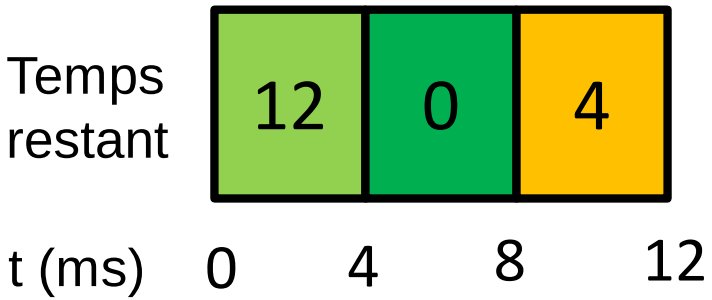
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

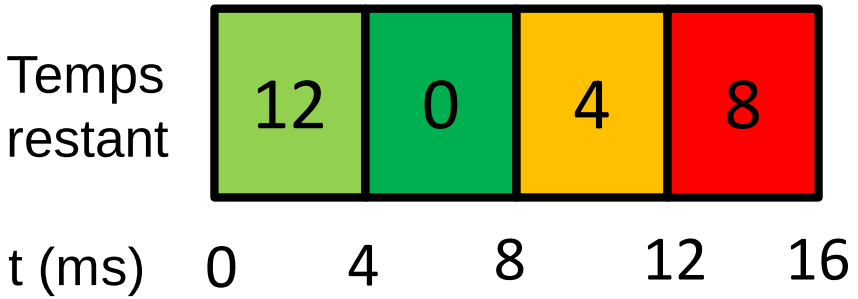
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

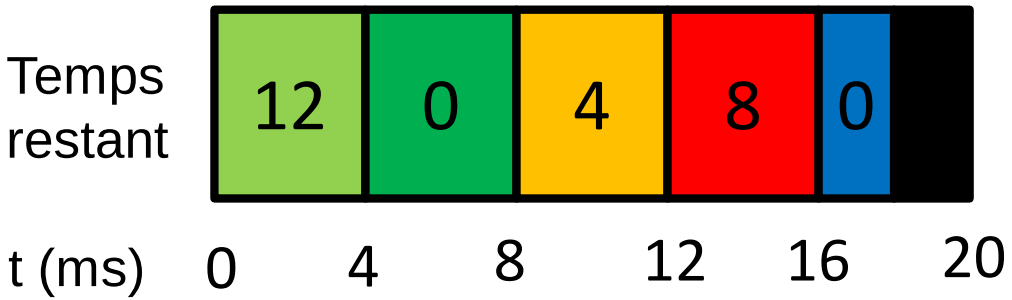
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

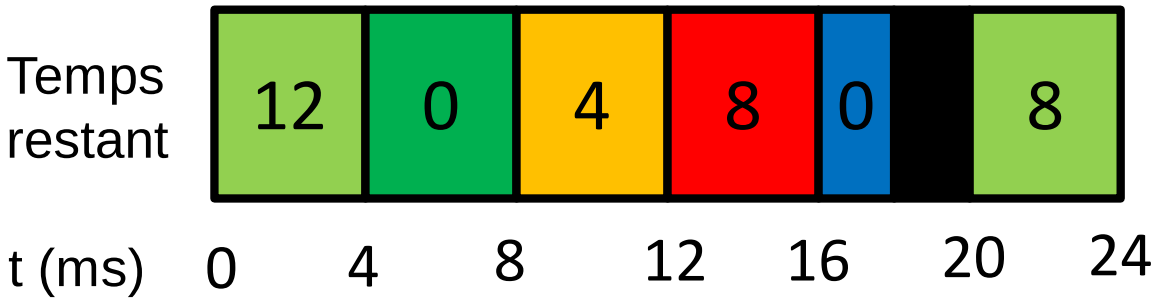
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

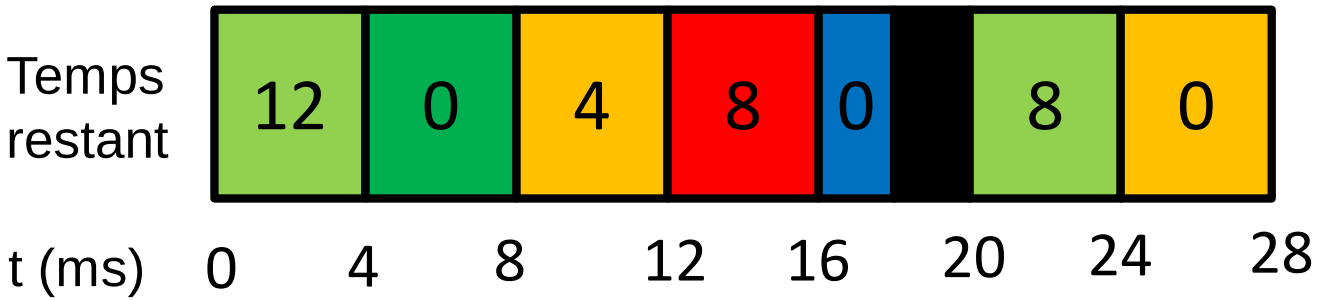
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

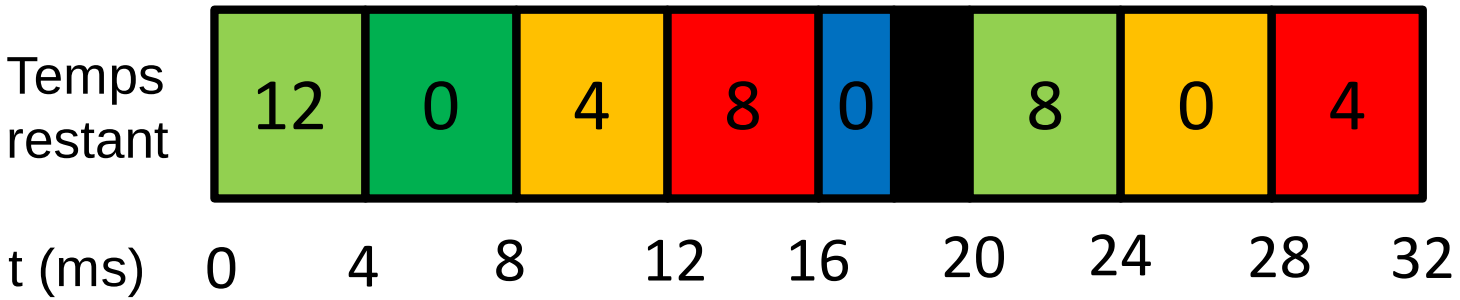
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

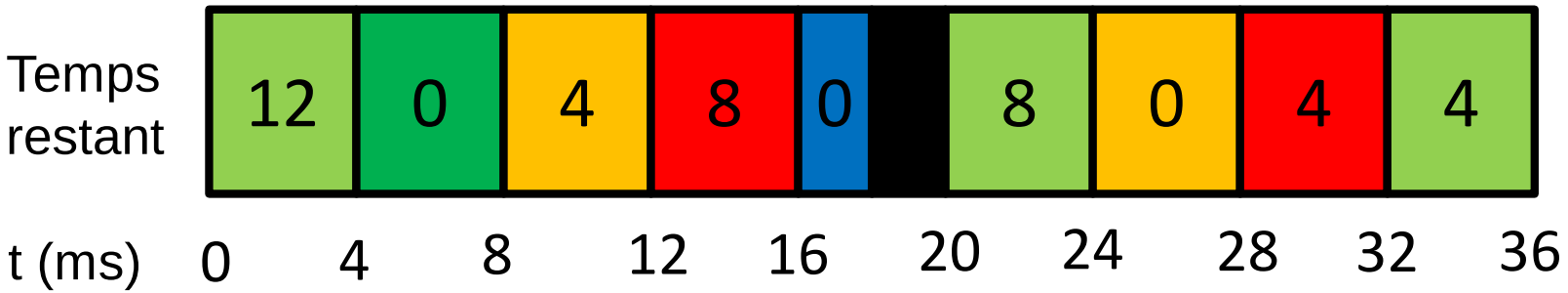
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

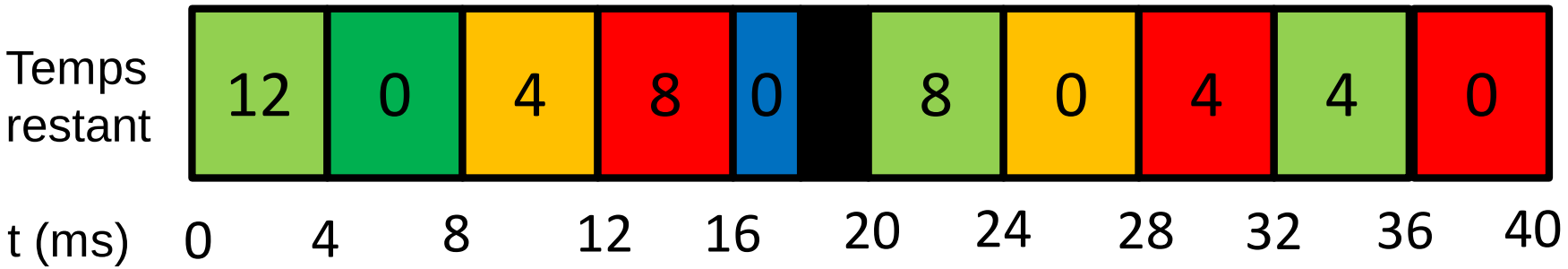
1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

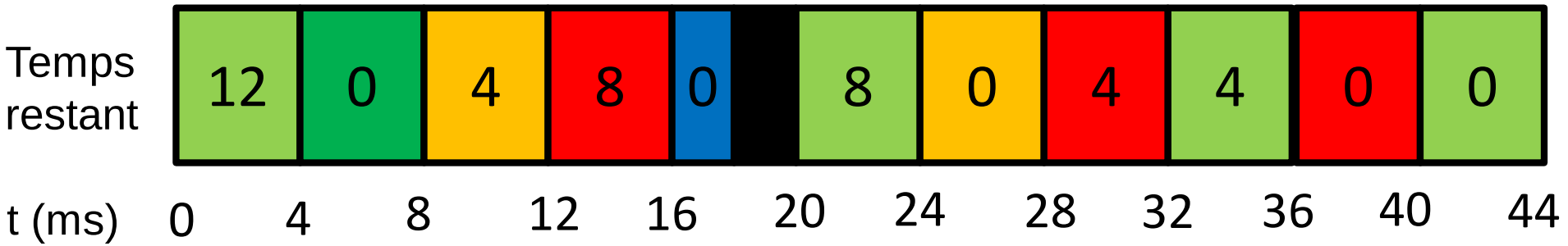
1 2 3 4 5



Autre exemple : Round Robin

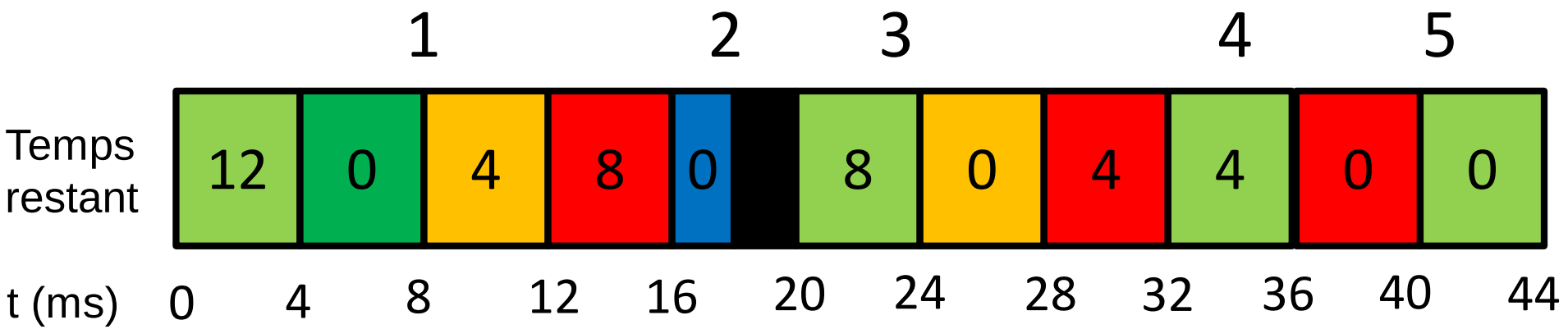
Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2

1 2 3 4 5



Autre exemple : Round Robin

Processus (dans l'ordre d'arrivée)	1	2	3	4	5
Temps de calcul demandé (ms)	16	4	8	12	2



Temps moyen de traitement : (8 + 18 + 28 + 40 + 44) / 5 = 27,6 ms

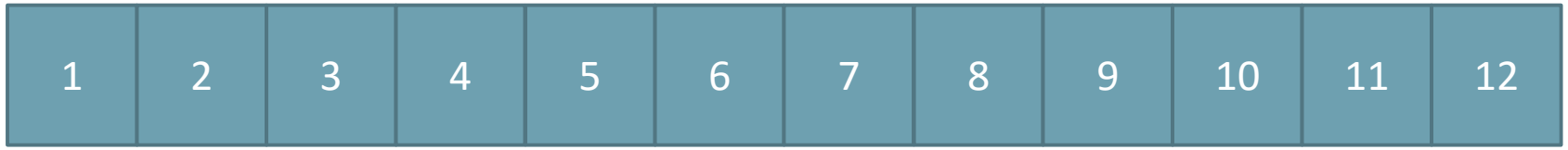
Evite la famine, mais les changements de contextes induisent un temps supplémentaire, et les quanta peuvent être inadéquats.

La mémoire aussi est illimitée

Tous les processus en attente ne tiennent pas dans la mémoire vive (parfois trop petite).

Mémoire virtuelle : on fait croire à l'ordinateur qu'une partie de son disque dur est de la mémoire. **(Le SWAP)**

Le défaut de page



Processus : il me faut la page 3
RAM : ah… attends je l'ai pas je vais la chercher
Processus : j'attends…….

Les données sont regroupées par pages.

La page dont le processus a besoin doit se trouver dans la RAM.

Si elle n'y est pas, il y a défaut de page, il faut charger la page dans la RAM.

Un défaut de page représente un temps d'inactivité pour le processus, le temps d'accéder au disque.

La mémoire aussi est limitée

MAIS : Le disque dur est **LENT** d'accès !

Conséquence :
Ralentissement considérable !
Il faut optimiser (moins d'accès au disque dur possible en minimisant le nombre de défauts de page)

Exemples d’algorithmes

FIFO (First In, First Out) :
On remplace la page qui a été chargée il y a le plus longtemps.

LRU (Least Recently Used) :
On remplace la page qui a été utilisée il y a le plus longtemps.

LFU (Least Frequently Used) :
On remplace la page la moins souvent utilisée.

Un exemple avec FIFO

Trois cases dans la mémoire centrale

On demande les pages 1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page
1	1	X

1

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page
1	1	X
2	12	X

1

1, 2

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		1, 2, 4

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		1, 2, 4
2	4 1 2		1, 2, 4

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		1, 2, 4
2	4 1 2		1, 2, 4
5	4 5 2	X	2, 4, 5

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		1, 2, 4
2	4 1 2		1, 2, 4
5	4 5 2	X	2, 4, 5
3	4 5 3	X	4, 5,3

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		1, 2, 4
2	4 1 2		1, 2, 4
5	4 5 2	X	2, 4, 5
3	4 5 3	X	4, 5, 3
4	4 5 3		4, 5, 3

Un exemple avec FIFO

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

9 défauts de page.

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		1, 2, 4
2	4 1 2		1, 2, 4
5	4 5 2	X	2, 4, 5
3	4 5 3	X	4, 5, 3
4	4 5 3		4, 5, 3

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page
1	1	X

1

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page
1	1	X
2	1 2	X

1

1, 2

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page
1	1	X
2	1 2	X
5	1 2 5	X

1

1, 2

1, 2, 5

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demanedé	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		2, 4, 1

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		2, 4, 1
2	4 1 2		4, 1, 2

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		2, 4, 1
2	4 1 2		4, 1, 2
5	5 1 2	X	1, 2, 5

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		2, 4, 1
2	4 1 2		4, 1, 2
5	5 1 2	X	1, 2, 5
3	5 3 2	X	2, 5, 3

Un exemple avec LRU

1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		2, 4, 1
2	4 1 2		4, 1, 2
5	5 1 2	X	1, 2, 5
3	5 3 2	X	2, 5, 3
4	5 3 4	X	5, 3, 4

Un exemple avec LRU

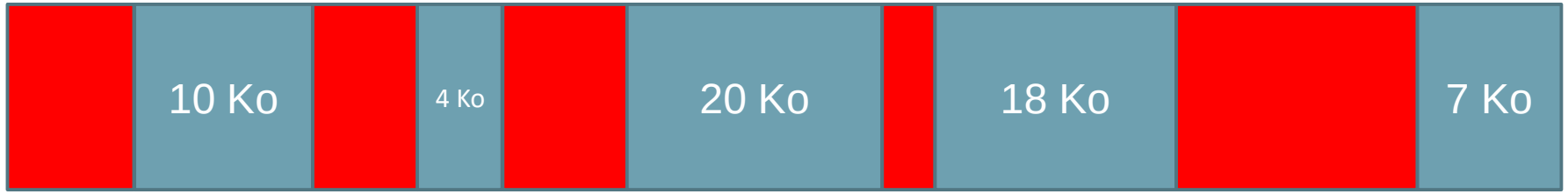
1, 2, 5, 3, 1, 2, 4, 1, 2, 5, 3, 4

10 défauts de page

page demandée	État de la mémoire	défaut de page	
1	1	X	1
2	1 2	X	1, 2
5	1 2 5	X	1, 2, 5
3	3 2 5	X	2, 5, 3
1	3 1 5	X	5, 3, 1
2	3 1 2	X	3, 1, 2
4	4 1 2	X	1, 2, 4
1	4 1 2		2, 4, 1
2	4 1 2		4, 1, 2
5	5 1 2	X	1, 2, 5
3	5 3 2	X	2, 5, 3
4	5 3 4	X	5, 3, 4

Fragmentation

Au fur et à mesure des allocations et désallocations de mémoire, celle-ci devient fragmentée :



Comment choisir les emplacements où l'on va allouer la mémoire ?

Les Algrorithmes d'Allocation

First Fit

Next Fit

Best Fit

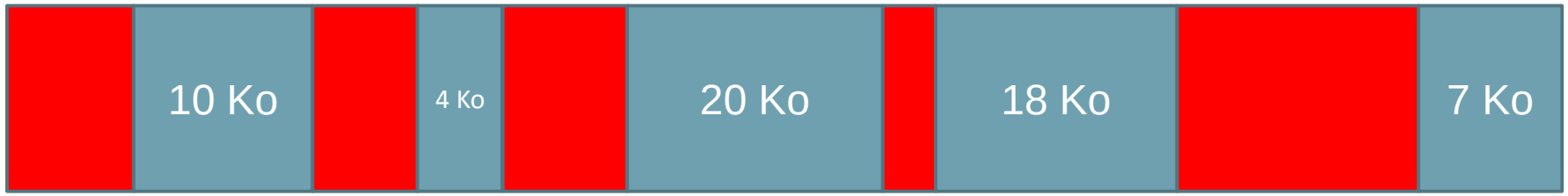
Worst Fit

Faire une série de « mauvais » choix peut mener à une meilleure solution

First Fit

La mémoire est allouée dans la première zone libre disponible

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko



First Fit

La mémoire est allouée dans la première zone libre disponible

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

First Fit

La mémoire est allouée dans la première zone libre disponible

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

First Fit

La mémoire est allouée dans la première zone libre disponible

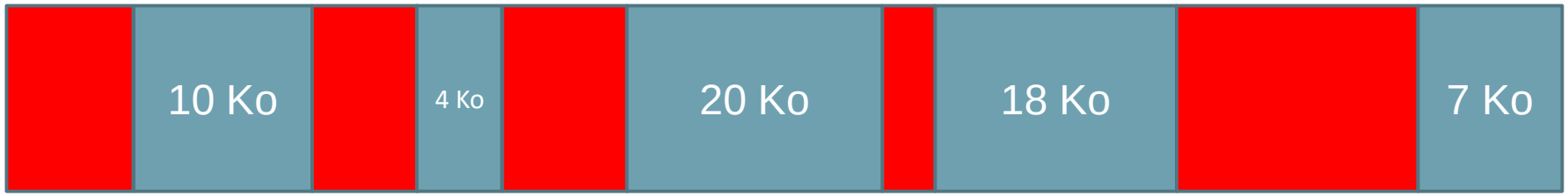
Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

First Fit n’est pas optimisé !

Next fit

Petite variante du First Fit, on choisit la première case à partir de la dernière allouée

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko



Next fit

Petite variante du First Fit, on choisit la première case à partir de la dernière allouée

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Next fit

Petite variante du First Fit, on choisit la première case à partir de la dernière allouée

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Next fit

Petite variante du First Fit, on choisit la première case à partir de la dernière allouée

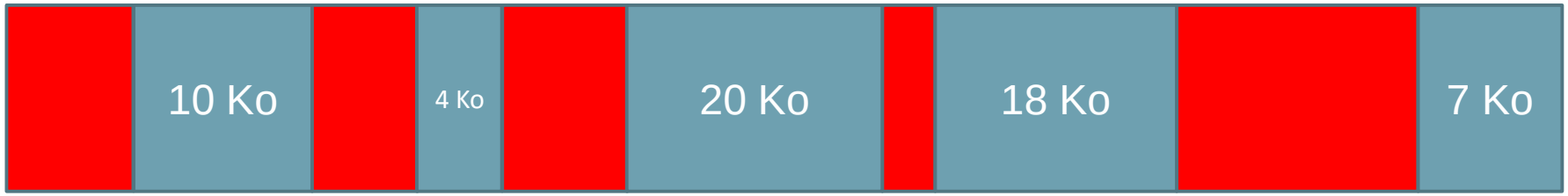
Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Pas plus optimisé que First Fit…

Best Fit

On choisit l’emplacement qui minimise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko



Best Fit

On choisit l’emplacement qui minimise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Best Fit

On choisit l’emplacement qui minimise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Best Fit

On choisit l’emplacement qui minimise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Worst fit

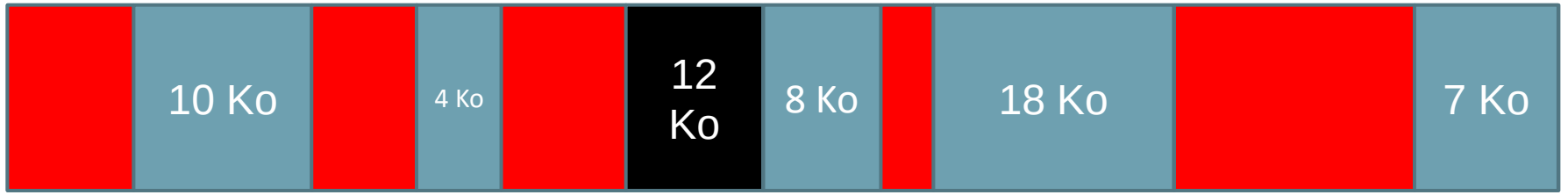
On choisit l’emplacement qui maximise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

Worst fit

On choisit l’emplacement qui maximise l’espace résiduel

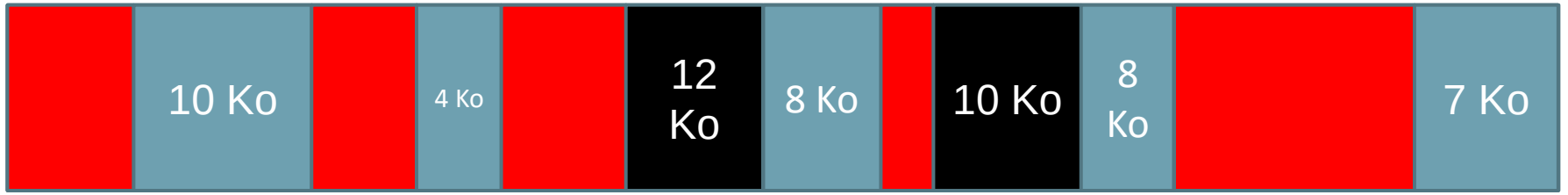
Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko



Worst fit

On choisit l’emplacement qui maximise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko



Worst fit

On choisit l’emplacement qui maximise l’espace résiduel

Ex : On doit allouer 12 Ko, 10 Ko et 9 Ko

CENTRALE

RESEAUX

La sécurité

Par Stanislas ‘Over’ Plessia

Avec toujours un magnifique plagiat de ‘*Kapi*’

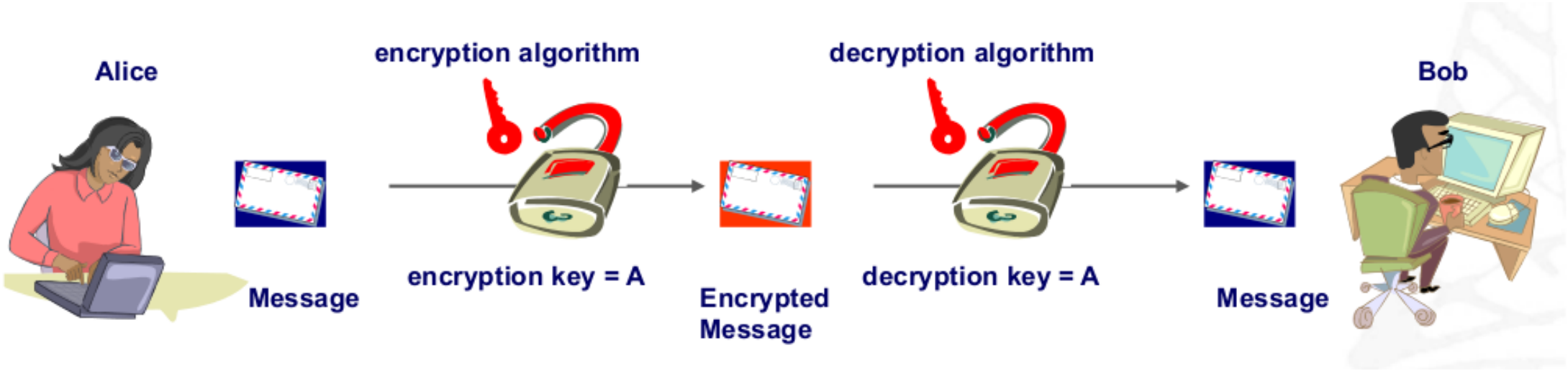
Sécurité

La plupart des exercices de Sécurité demandent de savoir réfléchir sans lien avec la SI. Voici l'essentiel des choses à savoir :

Codage :
- symétrique
- asymétrique

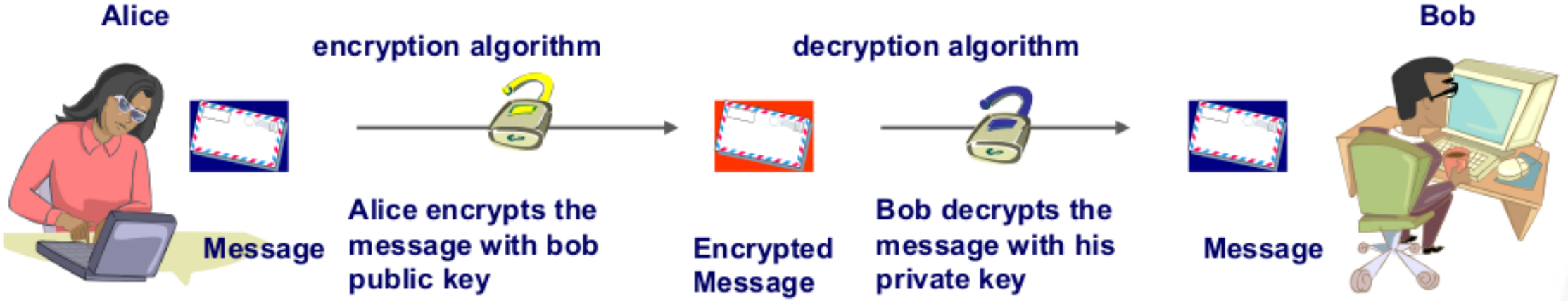
Les attaques par déni de service
- Smurf
- DDoS

Cryptage Symétrique



En cryptographie symétrique, la même clé sert à coder et décoder le message
Problème : comment envoyer la clef à son destinataire ?

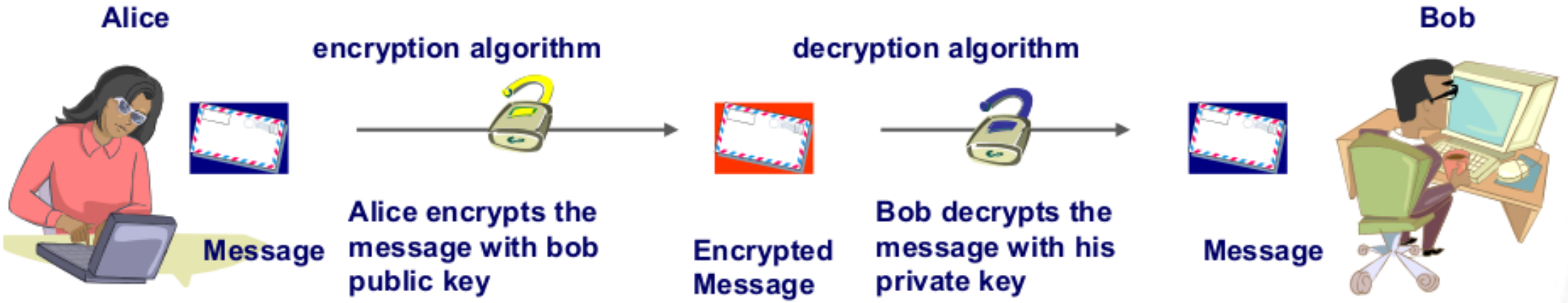
Cryptage asymétrique



Bob crée deux clés liées l'une à l'autre ; si l'une est publique, l'autre doit absolument rester privée
Il est théoriquement possible de revenir à la clé privée depuis la clé publique (mathématiques) mais pas dans un temps raisonnable

- La clé publique est envoyée à Alice (et à tout le monde) via un serveur de clés publiques

Cryptage asymétrique



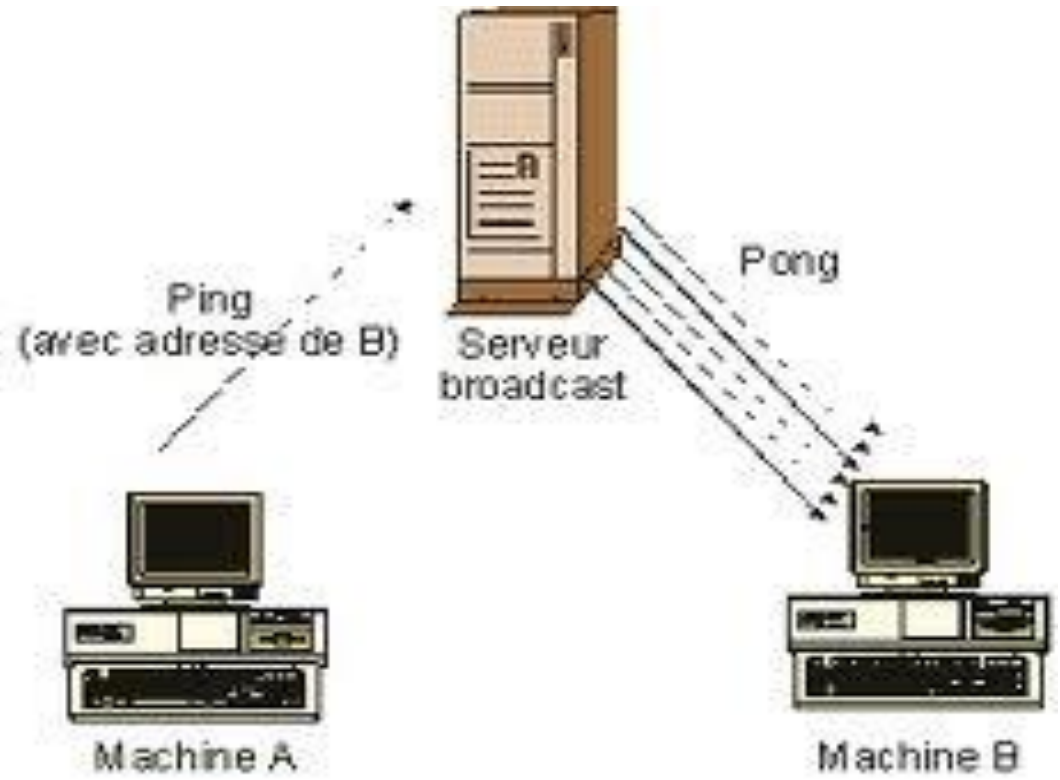
Alice ne connaît pas la clé privée de Bob : elle n'en a pas besoin pour lui envoyer son message
N'importe qui peut envoyer un message crypté à Bob, mais lui seul sera capable de le décrypter avec sa clé privée
Si quelqu'un intercepte le message, il ne pourra pas le lire, même avec la clé publique

Cryptage asymétrique

Bob est alors seul à pouvoir déchiffrer le message avec sa clé privée.
S'il veut répondre, Bob doit demander à Alice de créer elle aussi une combinaison de deux clés, et en partager une à tout le monde (donc à Bob), pour qu'il encrypte sa réponse.
Alors Bob pourra encrypter la réponse avec la clé publique d'Alice, qui sera seule à pouvoir la lire avec la clé privée qu'elle a créé.

SMURF

Dans un Smurf 'attaque par réflexion',
On envoie une quantité de requêtes ping 'tu me reçois ?'
avec une fausse adresse de réponse : celle de la victime
La victime est alors dépassée par la quantité d'information
qui lui arrive

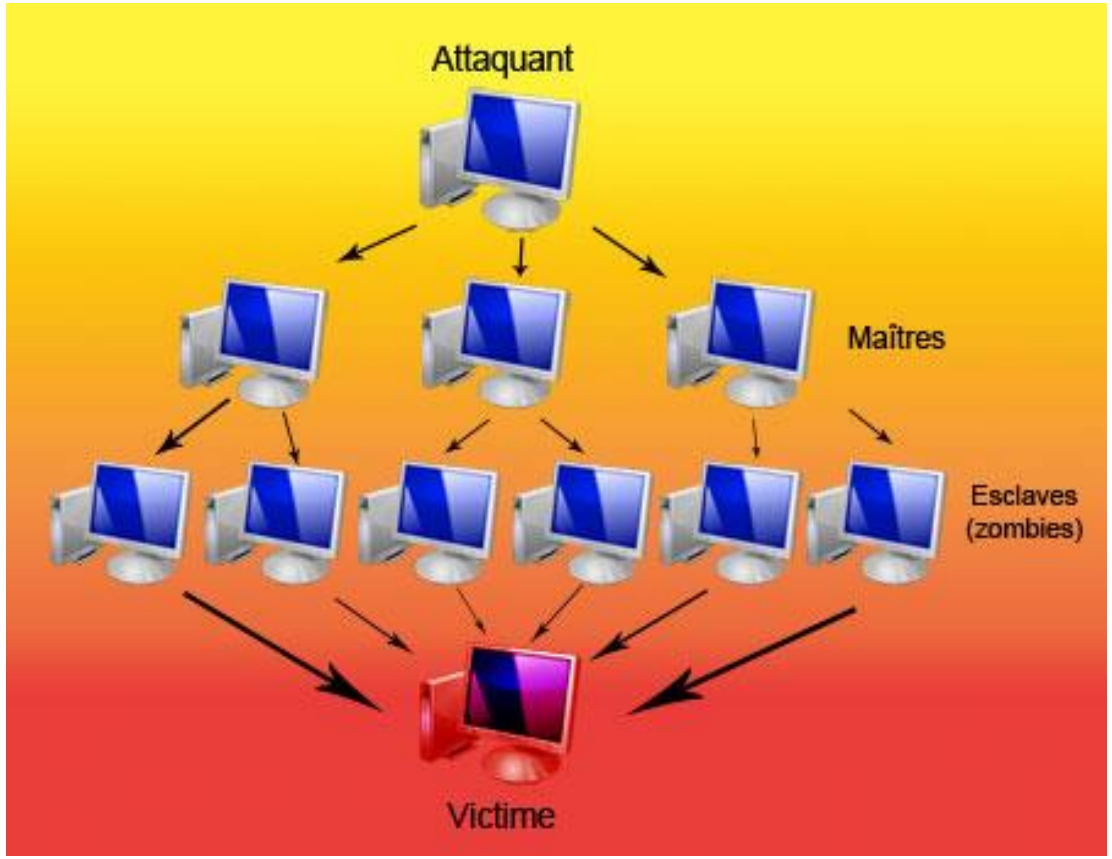


DDoS

Un pirate envoie des ordres à un grand nombre d'ordinateurs infectés 'botnet' : ce sont eux qui envoient systématiquement toutes les requêtes de ping.

Souvent les 'zombies' sont des particuliers infestés à leur insu.

Très difficile de remonter à l'auteur.



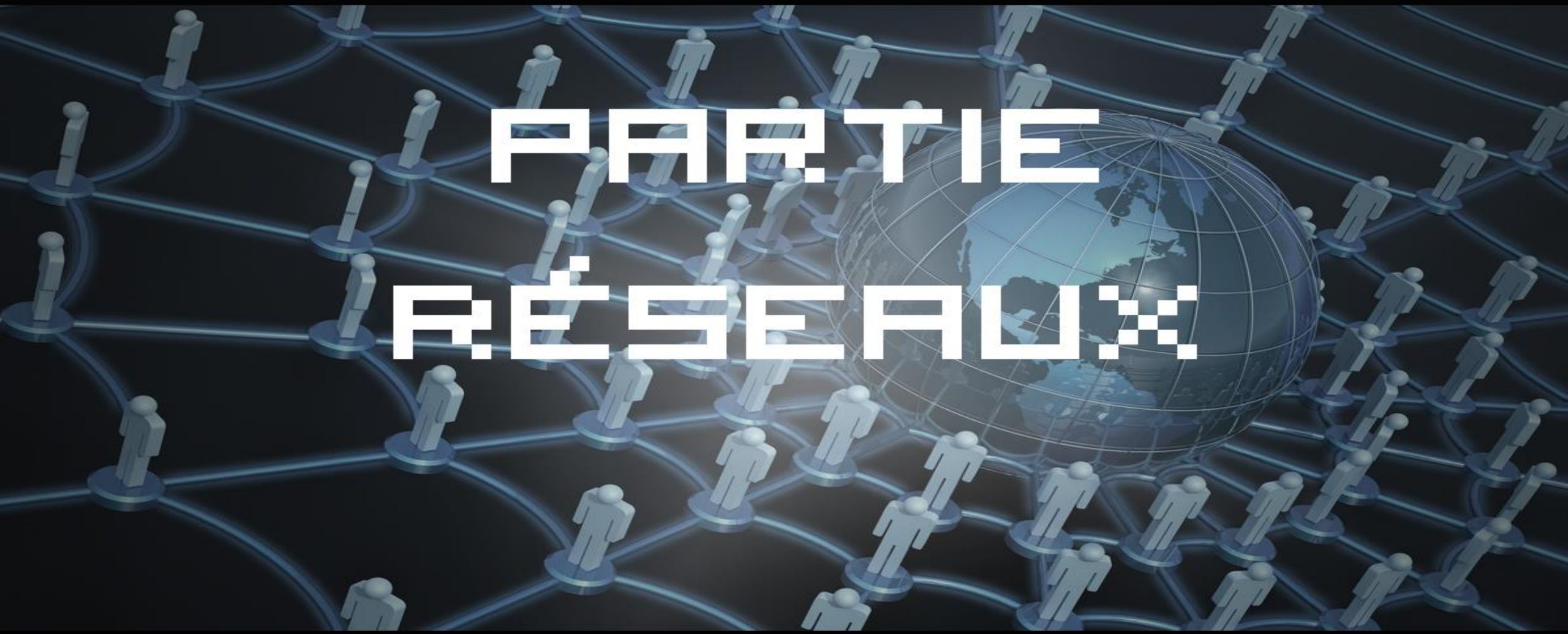
Les conséquences légales

Article 323-2 du code pénal :

'Le fait d'entraver ou de fausser le fonctionnement d'un système de traitement automatisé de données est puni de **cinq ans** d'emprisonnement et de **75000 euros** d'amende.’

Cordialement.

CENTRALE RESEAUX



PARTIE RÉSEAUX

Slides de Samy Ghribi, Sarah G. et Nayef Ghattas

INTRODUCTION

Objectifs :

- ✓ Comprendre comment est construit un réseau
- ✓ Comprendre comment les informations circulent sur un réseau

INTRODUCTION

Objectifs :

- ✓ Comprendre comment est construit un réseau
- ✓ Comprendre comment les informations circulent sur un réseau

- ✓ Et surtout… réussir le CF !

PLAN

I. Architecture d'un réseau
II. Les couches OSI et les équipements associés
III. Autres équipements

QU'EST-CE QU'UN RÉSEAU ?

- ✓ Un **réseau** est un **groupe de dispositifs** (ordinateurs, imprimantes, téléphones…) connectés afin d'échanger entre eux des **données**.

QU'EST-CE QU'UN RÉSEAU ?

- ✓ Un **réseau** est un **groupe de dispositifs** (ordinateurs, imprimantes, téléphones…) connectés afin d'échanger entre eux des **données**.

- ✓ Chaque dispositif du réseau est appelé un **nœud**.

QU'EST-CE QU'UN RÉSEAU ?

- ✓ Un **réseau** est un **groupe de dispositifs** (ordinateurs, imprimantes, téléphones…) connectés afin d'échanger entre eux des **données**.

- ✓ Chaque dispositif du réseau est appelé un **nœud**.

- ✓ Ces échanges ne peuvent pas être effectués n'importe comment: il faut suivre certaines **règles**, appelées **protocoles**. (voir partie II)

LES INDICATEURS-CLÉS D'UN RÉSEAU

- ✓Bande passante: qté d'information/qté de temps (en Mb/s, GB/s ...)
  - débit montant: on dépose sur un serveur distant (upload)
  - débit descendant: on télécharge (download)

LES INDICATEURS-CLÉS D'UN RÉSEAU

- ✓ Bande passante: qté d'information/qté de temps (en Mb/s, GB/s ...)
  - débit montant: on dépose sur un serveur distant (upload)
  - débit descendant: on télécharge (download)

- ✓ Temps de réponse: temps entre la fin d'une requête et le début de la réponse

LES INDICATEURS-CLÉS D'UN RÉSEAU

- ✓ Bande passante: qté d'information/qté de temps (en Mb/s, GB/s ...)
  - débit montant: on dépose sur un serveur distant (upload)
  - débit descendant: on télécharge (download)

- ✓ Temps de réponse: temps entre la fin d'une requête et le début de la réponse

- ✓ Sûreté, fiabilité…

I. ARCHITECTURE D'UN RÉSEAU

1. Relier des dispositifs
2. Communiquer sur un réseau
3. Echelles de réseaux

1. RELIER DES DISPOSITIFS

Deux moyens de relier des dispositifs (on parle de la **topologie du réseau**) :

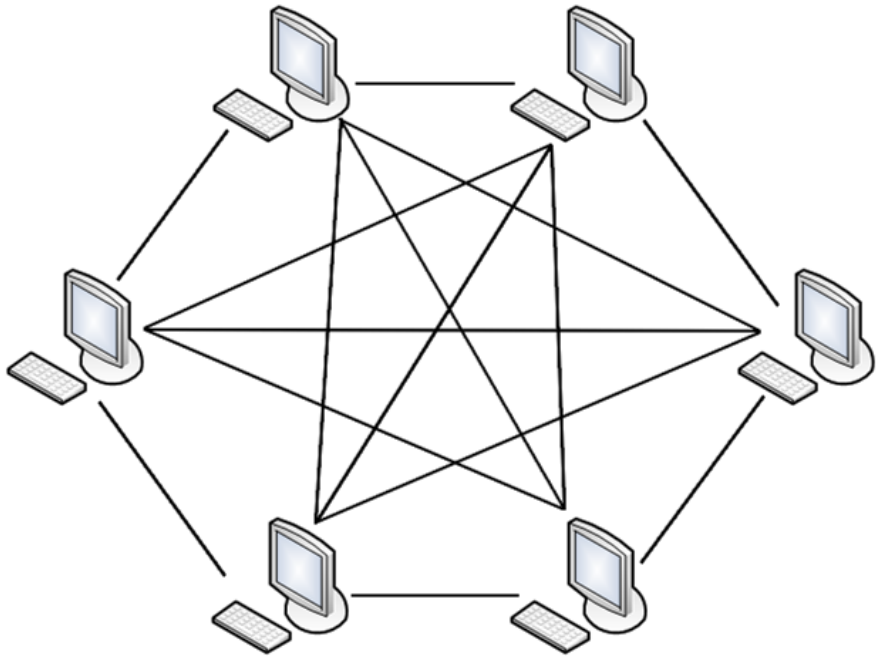
- ✓ Sans équipement réseau central (‘*Network Equipment*’)
- ✓ Avec un équipement réseau, appelé un **central**

  Les différents types de centraux (commutation, routage) seront détaillés en partie II.

Ici, nous ferons les exemples avec des ordinateurs.

1. RELIER DES DISPOSITIFS

Réseau en maille (sans central)

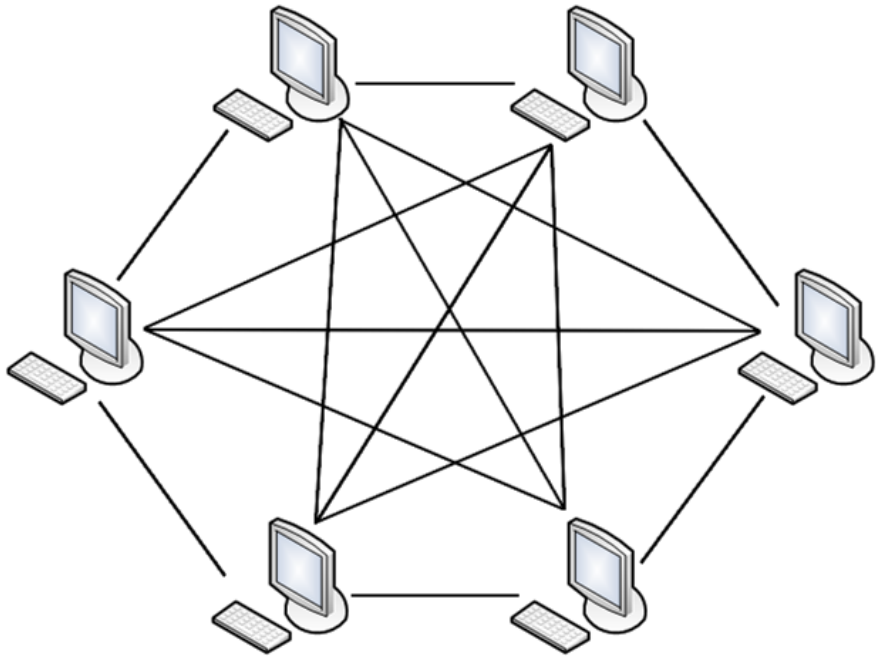


1. RELIER DES DISPOSITIFS

Réseau en maille (sans central)

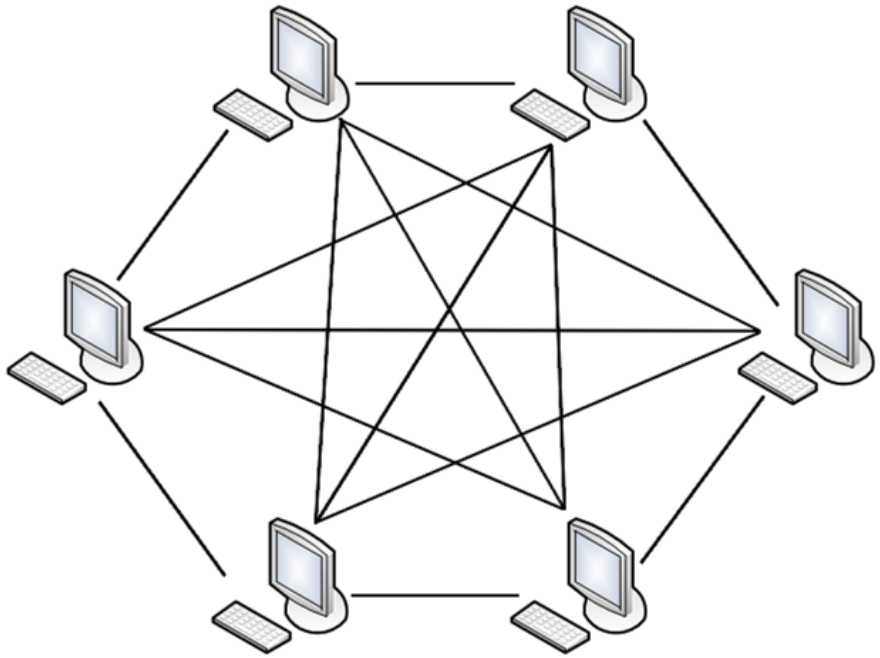
Avantages :

- ✓ Très fiable



1. RELIER DES DISPOSITIFS

Réseau en maille (sans central)



Avantages :

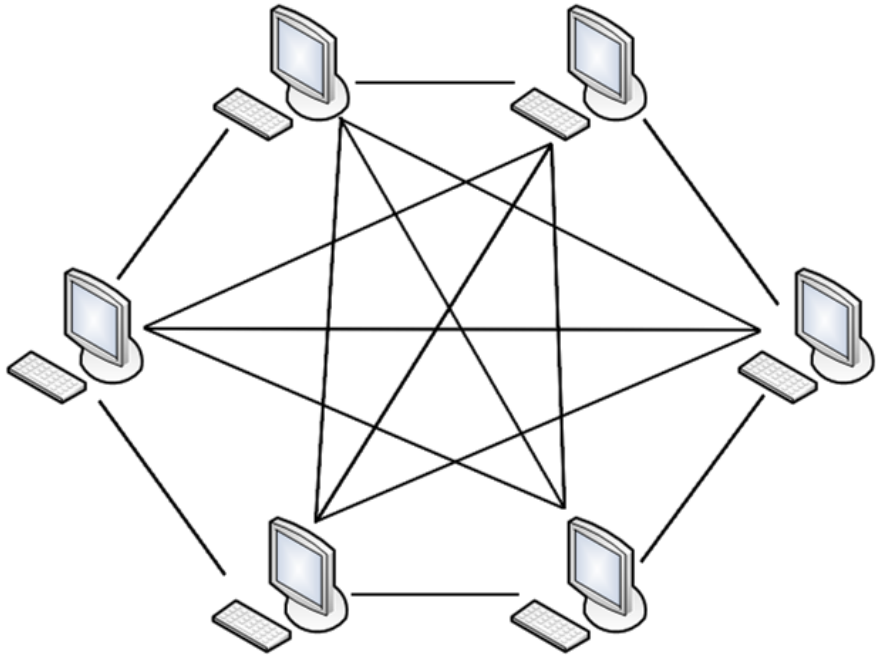
- ✓ Très fiable

Inconvénients :

- ✓ Trop de liens : $\frac{n(n-1)}{2}$
- ✓ Trop d'interfaces : $n-1$ par ordinateur

1. RELIER DES DISPOSITIFS

Réseau en maille (sans central)



Avantages :

- ✓ Très fiable

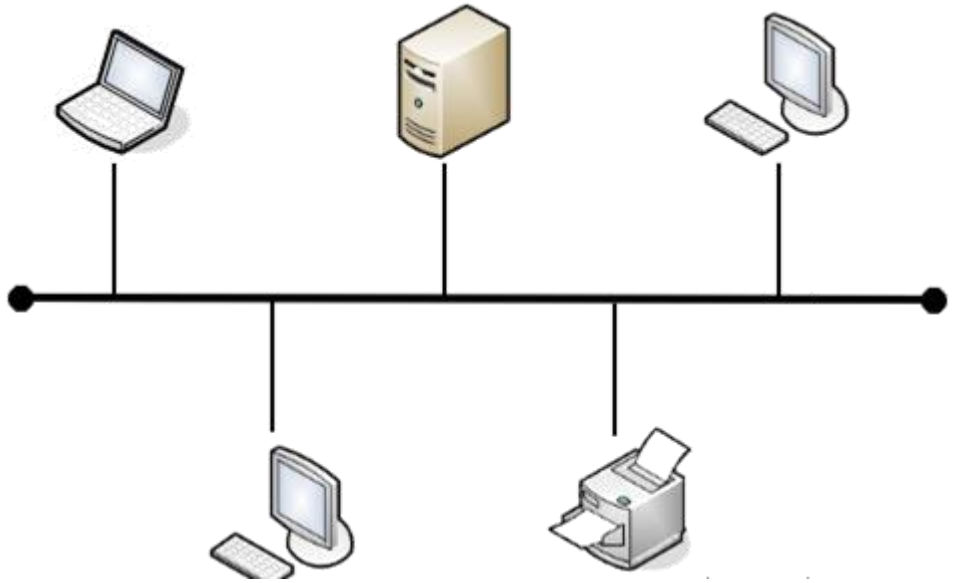
Inconvénients :

- ✓ Trop de liens : $\frac{n(n-1)}{2}$
- ✓ Trop d'interfaces : $n-1$ par ordinateur

➔ Adapté pour des **usages critiques**

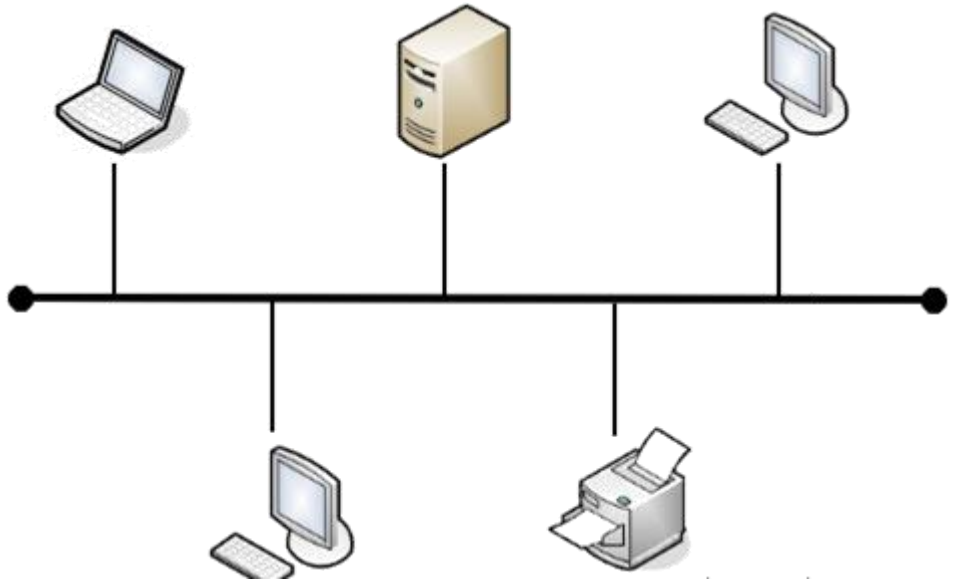
1. RELIER DES DISPOSITIFS

Réseau en bus (sans central)



1. RELIER DES DISPOSITIFS

Réseau en bus (sans central)

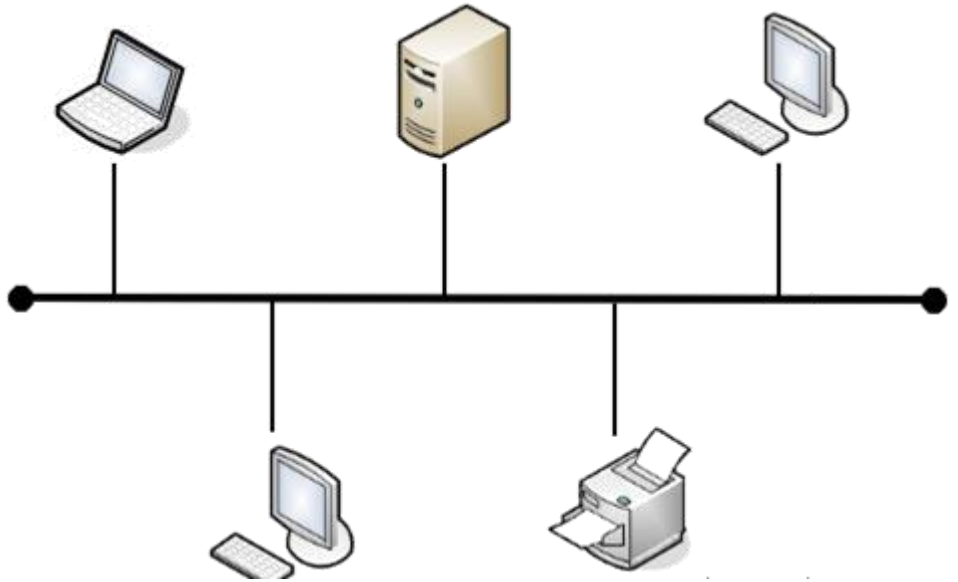


Avantages :

- ✓ Structure simple

1. RELIER DES DISPOSITIFS

Réseau en bus (sans central)



Avantages :

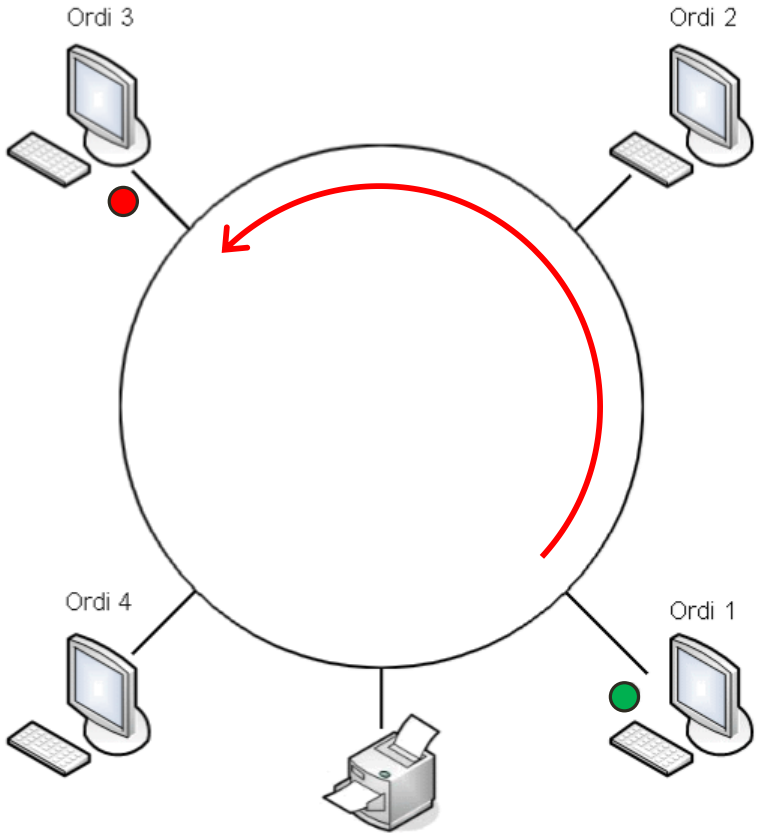
- ✓ Structure simple

Inconvénients :

- ✓ Tout le monde reçoit les données
- ✓ Si coupure du câble : réseau presque HS

1. RELIER DES DISPOSITIFS

Réseau en anneau ou '*ring*' (sans central)

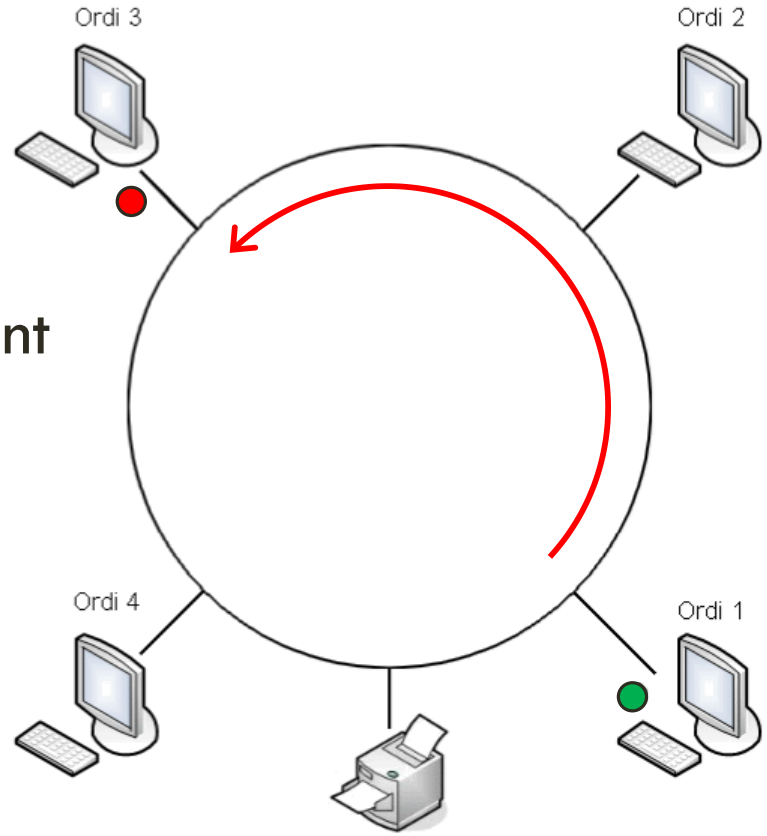


1. RELIER DES DISPOSITIFS

Réseau en anneau ou '*ring*' (sans central)

Avantages :

- ✓ Structure simple
- ✓ Ajout d'un ordinateur automatiquement



1. RELIER DES DISPOSITIFS

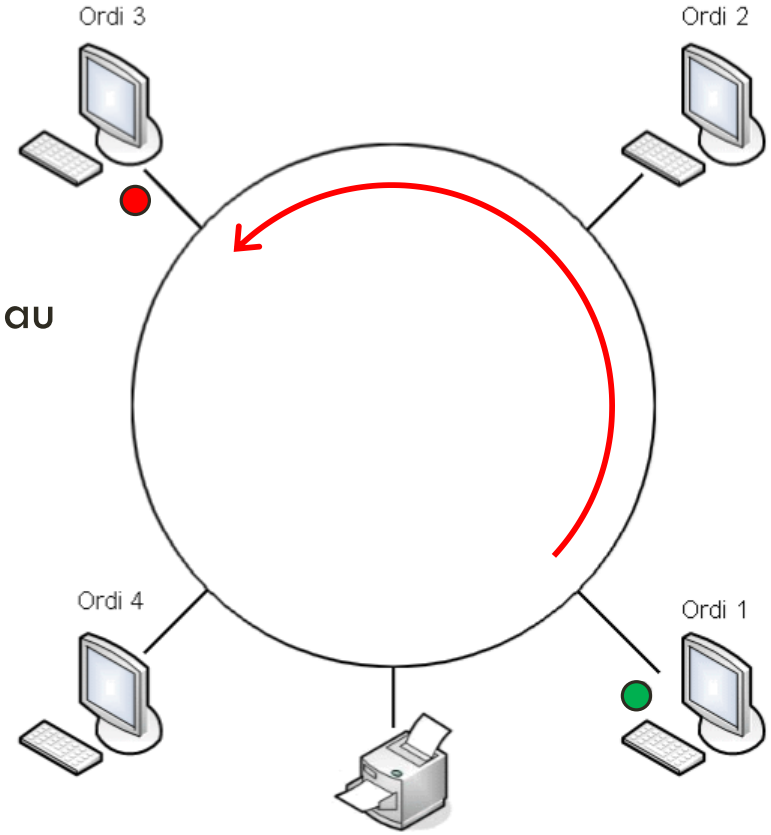
Réseau en anneau ou '*ring*' (sans central)

Avantages :

- ✓ Structure simple
- ✓ Ajout d'un ordinateur automatiquement
- ✓ Fonctionne même s'il y a une coupure dans l'anneau

Inconvénients :

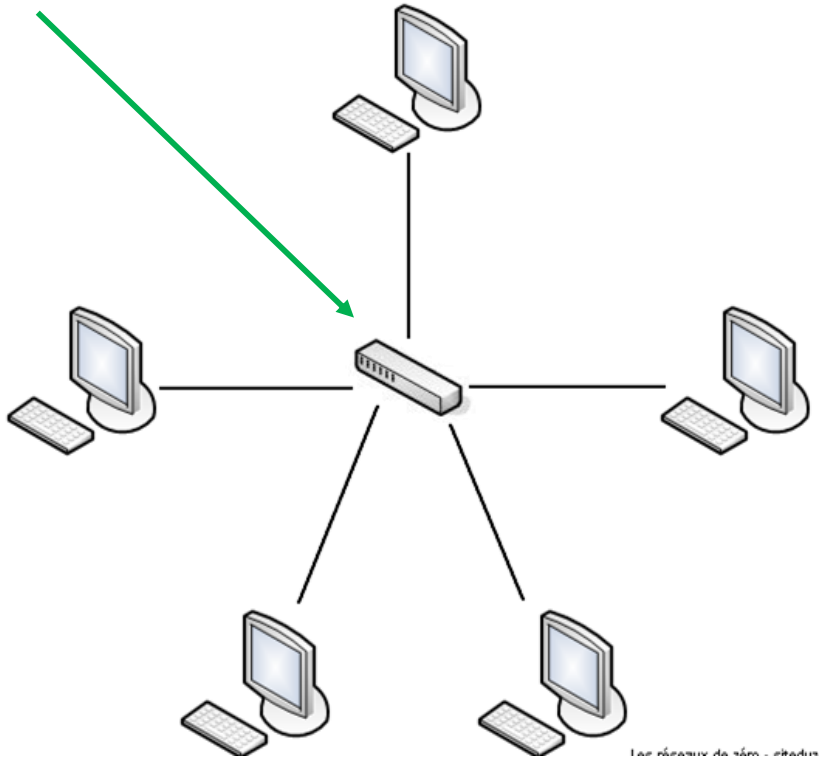
- ✓ Tout le monde reçoit les données et les transmet
- ✓ Si plus d'une coupure dans l'anneau : Réseau presque HS



1. RELIER DES DISPOSITIFS

Réseau en étoile ou '*star*' (avec central)

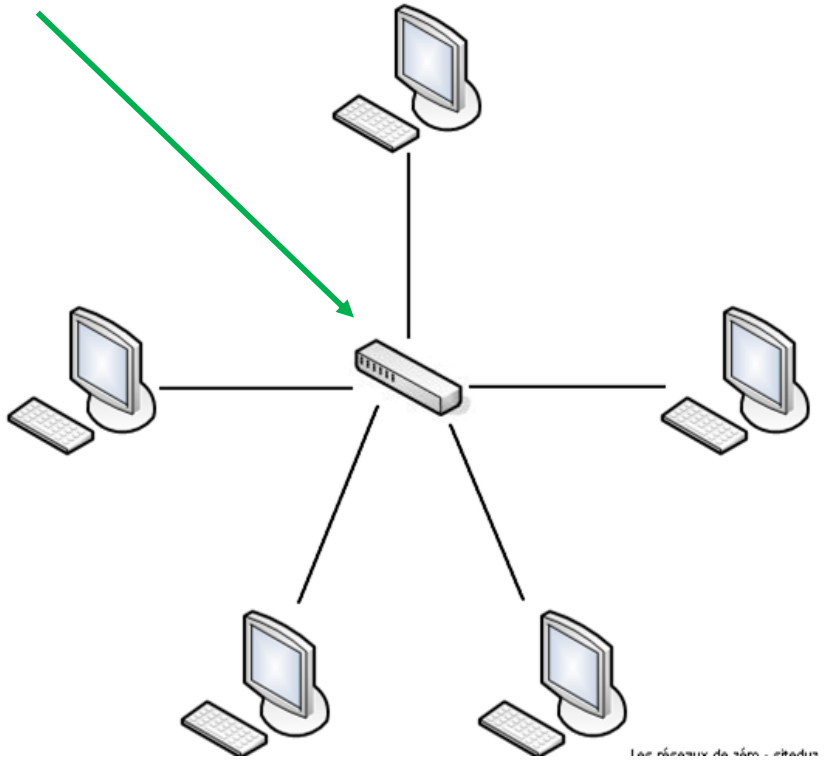
Toutes les communications passent par le **central.**



1. RELIER DES DISPOSITIFS

Réseau en étoile ou '*star*' (avec central)

Toutes les communications passent par le **central.**



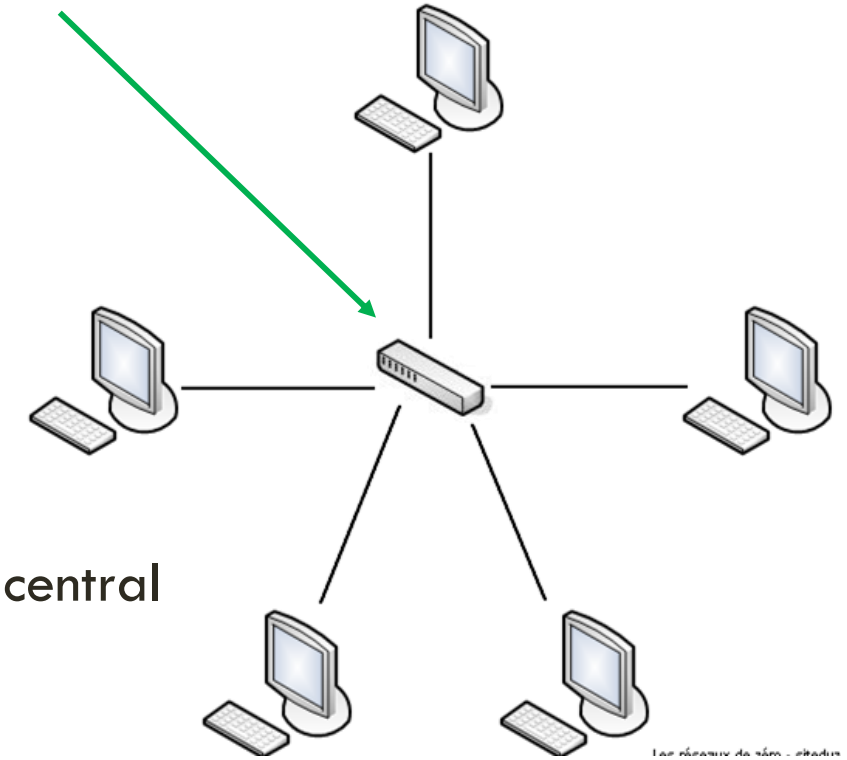
Avantages :

✓Un problème dans un terminal n'affecte pas le réseau entier

1. RELIER DES DISPOSITIFS

Réseau en étoile ou '*star*' (avec central)

Toutes les communications passent par le **central.**



Avantages :

✓Un problème dans un terminal n'affecte pas le réseau entier

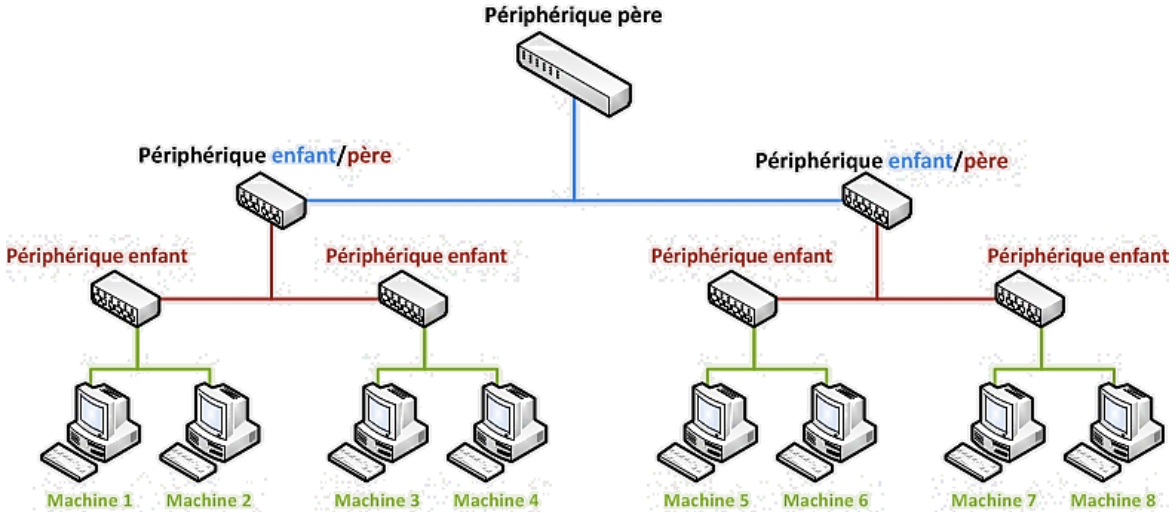
Inconvénients :

✓ Beaucoup de câbles : tout passe par le central

✓ Si le central est HS : le réseau est HS

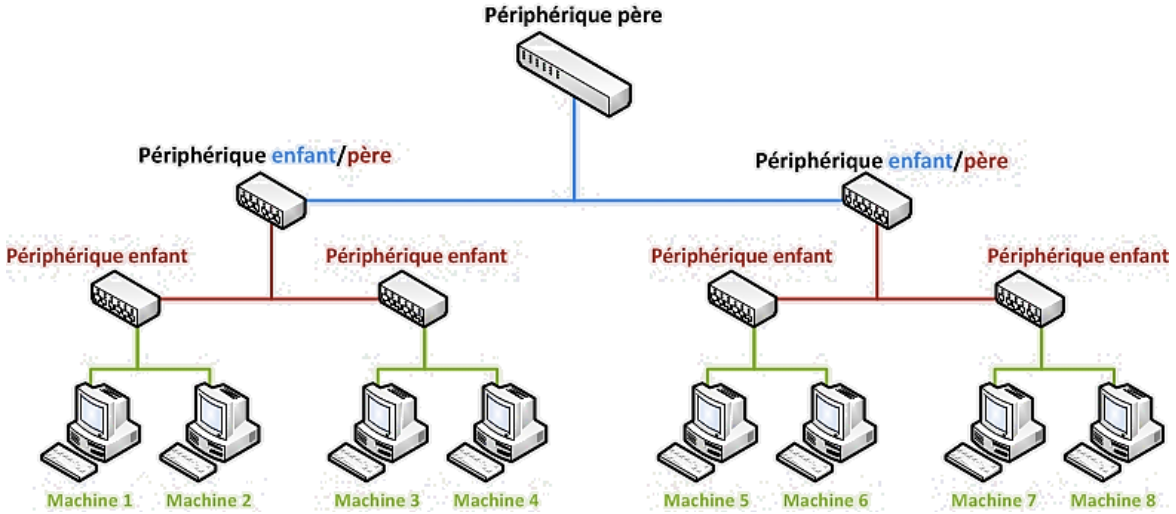
1. RELIER DES DISPOSITIFS

Réseau en arbre ou '*tree*' (avec centraux)



1. RELIER DES DISPOSITIFS

Réseau en arbre ou '*tree*' (avec centraux)

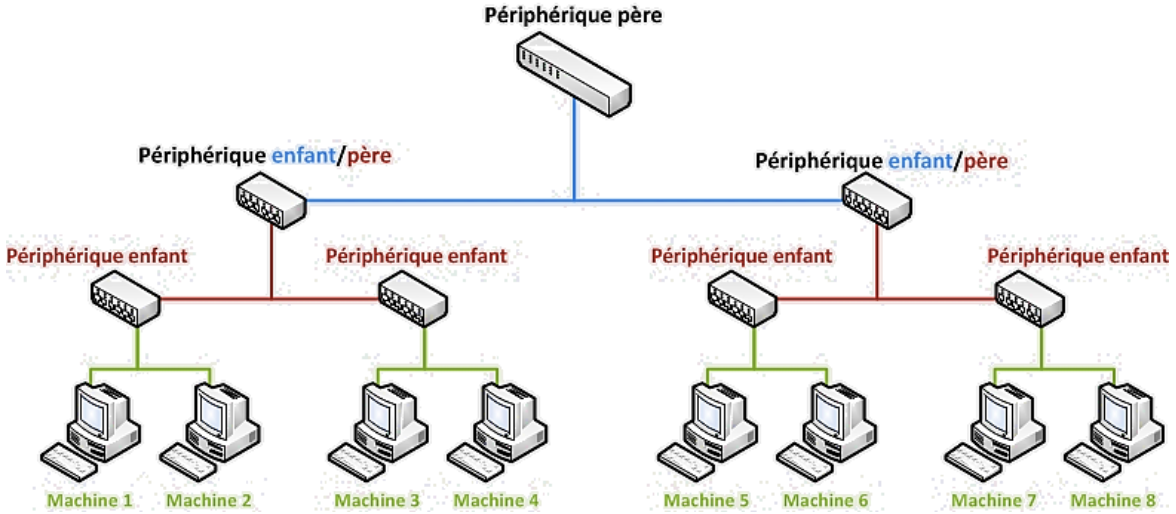


Avantages :

- ✓ Hiérarchisation, organisation

1. RELIER DES DISPOSITIFS

Réseau en arbre ou '*tree*' (avec centraux)



Avantages :

- ✓ Hiérarchisation, organisation

Inconvénients :

- ✓ Gestion difficile du réseau
- ✓ Si un central est HS : ses descendants sont isolés

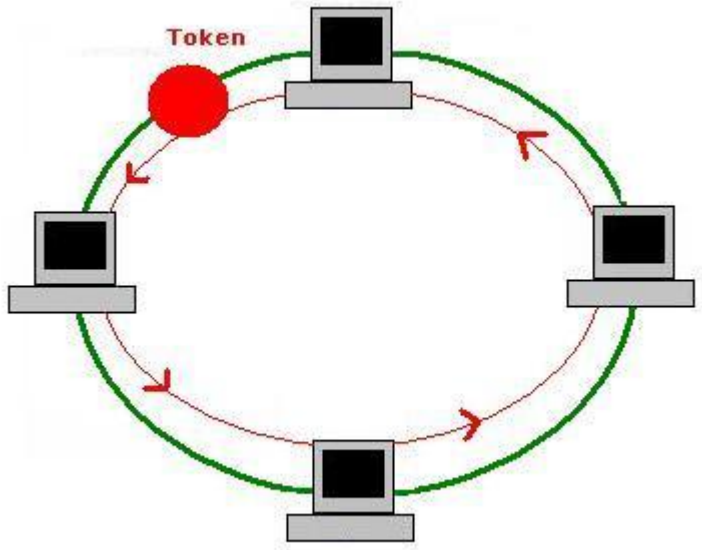
2. COMMUNIQUER SUR UN RÉSEAU

Deux types de **gestion** des communications sur un réseau :

2. COMMUNIQUER SUR UN RÉSEAU

Deux types de **gestion** des communications sur un réseau :

- ✓ Par la méthode du '*token ring*' : les données sont stockées dans un 'jeton' qui va passer d'un ordinateur au suivant, jusqu'à ce qu'il arrive au destinataire. L'ordinateur concerné vide le jeton, qui continue à voyager dans l'anneau jusqu'à être rempli de nouveau (réseau en anneau ou ring)



2. COMMUNIQUER SUR UN RÉSEAU

Deux types de **gestion** des communications sur un réseau :

✓Par une méthode aléatoire : protocole **CSMA/CD**

(Carrier Sense Multiple Acces with Collision Detection)

2. COMMUNIQUER SUR UN RÉSEAU

Deux types de **gestion** des communications sur un réseau :

✓Par une méthode aléatoire : protocole **CSMA/CD**

(Carrier Sense Multiple Acces with Collision Detection)

Le dispositif qui vient d'émettre 'écoute' le réseau.

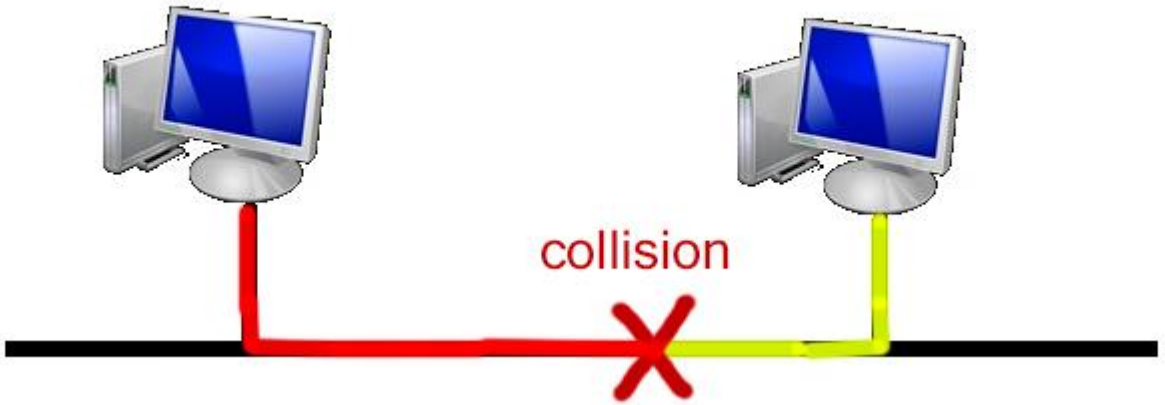
2. COMMUNIQUER SUR UN RÉSEAU

Deux types de **gestion** des communications sur un réseau :

✓Par une méthode aléatoire : protocole **CSMA/CD**

(Carrier Sense Multiple Acces with Collision Detection)

Si le signal est brouillé, c'est qu'il y a eu collision avec un paquet envoyé par un autre dispositif.



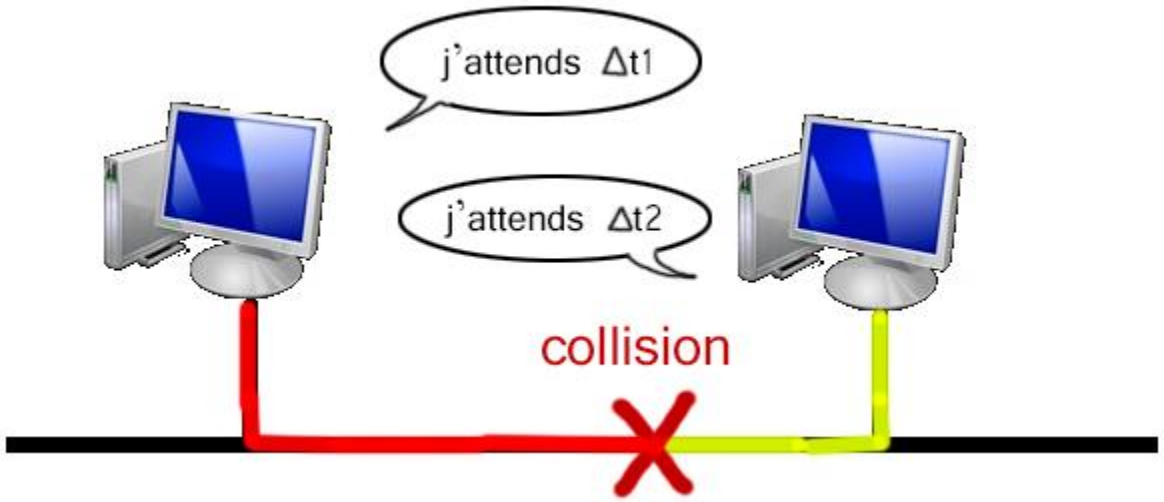
2. COMMUNIQUER SUR UN RÉSEAU

Deux types de **gestion** des communications sur un réseau :

✓Par une méthode aléatoire : protocole **CSMA/CD**

(Carrier Sense Multiple Acces with Collision Detection)

Les stations attendent donc chacune un temps aléatoire avant de réessayer de transmettre le message.



2. COMMUNIQUER SUR UN RÉSEAU

Plusieurs **directions** de communication :

2. COMMUNIQUER SUR UN RÉSEAU

Plusieurs **directions** de communication :

- ✓ Simplex :

2. COMMUNIQUER SUR UN RÉSEAU

Plusieurs **directions** de communication :

- ✓ Simplex :

- ✓ Half-duplex :

2. COMMUNIQUER SUR UN RÉSEAU

Plusieurs **directions** de communication :

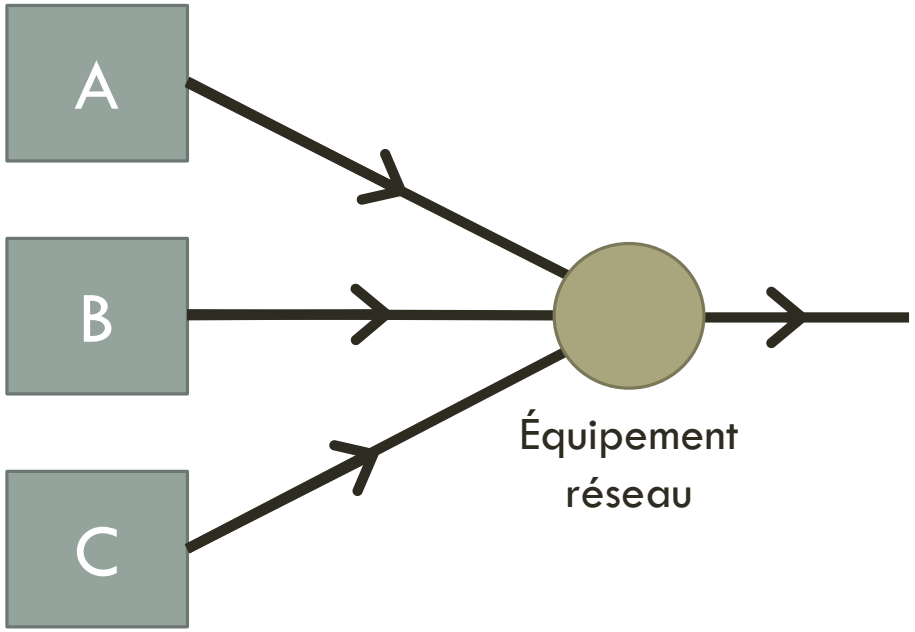
- ✓ Simplex :

- ✓ Half-duplex :

- ✓ Full-duplex :

2. COMMUNIQUER SUR UN RÉSEAU

- ✓ Multiplexage : communiquer à plusieurs sur la même ligne



3. ÉCHELLES DE RÉSEAUX

Il y a des **milliards** d'équipements réseau dans le monde.

3. ÉCHELLES DE RÉSEAUX

Il y a des **milliards** d'équipements réseau dans le monde.

On ne peut pas gérer de la même manière des réseaux de tailles très différentes.

On distingue 3 **échelles de réseaux :**

3. ÉCHELLES DE RÉSEAUX

Il y a des **milliards** d'équipements réseau dans le monde.

On ne peut pas gérer de la même manière des réseaux de tailles très différentes.

On distingue 3 **échelles de réseaux** :

LAN (Local Area)

salle, campus…

3. ÉCHELLES DE RÉSEAUX

Il y a des **milliards** d'équipements réseau dans le monde.

On ne peut pas gérer de la même manière des réseaux de tailles très différentes.

On distingue 3 **échelles de réseaux** :

LAN (Local Area)

salle, campus…

MAN (Metropolitan)

ville…

3. ÉCHELLES DE RÉSEAUX

Il y a des **milliards** d'équipements réseau dans le monde.

On ne peut pas gérer de la même manière des réseaux de tailles très différentes.

On distingue 3 **échelles de réseaux** :

LAN (Local Area)

salle, campus…

MAN (Metropolitan)

ville…

WAN (Wide Area)

pays, monde…

II. LES 7 COUCHES OSI ET LES ÉQUIPEMENTS ASSOCIÉS

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
5. Couche session
6. Couche présentation
7. Couche application

LE MODÈLE OSI

Principe du modèle Open System Interconnection

✓ Découpe les étapes de la transmission d'un message en 7 parties, appelées les **7 couches** du modèle OSI.

LE MODÈLE OSI

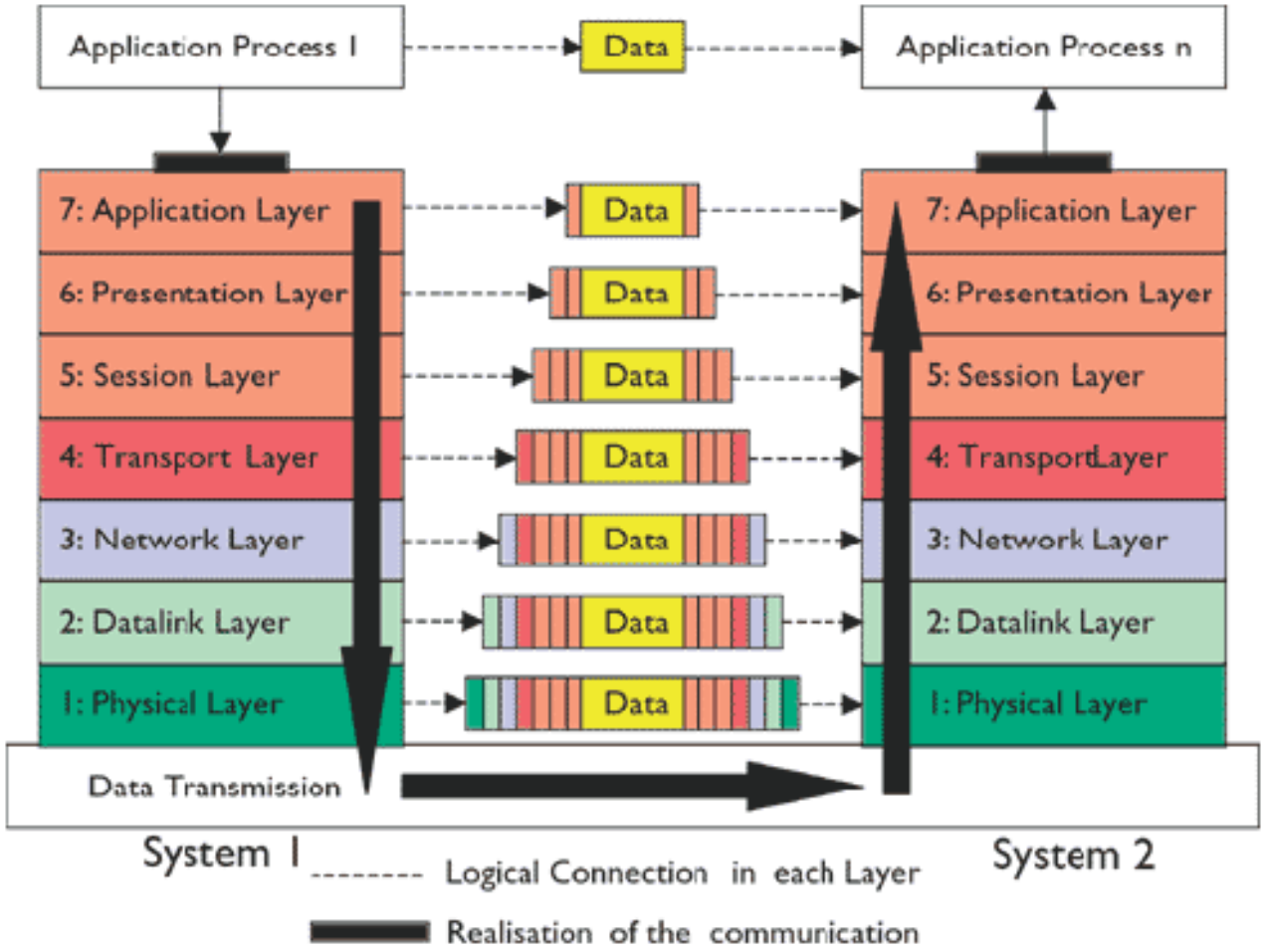
Principe du modèle Open System Interconnection

- ✓ Découpe les étapes de la transmission d'un message en 7 parties, appelées les **7 couches** du modèle OSI.
- ✓ Ces couches sont **indépendantes** les unes des autres.

LE MODÈLE OSI

Principe du modèle Open System Interconnection

- ✓ Découpe les étapes de la transmission d'un message en 7 parties, appelées les **7 couches** du modèle OSI.
- ✓ Ces couches sont **indépendantes** les unes des autres.
- ✓ Pour chaque couche, on associe plusieurs **protocoles** permettant de réaliser la fonction de la couche.

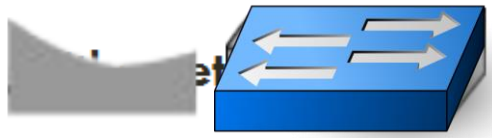


✓**Encapsulation** : l'émetteur ***enrobe*** le message à transmettre avec 7 **couches** contenant les **informations relatives à chaque protocole**. Le récepteur déballe successivement les couches, récupère les informations contenues dans chaque couche, puis retrouve le message initial.

LE MODÈLE OSI

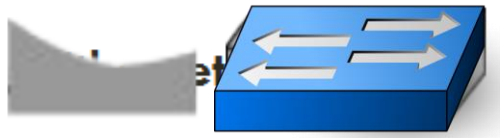
COUCHE	FONCTIONS	Protocoles
7 – Application	Interface entre l'utilisateur (application) et le réseau	**HTTP, DNS, SMTP, DHCP, …**
6 – Présentation	Manière dont les données sont chiffrées, compressés, codés, …	**GIF, JPEG, ASCII, …**
5 – Session	Gestion de l'établissement et du maintien de la connexion	
4 – Transport	Fiabilité de la réception d'un message, découpage et recollage des messages	**TCP, UDP, …**
3 – Réseau	Détermination de la (meilleure) route à employer pour transmettre un message	**IP (IPv4 et IPv6), OSPF, BGP …**
2 – Liaison	Adressage en local , détection d'erreurs	**CSMA/CD, MAC …**
1 – Physique	Déplacement de bits entres les équipements	**USB, Bluetooth, coaxial**

LE MODÈLE OSI

COUCHE	FONCTIONS	Protocoles
7 – Application	Interface entre l'utilisateur (application) et le réseau	**HTTP, DNS, SMTP, DHCP, …**
6 – Présentation	Manière dont les données sont chiffrées, compressés, codés, …	**SSL, JPEG, ASCII, …**
5 – Session	Gestion de l'établissement et du maintien de la connexion	
4 – Transport	Fiabilité de la réception d'un message, découpage et recollage des messages	**TCP, UDP, …**
3 – Réseau	Détermination de la (meilleure) route à employer pour transmettre un message	**IP**  **6),**
2 – Liaison	Adressage en local , détection d'erreurs	
1 – Physique	Déplacement de bits entres les équipements	**USB**  **oth,**

LE MODÈLE OSI

COUCHE	FONCTIONS	Protocoles
7 – Application	Interface entre l'utilisateur (application) et l	HTTP, DNS, SMTP, DHCP, …
6 – Présentation	Manière dont les d chiffrées, compressés, codés, …	SSL, JPEG, ASCII, …
5 – Session	Gestion de l'établissement et du maintien de la connexion	
4 – Transport	Fiabilité de la réception d'un message, découpage et recollage des messages	TCP, UDP, …
3 – Réseau	Détermination de la (meilleure) route à employer pour transmettre un message	IP 6),
2 – Liaison	Adressage en local , détection d'erreurs	et
1 – Physique	Déplacement de bits entre les équipements	USB oth,



HISTOIRE D'UNE PETITE REQUÊTE WEB

John cherche à accéder à killer.via.ecp.fr

HISTOIRE D'UNE PETITE REQUÊTE WEB

John cherche à accéder à killer.via.ecp.fr

Avec les couches OSI, ça se passe comment ?

7. COUCHE APPLICATION

Fonction

Gérer ce qui concerne l'interface entre le réseau et l'utilisateur, quasiment tout ce qu'un utilisateur fait sur un ordinateur est en couche 7

Exemples : HTTP, DNS, DHCP, SMTP…

7. COUCHE APPLICATION

Fonction

Gérer ce qui concerne l'interface entre le réseau et l'utilisateur, quasiment tout ce qu'un utilisateur fait sur un ordinateur est en couche 7

Exemples : HTTP, DNS, DHCP, SMTP…

DNS (Domain Name System)

Permet de faire la correspondance entre un **nom de domaine** et une **adresse IP**

Exemple : si on tape killer.via.ecp.fr dans son navigateur, les DNS vont faire la correspondance avec l'adresse IP : **138.195.130.37**

7. COUCHE APPLICATION

Fonction

Gérer ce qui concerne l'interface entre le réseau et l'utilisateur, quasiment tout ce qu'un utilisateur fait sur un ordinateur est en couche 7

Exemples : HTTP, DNS, DHCP, SMTP…

DNS (Domain Name System)

Permet de faire la correspondance entre un **nom de domaine** et une **adresse IP**

Exemple : si on tape killer.via.ecp.fr dans son navigateur, les DNS vont faire la correspondance avec l'adresse IP : **138.195.130.37**

DHCP (Dynamic Host Configuration Protocol)

Permet d'**attribuer une adresse IP** à un équipement réseau.

HISTOIRE D'UNE PETITE REQUÊTE WEB – LES PREMIERS PAS

- ✓ John cherche à accéder à killer.via.ecp.fr

HISTOIRE D'UNE PETITE REQUÊTE WEB – LES PREMIERS PAS

- ✓ John cherche à accéder à killer.via.ecp.fr

- ✓ L'ordinateur, en utilisant le protocole DNS, regarde à quelle adresse ip le site correspond, il met cette adresse de côté.

HISTOIRE D'UNE PETITE REQUÊTE WEB – LES PREMIERS PAS

- ✓ John cherche à accéder à killer.via.ecp.fr

- ✓ L'ordinateur, en utilisant le protocole DNS, regarde à quelle adresse ip le site correspond, il met cette adresse de côté.

- ✓ Le navigateur de John rédige une requête HTTP qu'il va envoyer au serveur de google !

Le contenu de la requête ?

« Coucou je m'appelle "navigateur de John", il est 21h chez moi, tu pourrais m'envoyer la dernière version de la page web google.com, s'il te plaît ? »

6. COUCHE PRÉSENTATION

Fonction

Gérer la manière dont les données sont présentées (chiffrées, compressés, codés, …)

Exemples : JPEG (images), ASCII (texte), SSL (données sécurisées)

6. COUCHE PRÉSENTATION

Fonction

Gérer la manière dont les données sont présentées (chiffrées, compressés, codés, …)

Exemples : JPEG (images), ASCII (texte), SSL (données sécurisées)

Rien d'autre à savoir sur cette couche !

HISTOIRE D'UNE PETITE REQUÊTE WEB – L'ENCODAGE

- ✓ Pour que le transport de la requête soit super rapide, une partie de la requête est compressée, et un encodage est choisi !

HISTOIRE D'UNE PETITE REQUÊTE WEB – L'ENCODAGE

- ✓ Pour que le transport de la requête soit super rapide, une partie de la requête est compressée, et un encodage est choisi !
- ✓ Les encodages c'est très simple (utf-8, ascii, etc… cf. TD1 cours d'algo-prog):

  ça sert à dire à s'assurer que les deux ordinateurs parlent « la même langue »

  Par exemple, en utf-8 : c3 a9 (code hexadécimal) => é

  Si mon ordinateur ne parle pas l'utf-8 par défaut, il va plutôt traduire en charabia : /#{|\@88!

5. COUCHE SESSION

Fonction

Gérer l'établissement et le maintien d'une connexion

5. COUCHE SESSION

Fonction

Gérer l'établissement et le maintien d'une connexion

Rien d'autre à savoir sur cette couche !

HISTOIRE D'UNE PETITE REQUÊTE WEB – LA SESSION

- ✓ Comment est-ce que les données vont transiter entre l'ordinateur de John et le serveur du killer?

- ✓ Est-ce que tout va se faire d'une manière sécurisée ?

4. COUCHE TRANSPORT

Fonctions

Découper/Recoller les messages à transmettre en **paquets.**

S’assurer de la bonne réception de ces paquets

4. COUCHE TRANSPORT

Fonctions

Découper/Recoller les messages à transmettre en **paquets.**

S'assurer de la bonne réception de ces paquets

Transmission Control Protocol et User Datagram Protocol

- ✓ **TCP** : si on n'a pas d'*accusé de réception*, on renvoie le paquet (pages Web, …)
- ✓ **UDP** : on envoie sans vérifier la bonne réception (TV sur Internet, …)

4. COUCHE TRANSPORT

Fonctions

Découper/Recoller les messages à transmettre en **paquets.**

S'assurer de la bonne réception de ces paquets

Transmission Control Protocol et User Datagram Protocol

- ✓ **TCP** : si on n'a pas d'*accusé de réception*, on renvoie le paquet (pages Web, …)
- ✓ **UDP** : on envoie sans vérifier la bonne réception (TV sur Internet, …)

Pourquoi découper en paquets ?

On envoie simplement (beaucoup) de **petites quantités** d'informations, et on attend que celles-ci arrivent.

On n'a **pas besoin de monopoliser la disponibilité de la ligne** (≠ téléphonie).

4. COUCHE TRANSPORT

Les ports

Numéro (entre 0 et 65535) permettant de classer les types d'interlocuteurs lors de l'échange d'informations.

Pas de réalité physique !

4. COUCHE TRANSPORT

Les ports

Numéro (entre 0 et 65535) permettant de classer les types d'interlocuteurs lors de l'échange d'informations.

Pas de réalité physique !

N° de port	Protocoles	Descriptions
21	**FTP** : File Transfer Protocol	Transfert de fichiers
22	**SSH** : Secure Shell	Communication sécurisée
25	**STMP:** Simple Mail Transfer Protocol	Envoi de mails
80	**HTTP** : HyperText Transfer Protocol	Pages Web
110/143	**POP3 /IMAP4**	Réception de mails
443	**HTTPS** : HyperText Transfer Protocol Secure	Données chiffrées

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉCOUPAGE

- ✓ La requête de John va être découpée en plusieurs petits morceaux : **les paquets** !

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉCOUPAGE

- ✓ La requête de John va être découpée en plusieurs petits morceaux : **les paquets** !

- ✓ C'est sur cette couche où on va décider de l'importance des paquets et de leur découpage et assemblage !

Pour la requête de John : **TCP**

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉCOUPAGE

- ✓ La requête de John va être découpée en plusieurs petits morceaux : **les paquets** !

- ✓ C'est sur cette couche où on va décider de l'importance des paquets et de leur découpage et assemblage !

  Pour la requête de John : **TCP**

- ✓ **Et puis c'est bon, les paquets s'envolent sur le net !**

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉCOUPAGE

- ✓ La requête de John va être découpée en plusieurs petits morceaux : **les paquets** !

- ✓ C'est sur cette couche où on va décider de l'importance des paquets et de leur découpage et assemblage !

  Pour la requête de John : **TCP**

- ✓ **Et puis c'est bon, les paquets s'envolent sur le net ! (ou presque… on ne sait pas encore à qui il faut l'envoyer)**

3. COUCHE RÉSEAU

Fonctions

Détecter la *meilleure* route possible sur un réseau pour transmettre un message

Interconnecter différents réseaux à une grande échelle : protocole **IP**

3. COUCHE RÉSEAU

Fonctions

Détecter la *meilleure* route possible sur un réseau pour transmettre un message

Interconnecter différents réseaux à une grande échelle : protocole **IP**

L'adresse IP (Internet Protocol)

✓ Chaque équipement réseau se voit attribuer une adresse codée sur 32 bits (**IPv4**) ou 128 bits (**IPv6**) qui permet une identification sur un réseau.

3. COUCHE RÉSEAU

Fonctions

Détecter la *meilleure* route possible sur un réseau pour transmettre un message

Interconnecter différents réseaux à une grande échelle : protocole **IP**

L'adresse IP (Internet Protocol)

✓ Chaque équipement réseau se voit attribuer une adresse codée sur 32 bits (**IPv4**) ou 128 bits (**IPv6**) qui permet une identification sur un réseau.

IPv4 : de 0.0.0.0 à 255.255.255.255

($2^{32} \approx$ 4 milliards adresses en IPv4)

IPv6 : de 0000:0000:0000:0000:0000:0000:0000:0000 à FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

($2^{128} \approx 3 \cdot 10^{38}$ adresses en IPv6)

3. COUCHE RÉSEAU

Fonctions

Détecter la *meilleure* route possible sur un réseau pour transmettre un message

Interconnecter différents réseaux à une grande échelle : protocole **IP**

L'adresse IP (Internet Protocol)

✓ Chaque équipement réseau se voit attribuer une adresse codée sur 32 bits (**IPv4**) ou 128 bits (**IPv6**) qui permet une identification sur un réseau.

IPv4 : de 0.0.0.0 à 255.255.255.255

($2^{32} \approx 4$ milliards adresses en IPv4)

IPv6 : de 0000:0000:0000:0000:0000:0000:0000:0000 à FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

($2^{128} \approx 3 \cdot 10^{38}$ adresses en IPv6)

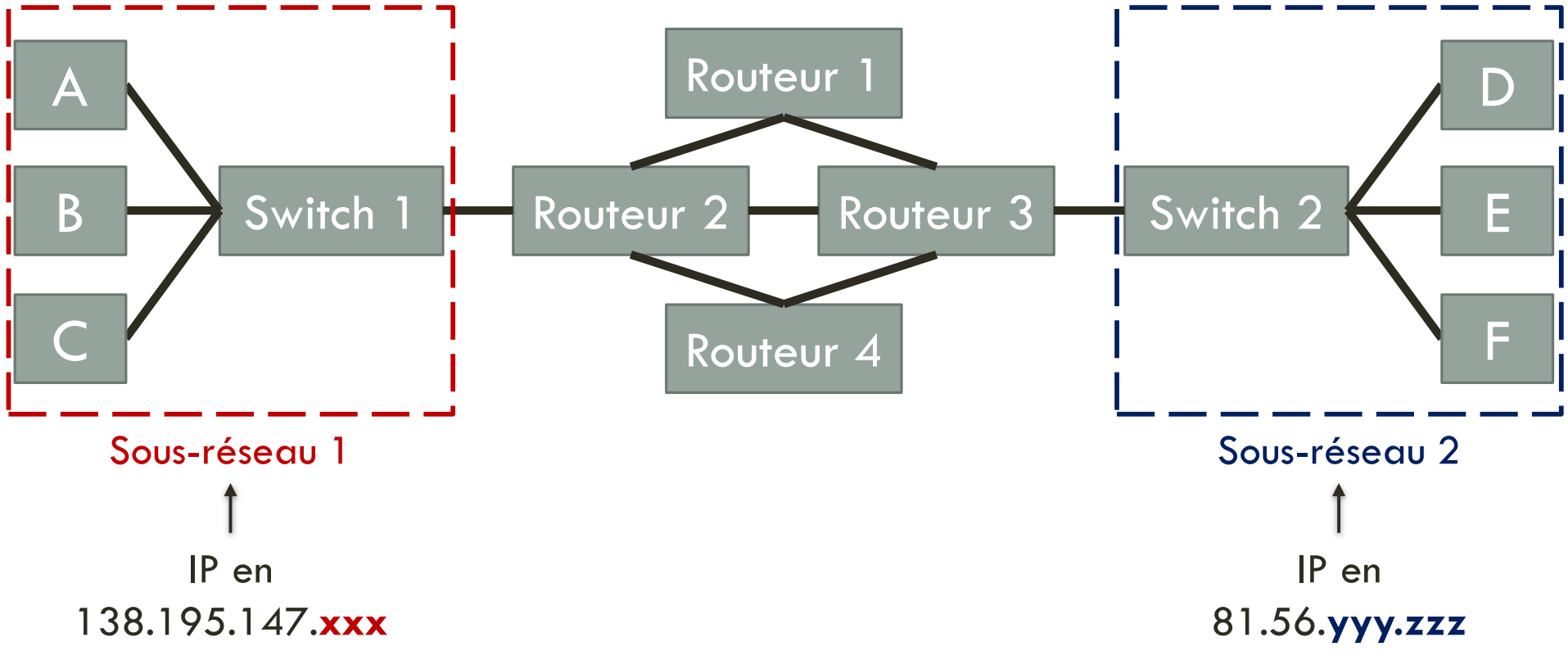
✓ Un début de **hiérarchisation :** $\underbrace{138.195}_{\text{É.C.P.}}.\ \underbrace{147}_{\text{Bât. D}}\ .\ \underbrace{24}_{\text{Ch. 006}}$

3. COUCHE RÉSEAU

Types d'équipements (suite)

✓ Le **routeur :**

Permet d'établir une connexion entre plusieurs sous-réseaux différents (éventuellement de couche 2 différente).



HISTOIRE D'UNE PETITE REQUÊTE WEB – L'ENVOI

- ✓ On note sur chaque paquet l'adresse IP à laquelle ils doivent être envoyés !

HISTOIRE D'UNE PETITE REQUÊTE WEB – L'ENVOI

- ✓ On note sur chaque paquet l'adresse IP à laquelle ils doivent être envoyés !
- ✓ Et puis c'est presque quasiment bon !

HISTOIRE D'UNE PETITE REQUÊTE WEB – L'ENVOI (OU PRESQUE)

- ✓ On note sur chaque paquet l'adresse IP à laquelle ils doivent être envoyés !

- ✓ Et puis c'est presque quasiment bon ! (hé oui, l'adresse IP n'identifie pas complètement une machine sur internet !)

2. COUCHE LIAISON

Fonctions

Assembler les bits en octets puis en cadres

Détecter les collisions lors de la transmission (**CSMA/CD** sur un réseau **Ethernet**)

Adresser des messages aux équipements selon leur adresse MAC : **commutation**

2. COUCHE LIAISON

Fonctions

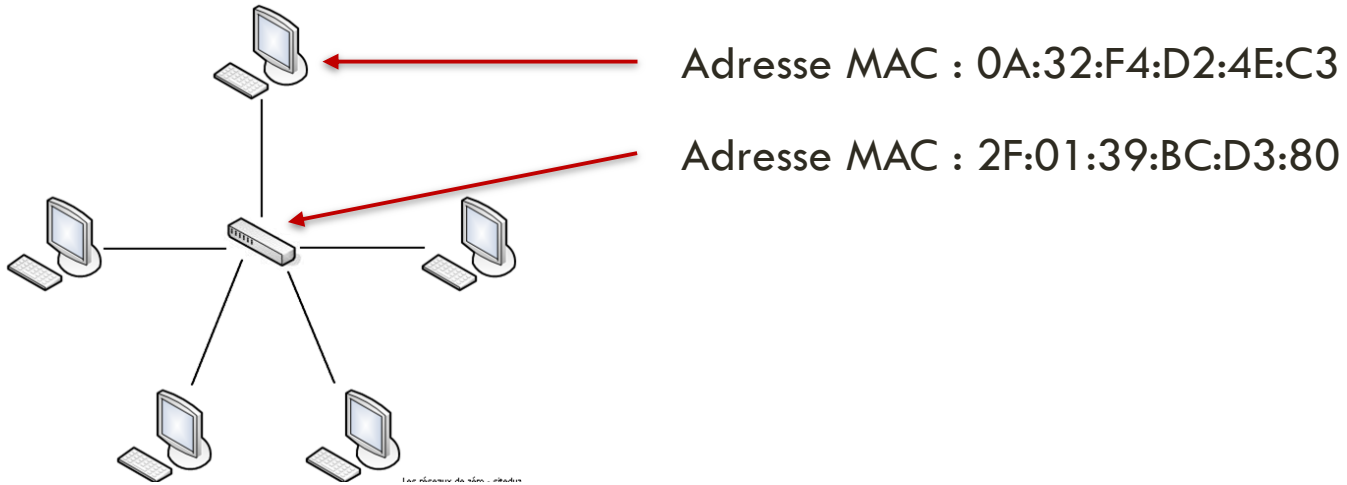
Assembler les bits en octets puis en cadres

Détecter les collisions lors de la transmission (**CSMA/CD** sur un réseau **Ethernet**)

Adresser des messages aux équipements selon leur adresse MAC : **commutation**

Un identifiant unique : l'adresse MAC (Media Access Control)

Identifie de manière **unique** tous les équipements du réseau.

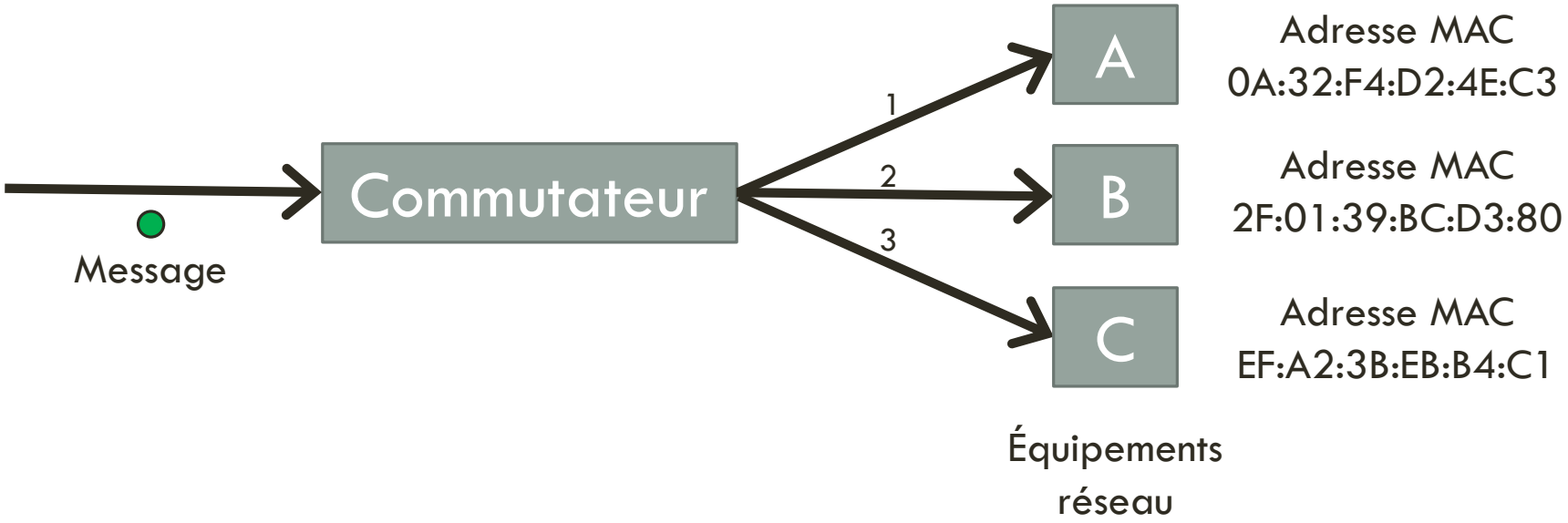


2. COUCHE LIAISON

Types d'équipements

✓ Le **commutateur ou *'switch'* :**

Examine l'adresse MAC contenue dans la couche 2 d'un message pour déterminer vers quel port diriger le message.

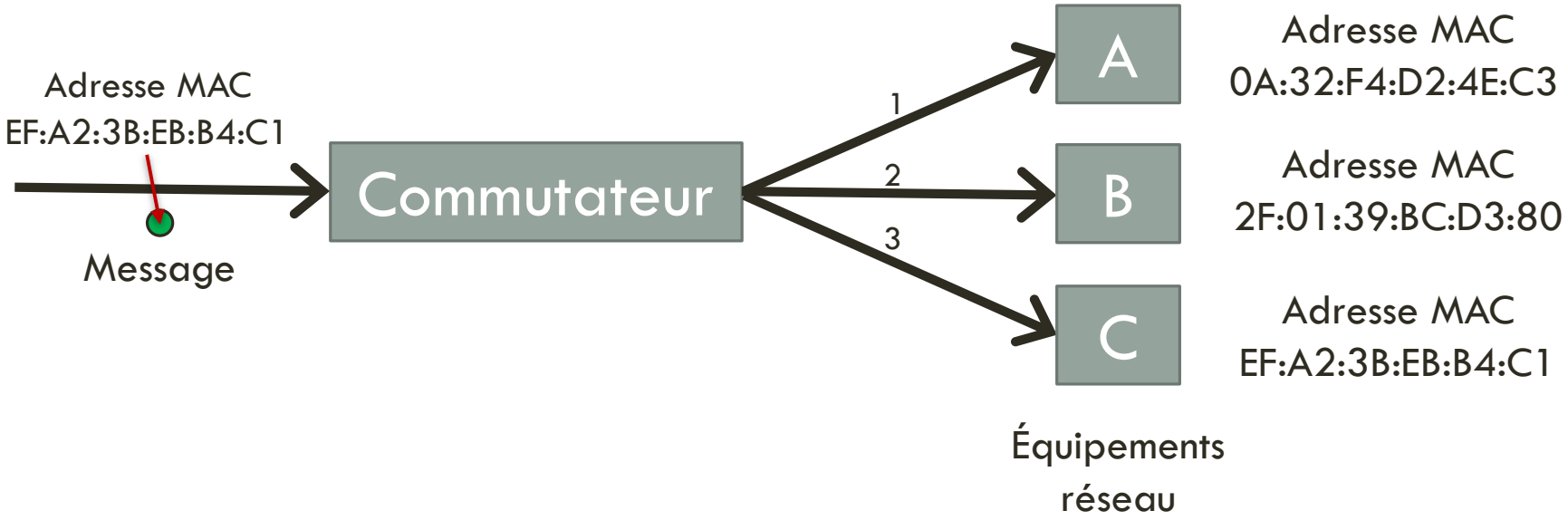


2. COUCHE LIAISON

Types d'équipements

✓ Le **commutateur ou *'switch'* :**

Examine l'adresse MAC contenue dans la couche 2 d'un message pour déterminer vers quel port diriger le message.

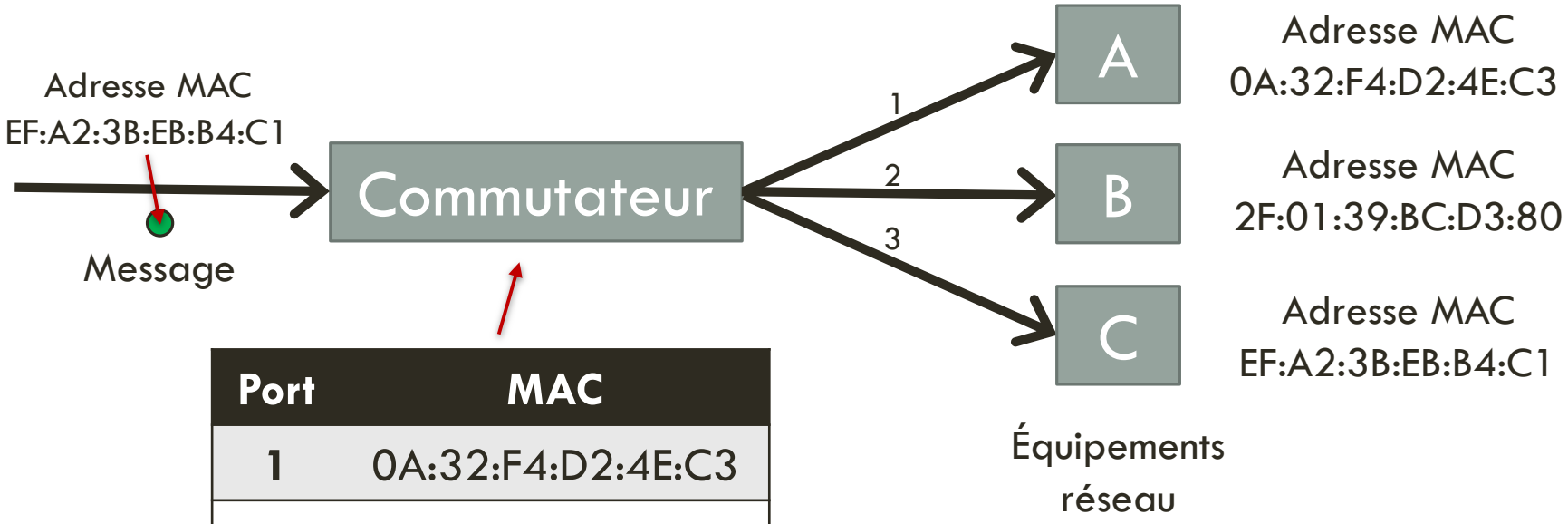


2. COUCHE LIAISON

Types d'équipements

✓ Le **commutateur ou *'switch'*** :

Examine l'adresse MAC contenue dans la couche 2 d'un message pour déterminer vers quel port diriger le message.



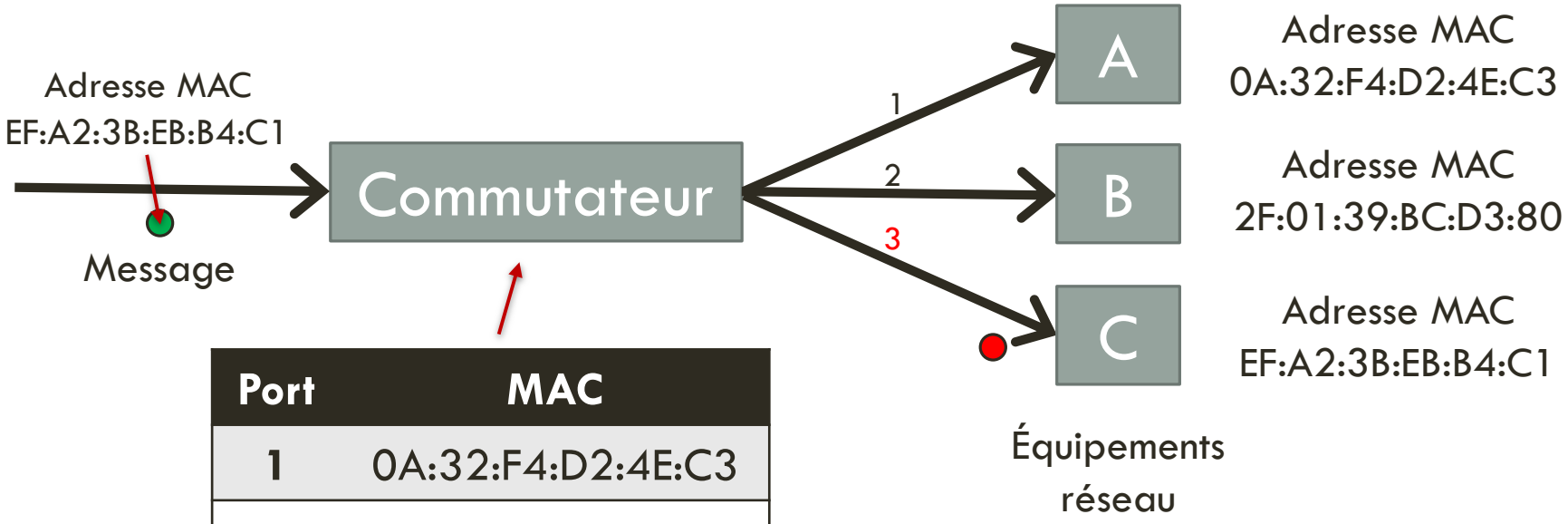
Port	MAC
1	0A:32:F4:D2:4E:C3
2	2F:01:39:BC:D3:80
3	EF:A2:3B:EB:B4:C1

2. COUCHE LIAISON

Types d'équipements

✓ Le **commutateur ou *'switch'*** :

Examine l'adresse MAC contenue dans la couche 2 d'un message pour déterminer vers quel port diriger le message.



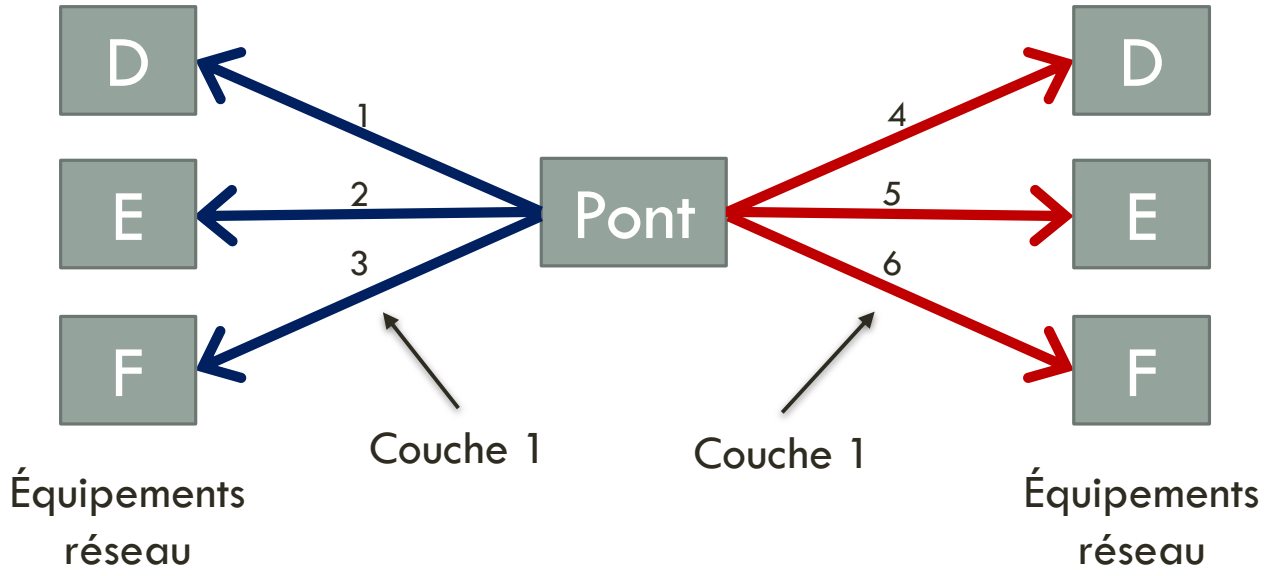
Port	MAC
1	0A:32:F4:D2:4E:C3
2	2F:01:39:BC:D3:80
3	EF:A2:3B:EB:B4:C1

2. COUCHE LIAISON

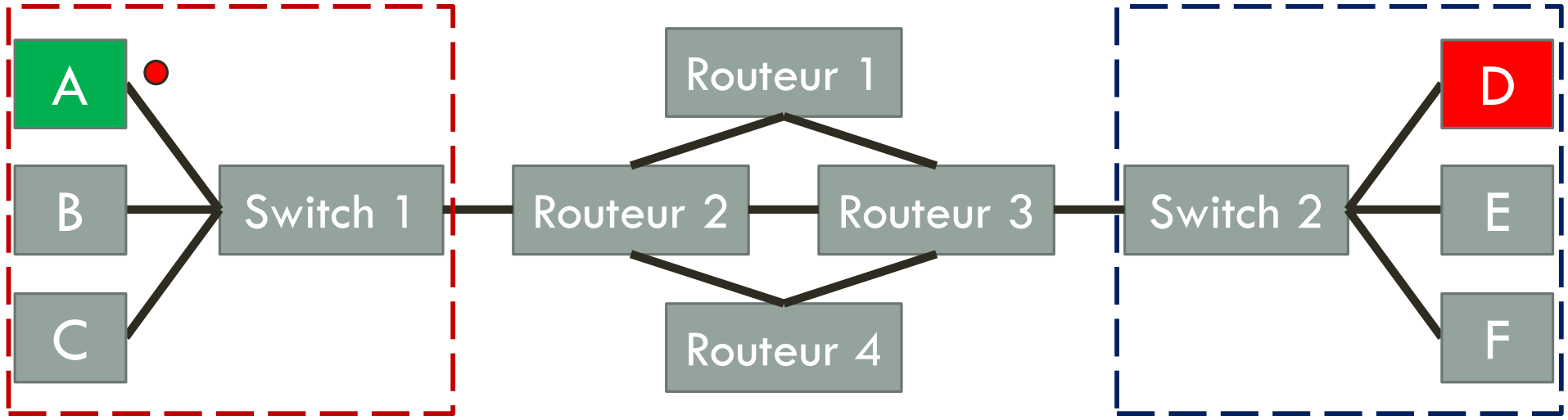
Types d'équipements (suite)

✓ Le **pont ou *'bridge'* :**

Effectue la connexion entre deux réseaux (éventuellement de couche 1 différente)



L'ENVOI D'UN PAQUET

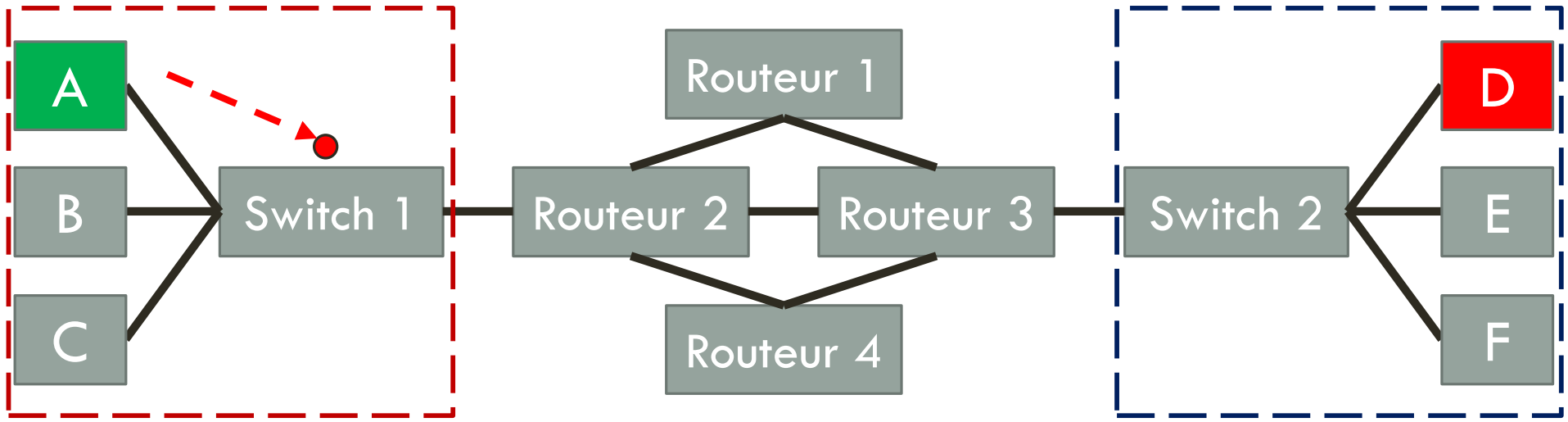


IP en
138.195.147.**xxx**

IP en
81.56.**yyy.zzz**

Comment A peut envoyer un message à D ?

L'ENVOI D'UN PAQUET



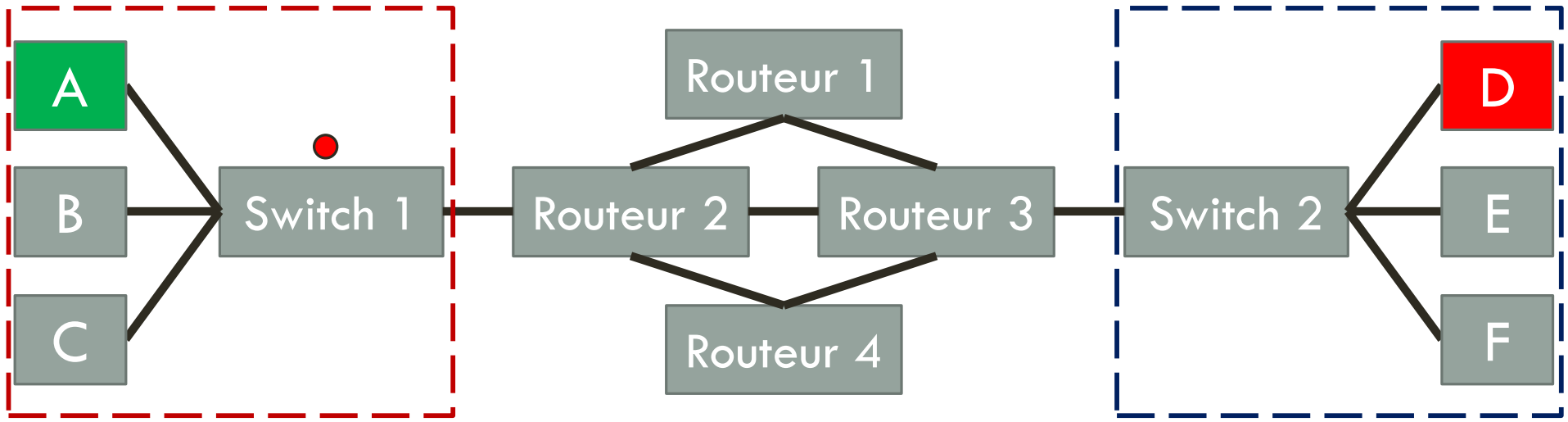
Comment A peut envoyer un message à D ?

1. **A** envoie le message à **Switch 1** :

   ➔ **Switch 1** connaît-il **D** ?

   ➔ Que fait **Switch 1** ?

L'ENVOI D'UN PAQUET



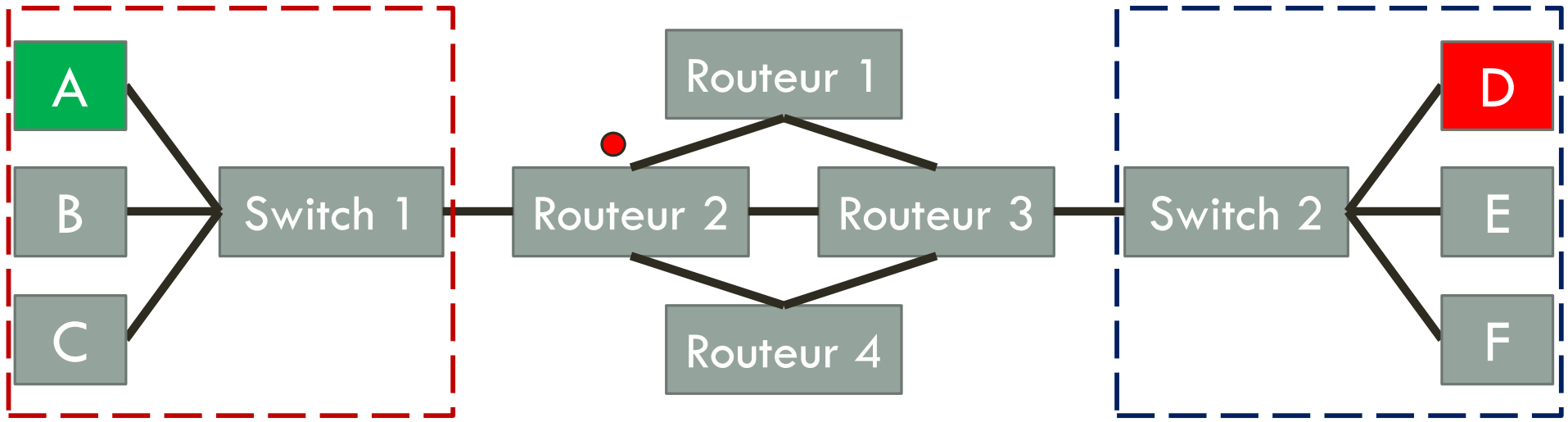
Comment A peut envoyer un message à D ?

1. **A** envoie le message à **Switch 1** :

   ➔ **Switch 1** connaît-il **D** ? NON, car **D** n'est **pas dans mon sous-réseau.**

   ➔ Que fait **Switch 1** ?

L'ENVOI D'UN PAQUET



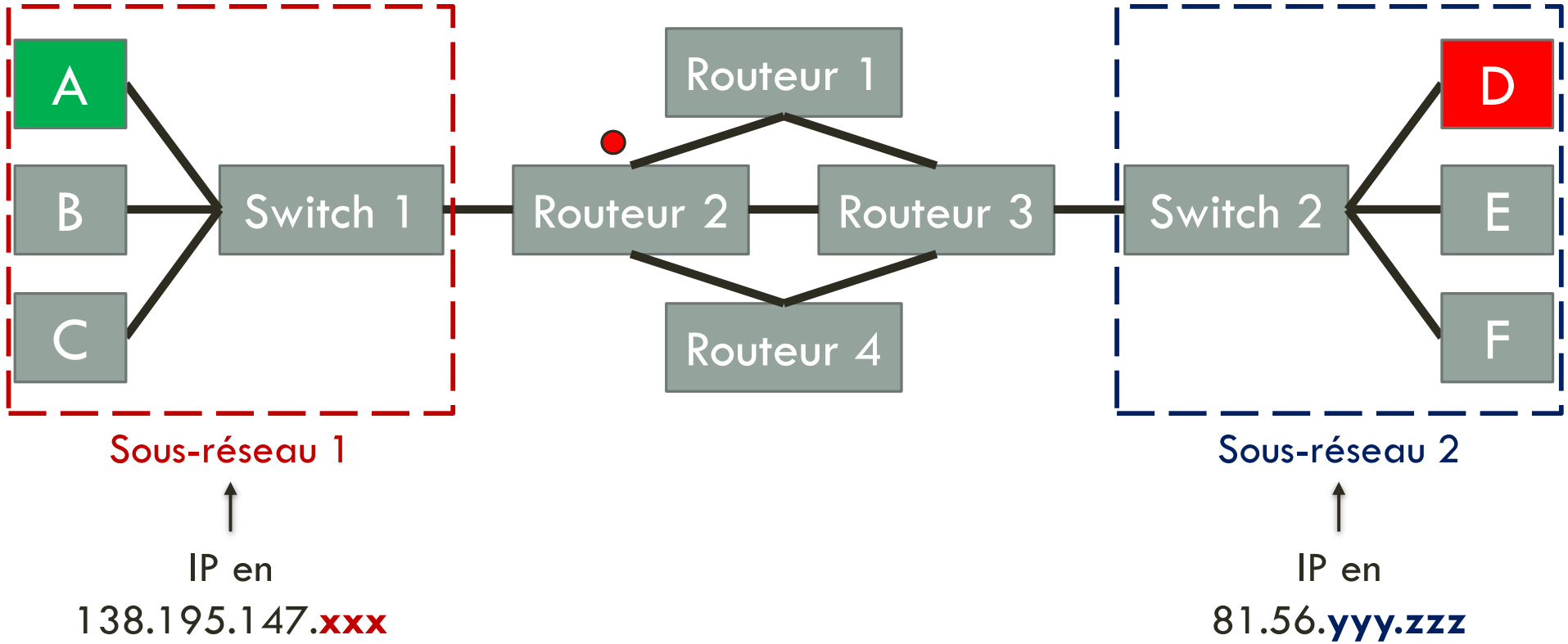
Comment A peut envoyer un message à D ?

1. **A** envoie le message à **Switch 1** :

   ➔ **Switch 1** connaît-il **D** ? NON, car **D** n'est **pas dans mon sous-réseau.**

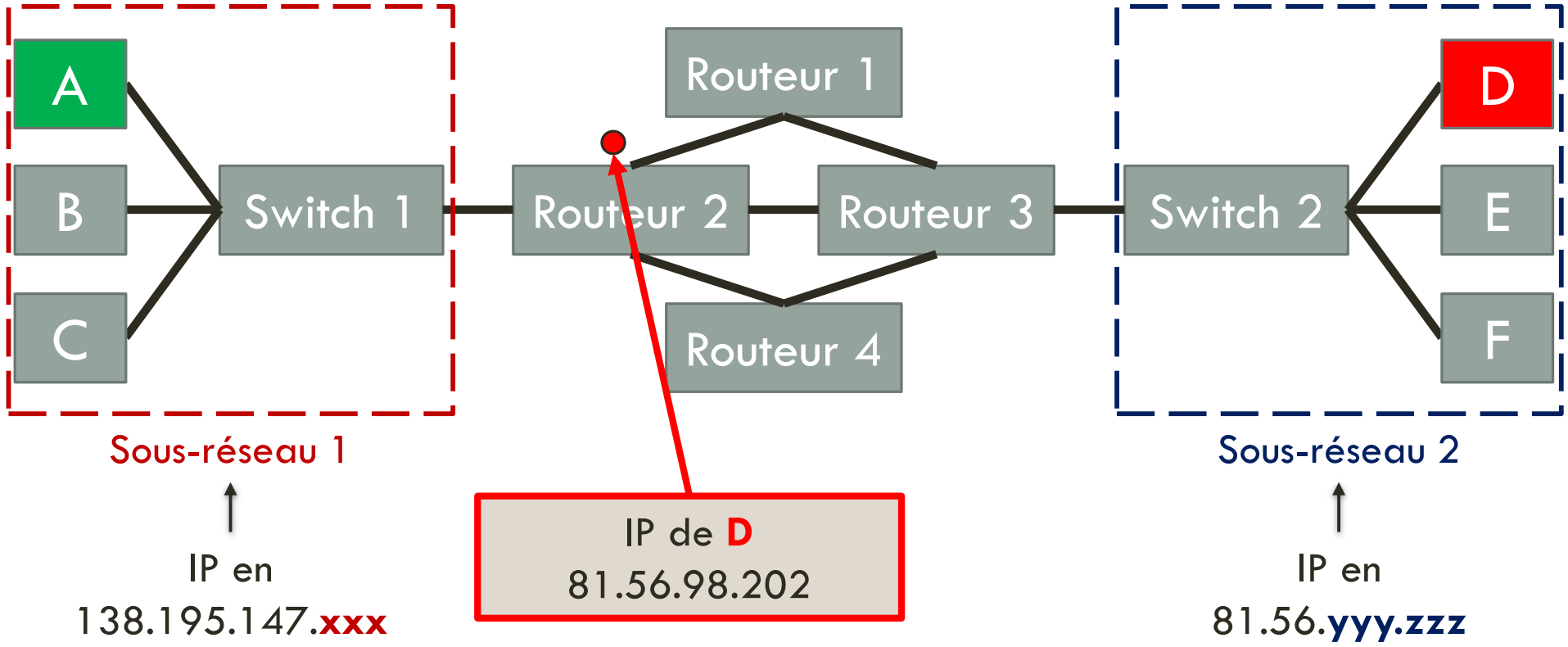
   ➔ Que fait **Switch 1** ? Il envoie le message à **Routeur 2**, il saura *peut-être* quoi faire

L'ENVOI D'UN PAQUET



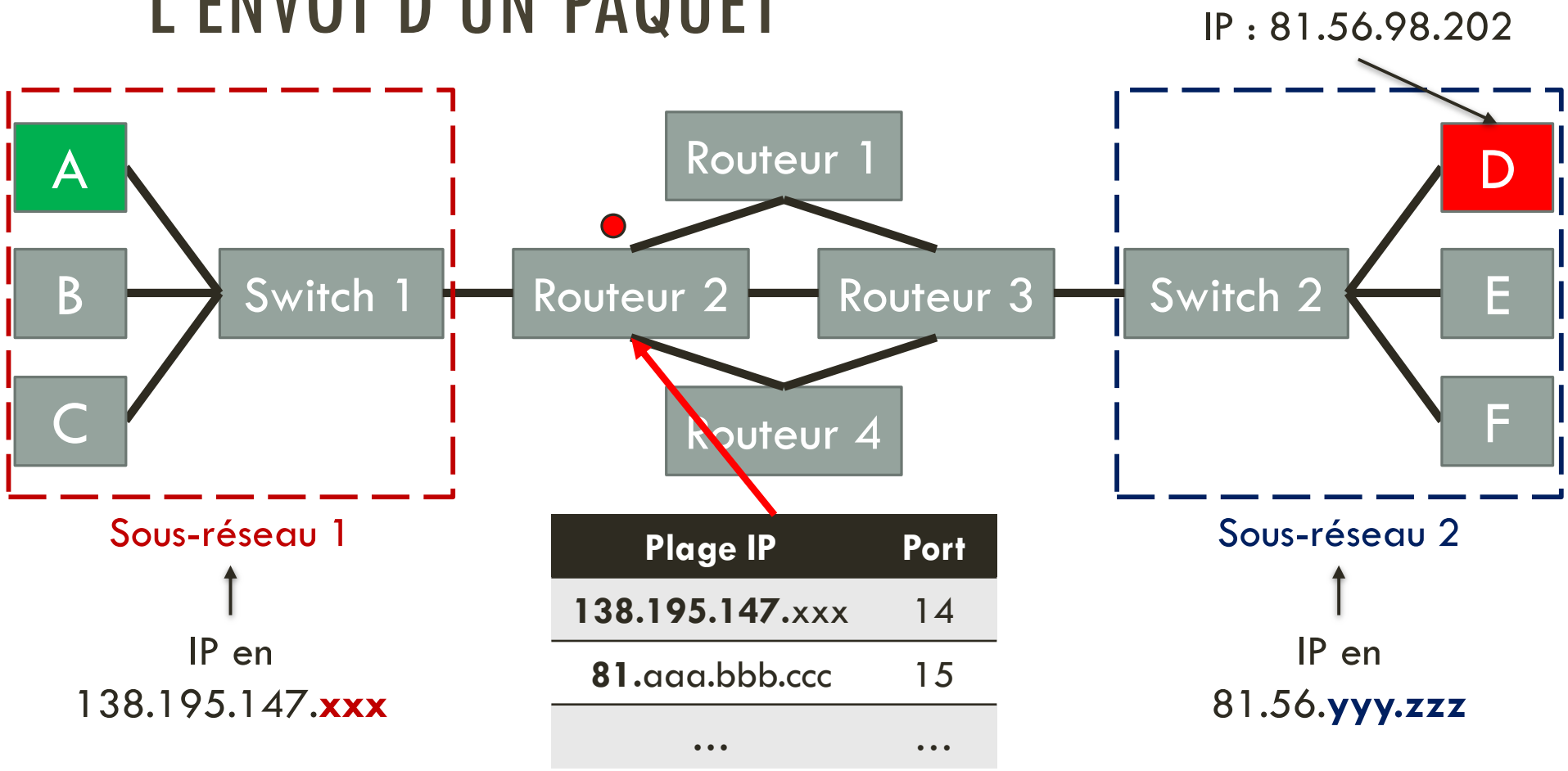
2. **Routeur 2** a reçu le message de **Switch 1** :
   - ➔ **Routeur 2** connaît-il **D** ?
   - ➔ Que fait **Routeur 2** ?

L'ENVOI D'UN PAQUET



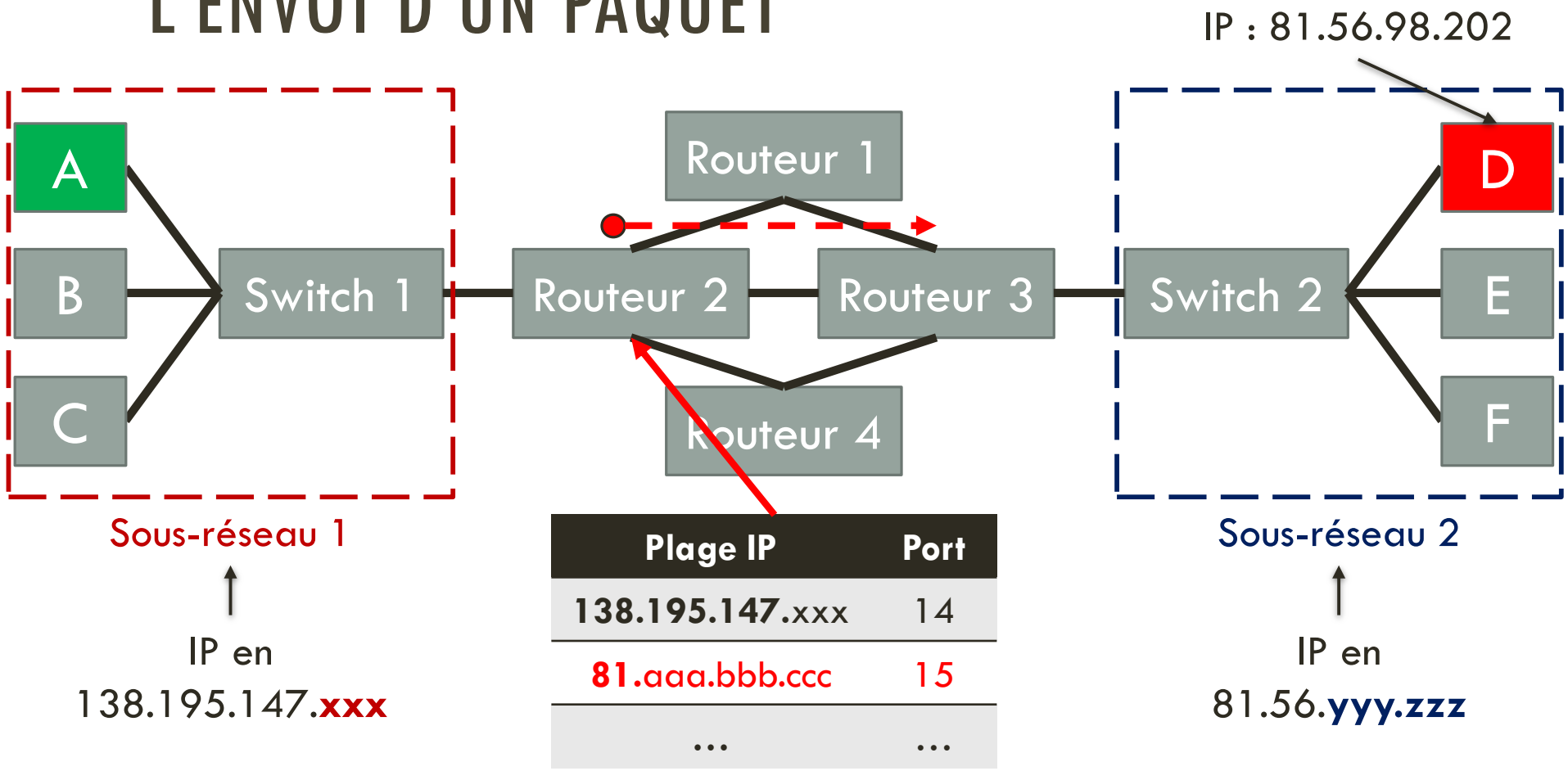
2. **Routeur 2** a reçu le message de **Switch 1** :
   - ➔ **Routeur 2** connaît-il **D** ? NON, mais il sait **lire son adresse IP** (couche 3)
   - ➔ Que fait **Routeur 2** ?

L'ENVOI D'UN PAQUET



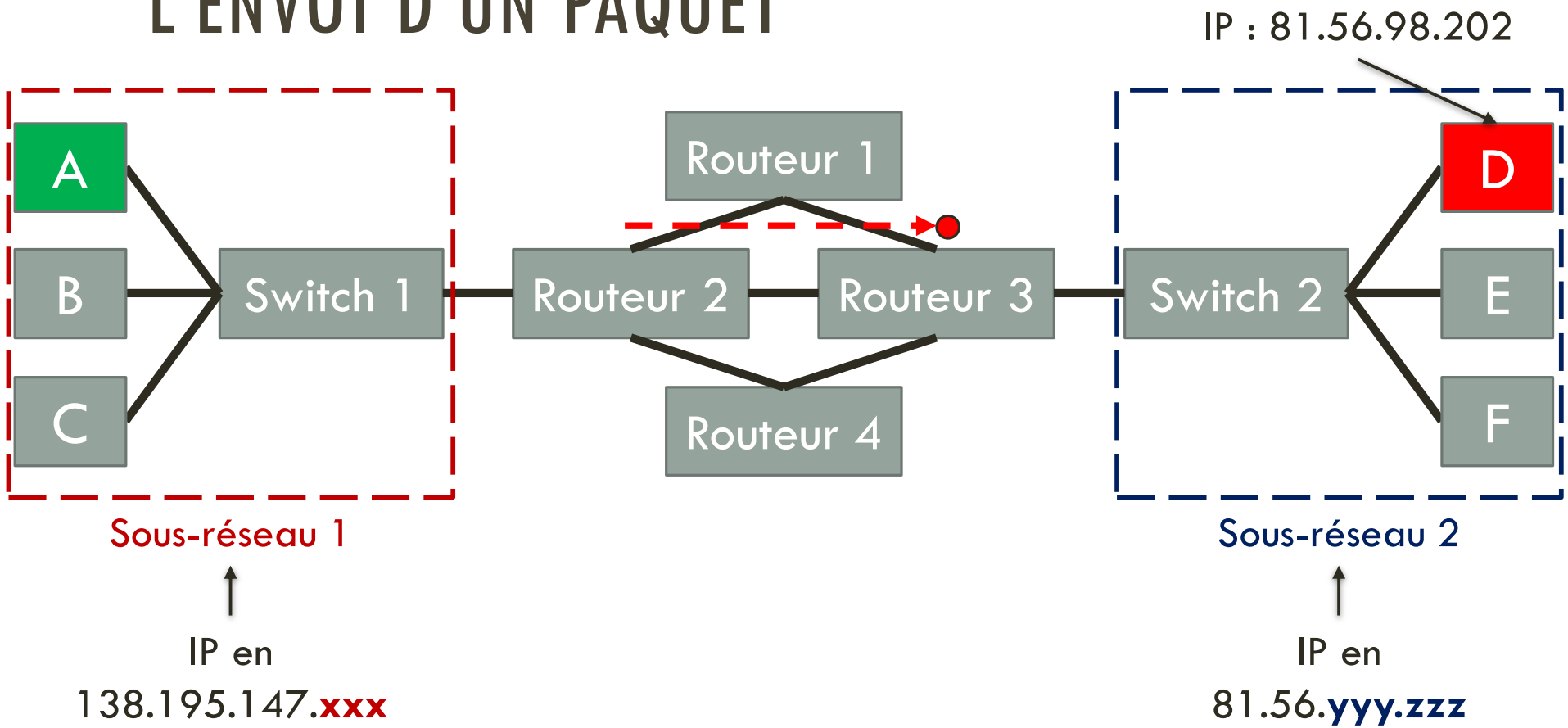
2. **Routeur 2** a reçu le message de **Switch 1** :
   - ➔ **Routeur 2** connaît-il **D** ? NON, mais il sait **lire son adresse IP** (couche 3)
   - ➔ Que fait **Routeur 2** ? Il regarde dans sa **table de routage** et envoie le message là où celui-ci vers la **meilleure route** (par exemple la plus rapide)

L'ENVOI D'UN PAQUET



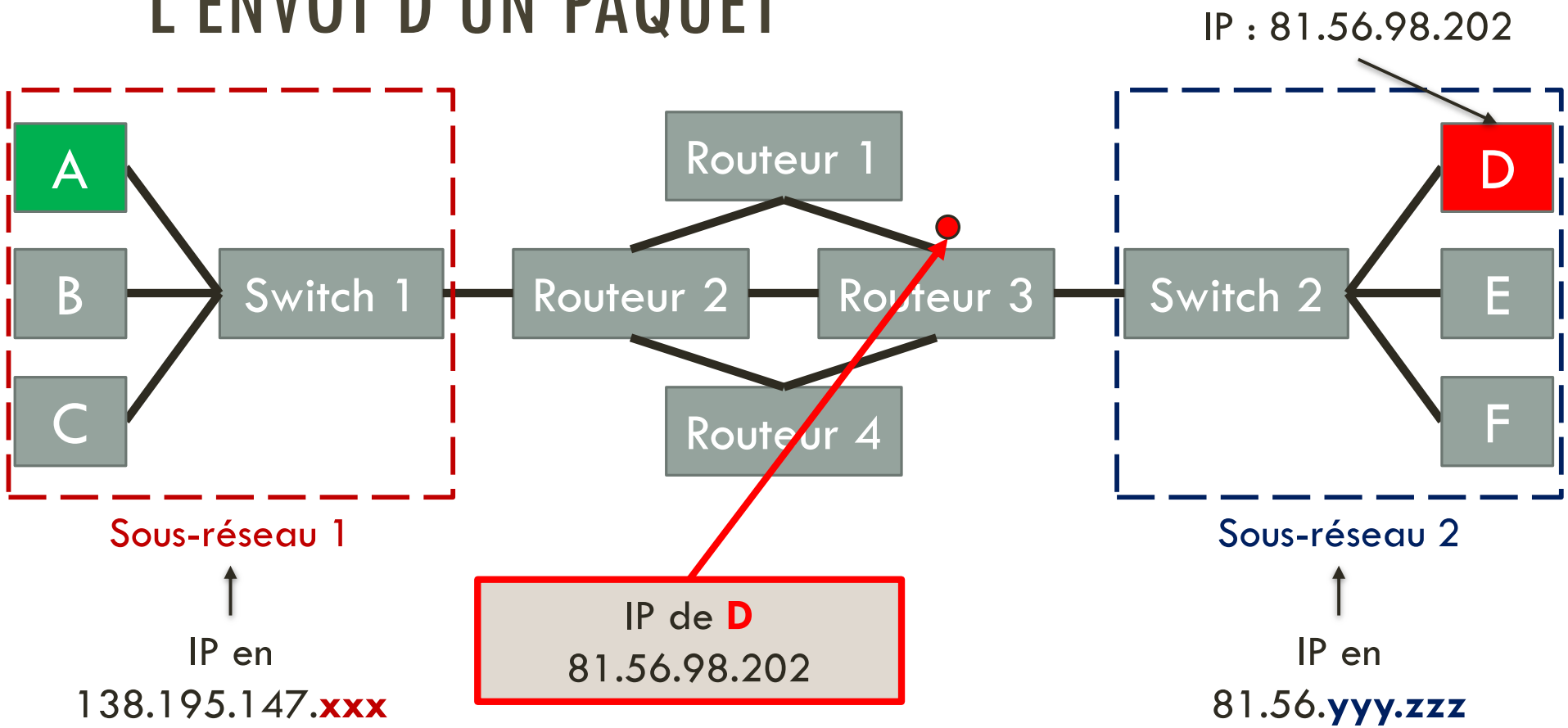
2. **Routeur 2** a reçu le message de **Switch 1** :
   - ➔ **Routeur 2** connaît-il **D** ? NON, mais il sait **lire son adresse IP** (couche 3)
   - ➔ Que fait **Routeur 2** ? Il regarde dans sa **table de routage** et envoie le message là où celui-ci vers la **meilleure route** (par exemple la plus rapide)
   - ➔ Il envoie le message sur son **Port 15**, ç'est-à-dire au **Routeur 3**

L'ENVOI D'UN PAQUET



3. **Routeur 3** a reçu le message de **Routeur 2** :
   - ➔ **Routeur 3** connaît-il **D** ?
   - ➔ Que fait **Routeur 3** ?

L'ENVOI D'UN PAQUET

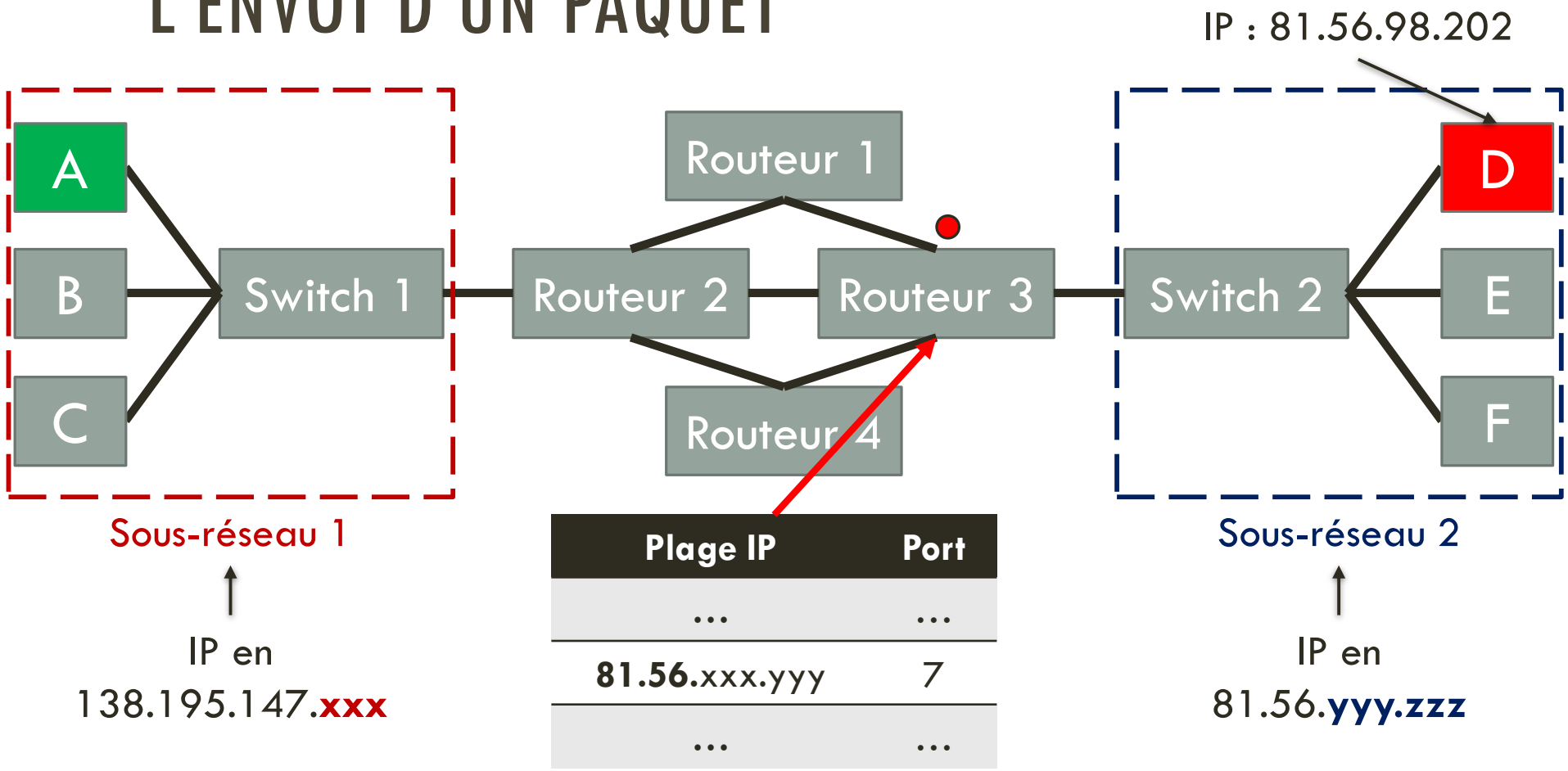


3. **Routeur 3** a reçu le message de **Routeur 2** :

   ➔ **Routeur 3** connaît-il **D** ? NON, mais il sait **lire son adresse IP** (couche 3)

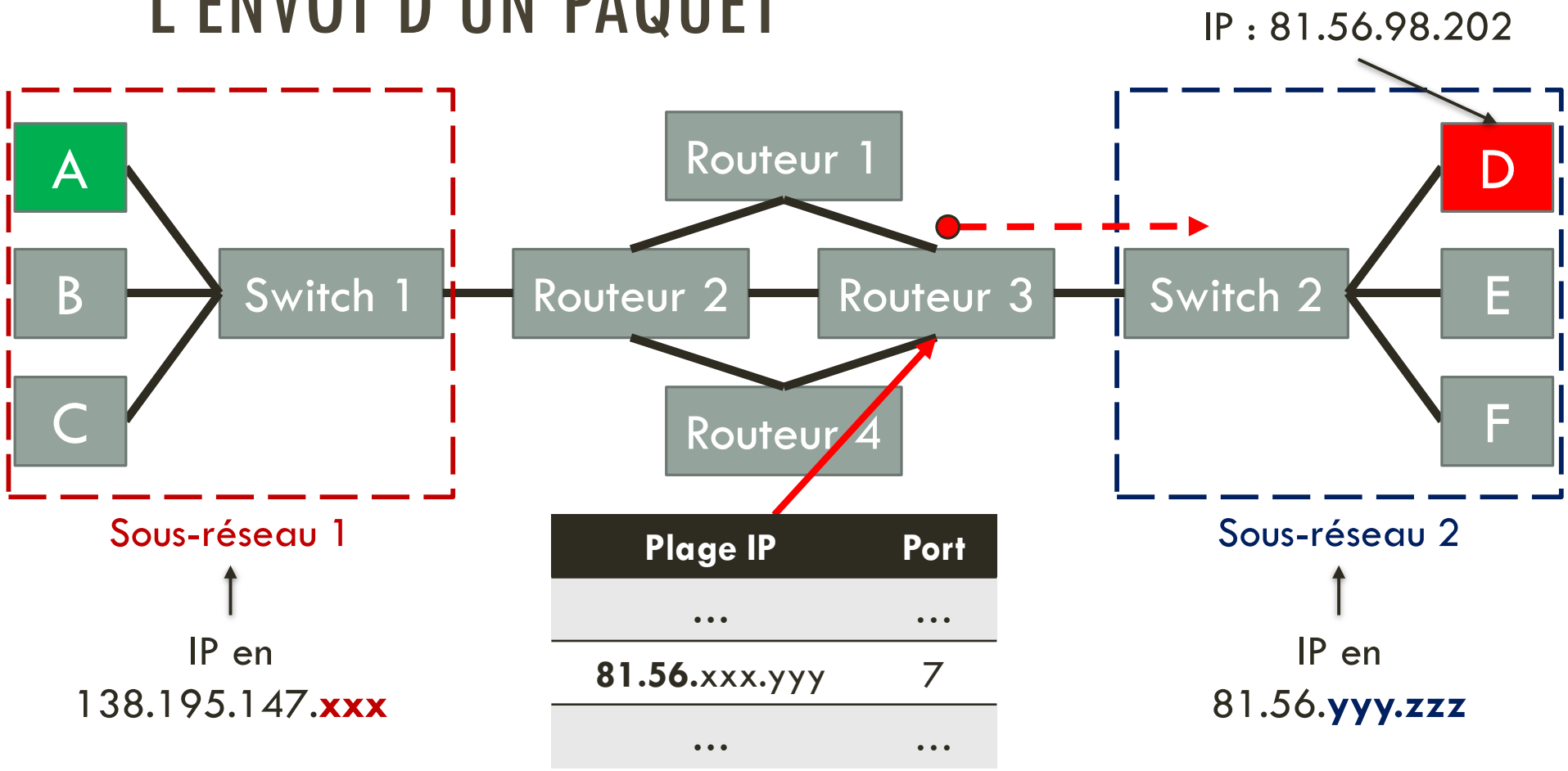
   ➔ Que fait **Routeur 3** ?

L'ENVOI D'UN PAQUET



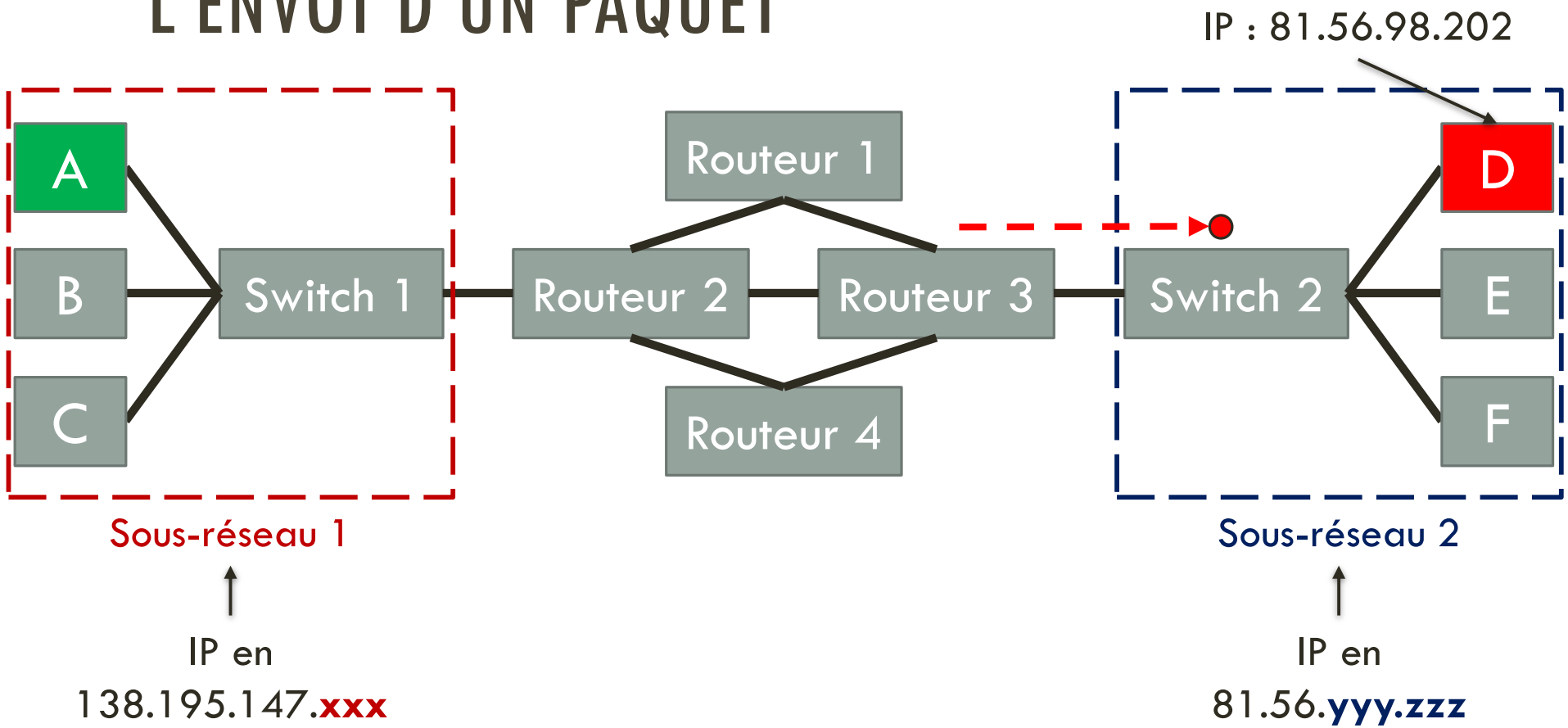
3. **Routeur 3** a reçu le message de **Routeur 2** :
   - ➔ **Routeur 3** connaît-il **D** ? NON, mais il sait **lire son adresse IP** (couche 3)
   - ➔ Que fait **Routeur 3** ? Il regarde dans sa **table de routage** et envoie le message là où celui-ci vers la **meilleure route** (par exemple la plus rapide)

L'ENVOI D'UN PAQUET



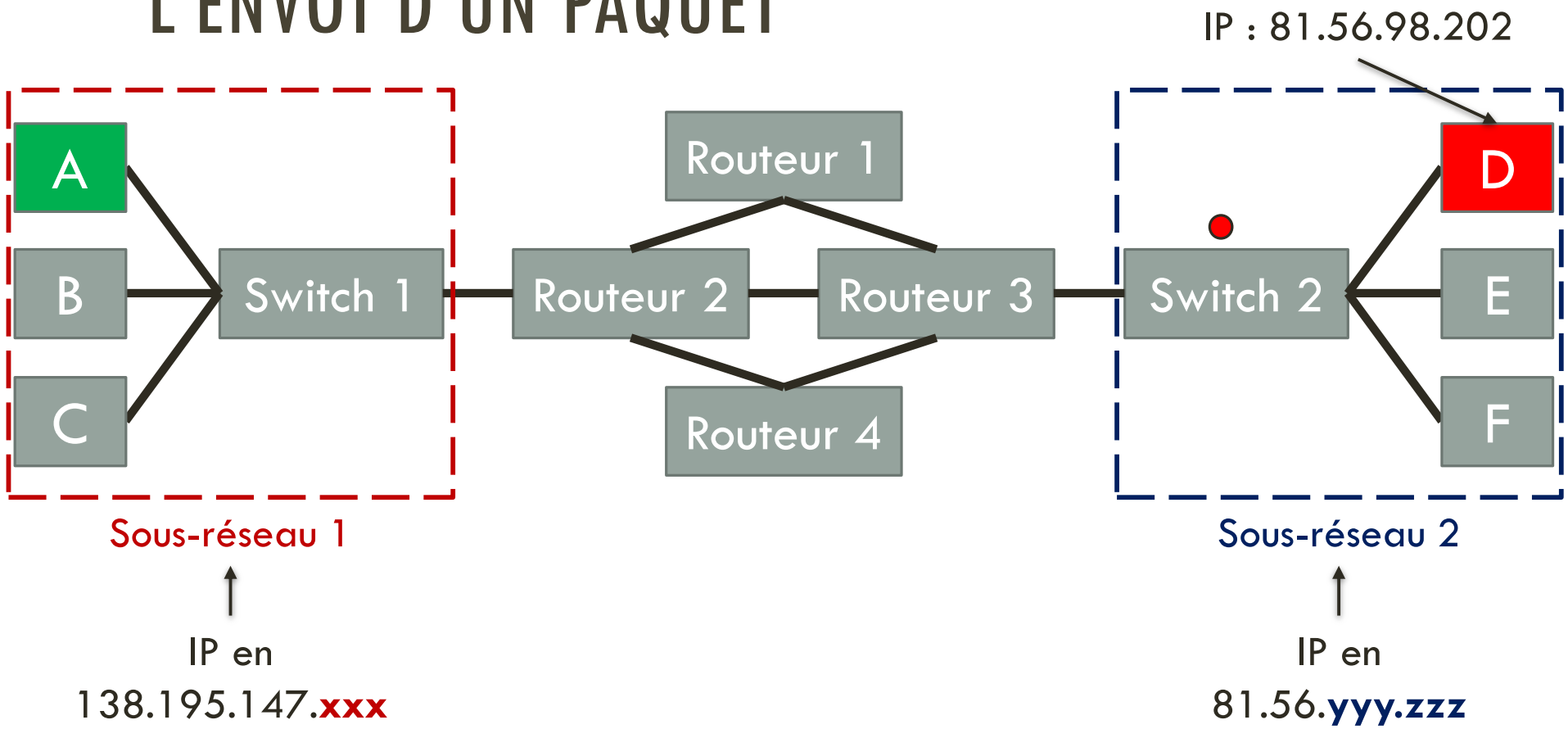
3. **Routeur 3** a reçu le message de **Routeur 2** :
   - ➔ **Routeur 3** connaît-il **D** ? NON, mais il sait **lire son adresse IP** (couche 3)
   - ➔ Que fait **Routeur 3** ? Il regarde dans sa **table de routage** et envoie le message là où celui-ci vers la **meilleure route** (par exemple la plus rapide)
   - ➔ Il envoie le message sur son **Port 7**, ç'est-à-dire au **Switch 2**

L'ENVOI D'UN PAQUET



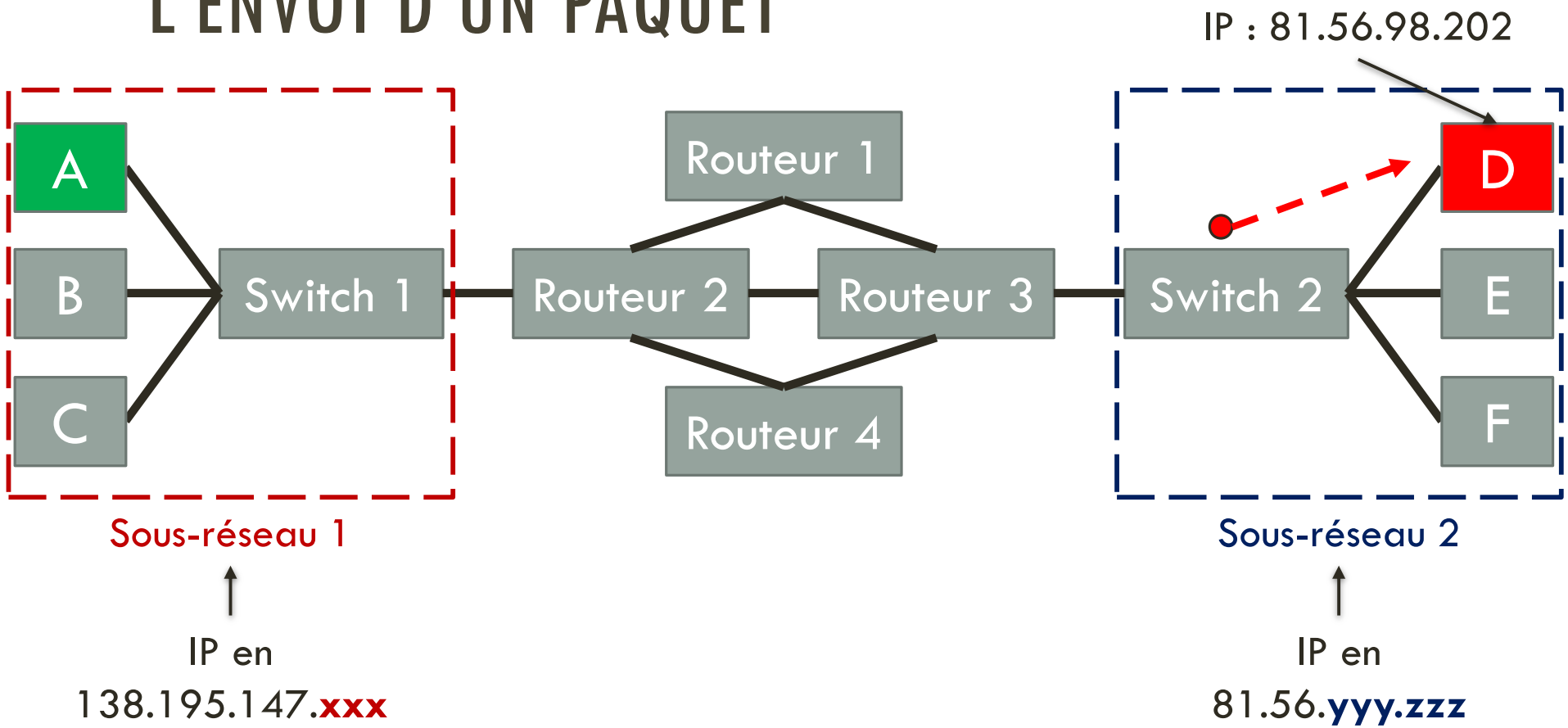
4. **Switch 2** a reçu le message de **Routeur 3** :
   - ➔ **Switch 2** connaît-il **D** ?
   - ➔ Que fait **Switch 2** ?

L'ENVOI D'UN PAQUET



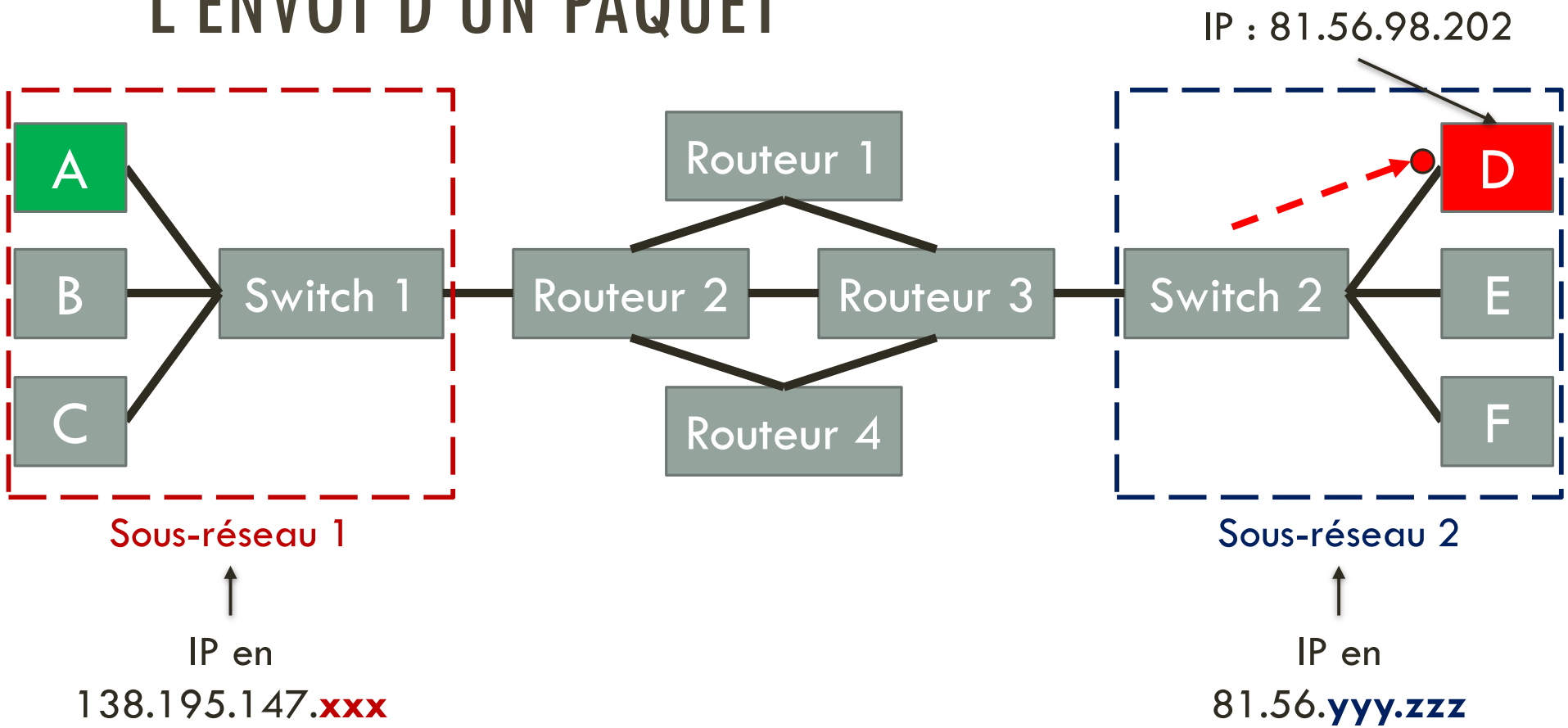
4. **Switch 2** a reçu le message de **Routeur 3** :
   - ➔ **Switch 2** connaît-il **D** ? OUI, car il voit dans la **couche 2** l'adresse MAC de **D**
   - ➔ Que fait **Switch 2** ?

L'ENVOI D'UN PAQUET



4. **Switch 2** a reçu le message de **Routeur 3** :
   - ➔ **Switch 2** connaît-il **D** ? OUI, car il voit dans la **couche 2** l'adresse MAC de **D**
   - ➔ Que fait **Switch 2** ? Il l'envoie à **D**, car **Switch 2** connait son adresse MAC et donc le port sur lequel envoyer le message

L'ENVOI D'UN PAQUET



4. **Switch 2** a reçu le message de **Routeur 3** :

➔ **Switch 2** connaît-il **D** ? OUI, car il voit dans la **couche 2** l'adresse MAC de **D**

➔ Que fait **Switch 2** ? Il l'envoie à **D**, car **Switch 2** connait son adresse MAC et donc le port sur lequel envoyer le message

HISTOIRE D'UNE PETITE REQUÊTE WEB – L'ENVOI FINAL ET LA RÉCEPTION

- ✓ Et voilà ! La requête de John est envoyée et reçue par le serveur de google !
- ✓ Mais comment tout cela transite, physiquement ?

1. COUCHE PHYSIQUE

Fonction

Convertir un message en un signal binaire transportable électriquement

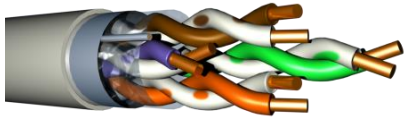
1. COUCHE PHYSIQUE

Fonction

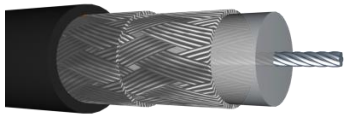
Convertir un message en un signal binaire transportable électriquement

Types d'équipements

✓ Pour transporter les signaux binaires :



- **Paire torsadée :** faibles distances, débit faible



- **Câble coaxial :** moyennes distances, bon débit, coûteux

- **Fibres optiques :** grandes distances, excellent débit, très coûteux

- **Ondes radios :** faible portée, perturbations , pratique

1. COUCHE PHYSIQUE

Types d'équipements (suite)

- ✓ Pour **garder la qualité** d'un signal binaire :

1. COUCHE PHYSIQUE

Types d'équipements (suite)

✓ Pour **garder la qualité** d'un signal binaire :

- **Répéteur :** retransmet passivement un message en amplifiant son signal

1. COUCHE PHYSIQUE

Types d'équipements (suite)

✓ Pour **garder la qualité** d'un signal binaire :

- **Répéteur :** retransmet passivement un message en amplifiant son signal

- **Concentrateur ou** '*hub*' **:** répéteur à plusieurs sorties

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉPAQUETAGE

- ✓ Couche 3 : Le serveur note l'adresse IP de l'envoyeur de paquet (donc du sous-réseau de John)

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉPAQUETAGE

- ✓ Couche 3 : Le serveur note l'adresse IP de l'envoyeur de paquet (donc du sous-réseau de John)
- ✓ Couche 4 : Les paquets sont réassemblés (protocole TCP). Si un paquet n'est pas reçu, il va être renvoyé par l'ordinateur de John !

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉPAQUETAGE

- ✓ Couche 3 : Le serveur note l'adresse IP de l'envoyeur de paquet (donc du sous-réseau de John)
- ✓ Couche 4 : Les paquets sont réassemblés (protocole TCP). Si un paquet n'est pas reçu, il va être renvoyé par l'ordinateur de John !
- ✓ Couche 5 : Tout est bon, le serveur accepte d'initier une session sécurisée ou non !

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉPAQUETAGE

- ✓ Couche 3 : Le serveur note l'adresse IP de l'envoyeur de paquet (donc du sous-réseau de John)
- ✓ Couche 4 : Les paquets sont réassemblés (protocole TCP). Si un paquet n'est pas reçu, il va être renvoyé par l'ordinateur de John !
- ✓ Couche 5 : Tout est bon, le serveur accepte d'initier une session sécurisée ou non !
- ✓ Couche 6 : Le serveur lit l'encodage de la requête.

HISTOIRE D'UNE PETITE REQUÊTE WEB – LE DÉPAQUETAGE

- ✓ Couche 3 : Le serveur note l'adresse IP de l'envoyeur de paquet (donc du sous-réseau de John)
- ✓ Couche 4 : Les paquets sont réassemblés (protocole TCP). Si un paquet n'est pas reçu, il va être renvoyé par l'ordinateur de John !
- ✓ Couche 5 : Tout est bon, le serveur accepte d'initier une session sécurisée ou non !
- ✓ Couche 6 : Le serveur lit l'encodage de la requête.
- ✓ Couche 7 : Le serveur comprend la requête HTTP, et il va envoyer une réponse, qui contient entre autre la page web demandée, celle de google.com ! Ouf!

III. AUTRES ÉQUIPEMENTS

1. Pare-Feu
2. Mandataire

1. PARE-FEU

Pare-feu ou *'firewall'* :

Équipement permettant de **protéger un réseau en bloquant certains accès** et en autorisant d'autres.

1. PARE-FEU

Pare-feu ou *'firewall'* :

Équipement permettant de **protéger un réseau en bloquant certains accès** et en autorisant d'autres.

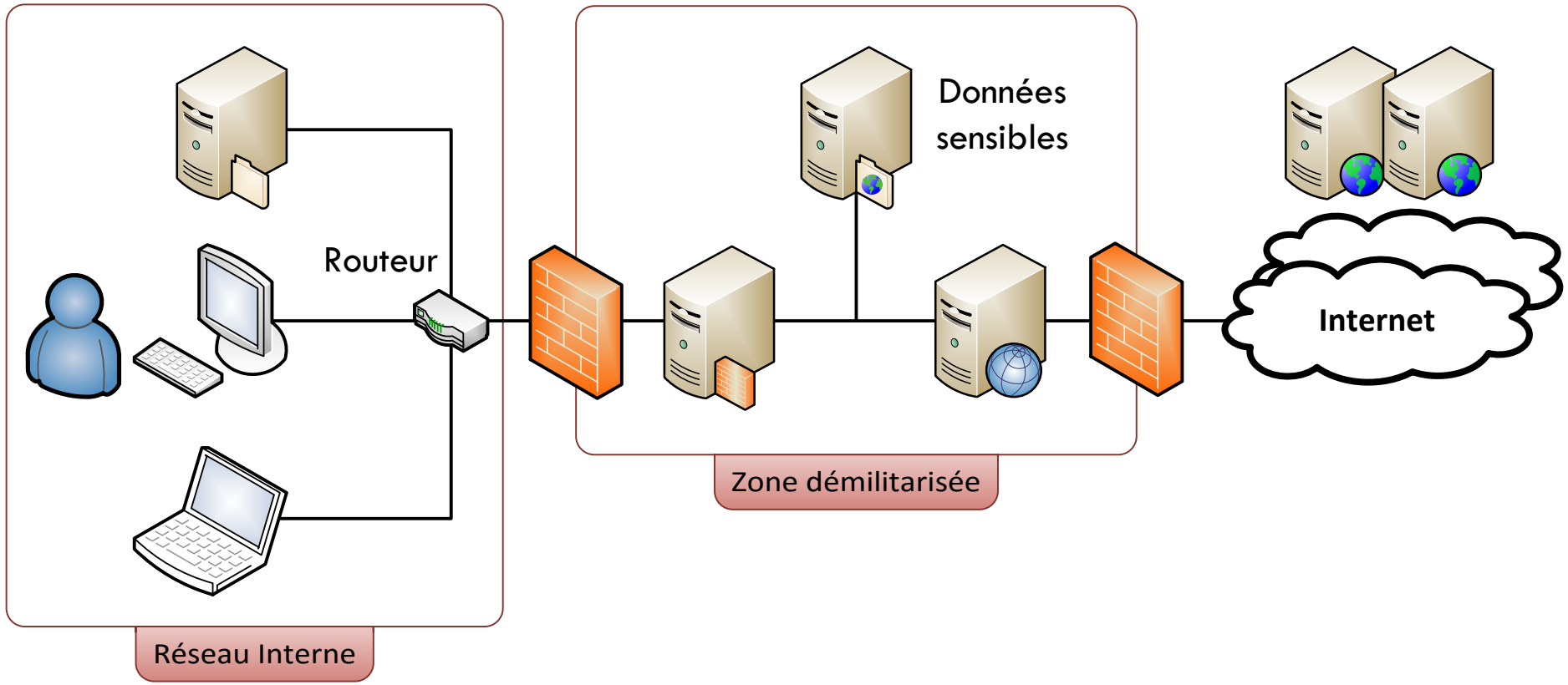
Il crée une **zone démilitarisée** qui fait l'interface entre Internet et le réseau interne.

1. PARE-FEU

Pare-feu ou *'firewall'* :

Équipement permettant de **protéger un réseau en bloquant certains accès** et en autorisant d'autres.

Il crée une **zone démilitarisée** qui fait l'interface entre Internet et le réseau interne.

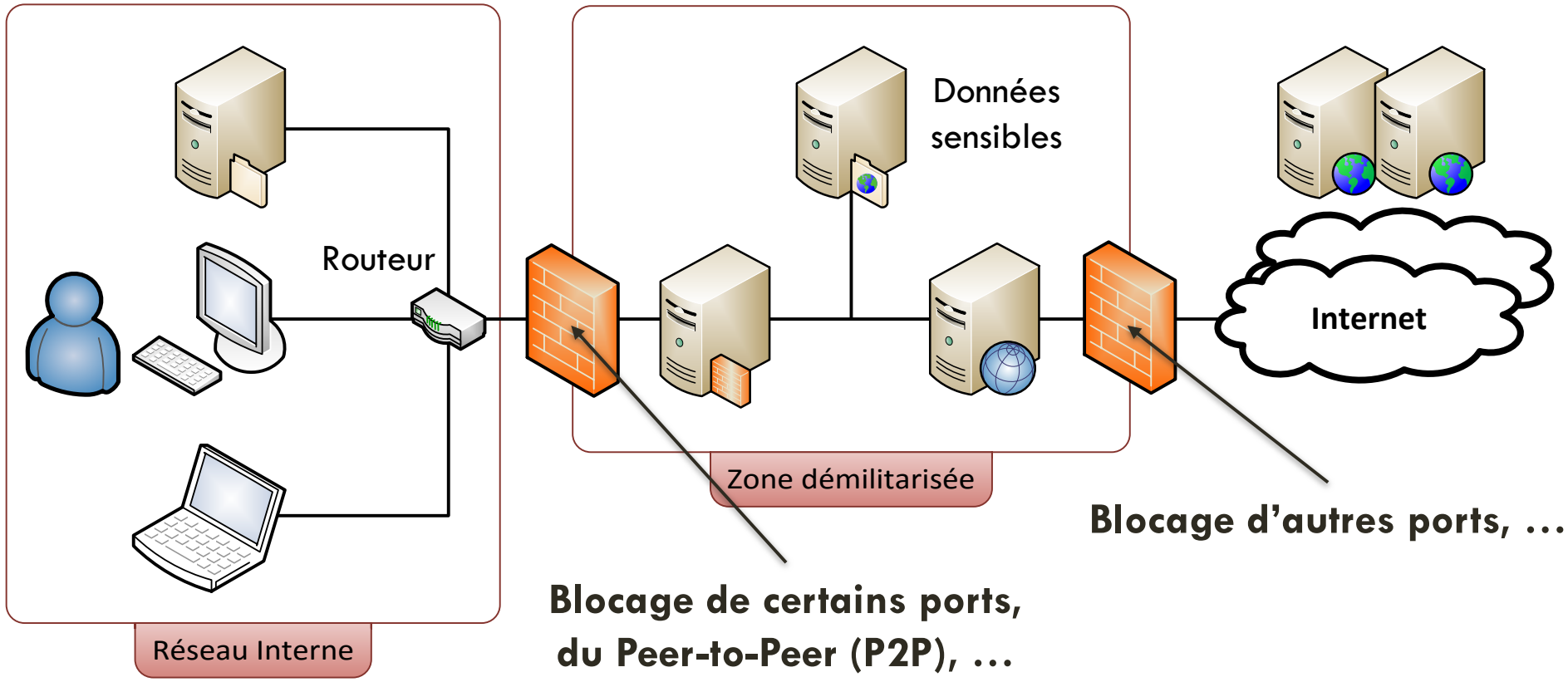


1. PARE-FEU

Pare-feu ou *'firewall'* :

Équipement permettant de **protéger un réseau en bloquant certains accès** et en autorisant d'autres.

Il crée une **zone démilitarisée** qui fait l'interface entre Internet et le réseau interne.



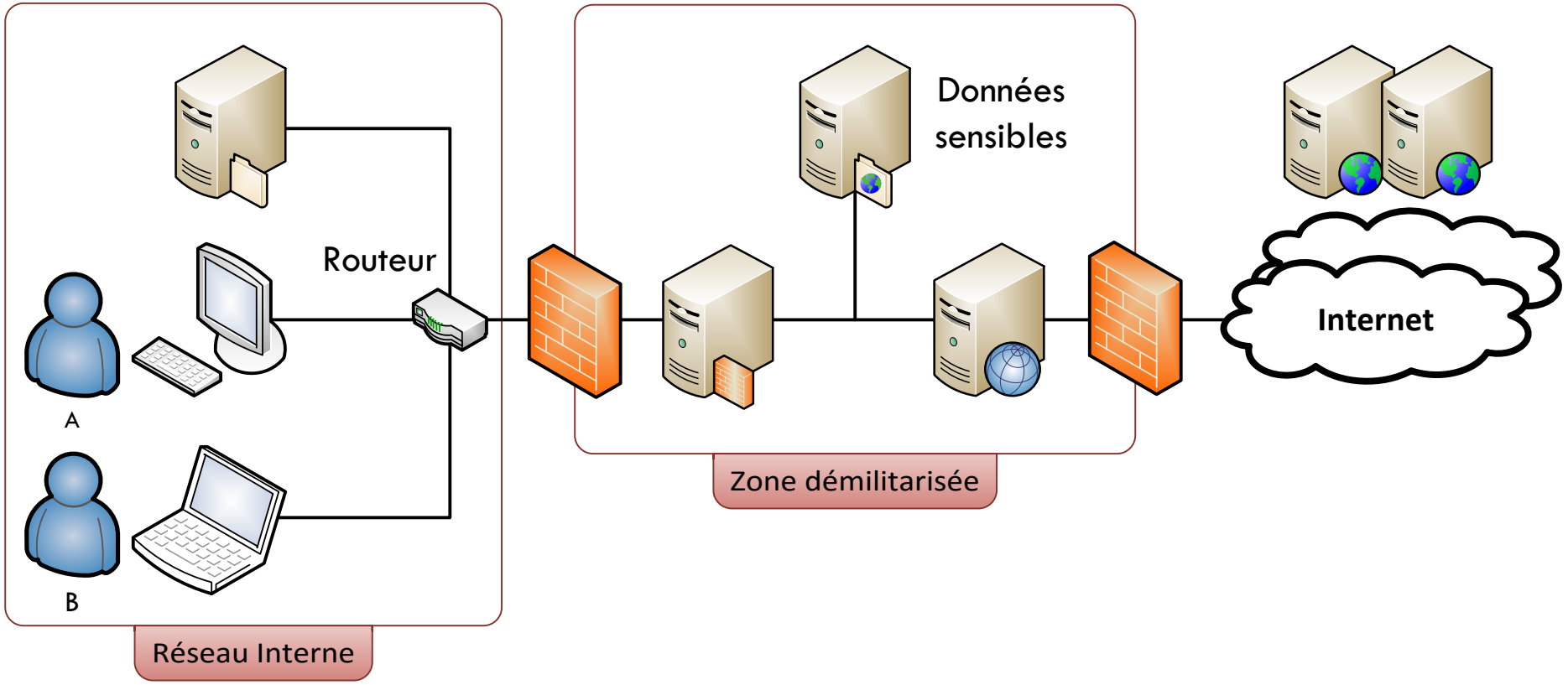
2. MANDATAIRE

Mandataire ou *'proxy'* :

Équipement servant **d'intermédiaire** pour les requêtes circulant sur un réseau.

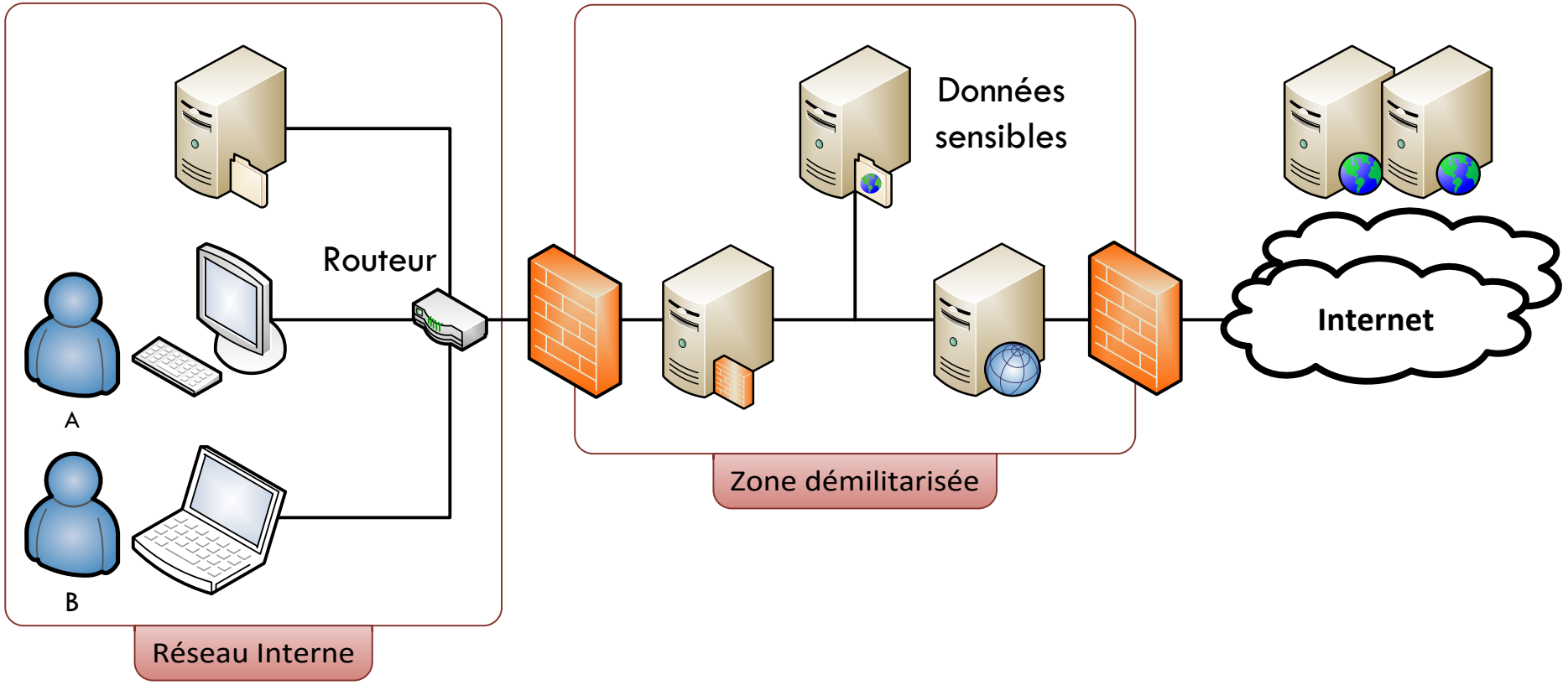
- ✓ **Préserve l'identité** des ordinateurs sur un réseau
- ✓ Accélère l'accès à certaines ressources en mettant **en cache** celles les plus utilisées

2. MANDATAIRE



A accède à une image à un instant t. Plus tard, B veut accéder à la **même image**.

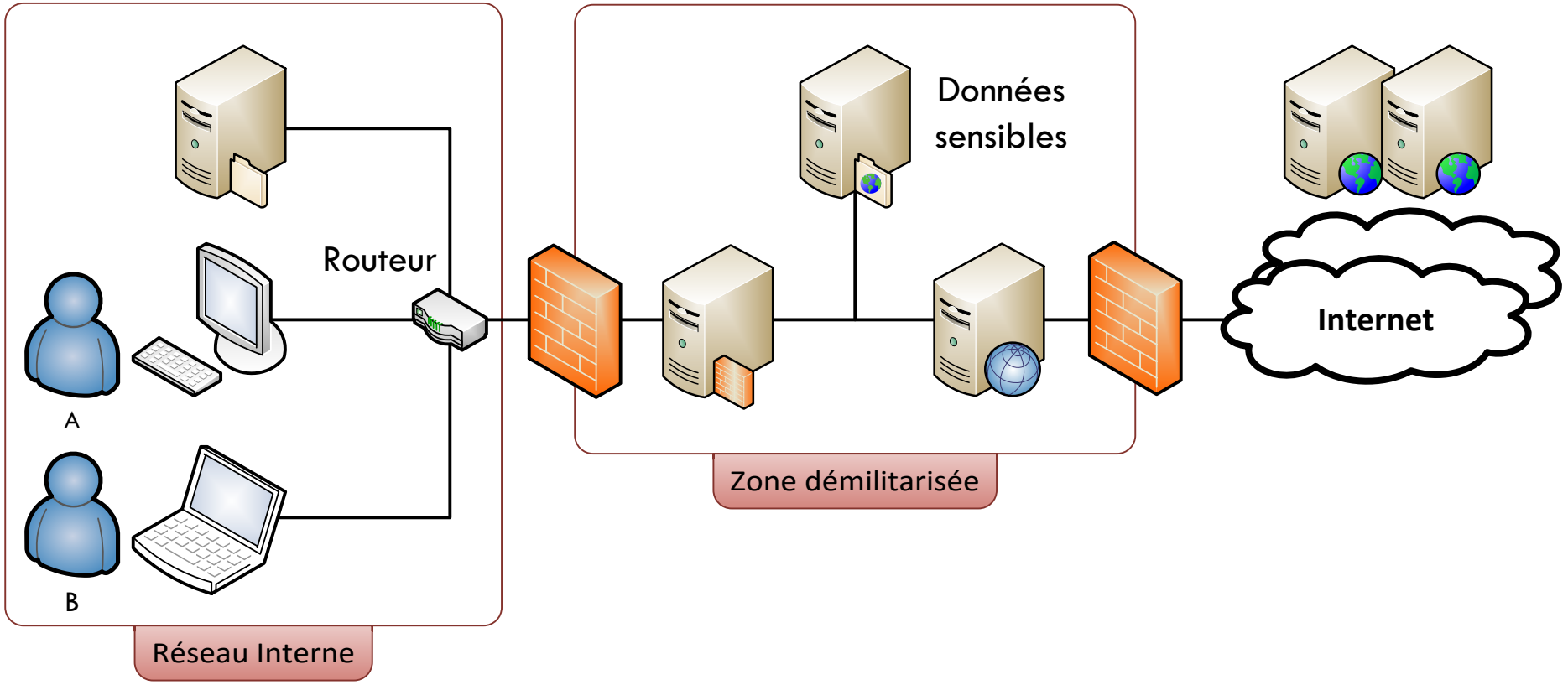
2. MANDATAIRE



A accède à une image à un instant t. Plus tard, B veut accéder à la **même image**.

Que fait le serveur mandataire ?

2. MANDATAIRE



A accède à une image à un instant t. Plus tard, B veut accéder à la **même image**.

Que fait le serveur mandataire ?

Il a gardé l'image dans une mémoire (**cache**) et peut donc envoyer directement une copie de cette image à B sans la récupérer sur Internet.

MERCI POUR VOTRE ATTENTION !

CENTRALE

RESEAUX

Télécom

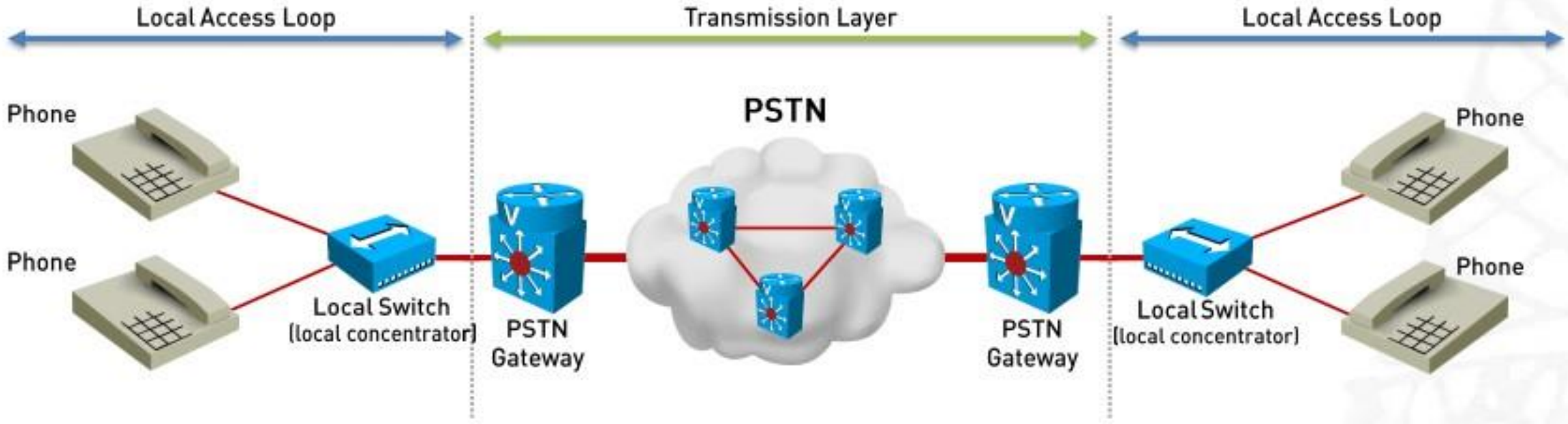
Par Edmond de Roffignac

D’après la fantastique œuvre
de Arnault Chazareix

Réseau de téléphonie fixe

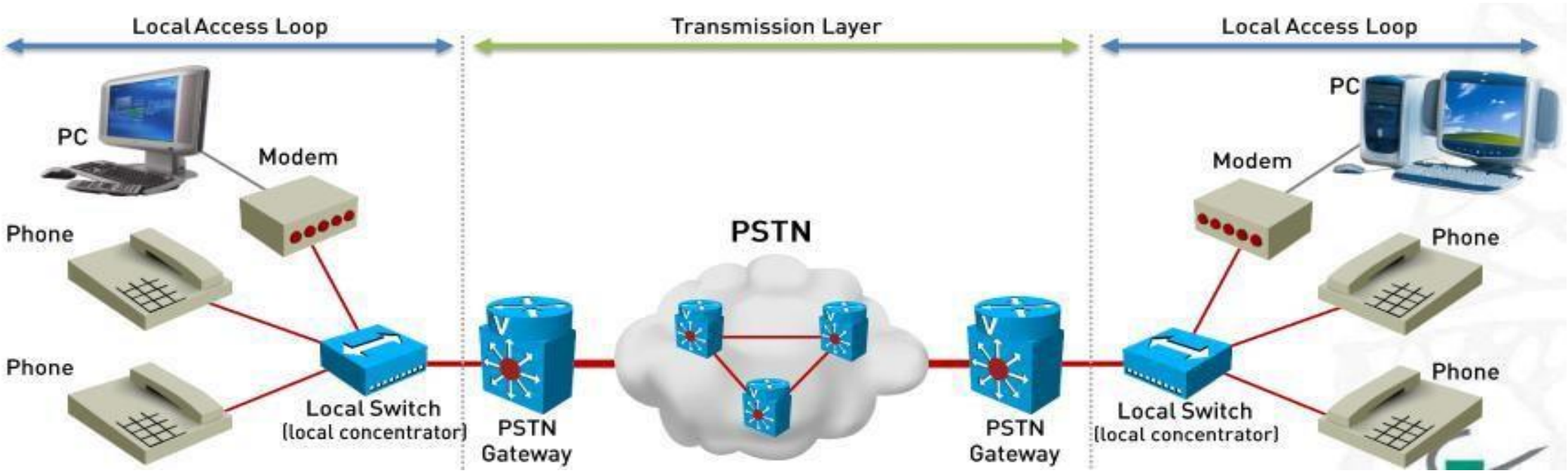
PSTN

- Public Switched Telephone Network
- RTC (réseau de Téléphone Commuté)
- 56kb/s
- Deux zones: boucles locales (analogique) et domaine de transmission (numérique)



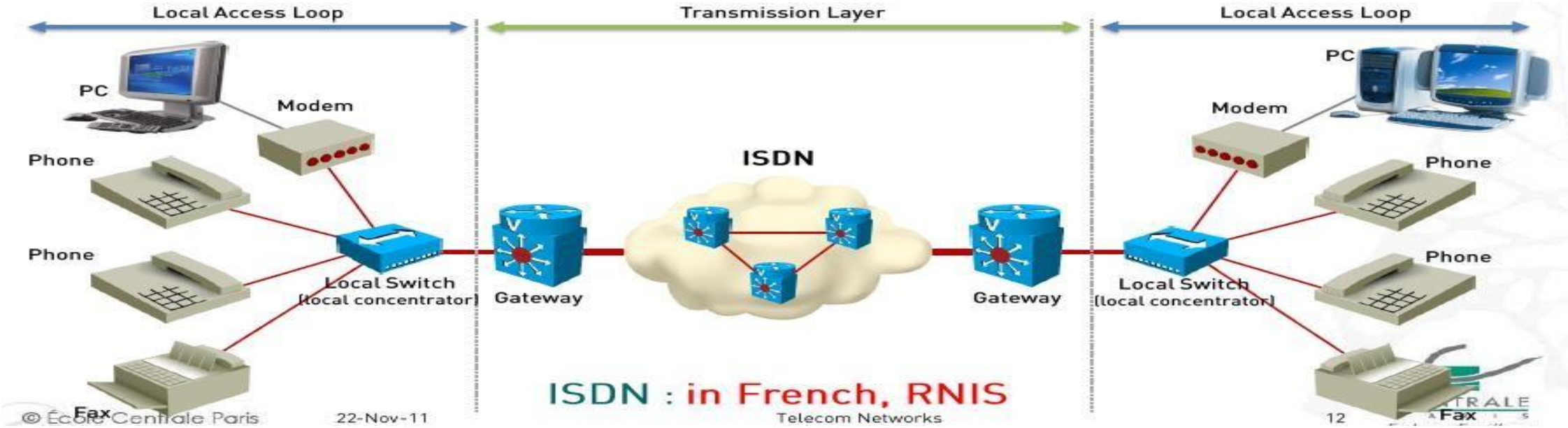
PSTN et Internet

- Possible avec l'utilisation d'un modem
- Conversion de l'info numérique de l'ordi en signal analogique



ISDN

- Integrated Services Digital Network
- Complètement numérique avec système de paquet
- RNIS (Réseau National à Intégretion de Service)
- Transmission de tout type de données en simultané (remplacé par ADSL)



Réseau mobile

Attention: ZE différence!!

- **Mobile:** Connection possible en déplacement. Mécanisme pour assurer la continuité du service (handover).

Ex: internet par téléphone(3G….)

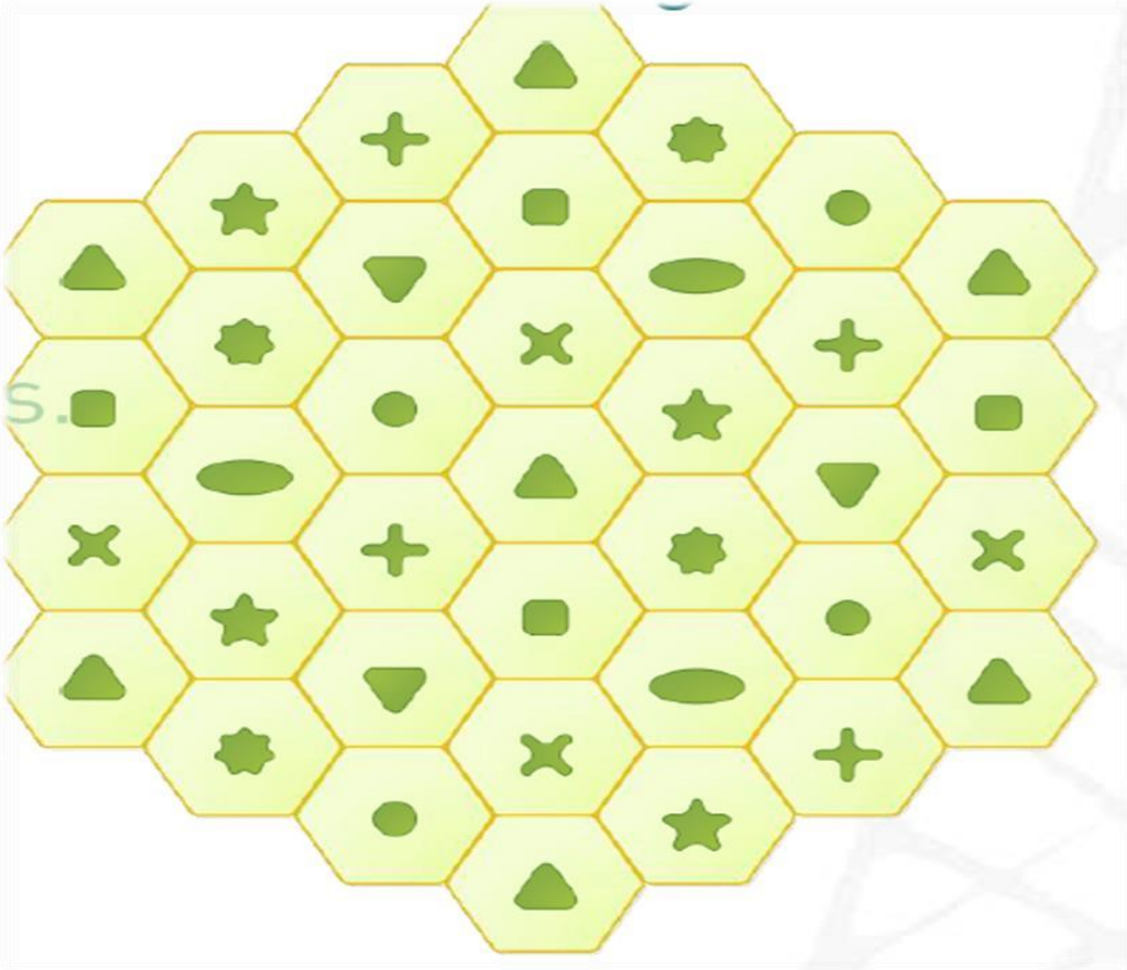
- **Nomade:** Connection VIA des terminaux fixes. La « continuité » est assurée par chaque terminal

Ex: Internet filaire, wifi

Structure

- Trois choses nécessaires pour le transport
  - Un signal radio (OSI 1)
  - Un format de données
  - Une structure de réseau

- Structure en cellules



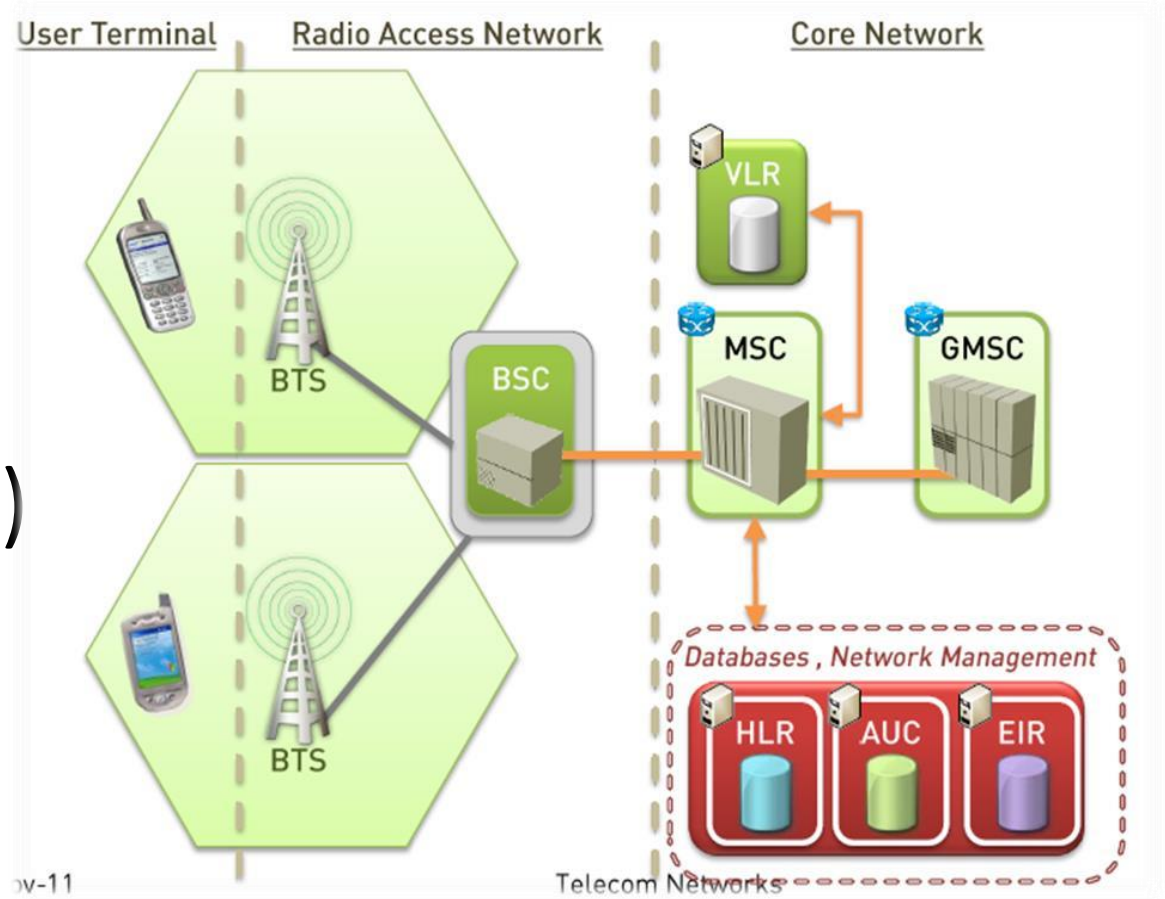
Cellule

- Region géographique dans laquelle tout point est atteint par l'antenne

  - La géométrie des cellules dépends de l'antenne
  - On utilise des fréquences différentes dans les cellules avoisinantes => éviter les interférences
  - Changement de cellule sans parte de communication (handover)

Le Réseau

- Trois parties:
  - Le terminal
  - Le RAN (Radio Access Network)
  - Le CN (Core Network)



Terminal

- 2 Identifiants:
  - **IMSI** (International Mobile Subscriber Identity)
    Carte SIM et abonnement
  - **IMEI** (International Mobile Equipement Identity)
    Téléphone

RAN

- Radio Access Network
- Assure le transfert du signal

  - **BTS** (Base Transceiver Station) ou plus communément appelée antennes

  - **BSC** (Base Station Controller) unité de contrôle d'un groupe de BTS

Core Network

- Interfaces avec le monde
  - **MSC** (Mobile Switching Center)

    lie les BSC entre elles
  - **GMSC** (Gateway MSC)

    lie le Core Network aux autres réseaux (PTSN,…)
  - **VLR** (Visitor Location Register)

    Copie partielle du HLR. Permet d'indentifier dans quelle cellule est l'utilisateur.

Core Network (suite)

- **HLR** (Home Location Register):
  Héberge les données des utilisateurs

- **AuC** (Authentification Center)
  Authentifie les utilisateurs. Fonctionne avec le HLR

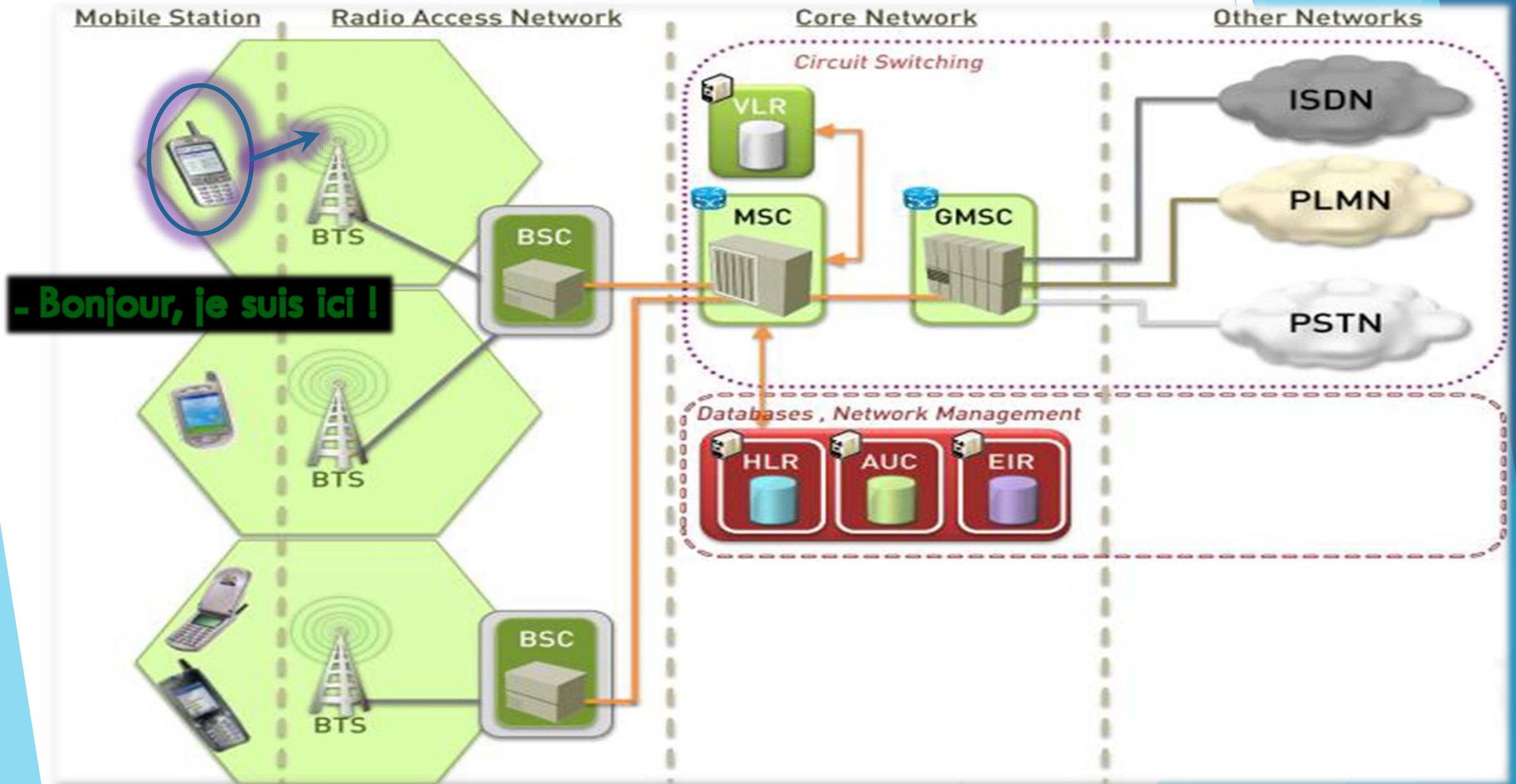
- **EIR** (Equipement Identity Register)
  Enregistre les IMEI des équipements interdits

GSM (2G)

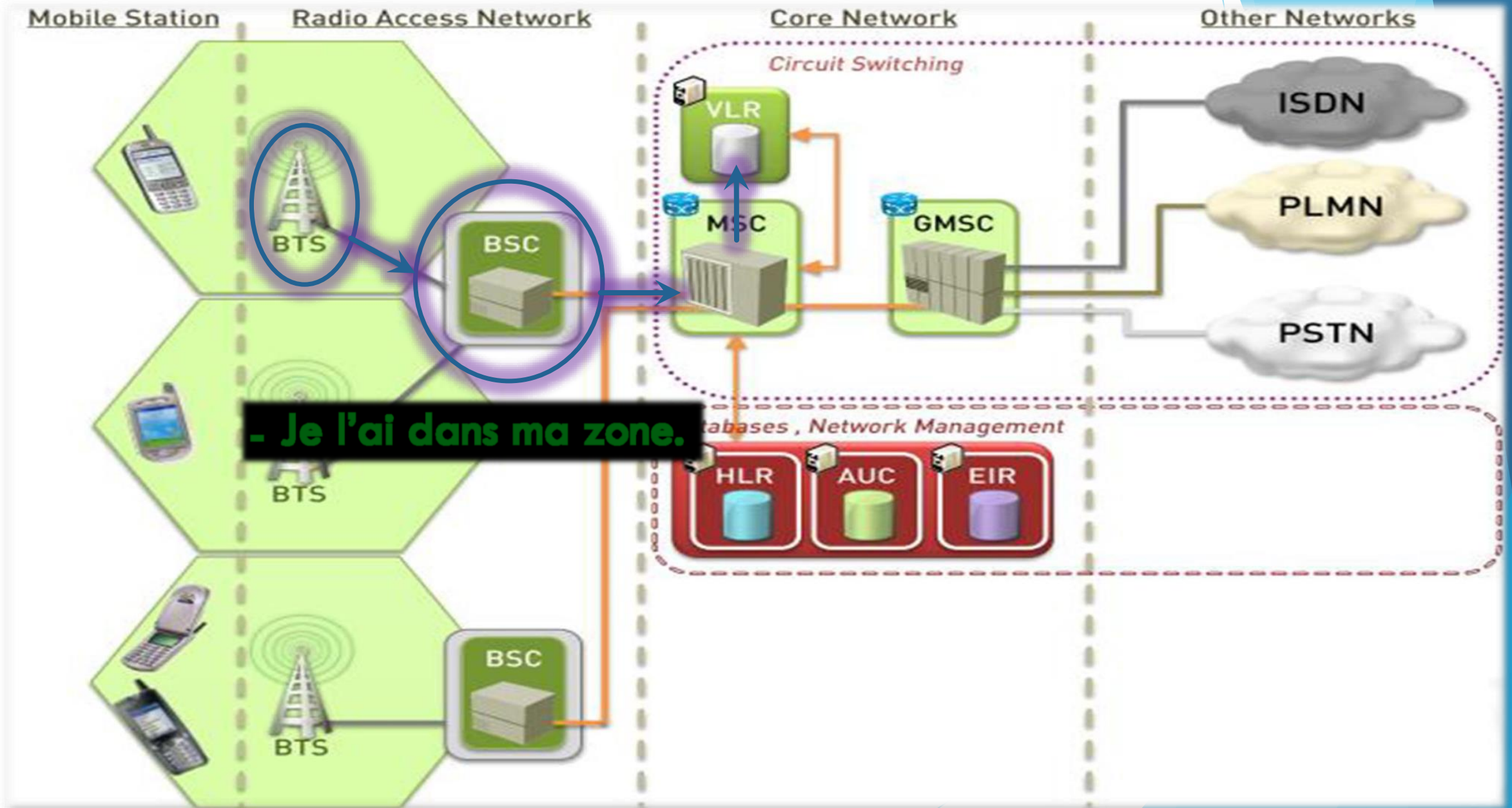
- Global System for Mobile communication
- Numérique
- Application : identification, voix, SMS
- 9,6kb/s

EX: Authentification sur le réseau

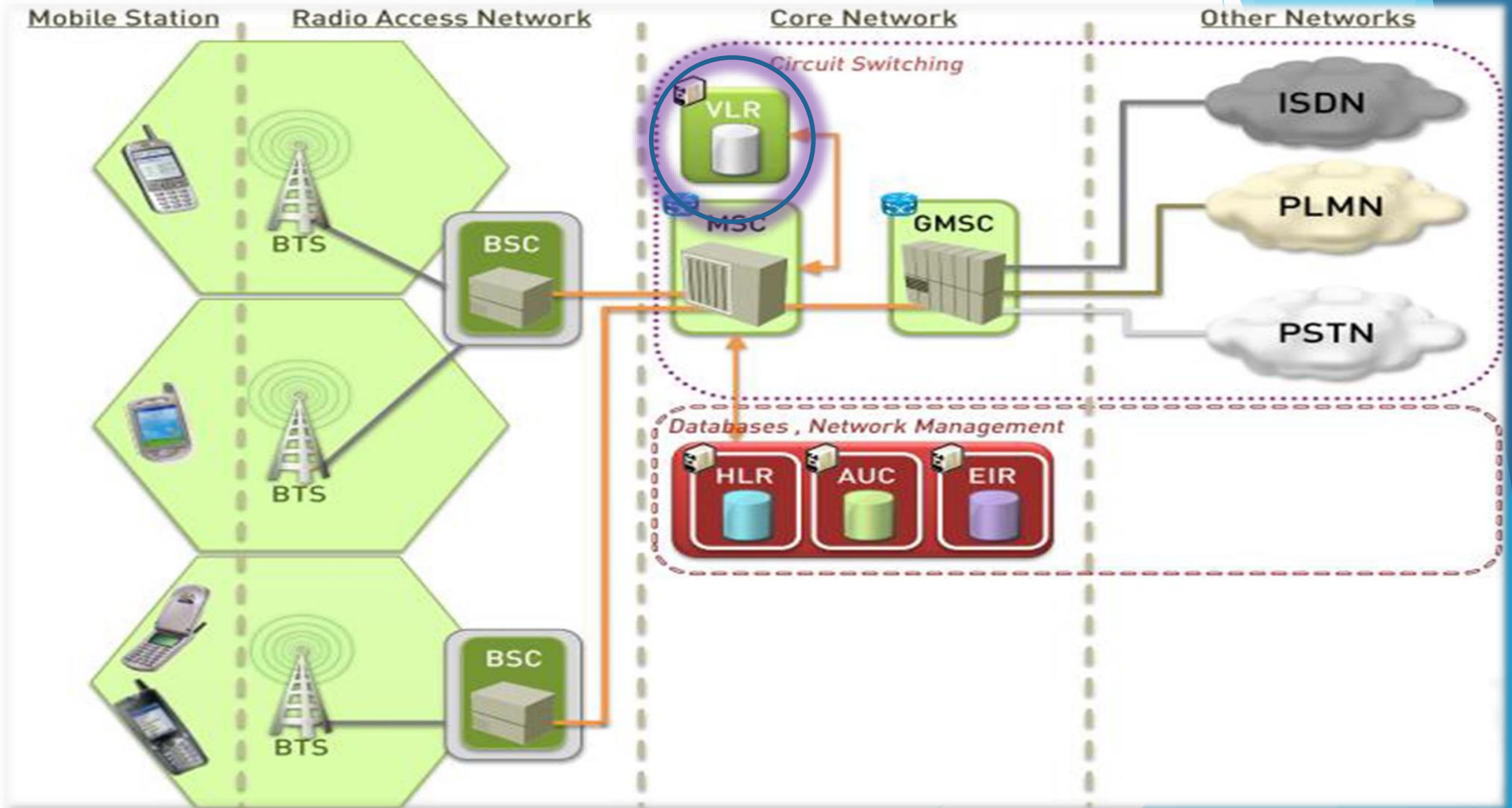
1) Au démarrage du téléphone



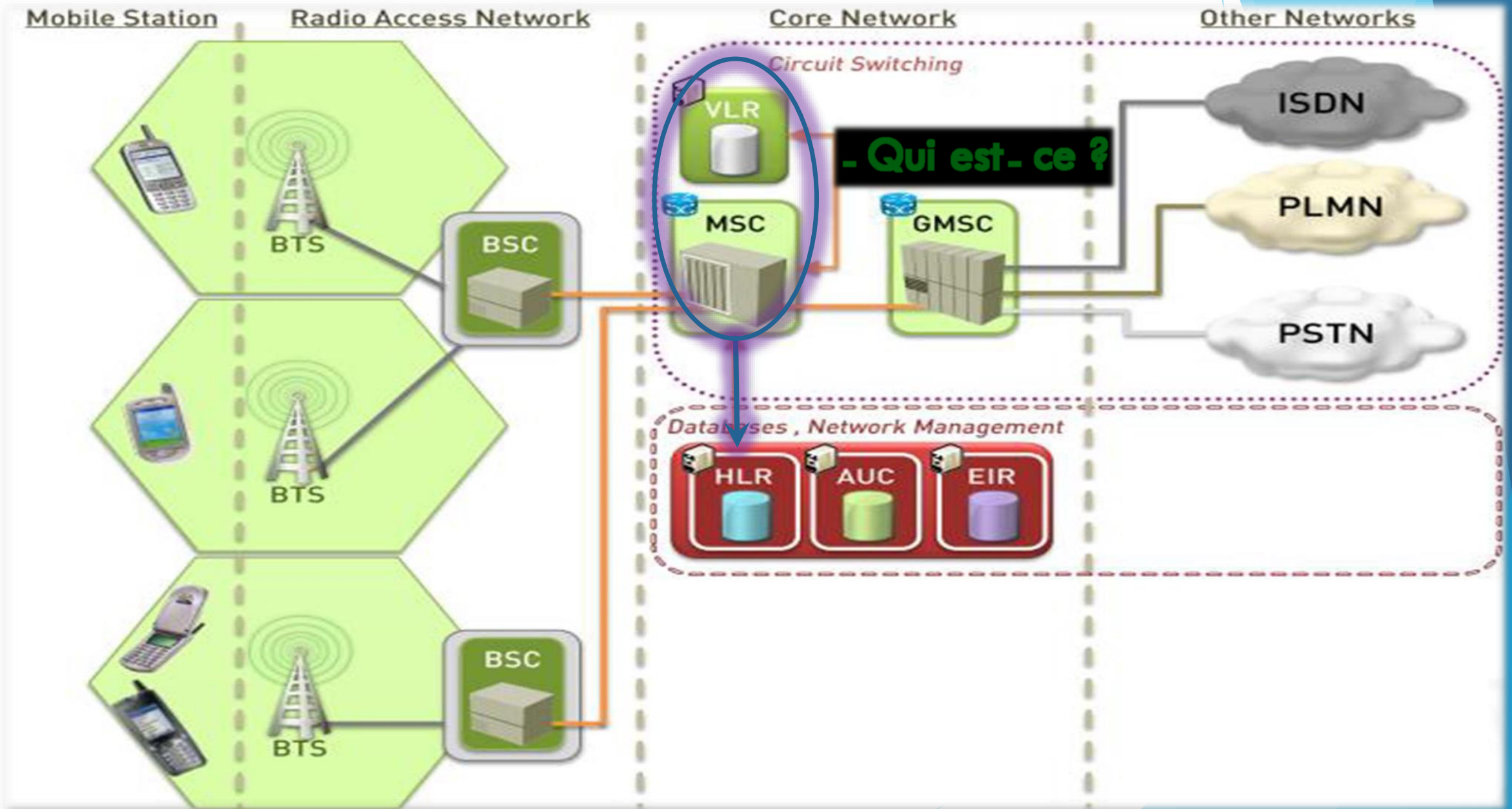
2) Le BTS transmet l'information au MSC /VLR



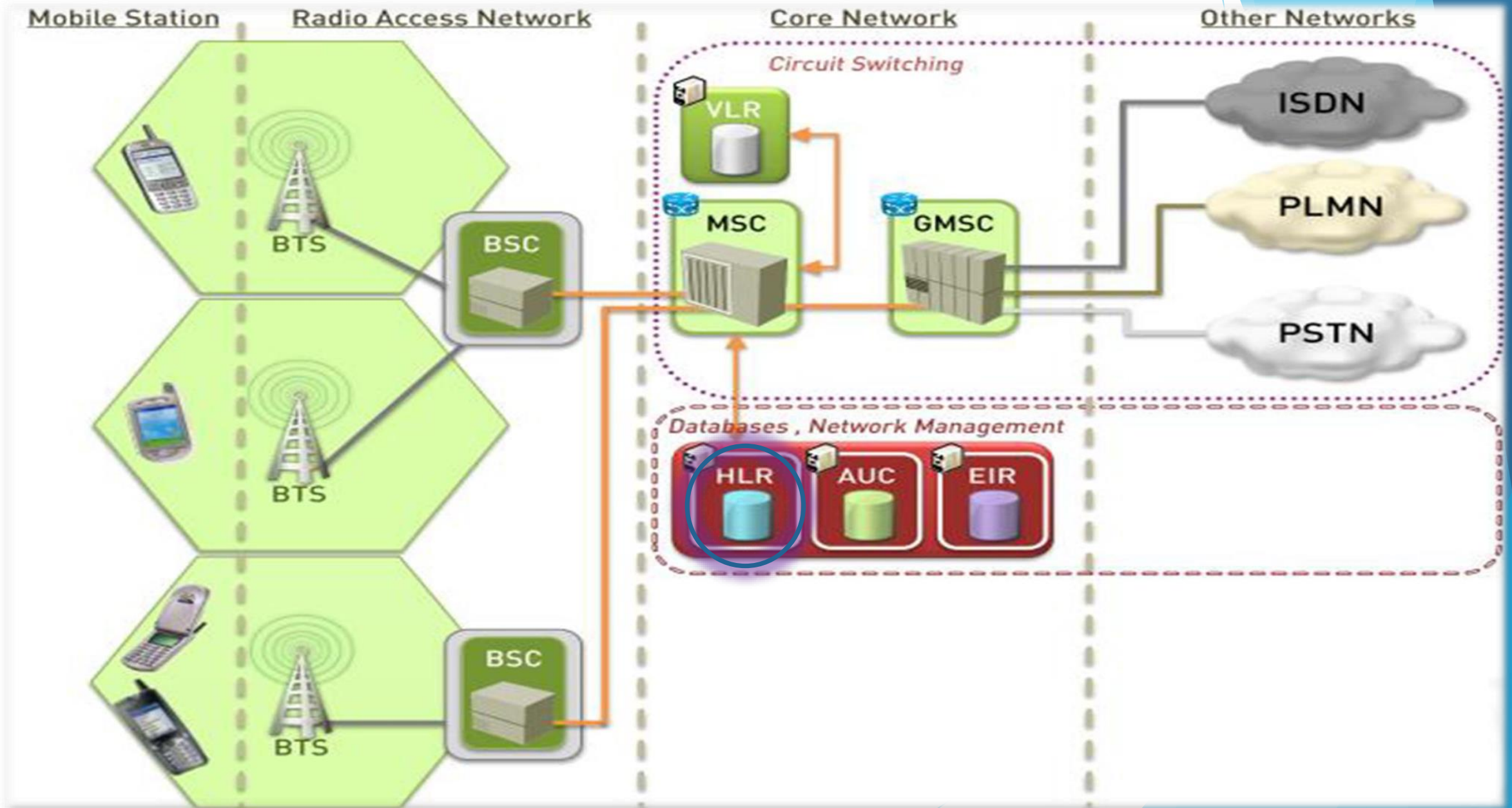
3) Le VLR reconnait la tentative de connexion



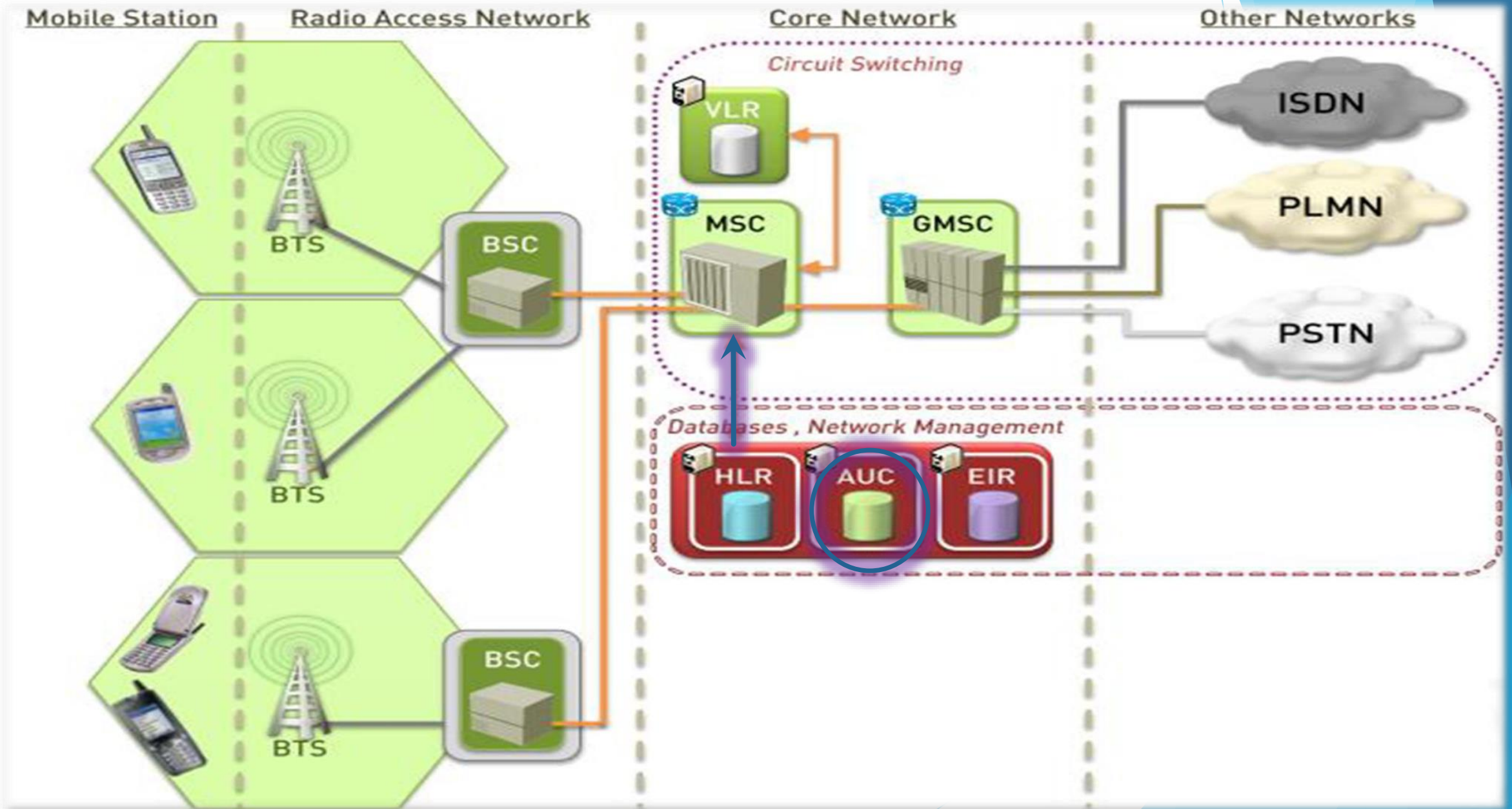
4) Le VLR demande des infos au HLR



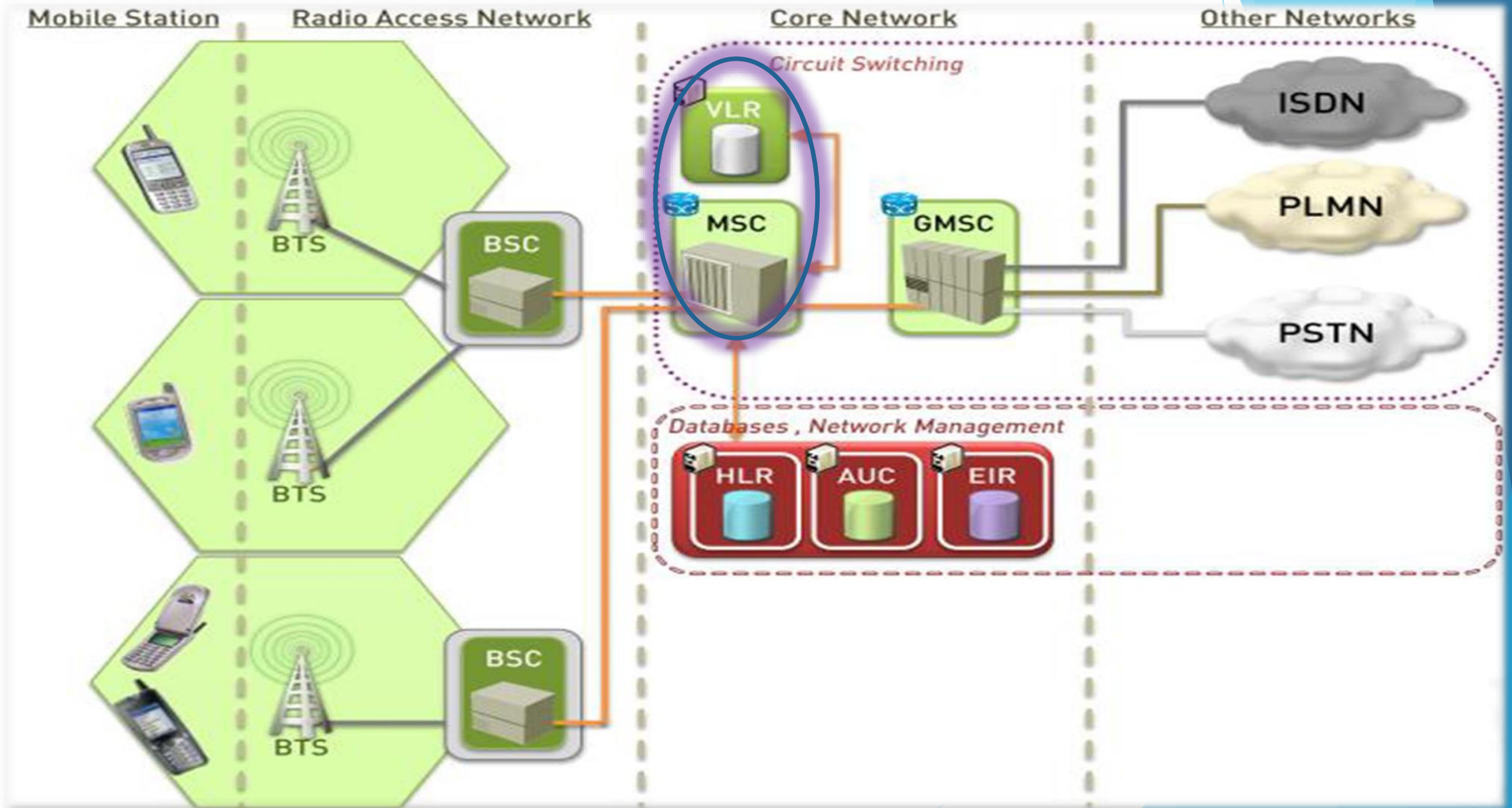
5) Le HLR identifie l'usager



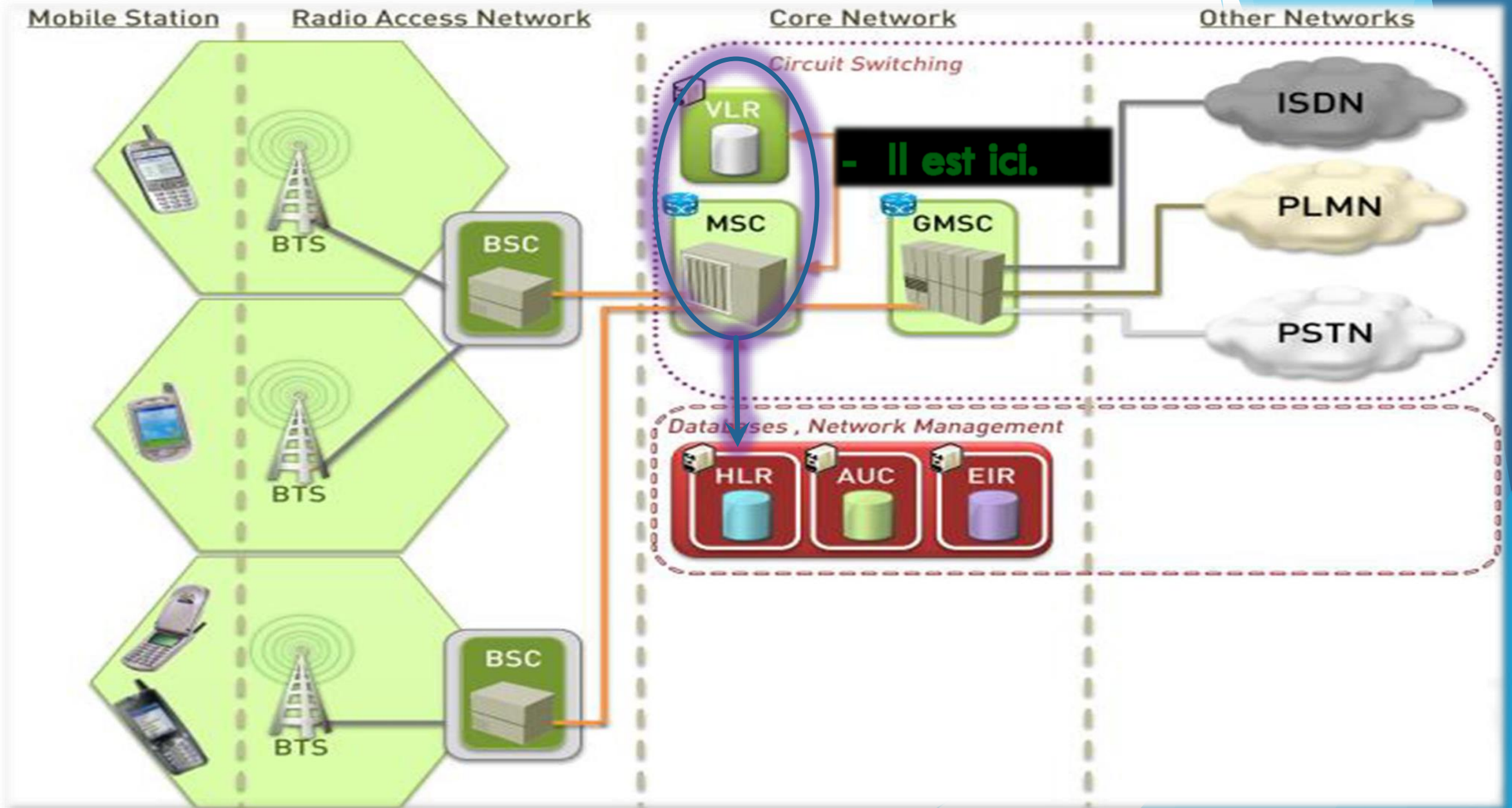
6) L' AuC génère et fournit des codes d'authentification au MSC/VLR



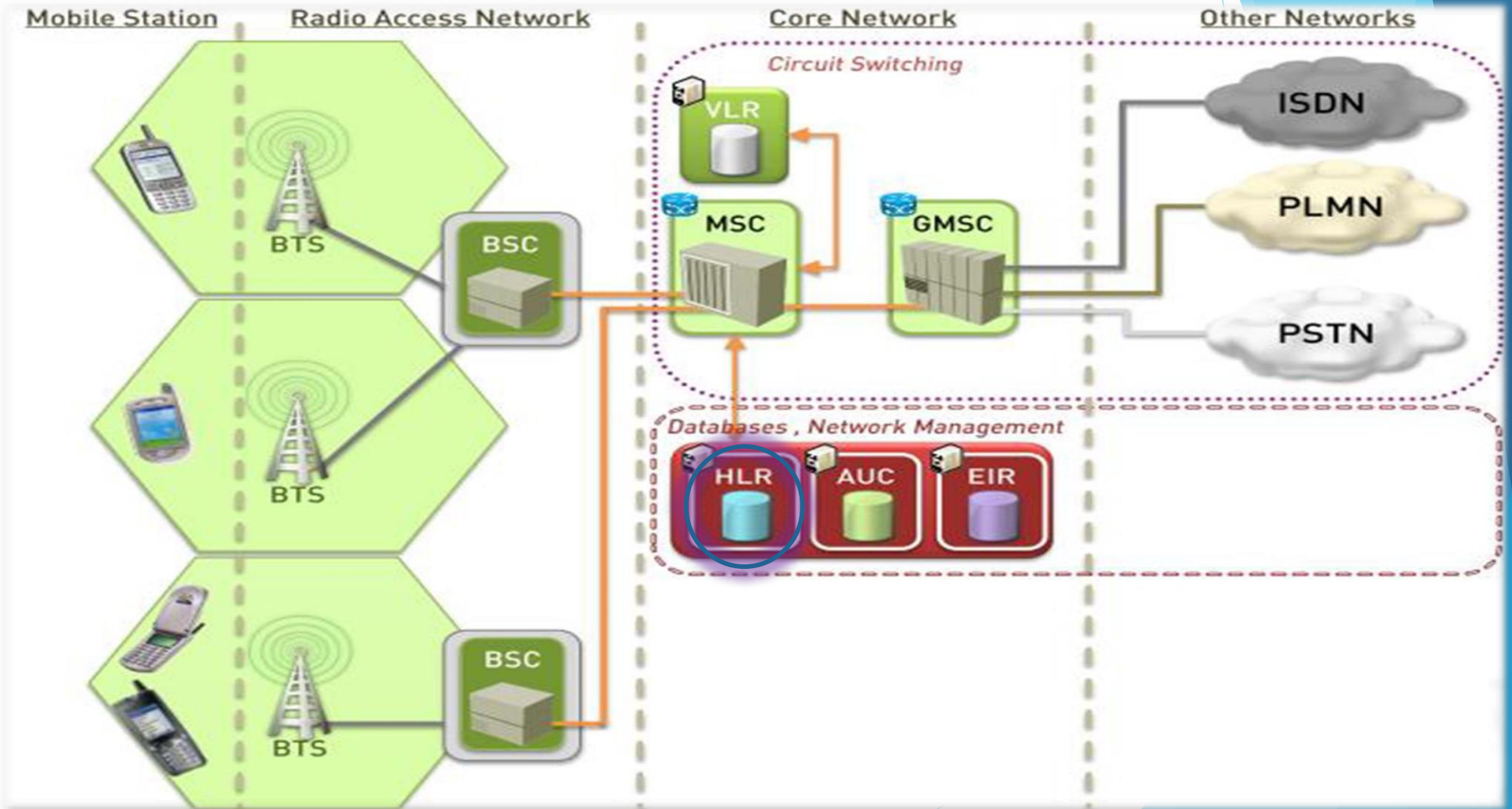
7) Le MSC/VLR vérifie les droits d'accès



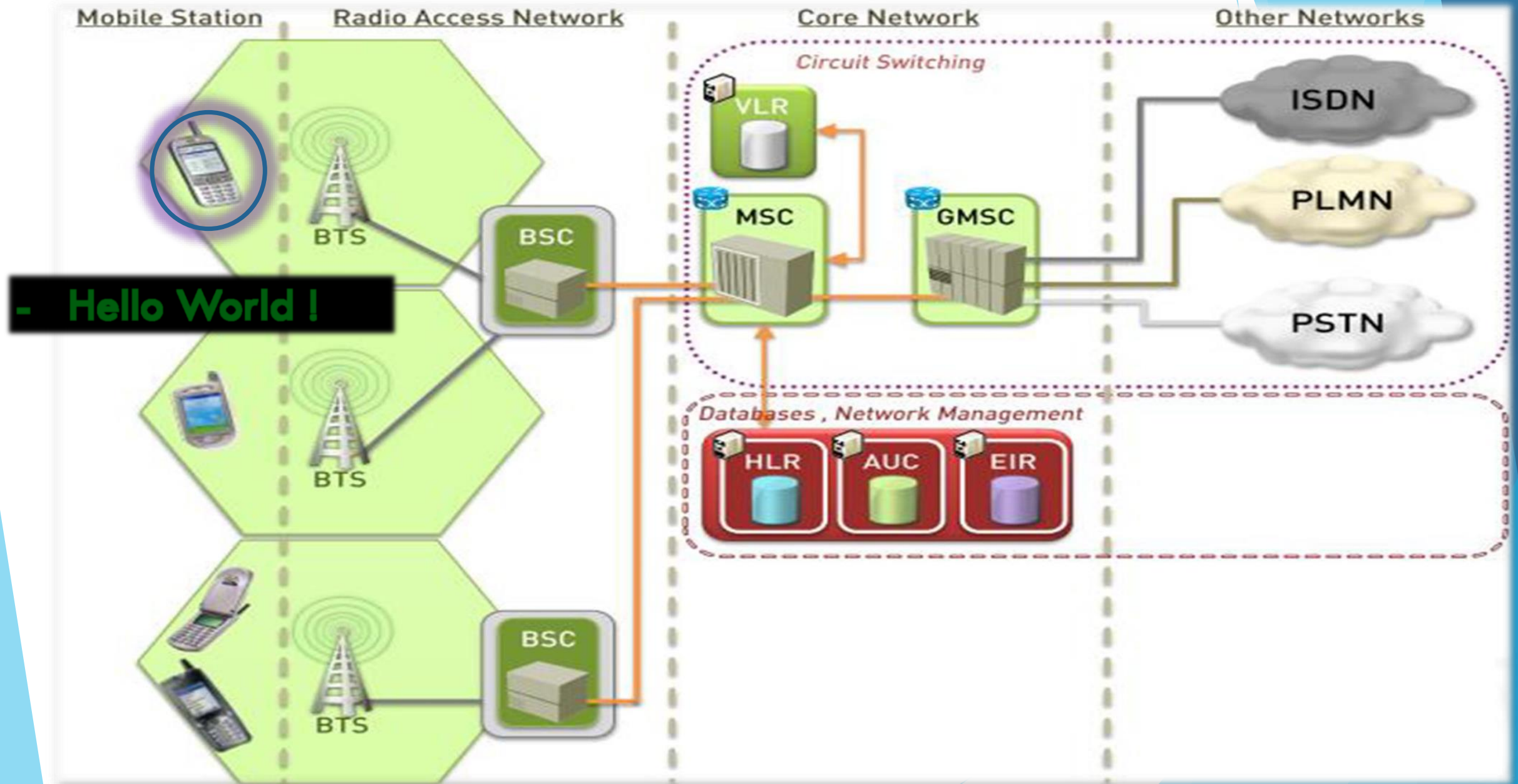
8) Le MSC/VLR informe le HLR de la position de l'usager



9) Le HLR actualise les données de l'utilisateur



10) Le téléphone est admis sur le réseau

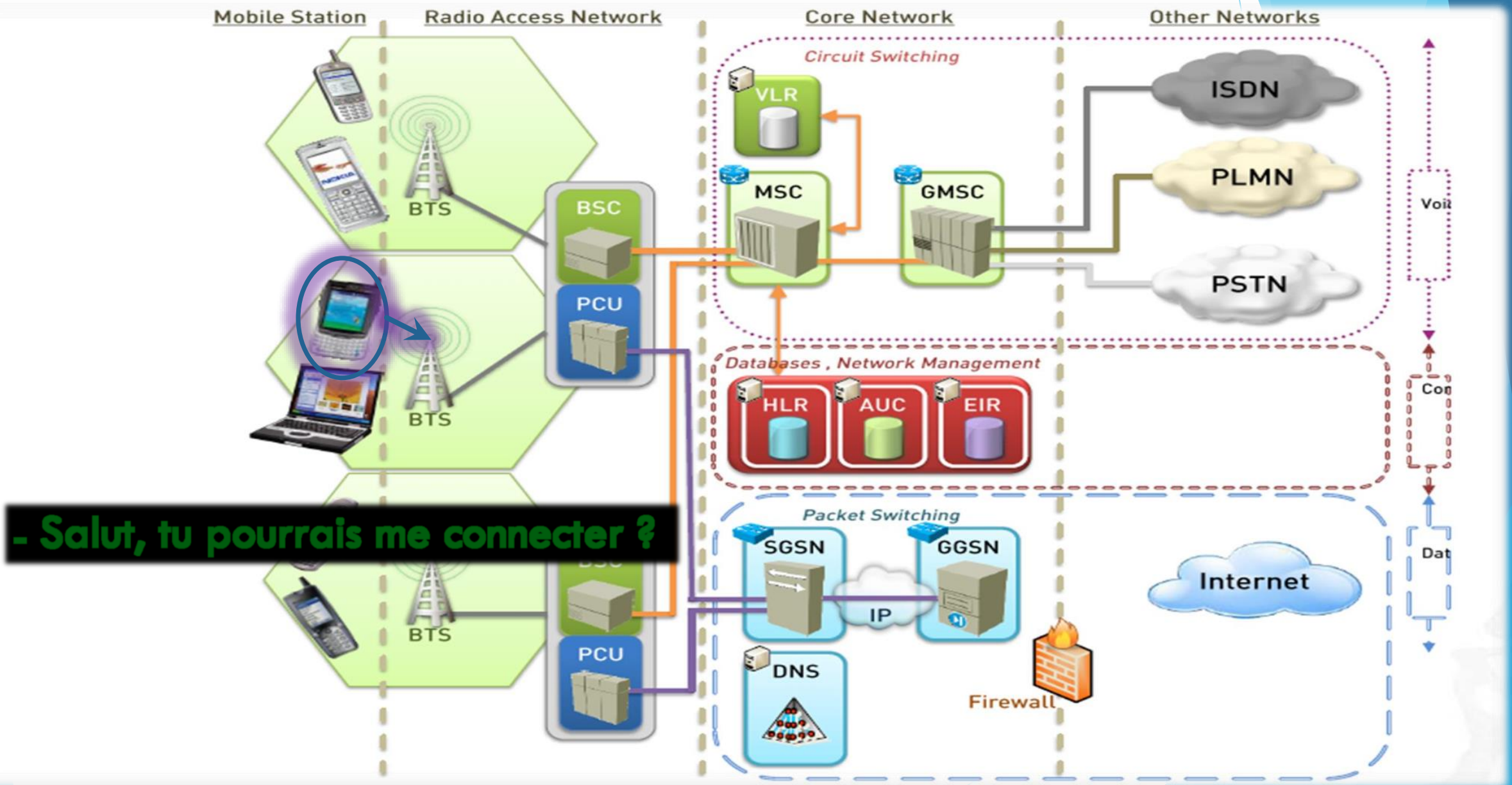


GPRS (2.5G)

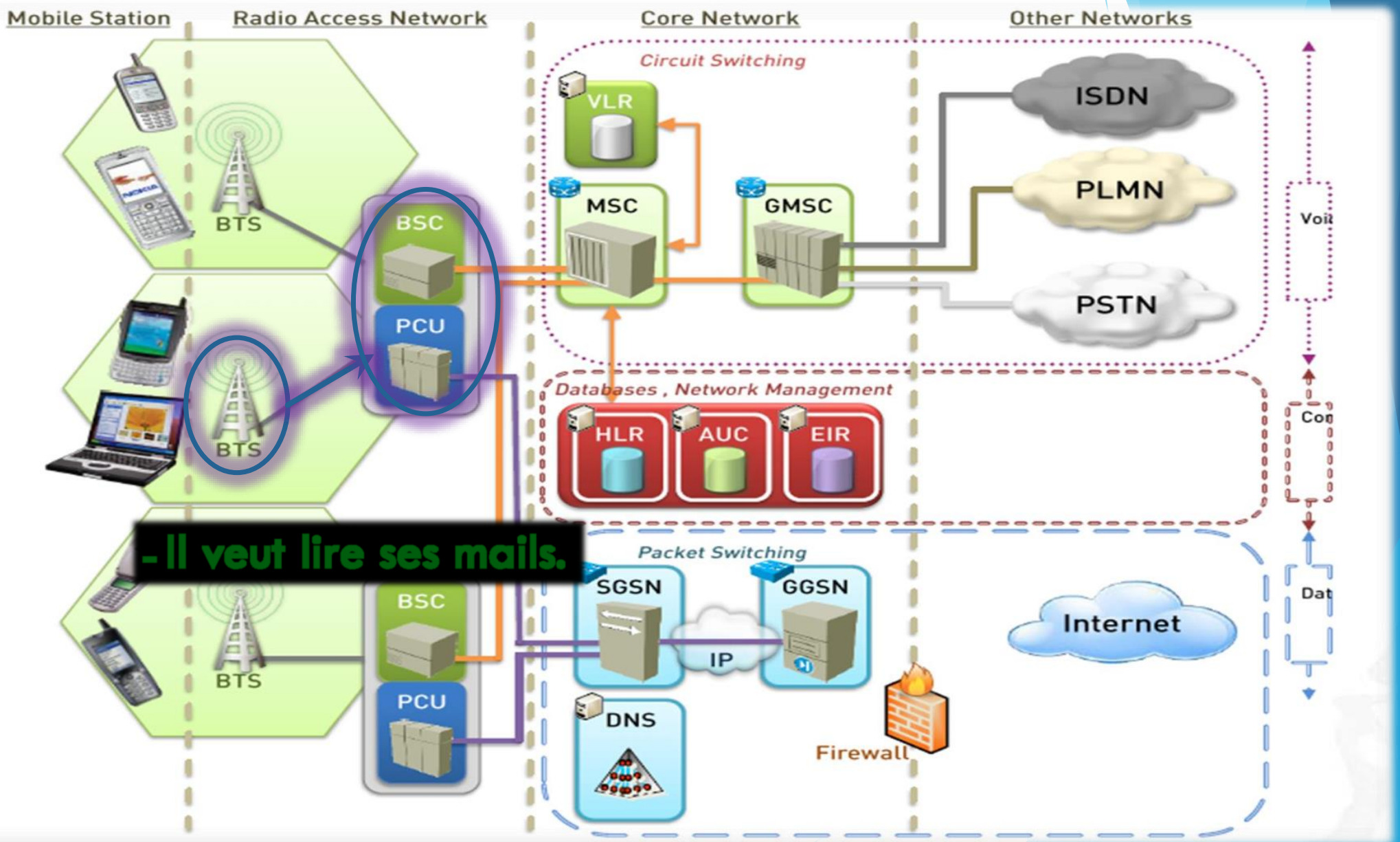
- General Packet Radio System
- Découpage des données en paquets réassemblés par la suite
- Nouveaux composants
  - **PCU** (Packet Control Unit): Traite les paquets pour la BSC
  - **SGSN** (Serving GPRS Support Node): Assure la livraison des paquets
  - **GGSN** (Gateway GPRS Support Node): Accès à Internet
  - **DNS** (Domain Name Server)

EX: Lire ses mails avec le système GPRS

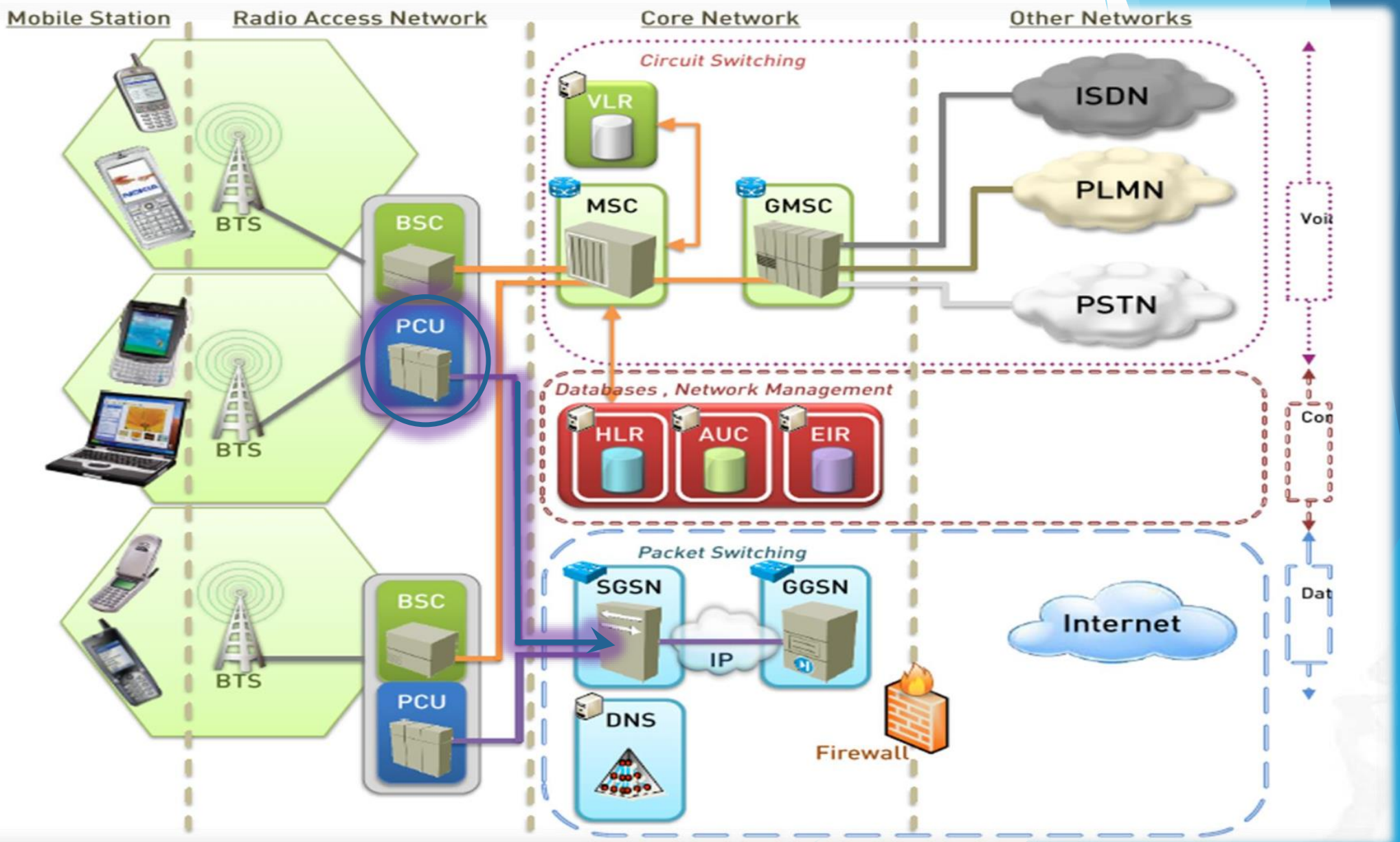
1) Le Smartphone demande une connexion réseau



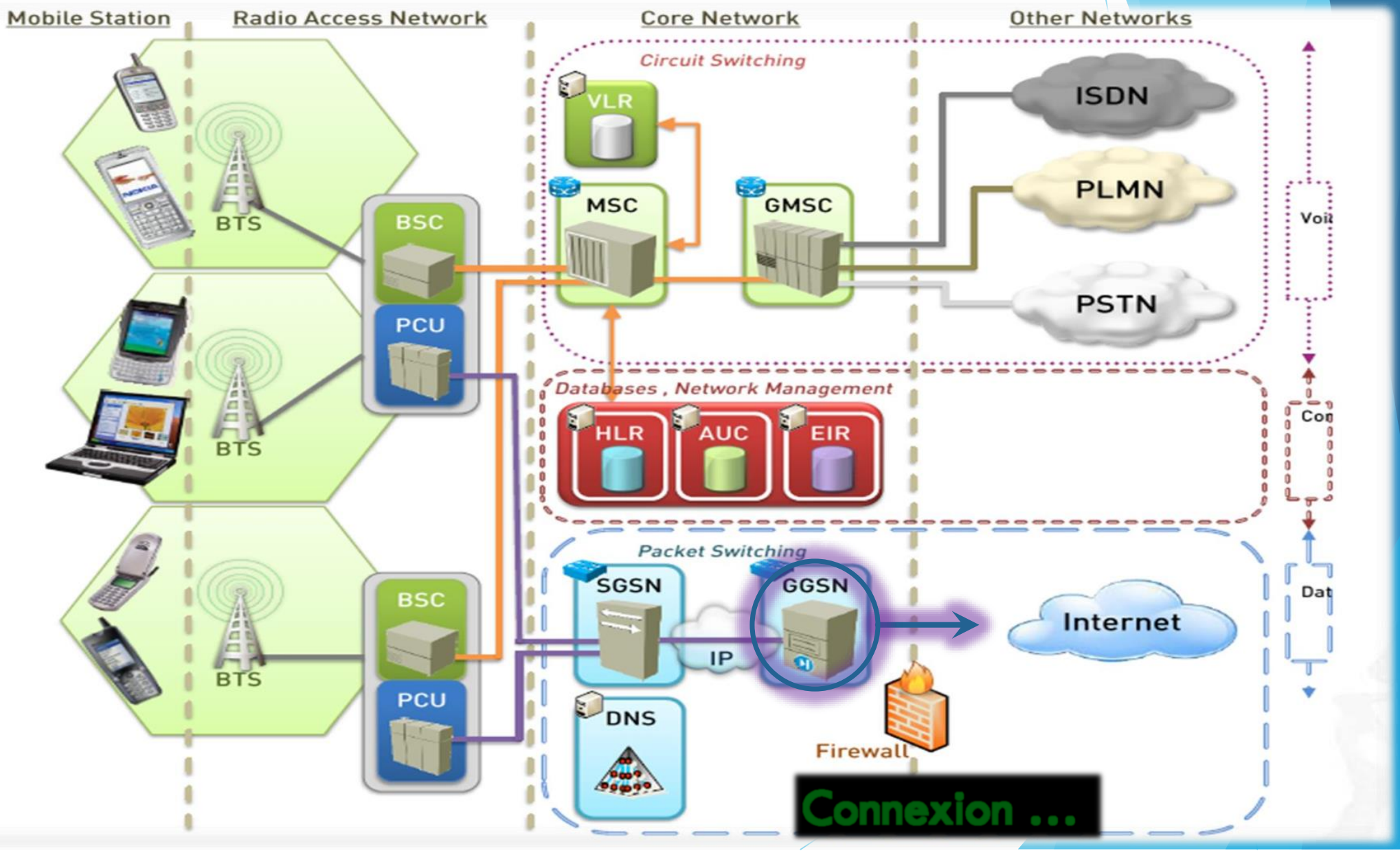
2) La demande passe par la BTS puis la BSC/PCU



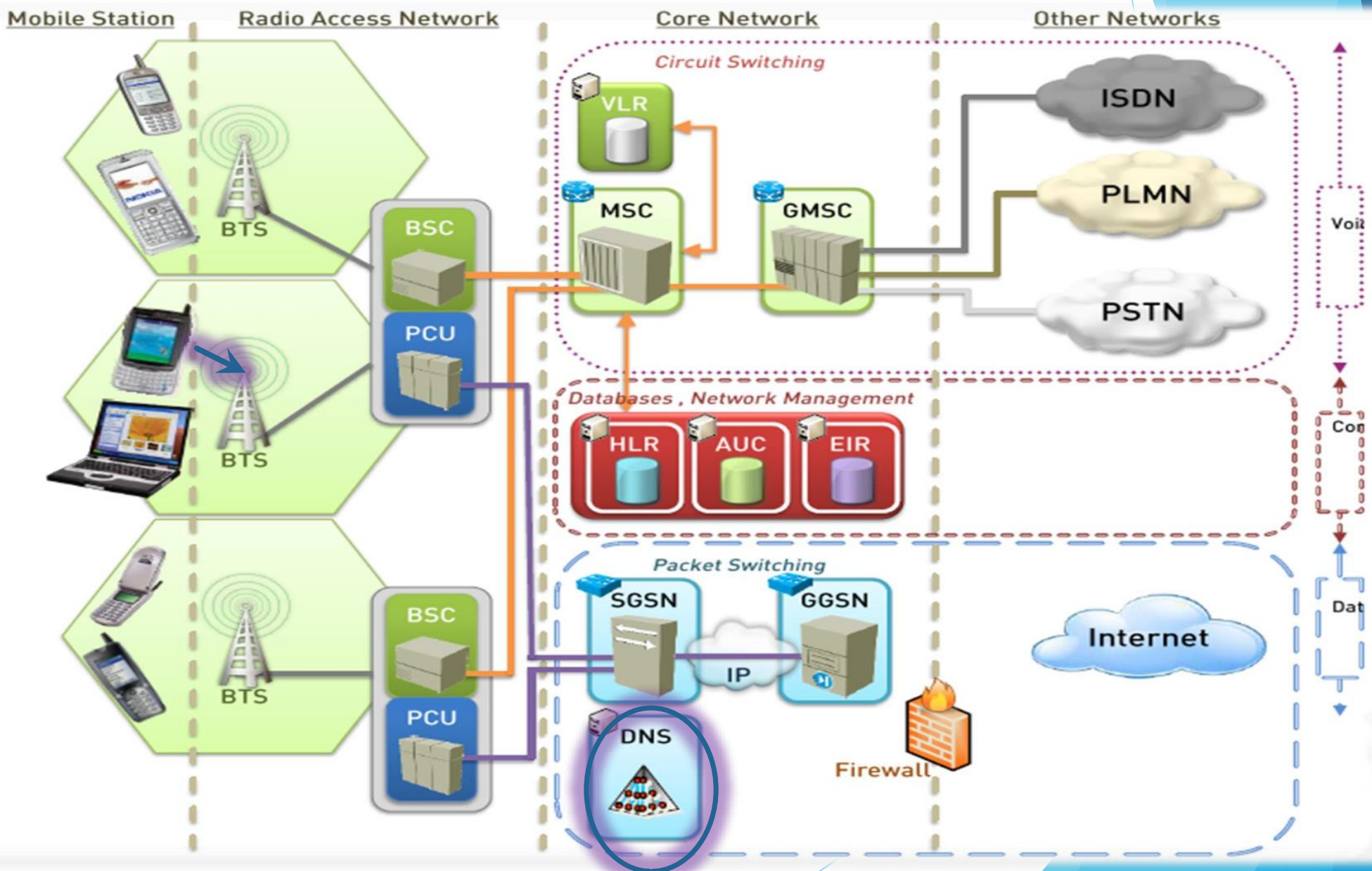
3) Le PCU transmet au SGSN



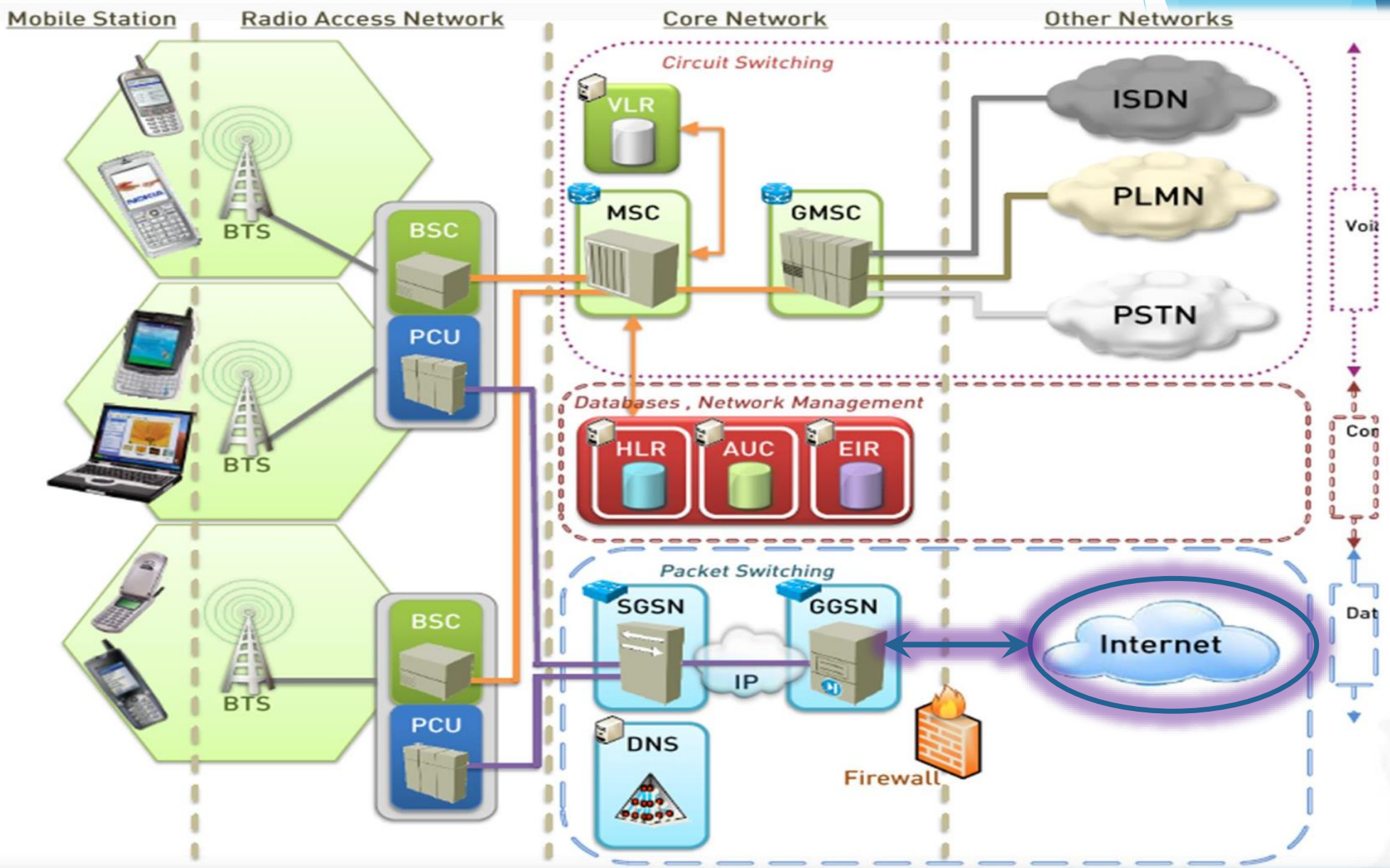
4) Le GGSN permet le passage à internet



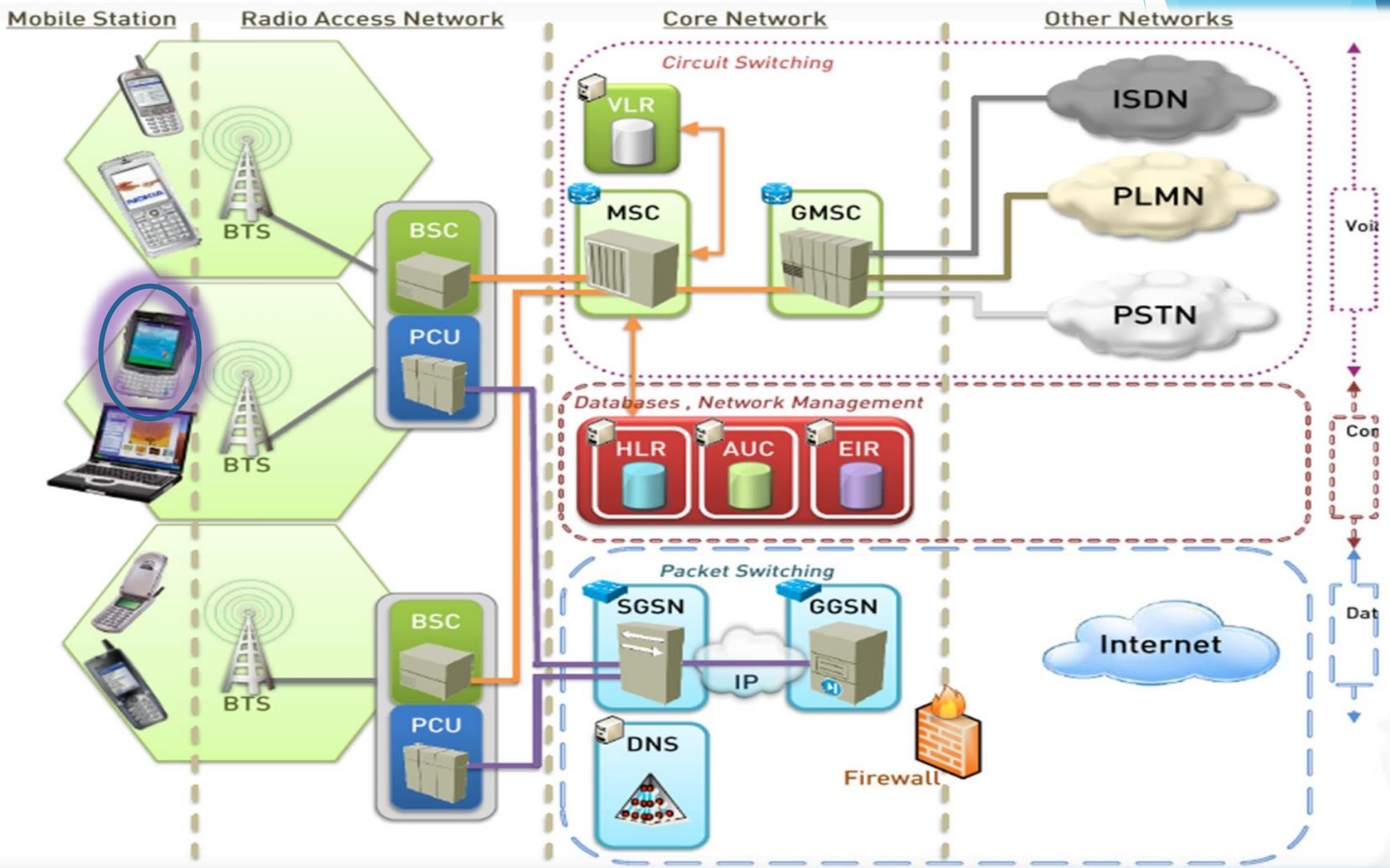
5) Le DNS trouve l'adresse IP du serveur mail



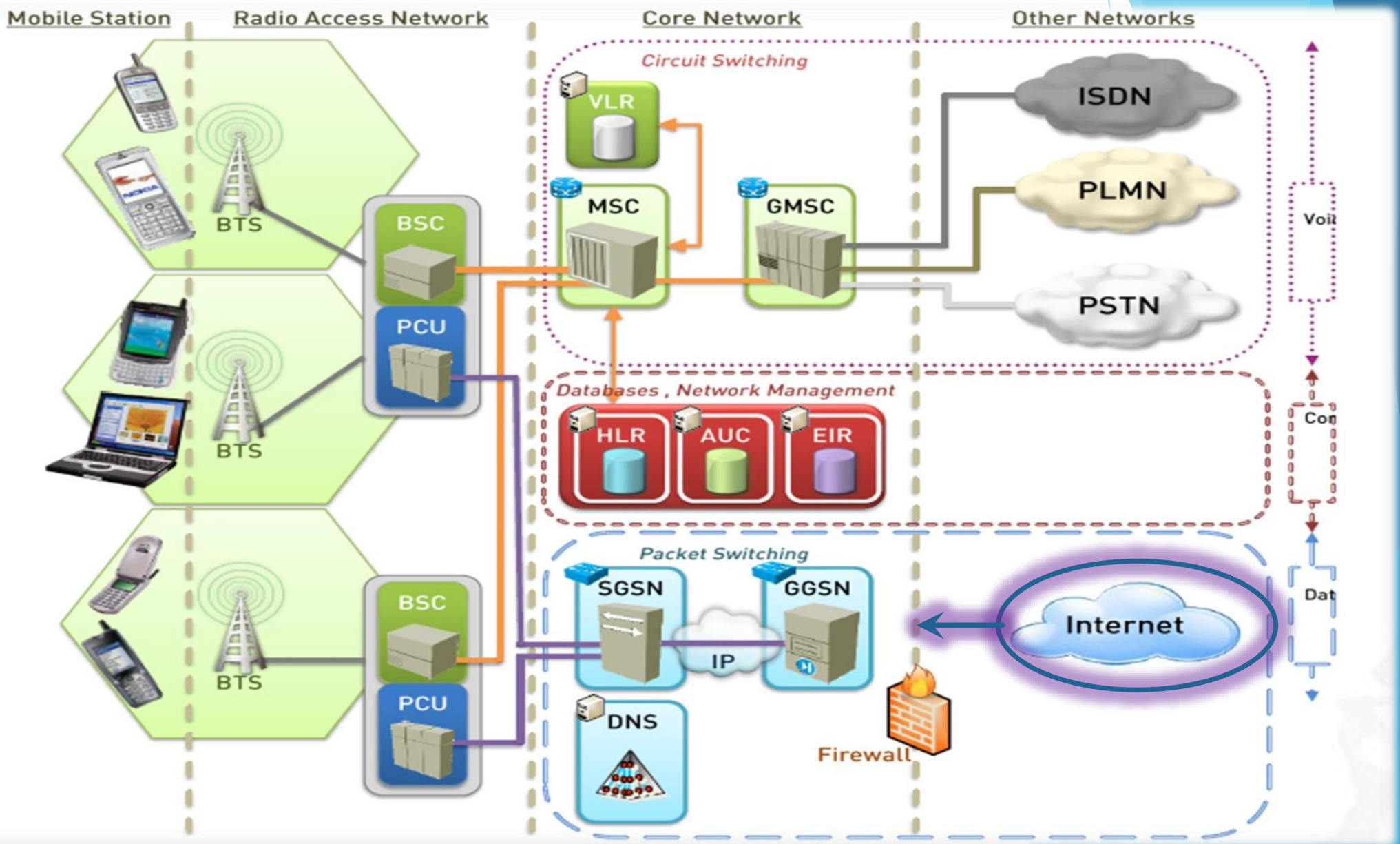
6) La connexion s'établit avec le serveur



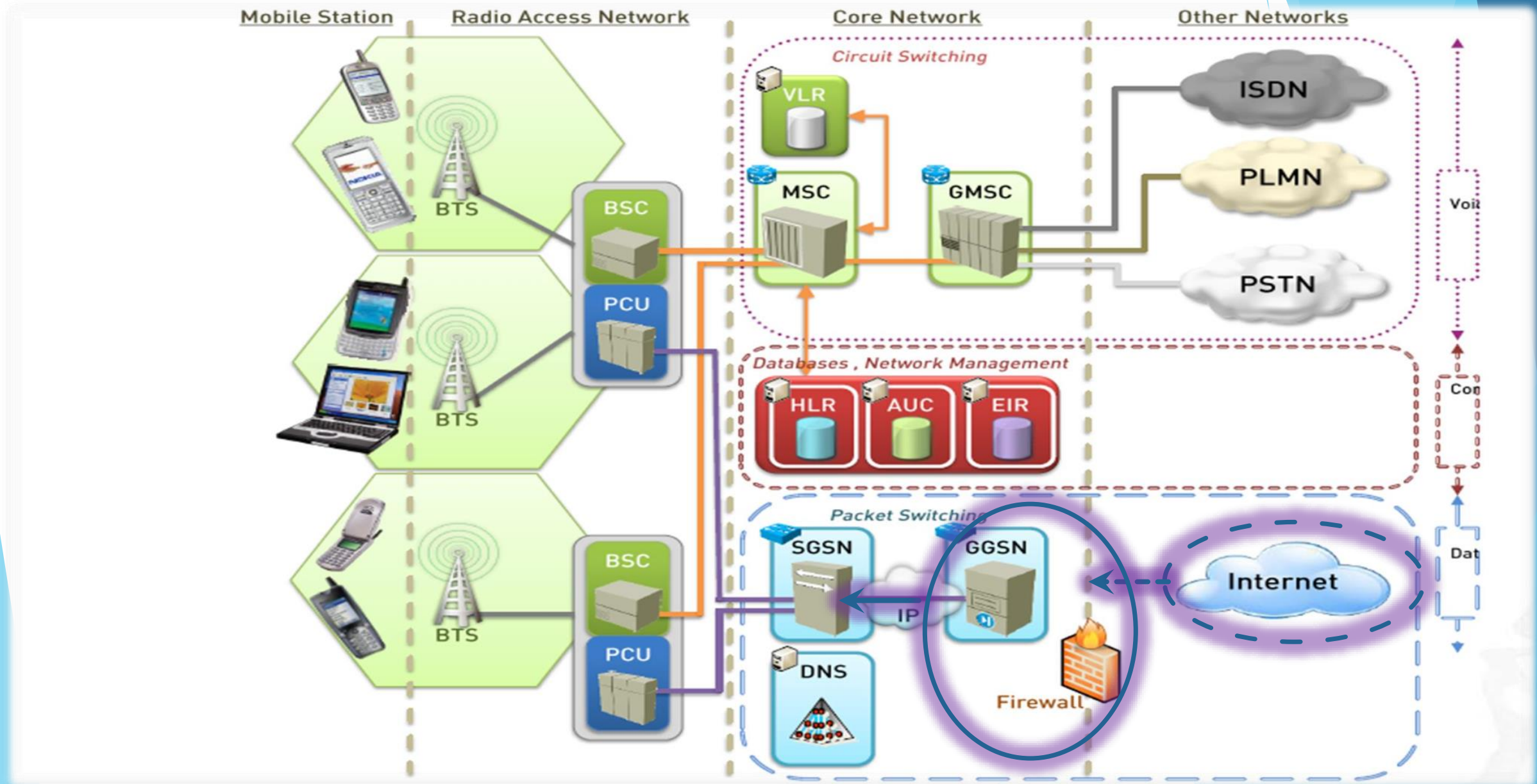
7) Vérification de la présence de nouveau mails



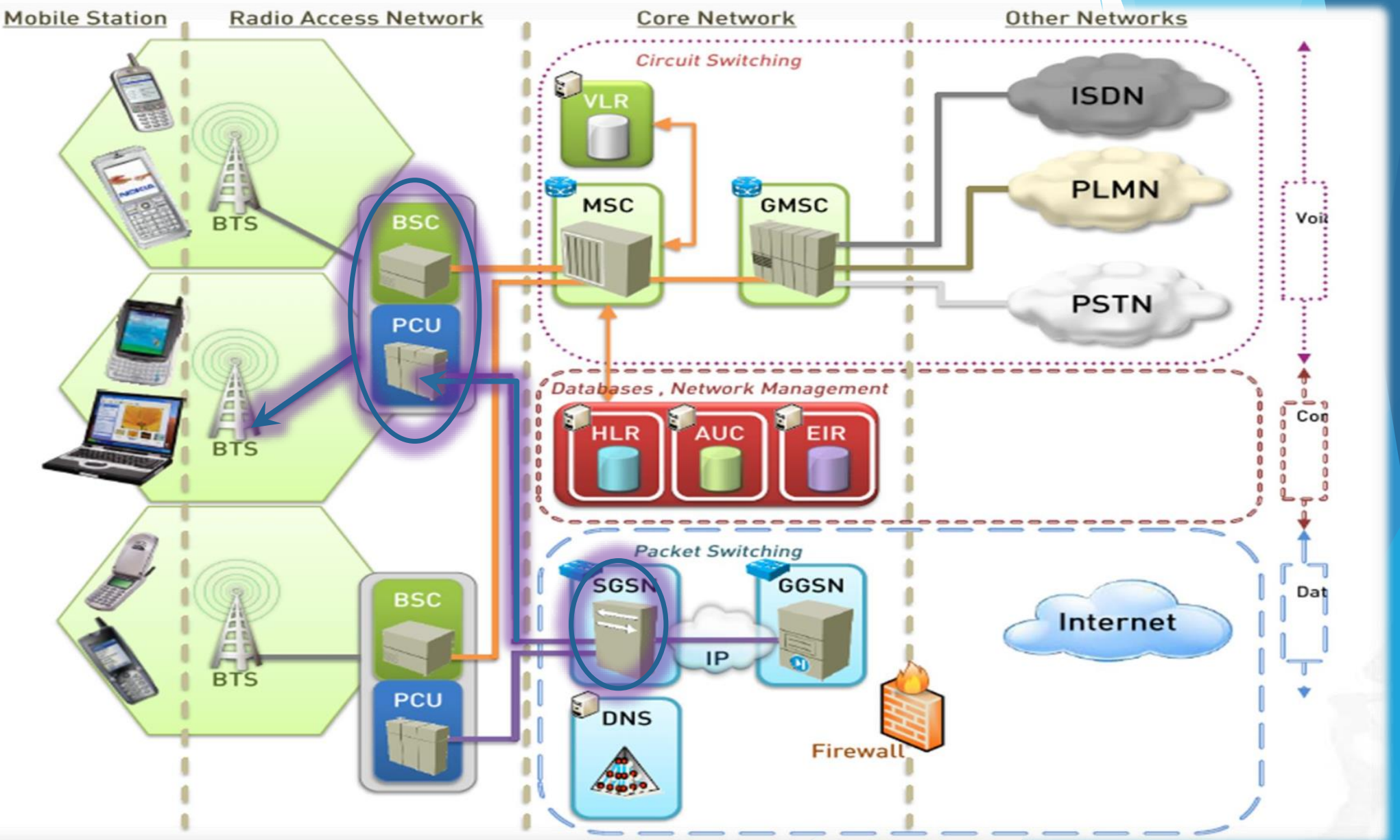
8) Le serveur envoie les nouveaux mails



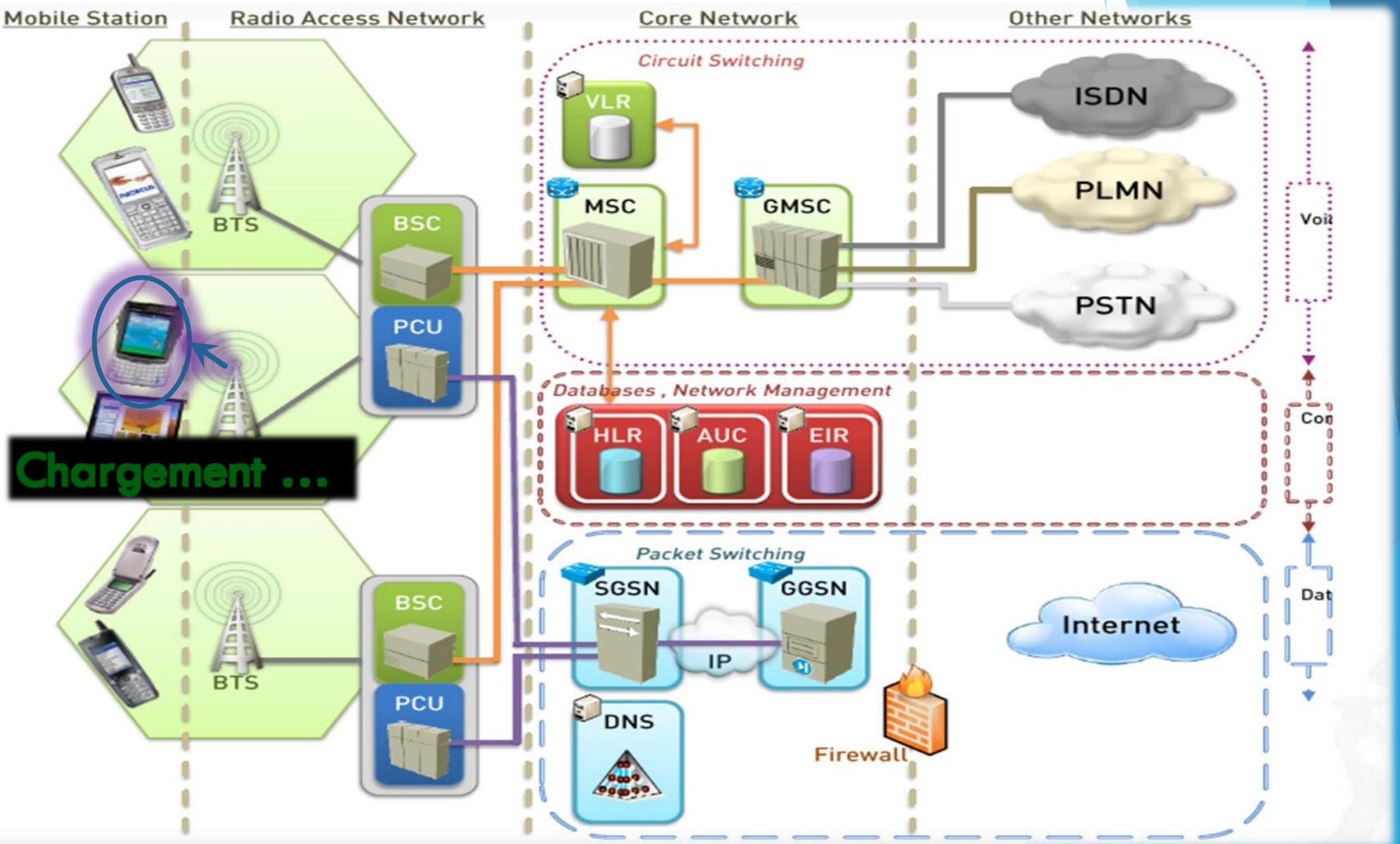
9) Le Firewall teste les paquets/Le GGSN les transmet



10) Les paquets sont transmis au PCU/BSC puis BTS



11) Le téléphone charge et affiche les mails



EDGE (2.75G)

- Enhanced Data rates for Global Evolution

- Identique au GPRS mais utilisation de meilleur algorithme de compression

UMTS (3G)

- Universal Mobile Telecommucation System

- Fonctionnement identique au GPRS
- Mais protocole différent dans les couches 1,2,3
- Antenne, fréquence, modulation différente
- Noms des composant différents (ex: BSC/PCU => RNC)

LTE (4G)

- Long-Term Evolution
- Protocoles améliorés et nouvelle structure
  - RAN => eNodeB
  - MME (Mobility Management Entity): gestion de flux
  - HLR => HSS (HomeSubscriberServer)
  - S/P-GW: porte vers les autres réseaux

- Voix en paquet (VoIP)
- 60Mb/s

CENTRALE

RESEAUX

Important:
Apprendre les acronymes pour le CF : POLY P.168
(plus il y a d'*' plus c'est important)

Merci de votre attention
Et bon courage à tous !!