

# Webservices

Introduction – Architectures Applicatives

# Composants applicatifs

Une application est généralement composée en 3 parties (appelées **Tiers**)

- **Les données** : c'est en général un SGBD, ce peut être des fichiers XML, un vulgaire fichier texte, etc.
- **Le traitement** : programme(s) qui traitent les données pour les servir
- **La présentation** : interface présentée à l'utilisateur final

# L'architecture

Un architecture décrit :

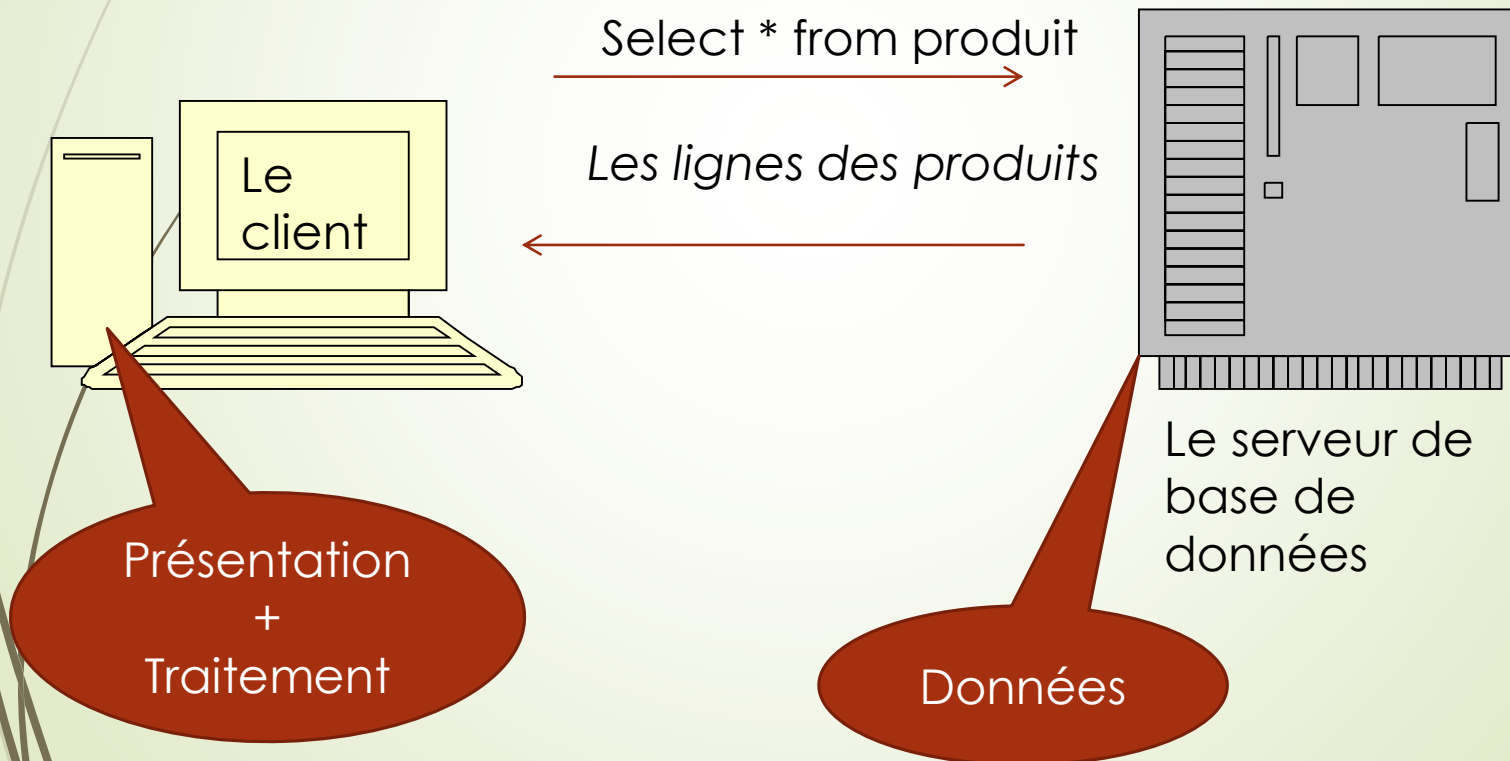
- L'implantation chaque tiers sur des machines
- La nature des *relations* entre eux
- Les *couches* logicielles à mettre en œuvre, car les composants/applications sont rarement accessibles directement mais au travers de couches logicielles

# Le classique : l'Architecture de type Client/Serveur

- Les composants sont au moins sur deux machines distinctes
- Les échanges sont de type maitre/esclave : un serveur qui fournit un service, un client qui demande un service

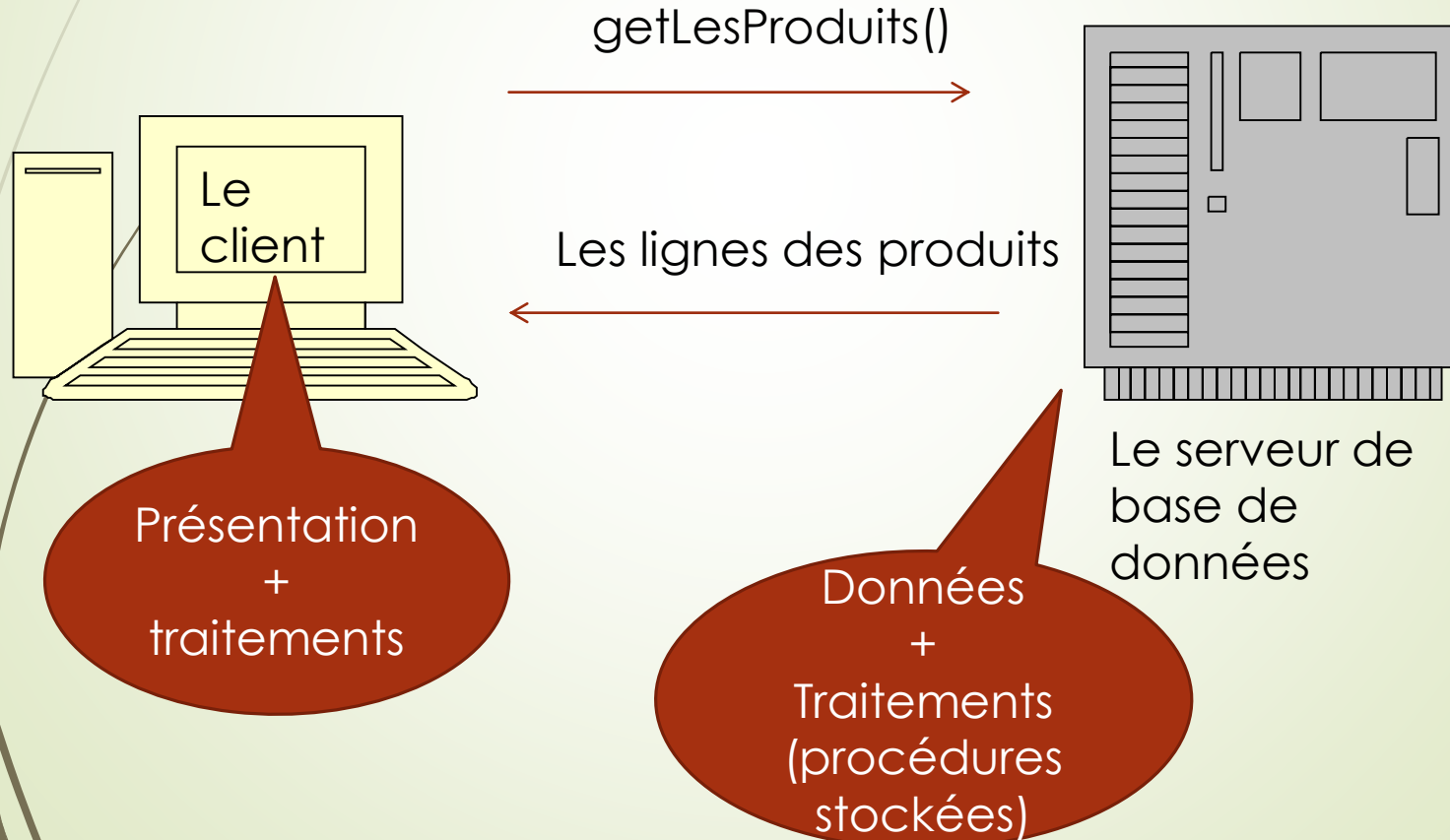
# Client/serveur : exemple 1

## Client/serveur de données



# Client/serveur : exemple 2

## appel de méthode distantes (RMI)



# Client/serveur

- **Les avantages :**

- Simple de mise en œuvre
- Présente une interface client conviviale riche

- **Les inconvénients :**

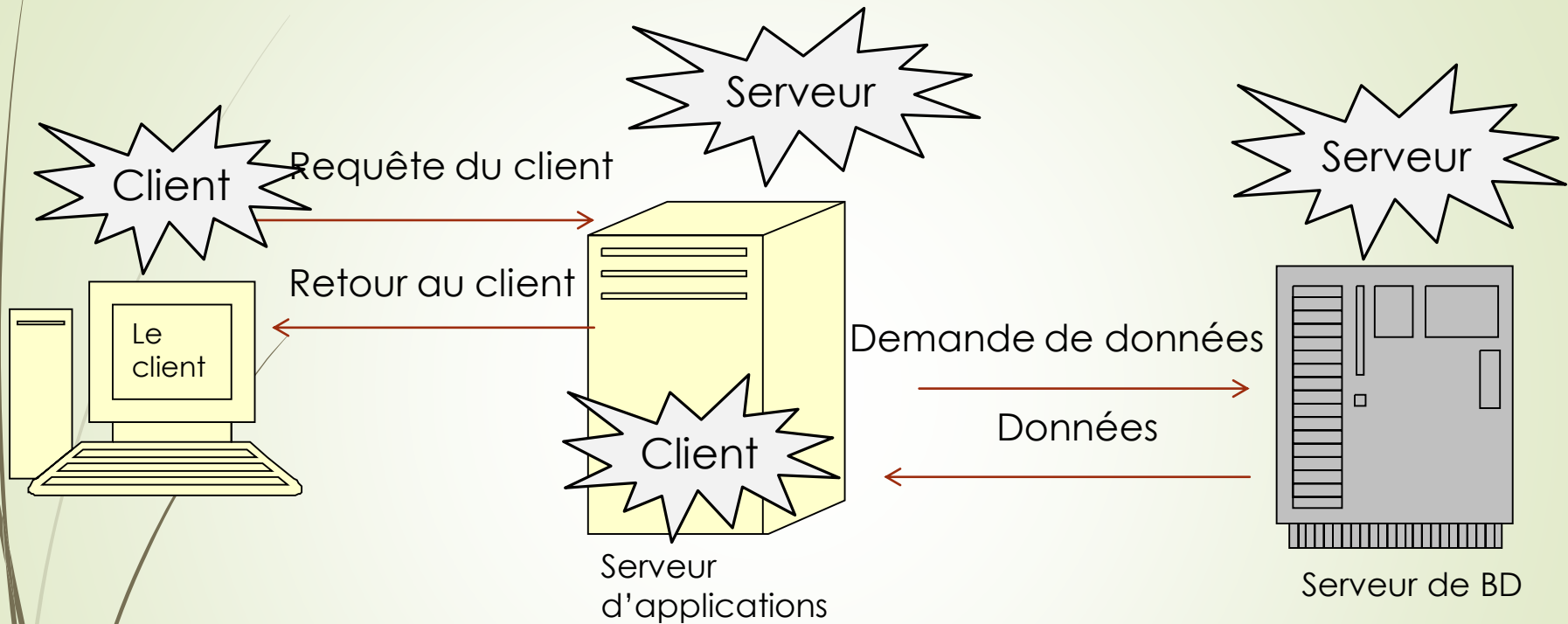
- Toute évolution doit être déployée sur tous les postes clients
- Dépendante souvent de logiciels propriétaires (Applications internes, Microsoft, Windev...)

# Architecture 3 tiers

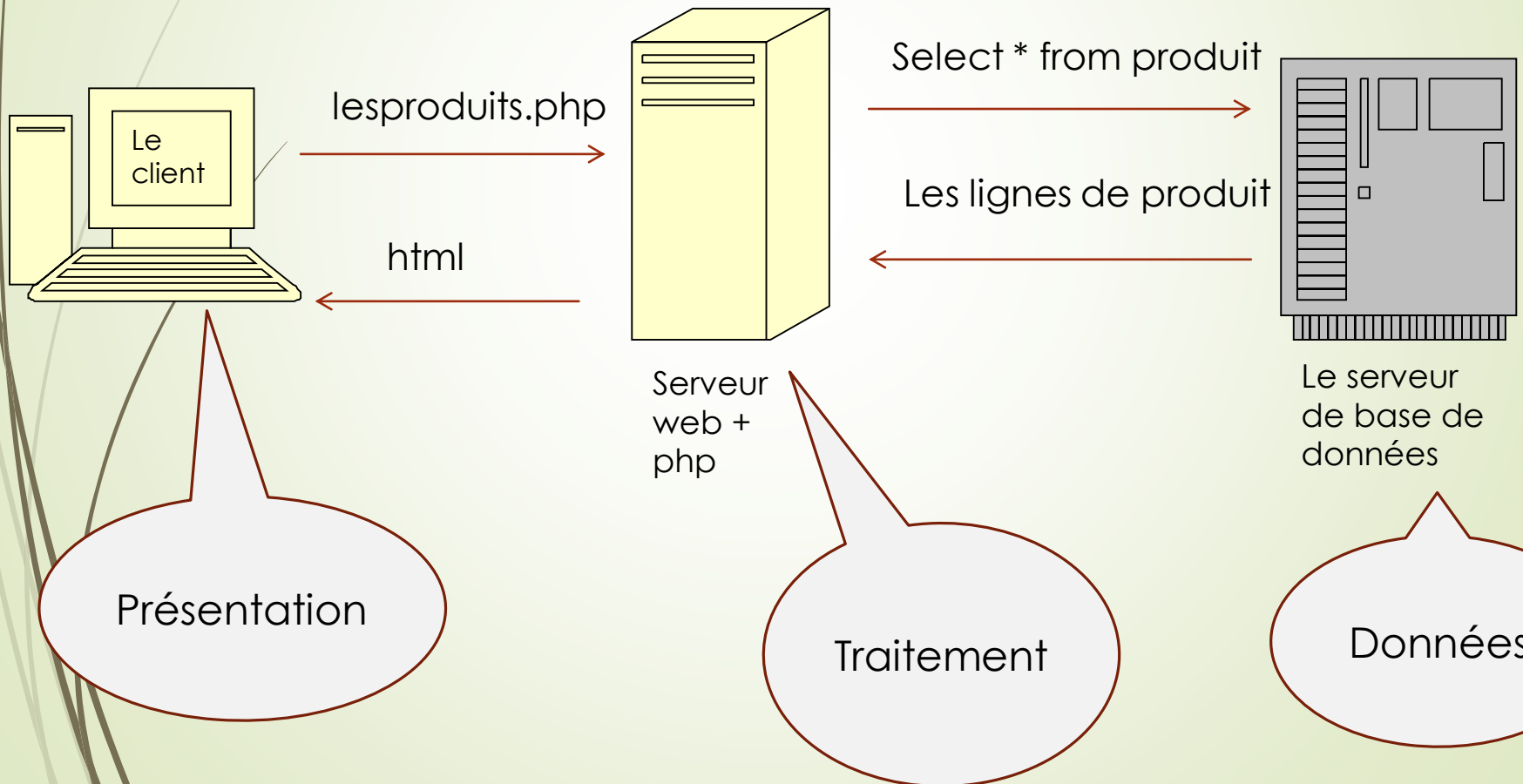
- Un autre « tier-niveau » est présent : un serveur d'application
- C'est l'architecture des applications web
- C'est aussi le cas d'applications sur gros système très sollicités



# Architecture 3 tiers



# Les applications web version classique



# Pourquoi la réussite d'une solution web

- S'exécute pour sa partie client dans un seul navigateur : pas de « déploiement »
- Utilise des protocoles standards IP, HTTP
- Gère la sécurité, HTTPS
- Sait s'adapter aux besoins nouveaux : interfaces « riches » (HTML5/CSS3)

# Les services web

- « Un **service web** est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués ».
- Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et de manière synchrone. »
- **Wikipédia**

# Les services web

- Architecture de type Client/serveur
- Présence d'un fournisseur du service
- Présence d'un client, consommateur du service
- Utilisent le protocole HTTP
- Le serveur retourne les données du service au format XML
- Très utilisé pour faire communiquer deux applications

# Les services web

➤ Concrètement...



# Les services web

- Exemple d'annuaire de services :
- <http://www.webserviceX.net/ws/default.aspx>

## Webservices Directory



Business and Commerce



Standards and Lookup Data



Graphics and Multimedia



Utilities



Messaging



Value Manipulation / Unit  
Converter



Other Web Services

# Un exemple de service

## GlobalWeather

The following operations are supported. For a formal definition, please review the WSDL.

- [GetCitiesByCountry](#)  
Get all major cities by country name(full / part).
- [GetWeather](#)  
Get weather report for all major cities around the world.

Fonctions disponibles

### GetWeather

Get weather report for all major cities around the world.

#### Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button

Parameter	Value
CityName:	<input type="text" value="Grenoble"/>
CountryName:	<input type="text" value="France"/>
<input type="button" value="Invoke"/>	

Appel de la fonction

```
<?xml xmlns="http://www.webserviceX.NET">
  <?xml version="1.0" encoding="utf-16"?> <CurrentWeather> <
  386M</Location> <Time>Mar 11, 2013 - 04:00 PM EDT / 2013.0
  (direction variable):0</Wind> <Visibility> greater than 7
  <Temperature> 44 F (7 C)</Temperature> <DewPoint> 39 F (4
  29.47 in. Hg (0998 hPa)</Pressure> <Status>Success</Status>
</string>
```

Réponse obtenue



# Les services web

- Utilisé en intranet ou extranet
- De très nombreux services sont payants
- Semble s'imposer comme un service standard de communication entre applications
- Est aujourd'hui très fortement implémenté par un mécanisme simple, sans protocole dédié et basé uniquement sur HTTP : les services REST

# Les services REST 1/4

- Le serveur va fournir directement la ressource demandé à *partir d'une URI*, au format XML ou JSON
- Ce type de service est très largement utilisé dans les applications mobiles qui utilisent des données publiques

# Un exemple de service REST

## 2/3

- La ville de Paris fournit les disponibilités des stations d'accueil de la ville.
- La liste des stations à :  
<http://www.velib.paris.fr/service/carto>

```
<carto>
- <markers>
  <marker bonus="0" open="1" lng="2.391225227186182" lat="48.892795924112306" fullAddress="ALLEE
    DU BELVEDERE PARIS 19 - 0 75000 Paris - 75000 PARIS" address="ALLEE DU BELVEDERE PARIS
    19 - 0 75000 Paris -" number="901" name="00901 - ALLEE DU BELVEDERE"/>
  <marker bonus="0" open="1" lng="2.374340554605615" lat="48.83713368945151" fullAddress="FETE
    DE L'OH (BERCY) - QUAI MAURIAC ANG PONT DE BERCY" address="FETE DE L'OH (BERCY) - QUAI
    MAURIAC ANG PONT DE BERCY" number="903" name="00903 - QUAI MAURIAC / PONT DE
    BERCY"/>
  <marker bonus="0" open="1" lng="2.301961227213259" lat="48.85213620522547" fullAddress="ECOLE
    MILITAIRE-AVENUE DE LA MOTTE PICQUET - 75007 PARIS" address="ECOLE MILITAIRE-AVENUE
    DE LA MOTTE PICQUET -" number="904" name="00904 - PLACE JOFFRE / ECOLE MILITAIRE"/>
  <marker bonus="0" open="1" lng="2.382472269083633" lat="48.83966087889425" fullAddress="GARE
    DE BERCY - ANGLE RUE CORBINEAU - 75012 PARIS" address="GARE DE BERCY - ANGLE RUE
    CORBINEAU -" number="905" name="00905 - GARE DE BERCY (STATION MOBILE 5)"/>
  <marker bonus="0" open="1" lng="2.358630064544601" lat="48.876419813641114"
```

# Un exemple de service REST

## 3/4

- Chaque station peut être accédée par son numéro
- <http://www.velib.paris.fr/service/stationdetails/901>



3 16 20 1 1 1484641451 1

# Un exemple de service REST

## 4/4

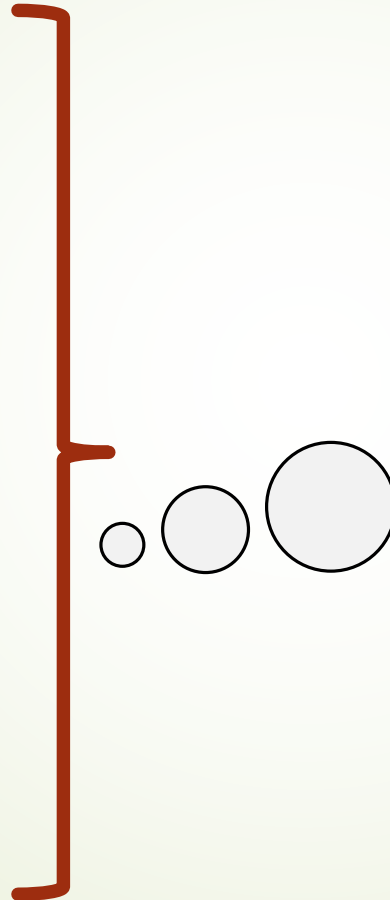
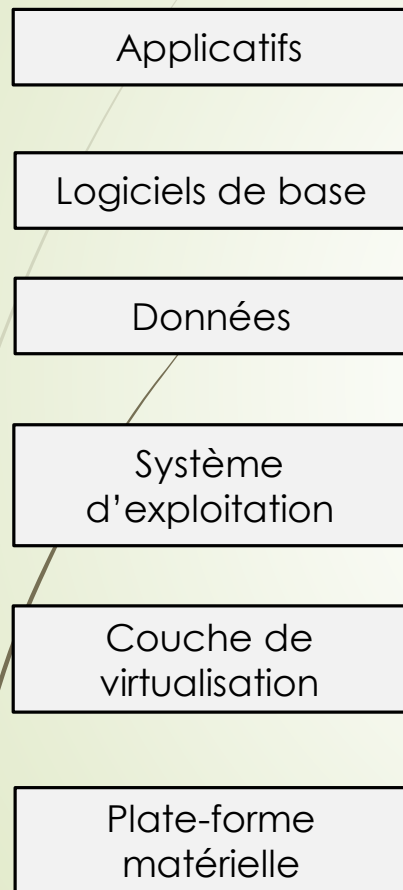
- Une application cliente peut ensuite exploiter ces données pour éventuellement les représenter sur une carte Google Maps en utilisant le service web de Google !

# Les services dans le Cloud

- « Le **Cloud computing** est l'accès via un réseau de télécommunication, à la demande et en libre-service, à des ressources informatiques partagées
- Le mot Cloud, qui signifie « nuage » en anglais, correspond à l'image généralement utilisée dans le monde informatique pour symboliser le réseau Internet.
- Il s'agit d'une forme particulière de gérance de l'informatique, dans laquelle l'emplacement et le fonctionnement dans le nuage ne sont pas portés à la connaissance des clients. »

**Wikipedia**

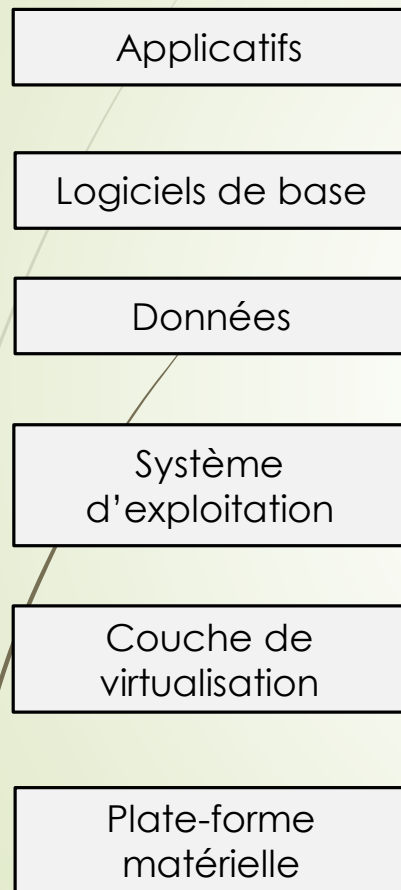
# Différents types de services



Architecture de type SAAS (Software As A Service), l'ensemble de l'architecture logicielle et matérielle est externalisée dans le Cloud



# Différents types de services



Architecture de type SAAS (Software As A Service), la partie applicative reste en interne

Google Apps propose ce type de service ou OneDrive de Microsoft



# Les avantages du Cloud

- Dépend du niveau d'externalisation;
- Pour une solution SAAS :
  - Pas d'investissement
  - Coût prévisible proportionnel à l'utilisation, ce qui est un gros avantage pour les activités cycliques
  - Favorise le travail collaboratif multi-sites

# Les inconvénients du Cloud

- Nécessite une connexion, si on utilise la couche applicative du cloud
- Dépendance à un réseau « public », donc vulnérable
- Dépendance à la qualité du service, rapidité...
- Plus de maîtrise du stockage et de la sécurité des données