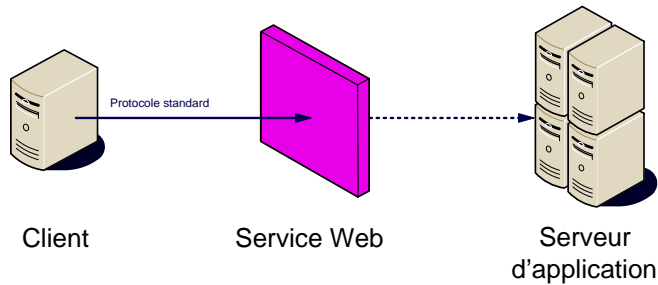


8. Services Web

8.1 Qu'est-ce qu'un service Web ?

8.1.1 Principe

Un service web est une interface basée sur des technologies standards de l'internet permettant d'accéder depuis le réseau à des fonctionnalités applicatives.



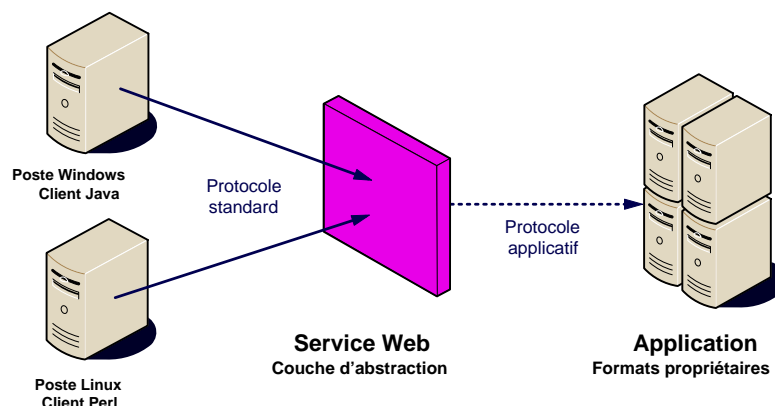
En résumé, dès lors qu'une application est accessible par le réseau à l'aide de protocoles et de formats tels que "*HTTP*", "*XML*", "*SMTP*"... on peut considérer qu'il s'agit d'un service Web.

Vus sous cet angle, les services Web n'apportent rien de particulièrement nouveau par rapport à des mécanismes comme les procédures **cgi** (*Common Gateway Interface*) qui existent depuis l'origine du Web, ou l'échange de documents électroniques (*EDI - Electronic Document Interchange*) tel qu'il est pratiqué par les entreprises depuis de nombreuses années.

Les services Web constituent toutefois la convergence et l'aboutissement de l'expérience accumulée dans les domaines de l'informatique et des réseaux pour l'appel de procédures distantes (*RPC - Remote Procedure Call*), et dans celui de l'industrie par des applications **EDI** basées sur des protocoles comme **CORBA** (*Common Object Request Broker Architecture*).

8.1.2 Interopérabilité

Dans son rôle d'interface, un service web constitue en fait une couche d'abstraction entre l'application et les clients qui accèdent à ses fonctionnalités :



L'interopérabilité est l'un des facteurs clés de succès des services Web.

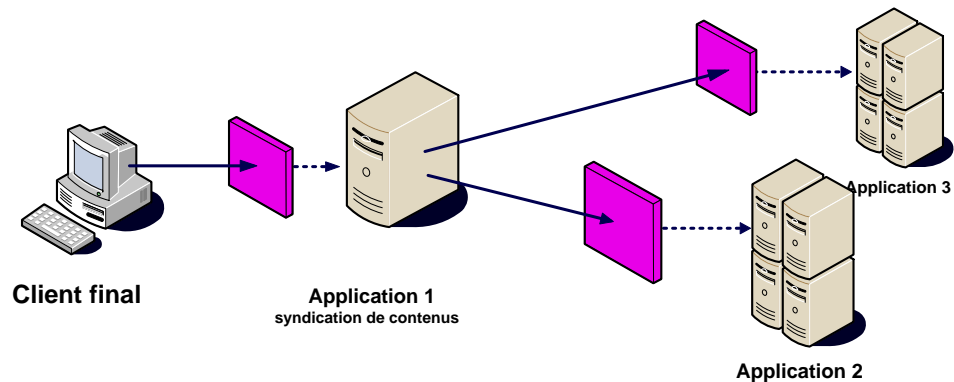
Comme illustré ci-dessus, le protocole standard et la publication des interfaces sont deux conditions qui permettent à des clients variés (*OS, plateformes matérielles, langages, ...*) d'accéder aux fonctionnalités publiques de l'application.

8.1.3 Applications réparties

En toute rigueur, on peut considérer qu'un serveur Web offre un service Web aux navigateurs qui lui adressent des requêtes **HTTP**.

Toutefois, les sites Web délivrent habituellement des informations formatées pour des humains, à la différence des services Web qui renvoient des données plus spécifiquement packagées à destination de procédures informatiques.

La syndication de contenus constitue un exemple classique où une application s'appuie sur des ressources disponibles ailleurs sur le réseau pour construire dynamiquement un ensemble cohérent à destination de l'utilisateur final :



Les services Web permettent ainsi l'existence d'applications réparties faiblement couplées, dont les divers modules peuvent être développés sur des plate-formes hétérogènes par des contributeurs indépendants.

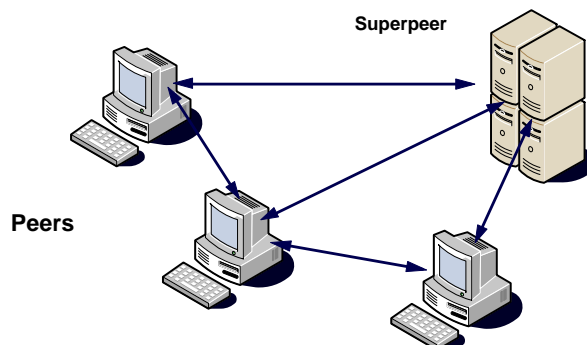
L'interopérabilité est encore une fois assurée par le respect de protocoles standards et la publication des interfaces offerts par chacun des services à ses clients.

8.1.4 Architectures

Les architectures suggérées jusqu'à présent respectaient peu ou prou le modèle client-serveur, à un seul niveau (*modèle classique*) ou multi-niveaux (*modèle n-tier*).

Cependant, la notion de client et de serveur n'est pas forcément statique, et une même machine (*et un même logiciel*) peut se retrouver alternativement dans le rôle de client ou celui de serveur (*architecture peer-to-peer*).

Ce type de fonctionnement est tout à fait compatible avec la notion générique de service Web (*la figure ci-dessous ne représente pas les interfaces*) :

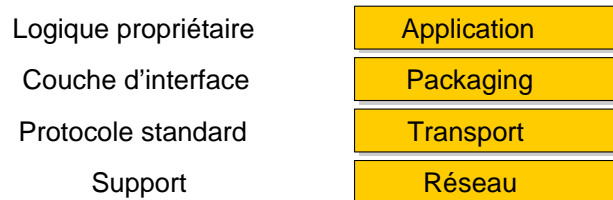


N.B. Ce type d'architecture est fréquent pour des applications comme la messagerie instantanée.

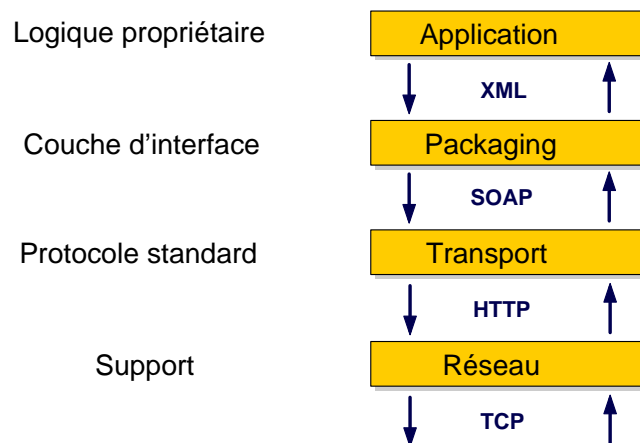
8.2 La pile revisitée

8.2.1 Fonctionnement d'un service Web

Les fonctions constitutives d'un service Web peuvent se décomposer en couches, organisées en pile. Le résultat ressemble aux piles réseau (comme OSI ou IP) :



L'intérêt de cette décomposition, et la clé de l'interopérabilité, sont que la communication entre couches s'effectue à chaque niveau à l'aide de protocoles bien établis, comme illustré ci-dessous.



Cet exemple montre un service Web basé sur l'échange de données **XML**, encapsulées par **SOAP** (*Simple Object Access Protocol*), transportées par **HTTP** sur un réseau **TCP**.

N.B. Ceci n'est bien sûr qu'un exemple. Il existe des formats concurrents pour chacun des niveaux de communication ci-dessus, sur lesquels peut se baser un service Web.

8.2.2 La fonction transport

> Alternatives

En théorie, un service Web peut virtuellement s'appuyer sur n'importe quel protocole de transport fiable.

Les protocoles existants comme **HTTP** (*Hypertext Transfer Protocol*) utilisé par les serveurs Web, ou **SMTP** (*Simple Mail Transfer Protocol*) conçu pour la messagerie, sont bien sûr de bons candidats...

Toutefois, certains services comme **Jabber** définissent leur propre protocole (*ici, XMPP - eXtensible Messaging and Presence Protocol*) qui est alors plus particulièrement adapté aux besoins de ce service (*ici, la messagerie instantanée*). **XMPP** s'appuie lui-même sur **TCP**.

> Critères de choix

Lors de la conception d'un service Web, le choix d'un protocole de transport parmi d'autres s'effectuera en fonction de critères comme la nécessité d'échanges synchrones, l'obligation de gérer des sessions ou la sensibilité aux pare-feux.

> Ressources

- 📄 RFC 793 - Transmission Control Protocol
[\[https://tools.ietf.org/html/rfc793\]](https://tools.ietf.org/html/rfc793)
- 📄 RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1
[\[https://tools.ietf.org/html/rfc2616\]](https://tools.ietf.org/html/rfc2616)
- 📄 RFC 2821 - Simple Mail Transfer Protocol
[\[https://tools.ietf.org/html/rfc2821\]](https://tools.ietf.org/html/rfc2821)
- 📄 RFC 3920 - eXtensible Messaging and Presence Protocol (XMPP): Core
[\[https://tools.ietf.org/html/rfc3920\]](https://tools.ietf.org/html/rfc3920)
- 📄 RFC 3921 - XMPP: Instant Messaging and Presence
[\[https://tools.ietf.org/html/rfc3921\]](https://tools.ietf.org/html/rfc3921)
- 📄 Jabber Software Foundation
[\[https://www.jabber.org/\]](https://www.jabber.org/)

8.2.3 La fonction packaging

> Position du problème

Afin de garantir l'interopérabilité au niveau applicatif, les données échangées par des applications hétérogènes (*OS, plateforme matérielle, langage ...*) doivent être soigneusement préparées et mises sous un format compris par tous.

Les questions qui doivent être résolues ici concernent le type des données échangées, les valeurs minimales et maximales, les codages utilisés, etc...

En langage informaticien on dira qu'il faut définir un format pour la sérialisation des données échangées par l'application.

> Alternatives

A l'heure actuelle (01-2005), **XML** constitue un candidat de choix pour la sérialisation de données.

En effet, comme son nom l'indique **XML** est eXtensible (*modularité, internationalisation...*), il existe énormément d'outils (*transformation, validation, présentation...*), il y a des environnements de développement pour tous les goûts (*commerciaux, publics, open source...*), et surtout, des parseurs **XML** sont disponibles dans tous les langages de programmation.

L'encapsulation de données **XML** aux fins de transport s'effectue alors elle aussi à l'aide d'éléments **XML**, dont la portée est en général cadrée par l'usage d'espaces de noms.

Dans ce contexte on peut considérer **XML-RPC**, plus spécifiquement conçu pour l'appel de procédures distantes (*Remote Procedure Call*), **XMPP** (*eXtensible Messaging and presence Protocol*) dédié au streaming de données **XML**, ou **SOAP** (*Simple Object Access Protocol*) recommandé par le **W3C**, qui s'appuie sur les types de données **XML Schema**.

> Ressources

- 📄 XML-RPC Specification
[\[http://www.xmlrpc.com/spec\]](http://www.xmlrpc.com/spec)
- 📄 RFC 3920 - eXtensible Messaging and Presence Protocol (XMPP): Core
[\[http://localhost/rfc/rfc3920.txt\]](http://localhost/rfc/rfc3920.txt)
- 📄 RFC 3921 - XMPP: Instant Messaging and Presence
[\[http://localhost/rfc/rfc3921.txt\]](http://localhost/rfc/rfc3921.txt)
- 📄 SOAP Version 1.2 Part 1: Messaging Framework
[\[http://www.w3.org/TR/soap12-part1/\]](http://www.w3.org/TR/soap12-part1/)
- 📄 SOAP Version 1.2 Part 2: Adjuncts
[\[http://www.w3.org/TR/soap12-part2/\]](http://www.w3.org/TR/soap12-part2/)
- 📄 XML Schema Part 2: Datatypes Second Edition
[\[http://www.w3.org/TR/xmlschema-2/\]](http://www.w3.org/TR/xmlschema-2/)

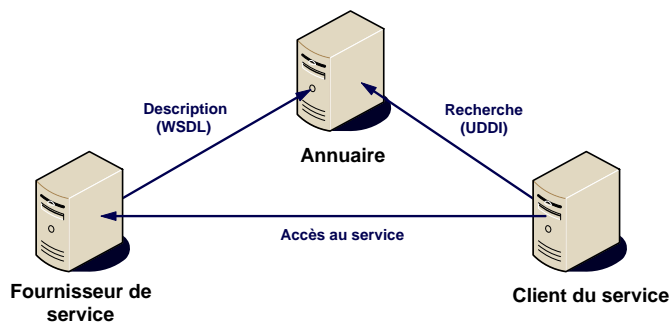
8.3 Environnement complet

8.3.1 Le paysage

Dans le monde du Web (*destiné aux humains*), il est courant de se servir d'un moteur de recherche ou d'un annuaire afin de pouvoir découvrir le site cherché.

D'autre part, afin de faire connaître un site plus rapidement, il est possible (*voire nécessaire*) d'effectuer une démarche afin de le référencer auprès des moteurs de recherche et des annuaires.

Dans le monde des services Web aussi, un service doit être décrit et référencé afin de pouvoir être découvert. Description et découverte sont des opérations qui doivent pouvoir s'effectuer de manière automatique, entre procédures de bonne compagnie :



Il existe des protocoles spécifiques pour chacune de ces étapes. La figure ci-dessus en suggère deux : **WSDL** (*Web Service Description Language*) et **UDDI** (*Universal Description, Discovery, and Integration*), mais il en existe d'autres (*vus plus loin*).

N.B. L'intérêt d'automatiser la découverte d'un service réside bien sûr dans la possibilité d'effectuer cette opération le plus tard possible, juste à temps, au moment où l'on ressent le besoin d'un service particulier (*just in time integration*).

8.3.2 Description d'un service Web

> Position du problème

Pour se référer encore à l'image du Web classique, destiné aux humains, le référencement d'un site nécessite de communiquer à l'annuaire l'**URL** du site, son nom, une liste de mots-clés, du texte décrivant son contenu, etc...

Dans le monde de services Web, où l'objectif est de pouvoir automatiser la découverte d'un service, puis son exploitation, les informations à fournir en plus de son adresse sont plus abstraites : type des données, nom des procédures, protocoles, liste des messages...

> Les solutions

WSDL (*Web Service Description Language*) constitue le standard de fait, pour la description de services Web. **WSDL** est actuellement (01/05) un standard en cours de mise place (*WD - Working Drafts*) auprès du **W3C**.

N.B. **WSDL** est conçu comme un mécanisme générique pour décrire toutes sortes de services Web. Toutefois, certains d'entre eux se prêtent plus facilement à cet exercice, et en particulier les services basés sur **SOAP** (*Simple Object access Protocol*).

Il existe également d'autres standards, moins employés, comme **RDF** (*Resource Description Framework*) ou **DAML** (*DARPA Agent Markup Language*). Ces deux langages sont nettement plus riches et plus puissants que **WSDL**, mais aussi plus complexes à mettre en oeuvre.

> Ressources

 W3C - Web Services Working Group

[\[http://www.w3.org/2002/ws/desc/\]](http://www.w3.org/2002/ws/desc/)

 W3C - XML Protocol Working Group

[\[http://www.w3.org/2000/xp/Group/\]](http://www.w3.org/2000/xp/Group/)

 The DARPA Agent Markup Language Homepage

[\[http://www.daml.org/\]](http://www.daml.org/)

 Resource Description Framework (RDF)

[\[http://www.w3.org/RDF/\]](http://www.w3.org/RDF/)

8.3.3 Recherche d'un service Web

> Position du problème

La description d'un service permet à ses clients potentiels de construire correctement les requêtes qui leur permettront d'en bénéficier. Cette étape est nécessaire mais pas suffisante : il manque toujours un annuaire où ces informations pourront être consultées afin de localiser les services disponibles.

Bien entendu, l'enregistrement d'un service, aussi bien que l'interrogation de l'annuaire devront être entièrement automatiques, et être accessibles sous forme de services Web.

> Les solutions

UDDI (*Universal Description, Discovery and Integration*) est un protocole de service qui répond à ce problème. **UDDI** est développé par le consortium **OASIS**.

L'enregistrement et la consultation constituent deux services Web qui s'appuient sur **SOAP** (*Simple Object Access Protocol*). Ils sont décrits à l'aide de documents **WSDL** (*Web Service Description Language*).

> Ressources

 UDDI Home page

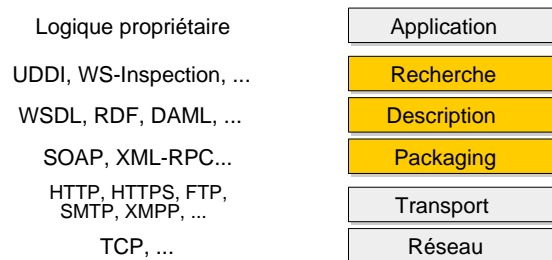
[\[http://uddi.xml.org/\]](http://uddi.xml.org/)

 W3C - Web Services Working Group

[\[http://www.w3.org/2002/ws/desc/\]](http://www.w3.org/2002/ws/desc/)

8.3.4 Retour sur la pile

La plupart des documents et auteurs chargent la pile des services Web avec l'étape de description et les mécanismes de publication et de recherche/découverte :



Cette vision est cohérente dans la mesure où chacune des couches sert d'abstraction à la suivante. La démarche associée consiste donc clairement à isoler les applications (*et leurs programmeurs*) de la mécanique **XML** sous-jacente.

> Remarques

Cette pile vient en général se compléter d'éléments concernant la sécurité, le routage, la gestion de transactions...

Il n'y a pas à l'heure actuelle de consensus sur l'ordre des opérations, les formats et protocoles ou l'intégration des mécanismes entre eux.

Les divers organismes et éditeurs en présence sur le marché des services Web (*Microsoft, IBM, W3C, ...*) proposent chacun leur vision et leur approche qui, si elles sont à peu près compatibles sur les fondamentaux (*HTTP, SOAP, WSDL*), ne le sont pas encore pour les éléments plus avancés.

 Web Services Architecture

[\[http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/\]](http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/)