International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

Déploiement D'une Architecture Client-Serveur Pour La Gestion Des Données à La Direction Générale Des Recettes Du Kasaï Central

ESOKOWA SHOKO Fiston

Assistant à la Faculté des Sciences Informatiques, Université de Kananga, ville de Kananga, en République démocratique du Congo.
Licencié en Informatique à la Faculté des Sciences Informatiques, Université de Kananga, Ville de Kananga, en République démocratique du Congo.
Contact : +243 827176153
E-mail : esokowaf@gmail.com

(cc) BY

Résumé – Le projet de déploiement d'une architecture client/serveur pour la gestion des données à la direction générale des Recettes du Kasaï central vise à moderniser et centraliser la gestion des recettes fiscales et autres données financières au sein de cette institution. L'objectif principal est de fournir une solution numérique efficace et sécurisée pour améliorer l'efficacité des services, réduire les erreurs humaines et optimiser la gestion des informations fiscales.
L'objectif général de ce projet est de concevoir et de déployer une application client/serveur permettant une gestion fluide et sécurisée des données. Cette solution centralisée vise à faciliter l'accès aux informations et à améliorer les processus administratifs tout en garantissant la sécurité et la transparence des données.
Les objectifs spécifiques incluent la création d'une architecture technique adaptée aux besoins de la direction, où un serveur central héberge la base de données et les services applicatifs, tandis que les clients accèdent à ces informations via un réseau sécurisé. Il s'agit également de développer des interfaces utilisateur simples et conviviales pour faciliter la prise en main de l'application par les agents de la direction. De plus, des mécanismes de sécurité, comme le chiffrement des communications et l'authentification des utilisateurs, seront intégrés pour protéger les données sensibles.
En conclusion, ce projet vise à automatiser les tâches de gestion des recettes, à améliorer l'efficacité administrative et à renforcer le contrôle fiscal dans la région du Kasaï Central. Il permettra de centraliser et de sécuriser les données, réduisant ainsi les erreurs humaines et facilitant un suivi plus rigoureux des recettes. Cette solution contribuera à une meilleure transparence et à une gestion plus efficace des finances publiques.

Mots Clés – Déploiement, Architecture, Conception,Réalisation, Application,Client,Serveur, Gestion ,Informatiques,Direction Générale,Recettes, Kasaï Central.

Abstract – The project to deploy a client/server architecture for data management at the Central Kasai Revenue Department aims to modernize and centralize the management of tax revenues and other financial data within this institution. The main objective is to provide an efficient and secure digital solution to improve service efficiency, reduce human error and optimize the management of tax information.
The overall objective of this project is to design and deploy a client/server application enabling fluid and secure data management. This centralized solution aims to facilitate access to information and improve administrative processes, while guaranteeing data security and transparency.
Specific objectives include the creation of a technical architecture tailored to management needs, where a central server hosts the database and application services, while customers access this information via a secure network. The aim was also to develop simple, user-friendly interfaces to make it easier for the department's staff to get to grips with the application. In addition, security mechanisms such as communication encryption and user authentication will be integrated to protect sensitive data.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

In conclusion, this project aims to automate revenue management tasks, improve administrative efficiency and strengthen tax control in the Kasaï Central region. It will centralize and secure data, reducing human error and facilitating more rigorous revenue tracking. This solution will contribute to greater transparency and more efficient management of public finances.

Keywords – Deployment, Architecture, Design,Realization, Application,Client,Server, Management ,IT,General Management,Revenue, Kasaï Central.

0.Introduction

L'essor technologique et l'augmentation des volumes de données ont transformé la manière dont les administrations publiques gèrent et traitent l'information. La Direction générale des Recettes du Kasaï Central, qui est en charge de la collecte, du suivi et de la gestion des recettes fiscales de la région, n'échappe pas à cette dynamique de modernisation. En effet, face aux défis de gestion manuelle et aux risques d'erreurs associés, il est devenu primordial de mettre en place une solution informatique adaptée. C'est dans ce contexte que le déploiement d'une architecture client/serveur pour la gestion des données informatiques prend tout son sens.

Ce projet vise à concevoir et réaliser une application performante et sécurisée qui permette à cette institution de centraliser et de rationaliser la gestion des données fiscales. L'architecture client/serveur a été choisie pour sa capacité à offrir une communication fluide et efficace entre le serveur central, qui regroupe toutes les données, et les différents clients (utilisateurs) répartis dans les bureaux de la direction. L'application ainsi déployée doit non seulement simplifier les processus de saisie et de traitement des données, mais aussi renforcer la sécurité des informations, en particulier celles relatives aux transactions fiscales.

Le but de cette initiative est d'améliorer l'efficacité des services fiscaux, en assurant une gestion rapide, précise et en temps réel des informations. Ce projet ambitionne également de réduire les risques d'erreurs humaines et de corruption, tout en facilitant la prise de décision grâce à des outils de reporting et de suivi. Dans cette optique, la mise en œuvre d'une telle architecture représente un pas important vers la digitalisation des services publics du Kasaï Central, offrant ainsi un modèle de gestion plus moderne et transparent.

0.1. État De La Question

L'idée de la configuration d'une architecture client-serveur pour la gestion des données dans la direction générale des recettes du Kasaï central nous est venue au cours de l'année 2024.

Jusqu'à ce temps-là, seules certaines entreprises étaient équipées d'applications informatiques chargées de gérer l'accès aux ressources de l'ordinateur sur base de temps alloué à l'Internet, au moment où d'autres utilisaient un système manuel.

Les applications utilisées par mes prédécesseurs fonctionnent soit en réseau en utilisant le fournisseur d'accès (FAI), C'est le cas du logiciel CHRISALID conçu par la maison DIGITAL CHRISALID basée à Kampala, utilisé dans les cybercafés tels que DATCO, Goma Mobile Center et pour ne citer que ceux-là ; ou soit hors réseau. C'est-à-dire monoposte, comme c'est le cas du CYBERGESTION conçu par MUTULWA KWABENE Ellis. [1]

Vu les différentes failles que présentent ces applications, à savoir : le manque d'une base de données pour celles travaillant en réseau et le contrôle quasi inexistant d'accès aux ressources pour celles travaillant hors réseau, nous envisageons de mettre en place une application de type client-serveur afin de suppléer tant soit peu au problème de gestion de la direction générale des recettes du Kasaï central dans la ville de Kananga en général et pour la direction générale des recettes du Kasaï central en particulier.

Ce système permettra au gestionnaire de contrôler les permissions d'exécution des tâches tout en évitant ainsi un manque à gagner de sa part.

0.2. Problématique

Vu le nombre d'internautes qui s'accroît de jour en jour à la direction générale des recettes du Kasaï central, le système informatique actuel de la direction générale des recettes ne permet pas d'avoir une bonne gestion. Cela se remarque par l'impossibilité à l'opérateur d'aider les internautes en mauvaise navigation suite aux demandes des nouveaux clients qui se présentent presque à chaque minute ; ou encore des clients qui, par oubli ou par mauvaise intention, en profitent pour naviguer sans être contrôlés.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

La réception enregistre journalièrement des pertes lors des remboursements suite à l'embrouillement causé par un grand nombre de clients qui arrivent presque au même moment.

De ce qui précède, nous nous posons la question suivante :

– Quels sont les besoins spécifiques en matière de gestion des données fiscales à la direction générale des recettes du Kasaï central et comment peuvent-ils être pris en compte dans la conception de l'architecture client-serveur ?

– Quels sont les outils et technologies les plus adaptés pour développer une application client-serveur efficace et sécurisée pour la gestion des données fiscales à la direction générale des recettes du Kasaï central ?

– Comment garantir la sécurité des données fiscales traitées par l'application client-serveur, notamment en termes de protection des données sensibles, de contrôle d'accès et de respect de la législation en matière de protection des données ?

0.3. Objectifs du travail

0.3.1. Objectif général

L'objectif général est celui de développer une architecture client-serveur pour la gestion des données informatiques à la direction générale des recettes du Kasaï central. Cette architecture permettra de collecter, stocker, transférer et traiter les données fiscales de manière efficace et sécurisée.

0.3.2. Objectifs spécifiques

– Identifier les besoins en matière de gestion des données fiscales à la direction générale des recettes du Kasaï central.

– Concevoir une architecture client-serveur appropriée pour répondre aux besoins identifiés.

– Développer une application pour le traitement et la gestion des données fiscales, en utilisant des technologies et des outils appropriés.

I.1. L'architecture client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur ; cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, …

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, de client de messagerie, …, lorsque l'on désigne un programme, tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès du serveur (dans le cas du client FTP, il s'agit de fichiers, tandis que pour le client de messagerie, il s'agit de courrier électronique).

Dans un environnement purement client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est l'un des principaux atouts de ce modèle. [2]

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

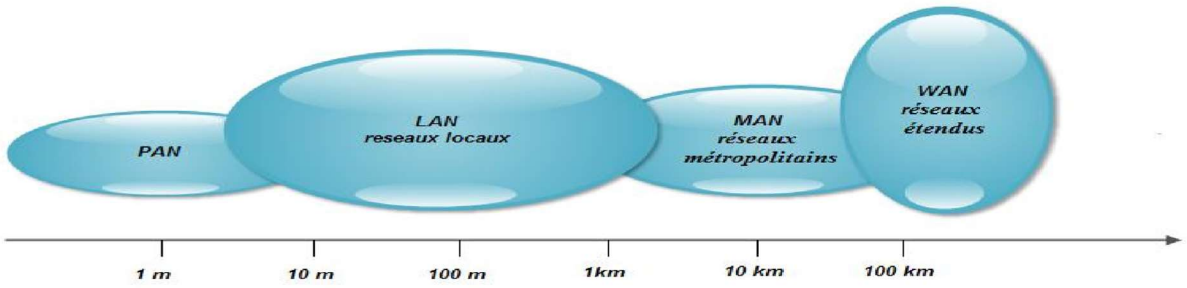


Figure I.1 : Classification des réseaux informatiques selon l'envergure

I.1.4.2- Selon la topologie :

On peut différencier les réseaux selon leur topologie, on trouve :

a. En bus :

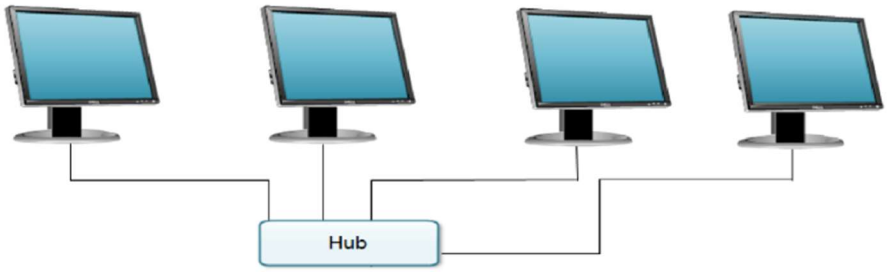
Cette topologie consiste à relier chaque ordinateur à un bus par l'intermédiaire de câbles coaxiaux, l'information envoyée par un poste est diffusée en même temps vers tous les utilisateurs mais seul le poste destinataire est censé le prendre en compte. [3]

Figure .I.2 : topologie en bus

b. en étoile :

Les stations du réseau communiquent directionnellement avec le serveur. C'est une liaison point à point entre le serveur (hub ou Switch) et chaque machine. Deux stations peuvent échanger des données à condition d'en passer par le serveur. [13]

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279



FigureI.3 : Topologie en étoile

c. en anneau :

Les stations sont connectées par un câble formant un anneau (ring). Dans cette topologie le message est transmis d'une station à une autre jusqu'à destination. Chaque station recevant un message vérifie s'il lui est destiné si c'est le cas elle le garde sinon elle le transmet à la station voisine. [13]

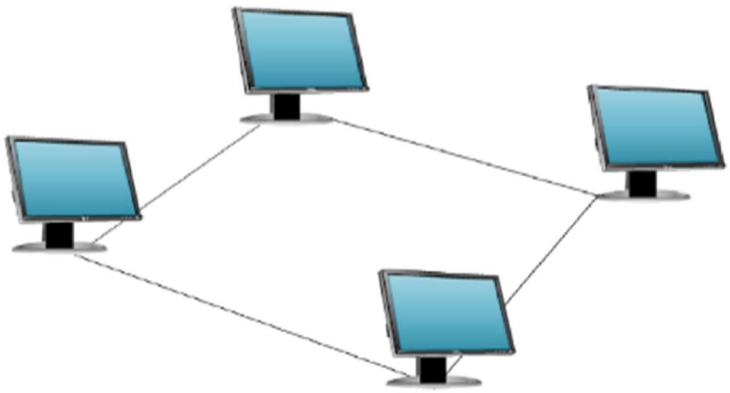


Figure I.4: Topologie en anneau

d. réseau maillé :

Un réseau maillé est caractérisé par le fait que deux nœuds quelconques soient reliés l'un à l'autre. Ce type de réseau permet plus de souplesse et de fiabilité dans son utilisation, mais il est difficilement envisageable pour un grand nombre de nœuds. Généralement, le maillage n'est pas parfait, on dit que c'est un réseau partiellement maillé. Tous les autres types de réseaux sont des sous-ensembles de ce type, obtenus en posant des conditions sur les liens entre les nœuds. [13]

Remarque : Toute topologie adoptée doit faire au préalable l'objet d'une étude prenant en compte plusieurs facteurs : Nombre de stations à connecter, flux des données, Coût, distance entre entités communicantes, évolution possible, Résistance aux pannes et lignes de secours, administration …etc.

I.1.5- Modes de transmission de données dans un réseau informatique :

Le but de la transmission de données est de les acheminer d'une source vers une destination d'une façon synchrone ou asynchrone, en parallèle ou en série.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

I.1.5.1- Transmission parallèle et transmission série :

- **Transmission parallèle :**

Les bits d'un même caractère sont envoyés sur des fils distincts pour arriver simultanément à destination. Elle est rapide mais très couteuse et très encombrante surtout sur les longues distances.

- **Transmission série :**

Contrairement à la transmission en parallèle, les bits sont envoyés l'un derrière l'autre sur un même fil de transmission

I.1.5.2- Transmission synchrone et transmission asynchrone :

- **Transmission synchrone :**

Les caractères sont envoyés de manière irrégulière, l'intervalle de temps entre deux caractères est aléatoire, ils sont accompagnés de deux signaux START et STOP permettant de spécifier le début et la fin de chaque caractère.

- **Transmission asynchrone :**

Un fil particulier reliant la source à la destination transporte le signal d'horloge et les bits sont transmis l'un à la suite de l'autre à chaque top d'horloge.

I.1.6- Modes d'exploitation :

Il existe plusieurs types d'exploitation des canaux de transmission :

I.1.6.1- Le mode simplex :

On parle de transmission unidirectionnelle si une seule extrémité peut émettre.

I.1.6.2- Le mode semi duplex :

Appelé aussi bidirectionnel à l'alternat, permet une transmission dans les deux sens mais pas en même temps.

I.1.6.3- Le mode duplex :

C'est du bidirectionnel simultané, permet une transmission dans les deux sens simultanément.

I.1.7- Architecture en couche :

Le transport des données d'une extrémité à l'autre d'un réseau nécessite un support physique de communication. Cependant, pour que ces données arrivent correctement à la destination avec la qualité de service exigée, il faut une architecture logicielle. L'ensemble de protocoles nécessaires constitue une architecture. Un protocole est l'ensemble de règles qui définissent les modalités de fonctionnement d'une communication entre deux ordinateurs. [9][10]

I.1.7.1 Modèle OSI (Open System Interconnexion) : Que l'on appelle Open System Interconnexion ou Systèmes Ouverts, c'est une architecture qui forme le modèle de référence qui comporte sept couches. Cette architecture est illustrée par la **figure I.5 [4]**

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

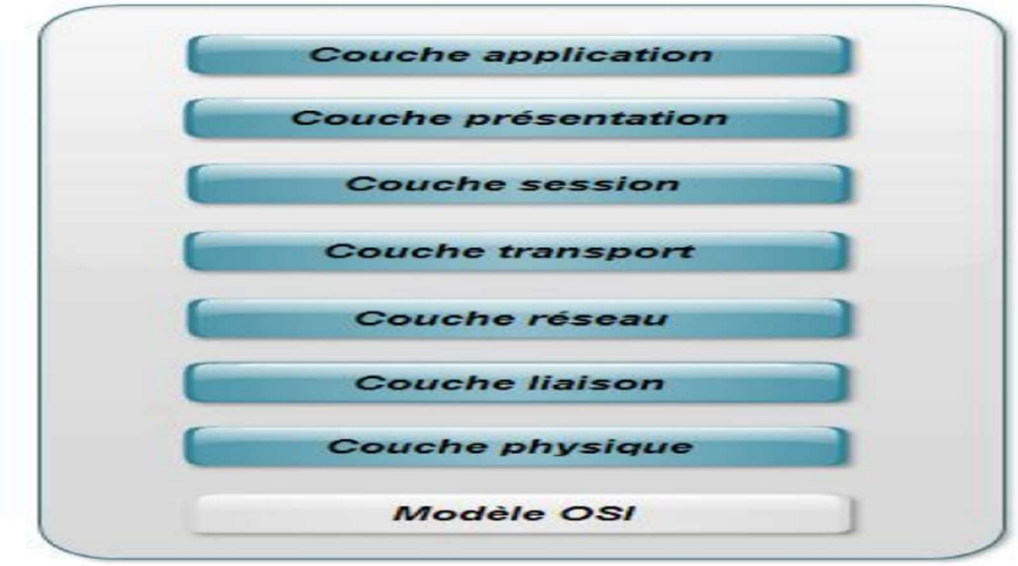


Figure I.5: Les couches du modèle OSI.

1. **La couche physique :**

Elle est responsable de la manipulation des détails mécaniques et électriques de la transmission physique d'un flot de bits sur le canal de communication.

2. **La couche de liaison :**

Sa tâche principale est de détecter et de corriger les erreurs de transmission, l'émetteur traite également les accusés de réception émis par le récepteur.

3. **La couche réseau :**

Elle s'occupe du routage et du contrôle de congestion. L'information stockée au niveau de cette couche est structurée sous forme de paquets de données.

4. **La couche transport :**

Elle est responsable du transfert de données entre les entités de session. Ce transport devant être transparent, c'est-à-dire indépendant de la succession des caractères et même des éléments binaires transportés.

5. **La couche session :**

Elle permet à des utilisateurs de différentes machines d'établir des sessions entre eux.

Parmi ses services : la gestion du dialogue, le transfert de fichiers, …

6. **La couche présentation :**

Elle gère la syntaxe et la sémantique des informations transmises (le codage, la compression, la cryptographie des données, …).

7. **La couche application :**

Elle est responsable de l'interaction directe avec les utilisateurs. Elle traite les protocoles de connexion à distance, le courrier électronique ainsi que la définition des terminaux virtuels. I.1.7.2- L'architecture TCP/IP (Transmission Control Protocol / Internet Protocol) : Dans les années 70, DOD (Department Of Defense), devant le foisonnement de machines utilisant des protocoles de

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

communication différents et incompatibles, décide de définir sa propre architecture. L'architecture TCP/IP est à la source du réseau Internet de nombreux réseaux privés. Cette architecture comme celle d'OSI est répartie en couches **(figure I.4)**.

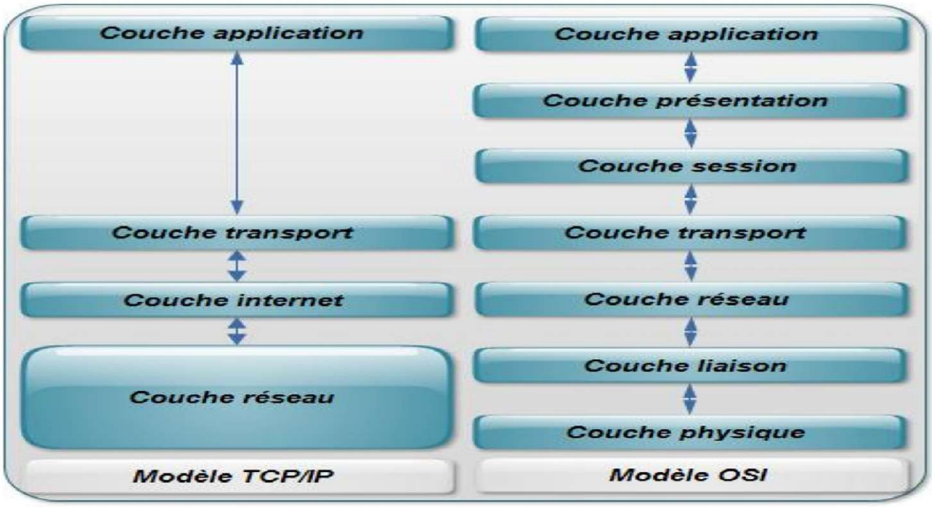


Figure I.6 : La pile de protocole TCP/IP avec sa correspondance OSI.

➢ **La couche application :**

Elle englobe les applications standard du réseau :

SMTP (Simple Mail Transport Protocol) : permet d'échanger du courrier entre deux serveurs de messagerie.

TELNET : Protocole permettant de se connecter sur une machine distante (serveur) en tant que utilisateur.

FTP (File Transfert Protocol) : Est un protocole permettant d'échanger des fichiers via Internet.

DNS (Domain Name Service) : Service de nom du domaine. Il permet de convertir le nom d'une machine en une adresse réseau (ou IP) et vice versa.

➢ **La couche transport :**

Elle assure l'acheminement des données sur le réseau local et les mécanismes permettant de connaitre l'état de la transmission. Les protocoles de cette couche permettent d'envoyer des informations d'une machine à une autre.

TCP : Il assure le contrôle de données, orienté connexion.

UDP (User Datagram Protocol) : Archaïque et non orienté connexion, il n'assure aucun contrôle de transmission de données.

➢ **Couche Internet :**

La couche Internet est chargée de fournir le paquet de données. Elle contient cinq protocoles, les trois premiers sont les plus importants :

IP : gère les destinations des messages, adresse du destinataire.

ARP (Address Resolution Protocol) : gère les adresses des cartes réseau. Chaque carte a sa propre adresse d'identification codée sur 48 bits.

ICMP (Internet Control Message Protocol) : gère les informations relatives aux erreurs de transmission. Il ne corrige pas les erreurs, mais signale aux autres couches que le message contient des erreurs.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

RARP (Reverse Address Resolution Protocol) : gère l'adresse IP pour les équipements, qui ne peuvent s'en procurer une, par lecture d'information dans un fichier de configuration. En effet, lorsque le PC démarre, la configuration réseau lit l'adresse IP qu'elle va utiliser.

Ceci n'est pas possible dans certains équipements qui ne possèdent pas de disques durs (imprimantes, terminaux).

IGMP (Internet Group Management Protocol) :

Permet d'envoyer le même message à des machines faisant parti d'un groupe. Ce protocole permet également à ces machines de s'abonner ou de se désabonner d'un groupe. Ceci est utilisé par exemple dans la vidéo conférence à plusieurs machines, envoi de vidéos, …

- les réseaux mobiles :

La première génération est apparue à la fin des années 70, le réseau est composé de cellules, c'est-à-dire de zones géographiques limitées à quelques kilomètres qui recouvrent le territoire de l'opérateur. Ces cellules se superposent partiellement pour assurer une couverture complète du territoire cible. Le mobile communique par une interface radio, avec l'antenne qui joue le rôle d'émetteur/récepteur de la cellule.

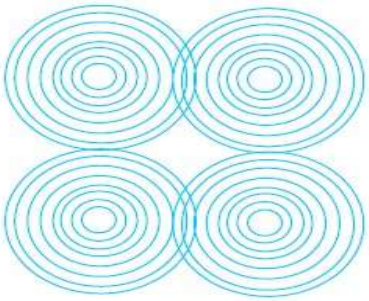


Figure I.7 : Les cellules des réseaux mobiles

I.2- L'architecture client/serveur :

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacité d'entrée-sortie, qui leur fournit des services. [7]

L'architecture client/serveur a deux composants principaux :

- **Un composant client** : c'est une machine qui fournit à l'utilisateur toute la gamme de ses services pour exécuter des applications.
- **Un composant serveur** : c'est une machine généralement très puissante en terme de capacité d'entrée/sortie, qui fournit aux clients des services de gestion de données, de partage d'information, d'administration du réseau et de sécurité.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

I.2.1- Fonctionnement d'une architecture client/serveur :

Une architecture client/serveur fonctionne selon le schéma suivant :

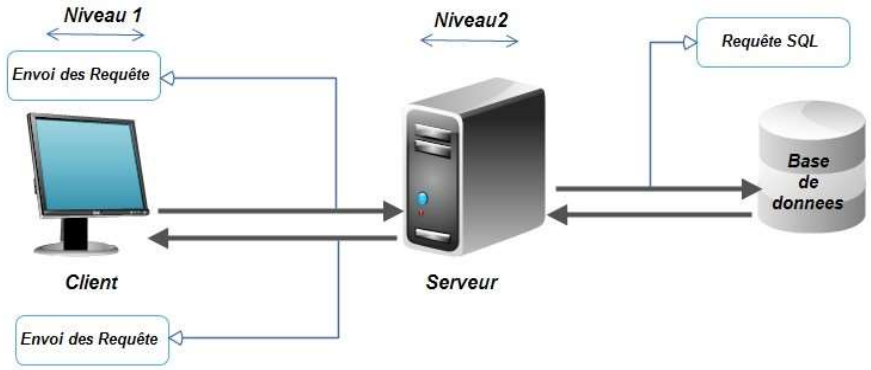


Figure I.8 : Architecture client/serveur.

1. Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur. [6]
2. Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port. [6]

I.2.2- Classification des architectures clients/serveurs :

I.2.2.1- Architecture à 2 niveaux :

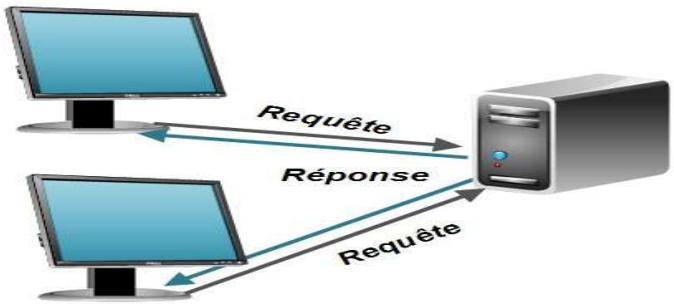


Figure I.9: Architecture client/serveur à 2 niveaux.

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tier signifiant étage en anglais) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service. [7]

Avantages :

- **Des ressources centralisées** : Etant donné que le serveur est au cœur de

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

L'infrastructure, il peut gérer des ressources communes à tous les clients, comme par exemple une base de donnée centralisée afin d'éviter les problèmes de redondance et d'incohérence.

- **Une meilleure sécurité** : le nombre de points d'accès aux données étant moins important.
- **Un réseau évolutif :** on peut supprimer et/ou rajouter des clients sans perturber le fonctionnement du réseau et sans faire des grandes modifications.

Inconvénients :

- Dépendance totale avec l'architecture des réseaux et les systèmes d'exploitation.

  Une modification de l'application ou de la structure de la base de données nécessite un redéploiement sur les postes clients.
- Coûts élevés, nécessite des serveurs très performants et une connexion réseau à haut débit.

I.2.2.2- Architecture à 3 niveaux :

- Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre
- Un **client** : le demandeur de ressources, équipé d'une interface utilisateur (généralement un navigateur Web) chargé de la présentation. [06]
- Le **serveur d'applications** (appelé également **middleware**) : chargé de fournir la ressource mais faisant appel à un autre serveur.
- Le **serveur de données** : fournissant au serveur d'application les données dont il a besoin.

En plus des avantages d'une architecture client/serveur, cette infrastructure procure :

•L'administration est centralisée au niveau serveur pour tous les clients ayant accès.

•Une plus grande sécurité : elle est définie pour chaque service.

•Permet une configuration matérielle et logicielle hétérogène des postes clients.

•Tendance au couplage faible : possibilité de remplacer un composant par un autre

•Passage à l'échelle, gestion d'utilisateurs concurrents.

Inconvénients :

•Ticket d'entrée technologique plus élevé

•Applications transactionnelles parfois complexes à développer.

I.2.2.3 – Architecture multi-niveaux :

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux. [06]

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

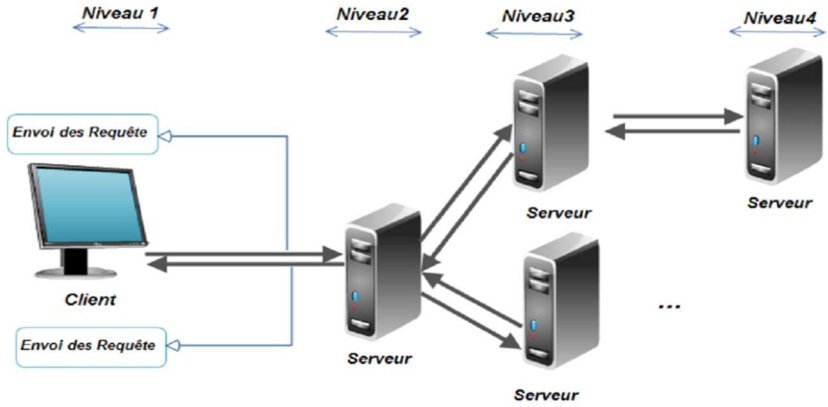


Figure I.11 : Architecture client/serveur à N niveaux.

I.2.3 – Les apports et les contraintes de l'architecture client/serveur

Dans ce qui suit, nous allons voir concrètement quels sont les avantages de l'architecture client/serveur par rapport aux systèmes classiques et quelles sont ses limites.

a) Les apports :

Capacité de traitement : le modèle client/serveur s'appuie sur les progrès réalisés dans le domaine des microprocesseurs. Les techniques qu'il met en œuvre nécessitent des capacités de traitement importantes que les microprocesseurs d'aujourd'hui sont capables de fournir.

Capacité de mémorisation : dans une architecture client/serveur, les postes serveurs doivent gérer simultanément les contextes applicatifs de plusieurs postes clients. Cette architecture sera d'autant mieux adaptée aux environnements transactionnels lourds où les serveurs disposent de grandes capacités mémoire.

Standardisation : en s'appuyant sur les normes standards, le modèle client/serveur assure la portabilité des applications entre systèmes distincts et garantit leur interopérabilité.

Productivité du développeur : l'environnement client/serveur facilite le processus de développement applicatif. Cela est dû notamment à la séparation fonctionnelle qui constitue le principe de base du modèle client/serveur.

Portabilité et interopérabilité : le modèle client/serveur assure le partage d'une même application sur différentes plates-formes, échangeer d'informations et l' interaction de plusieurs applications dans des environnements hétérogènes.

b) Les contraintes de complexité de mise en œuvre : une complexité qui se manifeste notamment lors du découpage des traitements entre partie cliente et partie serveur.

Fragilité relative du système : dans un environnement centralisé, les données sont concentrées en un seul endroit ; le contexte client/serveur peut nécessiter l'installation des données sur des serveurs distincts. Cette répartition géographique des données et l'utilisation même d'un réseau fragilise l'ensemble du système et pose alors le problème de la sécurité des données en environnement réparti.

Investissements initiaux élevés : la mise en place d'une architecture client/serveur nécessite des efforts financiers importants, investissement en matériels et logiciels tels que micro, interfaces graphiques, matériels de connexion.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2025 Scholar AI LLC.
https://ijpsat.org/
Vol. 49 No. 1 February 2025, pp. 266-279

I.2.4 – Comparaison entre l'architecture à deux niveaux et l'architecture à trois niveaux :

L'architecture client-serveur à deux niveaux ne présente aucun inconvénient majeur lorsqu'il s'agit de l'utiliser pour un système à un seul site, bien au contraire, elle offre des avantages concernant la simplicité et la facilité de développement. Mais lorsque le client/serveur s'est agrandi pour traiter des applications critiques d'ordres intergalactiques, l'architecture client-serveur à trois niveaux s'est révélée essentielle, car moins coûteuse. [11]

Conclusion

Le déploiement d'une architecture client-serveur pour la gestion des données informatiques à la direction générale des Recettes du Kasaï Central représente une avancée significative pour améliorer l'efficacité administrative et la gestion des ressources fiscales de la région. Cette solution a permis de centraliser les données tout en assurant une gestion sécurisée et optimale des informations critiques relatives aux recettes publiques.

À travers la conception et la réalisation de l'application, il a été possible de résoudre de nombreux défis auxquels la direction était confrontée, tels que la gestion manuelle des dossiers fiscaux, les risques d'erreurs humaines et l'inefficacité des processus de collecte et de traitement des données. L'architecture client-serveur, en permettant une communication fluide entre les utilisateurs et le serveur central, garantit une mise à jour en temps réel des informations, facilitant ainsi une gestion proactive des données fiscales.

De plus, l'application mise en place offre une interface intuitive et conviviale pour les utilisateurs, réduisant ainsi le besoin de formations complexes. Elle intègre également des mécanismes de sécurité robustes pour protéger les informations sensibles contre tout accès non autorisé et assure une traçabilité de toutes les opérations effectuées.

En conclusion, le déploiement de cette architecture client-serveur à la Direction générale des Recettes du Kasaï central marque un tournant important dans la modernisation des infrastructures fiscales de la région. Non seulement l'application répond aux besoins actuels de gestion des données, mais elle pose également les bases d'une évolution future vers une gestion numérique encore plus performante, adaptable et sécurisée. Cette initiative ouvre la voie à une administration publique plus moderne, réactive et transparente.

Références

[1] MUTULWA KWABENE Ellis Université LYON 1/UFR d'Informatique

Olivier.Gluck@enslyon.fr http://www710.univ-lyon1.fr/~ogluck.

[2] : livre Cisco

[3] : Les réseaux Edition 2008 par Goylujolle avec la colaboration de

Oliser Selvatori et la contribution de Jacque Nozick

[4] http://www.commentcamarche.net/contents/cs/cs3tier.php3.

[5] : http://www.commentcamarche.net/contents/oracle/oracproc.php3.

[6] : thèse ingénieur, université Mouloud MAMMERI Tizi-Ouzou,

[7] www.Oracle.com

[8]www.développer.com

[9]www.labo-oracle.com

[10]www.netbeans.org

[11] : thèse ingénieur, université Mouloud MAMMERI Tizi-Ouzou,conception et réalisation d'une application client-serveur 3tiers sous Oracle cas : gestion de la comptabilité général de l'ENIEM.,réalisé par NAIT BELAID Ouerdia et HABI Fetta.

[12] :BERGERON P.G., La gestion dynamique : concepts, méthodes et applications, 2éd, Gaétan Morin, Paris, 1995

[13] :TARDIEU, H., ROCHEFELD, A. et COLETTI R. *La méthode Merise principes et outils* édition d'organisation, 1986

[14] :MATHERON J.P ; *comprendre Merise : outils conceptuel et organisationnel*, 5e tirage, éditions Eyrolles, Paris, 1998

[15] :RAYMOND REIX ; *Informatique appliquée à la gestion*, Ed. Faucher, Paris, 1993

[16] :LENTZNER R., Visual Basic 6 et les bases des données, 3ème Edition, Ed. OEM, Paris, 2001