# **Chapitre II. État De L’art**

**Introduction**

Les étapes de résolution de notre problématique nécessitent de passer d’abord par la phase de définition des concepts qui contribuent à la résolution du problème posé, avant de s’attaquer au choix final. Cette approche a pour objectif de détailler les fonctionnalités de notre environnement. Ce chapitre est consacré à la présentation du modèle ISO/OSI et TCP/IP et des topologies existantes. De même, il comporte d’abord la proposition et étude comparative des solutions telles que le système d’exploitation et l’environnement d’implémentation ainsi que l’architecture de l’application et de réseau.

## II.1. Description de l’environnement réseau

### II.1.1. Le modèle ISO-OSI

Si chacune des personnes (physiques ou morales) ne devait échanger des informations qu’avec des gens de sa communauté, alors il n’y aurait pas besoin de normalisation, chaque entité pourra nit échanger ces informations avec des membres de la même entité. Il suffirait que chacune des personnes utilise le même "langage" (protocole) pour échanger ces informations.

Malheureusement (?), de plus en plus d'entité ont besoin d'échanger des informations entre elles (agence de voyage, organisme de recherche, école, militaires, ...). Si chacune de ces entités utilise son réseau (au sens protocole) pour que ces entités puissent communiquer ensemble il faudrait chaque fois réinventer des moyens pour échanger l'information. C'est ce qui se faisait au début.

Des gens ont eu l'idée de réfléchir à ce problème et on essaye de recenser les différents problèmes que l'on trouvait lorsqu'on veut mettre des machines en réseau. De cette réflexion est sortie le modèle OSI de l'ISO.

**1977** : ISO démarre une réflexion sur une architecture de réseau en couches,

**1983** : définition du modèle OSI

Open : systèmes ouverts à la communication avec d’autres systèmes

Systèmes : ensemble des moyens informatiques (matériel et logiciel) contribuant au traitement et au transfert de l’information

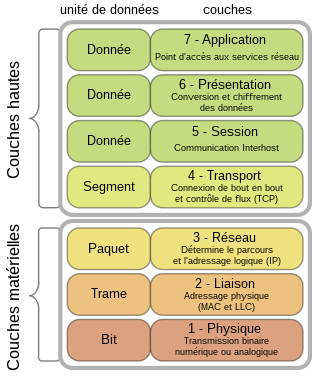
Interconnexion

OSI Garantit que 2 systèmes hétérogènes pourront communiquer si :

* même ensemble de fonctions de communication,
* fonctions organisées dans le même ensemble de couches,
* les couches paires partagent le même protocole.

Le modèle ISO-OSI permet de :

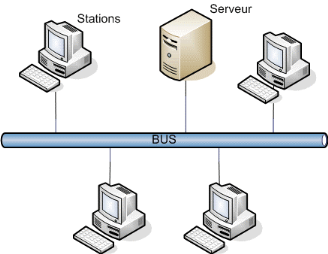
* relier les systèmes par un lien physique (couche PHYSIQUE) ;
* contrôler qu’une liaison peut être correctement établie sur ce lien (couche LAISON) ;
* assurer à travers le relais (réseau) l’acheminement des données et la délivrance au bon destinataire (couche RESEAU) ;
* contrôler, avant de délivrer les données à l’application que le transport s’est réalisé correctement de bout en bout (couche TRANSPORT) ;
* organiser le dialogue entre toutes les applications, en gérant des sessions d’échange (couche SESSION) ;
* traduire les données selon une syntaxe d’échange compréhensible par les deux entités d’application (couche PRESENTATION) ;
* fournir à l’application utilisateur tous les mécanismes nécessaires pour masquer à celle-ci les contraintes de transmission (couche APPLICATION).



**II.1.2. Les topologies existantes**

1. **La topologie : Bus**

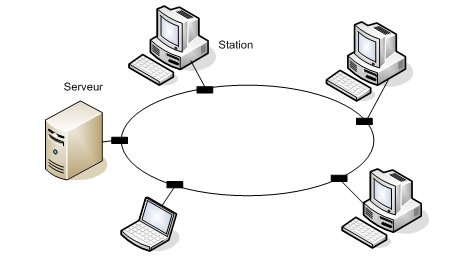
Tous les équipements sont branchés en série sur un câble appelé segment.Chaque poste reçoit l’information mais seul le poste pour lequel le message est adressé traite l’information.On utilise un câble coaxial pour ce type de topologie.L’avantage du bus est sa simplicité de mise en œuvre et sa bonne immunité aux perturbations électromagnétiques**.** Par contre, si le câble est interrompu, toute communication sur le réseau est impossible



1. **La topologie : Anneau**

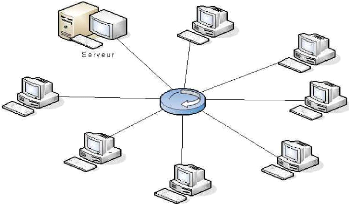
Les équipements sont reliés entre eux en formant une boucle. L’information est gérée comme dans la topologie bus. Chaque station reçoit le message, mais seule la station à qui le message est adressé la traite. On utilise un câble en paires torsadées ou de la fibre optique.

L’avantage est que l’anneau offre deux chemins pour aller d’un point à l’autre. Ceci permet à l’information de passer malgré une coupure sur le câble. On utilise cette topologie pour les réseaux de type Token Ring. Pour augmenter la sécurité, on peut utiliser un double anneau (si le premier anneau est interrompu, les données passent sur l’anneau secondaire, le temps de réparer le premier anneau).



1. **La topologie : Étoile**

Toutes les liaisons sont issues d’un point central. C’est une liaison dite « point à point », c’est à dire que les équipements sont reliés individuellement au nœud central et ne peuvent communiquer qu’à travers lui. On utilise les câbles en paires torsadées ou en fibre optique pour ce type de topologie. L’avantage est que les connexions sont centralisées et facilement modifiables en cas de problème. Si un câble est interrompu, le reste du réseau n’est pas perturbé. L’inconvénient de taille de cette topologie est l’importante quantité de câbles nécessaire.



1. **Classifications : La topologie**

|  |  |  |
| --- | --- | --- |
|  | **Avantages** | **Inconvénients** |
| Topologie en  bus | Facile à mettre en œuvre  Fonctionnement facile | Si l’une des connexions est endommagée l’ensemble du réseau est affecté |
| Topologie en étoile | Si un câble tombe en panne, seul cet ordinateur est isolé du reste du réseau  réseau supporte de fortes  charges | Plus de câbles  Plus cher  Le point central |
| Topologie en  anneau | Bon niveau de sécurité | Plus de câbles  La panne d’une seule machine isole les autres |

**II.2. Matériel de mise en place du réseau Ethernet**

En informatique, les médias d’accès sont les moyens utilisés pour rendre possible la communication (l’échange des informations) entre les ordinateurs. Voyons divers moyens de connecter des ordinateurs entre eux. Un des médias d’accès les plus utilisés est le câble. Les câbles sont des liaisons physiques entre ordinateurs. Mais il en existe différentes sortes, nous allons en voir 2 principales.

### II.2.1. Câble Ethernet

Le câble Ethernet est sûrement le type de câble le plus utilisé pour connecter des ordinateurs entre eux dans un réseau local. Il relie généralement un ordinateur personnel à un routeur (ce que l’on appelle parfois une « box »). Le nom formel de ces câbles est paire torsadée, en anglais twisted pair. Il existe deux types de câble Ethernet : les câbles Ethernet droits et les câbles Ethernet croisés. Ces derniers permettent de relier directement entre eux deux ordinateurs alors que les câbles droits servent à relier un ordinateur à un autre appareil comme un hub ou un switch.

### II.2.2. Le mode sans fil

Le mode sans fil est un réseau informatique numérique qui connecte différents postes ou systèmes entre eux par ondes radio. Il peut être associé à un réseau de télécommunications pour réaliser des interconnexions à distance entre nœuds.

|  |  |
| --- | --- |
| Matériels | Descriptions |
| Carte réseau | Les fonctions de la carte réseau sont :  -préparation pour le câble réseau des données qui seront transmises à partir de l’ordinateur  -envoi des données vers un autre ordinateur  -contrôle du flux de données entre l’ordinateur et le système de câblage |
| Concentrateur(hub) | Élément matériel qui permet de concentrer le trafic réseau provenant de plusieurs hôtes et de régénérer le signal. Servent à relier entre eux toutes les parties d'un même réseau physique. Lorsqu’une information arrive sur un Hub, elle est rediffusée vers toutes les destinations possibles c-à-d vers tous les ports. Les trames envoyées à destination d’une machine sont reçues par toutes les machines. |
| Commutateur(switch) | C’est un pont multi-ports qui analyse les trames arrivant sur les ports d’entrée et filtre les données. Ils assurent le filtrage et la connectivité. A un instant donné, ils ne laissent passer les informations que vers la destination voulue. Diffèrent du hub : les trames envoyées à une machine sont directement conduites vers la machine destinatrice. |
| Router | Le routeur permet d’assurer la communication entre différents réseaux pouvant être fondamentalement différents (réseau local et Internet). |
| Serveur | C’est 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... [1] |

**II.3. Les Protocoles et services standards**

Une norme est l’ensemble des règles qui doivent être respectées pour réaliser un échange entre deux ordinateurs.

Les protocoles sont un ensemble de règles et de procédures qui gèrent les communications entre les ordinateurs. Plusieurs protocoles fonctionnent ensemble afin que les données soient : préparées, transférées, reçues et Traitées.

Le fonctionnement des protocoles doit être coordonné afin d’éviter des conflits ou des opérations incomplètes. Pour garder une certaine cohérence, une structure et une organisation dans le domaine des réseaux, il faut des normes. Ces derniers sont créés et approuvées par un certain nombre d’organismes de normalisation.

### II.3.1. Le modèle TCP / IP

**TCP (protocole de contrôle de transmission) / IP (protocole Internet)**a été développé par le **Département de la Défense (DoD)** agence de projet. Contrairement au modèle OSI, il se compose de quatre couches, chacune ayant ses propres protocoles. Les protocoles Internet sont l'ensemble des règles définies pour la communication sur le réseau. TCP / IP est considéré comme le modèle de protocole standard pour la mise en réseau. TCP gère la transmission de données et IP gère les adresses.

La suite de protocoles TCP / IP comprend un ensemble de protocoles comprenant les protocoles TCP, UDP, ARP, DNS, HTTP, ICMP, etc. Il s’agit d’un modèle robuste et flexible. Le modèle TCP / IP est principalement utilisé pour interconnecter des ordinateurs via Internet.

### II.3.2. Couches de modèle TCP / IP

1. **Couche d'interface réseau**

Cette couche sert d'interface entre les hôtes et les liaisons de transmission et est utilisée pour la transmission de datagrammes. Il spécifie également quelle opération doit être effectuée par des liaisons telles que les liaisons série et Ethernet classique pour répondre aux exigences de la couche Internet sans connexion.

1. **Couche Internet**

Le but de cette couche est de transmettre un paquet indépendant à tout réseau qui se rend à la destination (pouvant résider sur un autre réseau). Il inclut IP (Internet Protocol), ICMP (Internet Control Protocol) et ARP (Address Resolution Protocol) en tant que format de paquet standard pour la couche.

1. **Couche de transport**

Il permet une livraison de bout en bout des données sans faille entre les hôtes source et de destination sous la forme de datagrammes. Les protocoles définis par cette couche sont les protocoles TCP (Transmission Control Protocol) et UDP (User Datagram Protocol).

1. **Couche d'application**

Cette couche permet aux utilisateurs d'accéder aux services d'internet global ou privé. Les différents protocoles décrits dans cette couche sont les suivants : terminal virtuel (TELNET), courrier électronique (SMTP) et transfert de fichiers (FTP). Certains protocoles supplémentaires tels que DNS (Domain Name System), HTTP (Hyper Transfer Protocol) et RTP (Real-time Transport Protocol). Le fonctionnement de cette couche est une combinaison des couches application, présentation et session du modèle OSI.



## II.4. Proposition et étude comparative des solutions

Comme tout système informatique bien conçu, la présentation de ses éléments est nécessaire pour sa mise en œuvre. Notre application sera créée suite aux choix du système d’exploitation, d’un langage de programmation, d’un IDE, d’un serveur Web, d’un client Web, d’un SGBD, d’un Framework.

**II.4.1. Système d’exploitation**

1. **Linux**

Linux est un système d'exploitation [Open Source](https://www.redhat.com/fr/topics/open-source/what-is-open-source). Un [système d'exploitation](https://www.redhat.com/fr/technologies/linux-platforms/old-enterprise-linux) est un logiciel qui gère directement les composants physiques du système ainsi que ses ressources, telles que le processeur, la mémoire et le [stockage](https://www.redhat.com/fr/topics/data-storage/software-defined-storage). Il représente l'interface entre les applications et le matériel. C'est lui qui établit les connexions entre tous les logiciels et les ressources physiques requises pour l'exécution des différentes tâches [b1]. États des sources : Open source

1. **Microsoft Windows**

Windows (littéralement « Fenêtres » en anglais) est une gamme de systèmes d’exploitation produite par Microsoft, principalement destinés aux ordinateurs compatibles PC. C’est le successeur de [MS-DOS](https://www.techno-science.net/glossaire-definition/MS-DOS.html). Depuis les années 1990, et notamment la sortie de Windows 95, son succès [commercial](https://www.techno-science.net/definition/4832.html) pour équiper les ordinateurs personnels est tel qu’il possède un statut de quasi-monopole [b2]. États des sources : Source fermée

**Choix :** Au cours de notre stage nous avons trouvé que l’Institut Africain de Technologie disposait déjà de deux (2) serveurs qui tournent sous Windows Server 2019 qui est un système d’exploitation pour serveurs développé par Microsoft et que tous les postes clients sont sous Windows avec des différentes versions.

### II.4.2. Environnement d’implémentation

### 1. Environnement de développement intégré (IDE)

C’est un logiciel de création d’application, qui rassemble des outils de développement fréquemment utilisés dans une seule interface utilisateur graphique. Voici quelques-unes des meilleures outils IDE :

1. **Visual Studio Code**

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippers, la factorisation du code et Git intégrer. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

1. **Sublime Text**

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour vim, riche en fonctionnalités. Depuis la version 2.0, sortie le 26 juin 20122, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares. [17]

**Choix :** L’IDE sur laquelle notre choix est fait est le **Visual Studio Code**.

### 2. Les langages de programmation

1. **Python**

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Small talk et Tcl.[10]

1. **Php**

PHP : HyperText Preprocessor, plus connu sous son sigle PHP (sigle autoréférentiel), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook et Wikipédia20. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur web. PHP peut également générer d'autres formats comme le WML, le SVG et le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. [12]

1. **Html**

Le HyperText Mark up Langage, généralement abrégé HTML ou dans sa dernière version

HTML, est le langage de balisage conçu pour représenter les pages web.

Ce langage permet : d’écrire de l’hypertexte, d’où son nom, de structurer sémantiquement la page, de mettre en forme le contenu, de créer des formulaires de saisie, d’inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques, de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l’accessibilité du web.

Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). [13]

1. **Css**

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d’un document écrit dans un langage de balisage tel que HTML. CSS est une technologie fondamentale du World Wide Web, aux côtés de HTML et JavaScript. CSS est conçu pour permettre la séparation de la présentation et du contenu, y compris la mise en page, les couleurs et les polices. Cette séparation peut améliorer l’accessibilité du contenu, fournir plus de flexibilité et de contrôle dans la spécification des caractéristiques de présentation. [14]

1. **Javascript**

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.

Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer 36 leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel.

JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environ 500 000 paquets en août 2017. JavaScript a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom d'ECMAScript en juin 1997 par ECMA International dans le standard ECMA-262. Le standard ECMA-262 en est actuellement à sa 8e édition. JavaScript est aussi employé pour les serveurs avec l'utilisation (par exemple) de Node.js ou de Deno.

**Choix :** Comme presque tous développement web les langages de programmation utilises sont **: Html, Css, JavaScript** qui seront notre choix parmi ceux énumérés.

### 3. Système de gestion de base des données

1. **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

SQL fait référence au Structured Query Language, le langage de requête utilisé.

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est

multi-thread et multi-utilisateur. Il fonctionne sur de nombreux systèmes d'exploitation différents, incluant AIX, IBM i-5, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetWare,

NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows. Les bases de données sont accessibles en utilisant les langages de programmation C, C++, VB, VB .NET, C#, Delphi/Kylix, Eiffel, Java, Perl, PHP, Python,

Windev, Ruby et Tcl ; une API spécifique est disponible pour chacun d'entre eux. Une interface

ODBC appelée MyODBC est aussi disponible. [8]

1. **Oracle**

Oracle (Oracle Corporation) est une entreprise américaine créée en 1977 par Larry Ellison. Ses produits phares so nt le système de gestion de base de données Oracle Database, le serveur d'applications Oracle Weblogic Server, le progiciel de gestion intégré Oracle E-Business Suite et l'offre de cloud computing Oracle Cloud Infrastructure. En 2019, Oracle était la deuxième plus grande entreprise de logiciels en matière de chiffre d'affaires et de capitalisation boursière.[9]

1. **PostgreSQL**

PostgreSQL est un système de gestion de bases de données relationnelles objet (ORDBMS) fondé sur POSTGRES. Ce dernier a été développé à l'université de Californie au département des sciences informatiques de Berkeley. POSTGRES est à l'origine de nombreux concepts qui ne seront rendus disponibles au sein de systèmes de gestion de bases de données commerciaux que bien plus tard.

PostgreSQL est un descendant libre du code original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes :

* requêtes complexes ;
* clés étrangères ;
* triggers ;
* vues modifiables ;
* intégrité transactionnelle ;
* contrôle des versions concurrentes (MVCC, acronyme de « MultiVersion Concurrency Control »).

De plus, PostgreSQL peut être étendu par l'utilisateur de multiples façons, en ajoutant, par exemple :

* de nouveaux types de données ;
* de nouvelles fonctions ;
* de nouveaux opérateurs ;
* de nouvelles fonctions d'agrégat ;
* de nouvelles méthodes d'indexage ;
* de nouveaux langages de procédure.

Et grâce à sa licence libérale, PostgreSQL peut être utilisé, modifié et distribué librement, quel que soit le but visé, qu'il soit privé, commercial ou académique.

**Choix :** Notre choix se porte sur **PostgreSQL** qui est un système de base de données relationnelles SGBDR) et gratuit.

**4. Serveur Web**

Un serveur web est un logiciel capable de répondre à des requêtes HTTP, c'est à dire de renvoyer des données (par exemple une page HTML), en réponse à des demandes écrites en HTTP (par exemple une requête GET). Un serveur web répond par défaut sur le port 80.

1. **Apache Tomcat**

Le logiciel Apache Tomcat est une implémentation open source des spécifications Jakarta

Servlet, Jakarta Server Pages, Jakarta Expression Langage, Jakarta Web Socket, Jakarta

Annotations et Jakarta Authentification. D'un point de vue global, Apache Tomcat est chargé de fournir un environnement d'exécution pour les servlets. Il permet donc aux développeurs d'exécuter leurs applications web Java. D'un point de vue plus détaillé, Tomcat sert à écouter toutes les requêtes des clients.

Un serveur web Apache peut être un excellent choix pour exécuter notre site web sur une plateforme stable et polyvalente. [4]

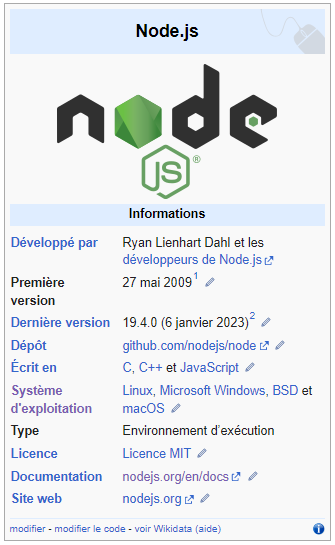
1. **Xampp Server**

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache Maria DB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. [5]

1. **Node JS**

Node.js est utilisé notamment comme plateforme de serveur Web [2]

Node.js est un environnement d’exécution single-thread, open-source et multi-plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d’exécution JavaScript V8 et utilise une architecture d’E / S non bloquante et pilotée par les événements, ce qui le rend efficace et adapté aux applications en temps réel.[3]



**Choix :**

Node.js est un environnement de serveur open source ;

Node.js est gratuit ;

Node.js fonctionne sur différentes plates-formes (Windows, Linux, Unix, Mac OS X, etc.) ;

Node.js utilise JavaScript sur le serveur avec le **Framework Express JS**

### 5. Client web

Un client web est un logiciel capable d'envoyer des requêtes HTTP à un serveur web et d'afficher les résultats. Les navigateurs web sont les clients web les plus répandus. Exemple : **Chrome**.

**6. Autre outil**

PowerAMC est un logiciel de conception créé par la société SDP, qui permet de modéliser les traitements informatiques et leurs bases de données associées. Il a été Créé par SDP sous le nom AMC\*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Il nous a permis de modéliser notre MCD et MPD.

**II.4.3. Architecture de l’application**

**Architecture Modèle/Vue/Contrôleur**

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Dans l'architecture MVC, les rôles des trois entités sont les suivants.

* modèle : données (accès et mise à jour)
* vue : interface utilisateur (entrées et sorties)
* contrôleur : gestion des événements et synchronisation

### Rôle du modèle

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient.

Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ses données. Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les étudiants, le modèle peut avoir des méthodes pour avoir, tous les étudiants d'une section, tous les cours d’un étudiant.

### Rôle de la vue

La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d’une entrée, boutons, …). Ses différents événements sont envoyés au contrôleur.

La vue peut aussi donner plusieurs vues, partielles ou non, des mêmes données. La vue peut aussi offrir la possibilité à l'utilisateur de changer de vue.

### Rôle du contrôleur

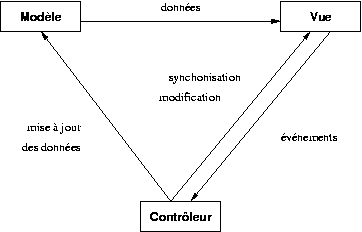
Le contrôleur est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Certains événements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier.

Dans le cas d'une base de données des e-learning. Une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner un nouvel étudiant pour visualiser tous ses cours. Ceci ne me modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cet étudiant.

Le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants. En effet si un même objet reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement.

### Interactions

Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante.



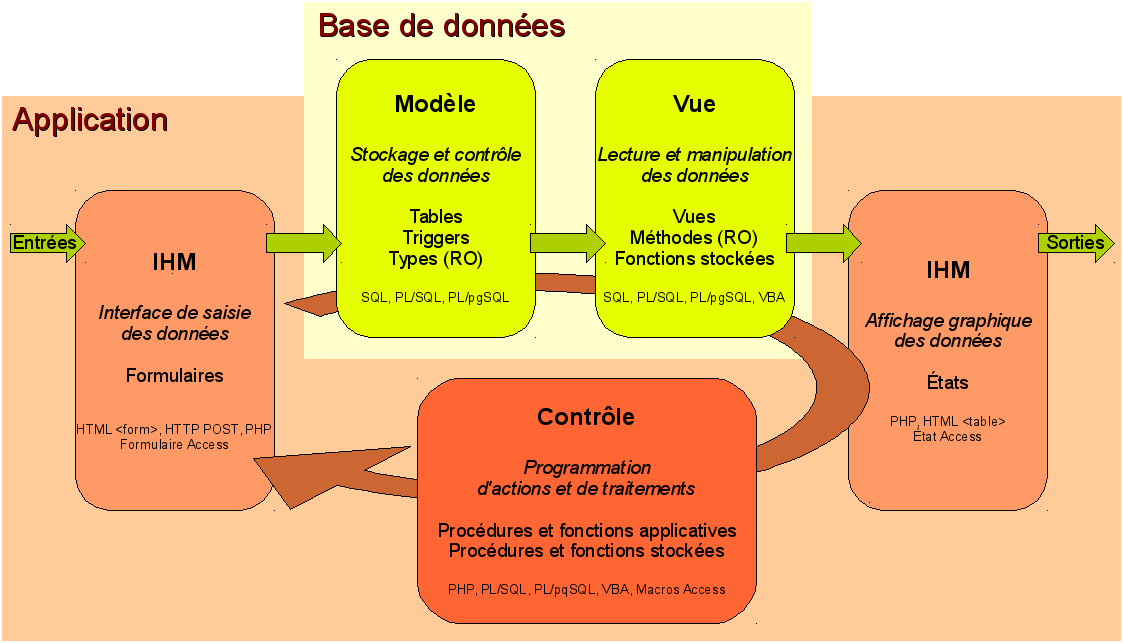


Fig. : Interactions entre le modèle, la vue et le contrôleur

**II.4.4 Architecture réseau**

1. **Notions d'architecture client-serveur**

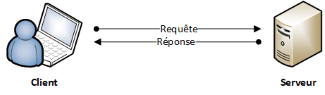
De nombreuses applications fonctionnent selon un environnement [clients](https://stph.scenari-community.org/bdd/lap2/co/webUC002archi.html#footnotesN64)/[serveur](https://stph.scenari-community.org/bdd/lap2/co/webUC002archi.html#footnotesN73), cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme 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, client de messagerie...

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

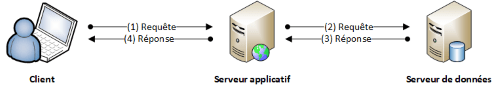
1. **Architecture 2-tier**

L'architecture à deux niveaux (aussi appelée architecture 2-tier,*tier* signifiant étage en anglais) caractérise les systèmes clients/serveurs dans lesquels **le**[**client**](https://stph.scenari-community.org/bdd/lap2/co/webUC003archi.html#footnotesN68) demande une ressource et **le**[**serveur**](https://stph.scenari-community.org/bdd/lap2/co/webUC003archi.html#footnotesN77) la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.



1. **Architecture 3-tier**

Dans l'architecture à 3 niveaux (appelée architecture 3-tier), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre : **le client** (le demandeur de ressources) et **le serveur d'application** (appelé aussi *middleware*, le serveur chargé de fournir la ressource mais faisant appel à un autre serveur) et **le serveur secondaire** (généralement un serveur de base de données, fournissant un service au premier serveur).



1. **Comparaison des deux types d'architecture**

|  |  |
| --- | --- |
| **Architecture 2-tier** | **Architecture 3-tier** |
| L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client. | Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web et serveur de base de données par exemple). Ainsi, l'architecture à trois niveaux permet :  - une plus grande flexibilité/souplesse  - une plus grande sécurité (la sécurité peut être définie pour chaque service)  - de meilleures performances (les tâches sont partagées). |

**Choix :** Pour un bon fonctionnement de notre application web l’architecture 3-tier prend le dessus, sur ce elle sera l’objet de notre choix comme architecture réseau.

Choix d’une solution

**Synthèse**

Le présent chapitre porte sur l’Etat de l’art englobant les outils disponibles qui contribueront à la résolution de notre problématique de départ. En rappel, notre sujet porte sur l’étude et la mise en place d’une application web de gestion des cours en ligne E-learning pour le compte de l’IAT. Pour mener à terme ce travail, l’apport de ce chapitre réside dans la synthèse des caractéristiques TCP/IP et ISO/OSI, des outils de développement web et celui de serveur web. Le choix de la solution qui permet dans la prochaine étape qui sera consacrée à l’implémentation de la nouvelle solution.

[b1] <https://www.redhat.com/fr/topics/linux/what-is-linux>

[b2] https://www.techno-science.net/glossaire-definition/Microsoft-Windows.html

[1]<https://stph.scenari-community.org/bdd/lap2/co/webUC002archi.html>

[2] <https://fr.wikipedia.org/wiki/Node.js>

[3] https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-node-js/