Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTERE DE L'INTERIEUR,
DE L'OUTRE-MER
ET DES COLLECTIVITES TERRITORIALES

CONCOURS D'INGENIEUR DES SYSTEMES D'INFORMATION ET DE COMMUNICATION

« SESSION 2010 »

TROISIEME CONCOURS

" Etude de cas à partir d'un dossier à caractère technique de 40 pages maximum, soumis au choix du candidat, faisant appel à des connaissances relatives à l'environnement et à la technique des systèmes d'information et de communication permettant de vérifier les capacités d'analyse et de synthèse du candidat ainsi que son aptitude à dégager des solutions appropriées, portant sur le thème suivant : "

Architecture et systèmes

Toute note inférieure à 8/21 est éliminatoire.

Durée : 4 h 00 (Coef. 3)

Le dossier documentaire comporte 21 pages.

L'utilisation de la calculatrice est interdite.

Rappel important :

Le candidat ne doit pas porter de signe distinctif sur sa copie ni sur les intercalaires.

SUJET

Contexte.

Vous êtes architecte de la société informatique « ArchiQVF » et votre directeur vous charge de réaliser une étude d'architecture auprès de la société « MissionPlus » spécialisée dans les missions d'intérim. Cette société souhaite faire une refonte totale de son système d'information qui s'avère ne plus être adapté à ses besoins.

Travail demandé.

Vous établirez un diagnostic du système d'information actuel de la société « MissionPlus » puis vous réaliserez une étude présentant une architecture couvrant l'intégralité des besoins qu'elle a exprimés, comportant un schéma d'architecture.

Tous vos choix techniques devront être argumentés et vous démontrerez en particulier comment les problèmes réglementaires et de sécurité sont couverts.

Documents joints :

Document n°1

Présentation de la société « MissionPlus »

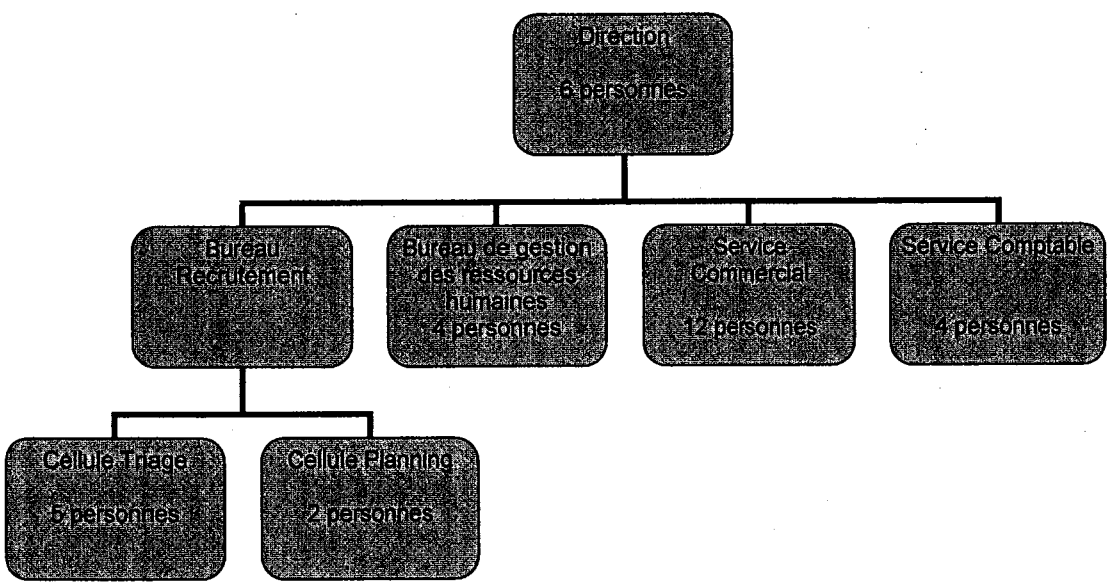
La société "MissionPlus" est une jeune société, créée en 2000, spécialisée dans le recrutement. Le principe est simple, un client de "MissionPlus" émet le besoin d'avoir un ou plusieurs personnels ayant un profil bien précis et "MissionPlus" se charge de lui trouver en moins de 72 heures la ou les personne(s) correspondante(s).

Au début, c'était une petite société et toute la gestion était réalisée sur un tableur. Des problèmes apparaissaient comme l'accès concurrent aux fichiers, l'intégrité, la perte de données,...etc. Ces problèmes étaient mineurs, mais la petite société du début qui gérait une centaine de profils, en gère maintenant plus de 10.000.

La situation est devenue difficile à gérer et l'équipe constituée maintenant de 33 personnes n'arrive plus à effectuer ses missions ce qui représente un frein au développement de la société.

Une première opération de modernisation a été mise en place sur le site de la société afin de pouvoir publier la liste des postes disponibles.

La société est structurée de la manière suivante :



Missions:

La cellule de triage :

Chargée de récupérer tous les mois les 5000 dossiers de candidatures, cette cellule renseigne dans un tableur l'intégralité des demandes afin de pouvoir renseigner les candidats sur la suite donnée à leur dossier.

Tous les dossiers sont étudiés rapidement afin de voir si un profil correspond au poste disponible. Si c'est le cas, le dossier est transmis sous forme papier et électronique via un fichier à la cellule des rendez-vous (cellule « planning »).

La cellule « Planning » :

Elle gère l'agenda des spécialistes en ressources humaines afin de fixer les rendez-vous avec les candidats sélectionnés par la cellule de triage.

Le bureau de gestion des ressources humaines :

Gère l'intégralité des profils sélectionnés (compétences, missions effectuées,...etc.), soit environ 10 000 dossiers.
Participe aux entretiens des candidats.

Le service commercial :

Les commerciaux sont souvent en déplacement chez les clients, ils analysent et anticipent les besoins de ces derniers.

Le service comptable :

Ce service assure la comptabilité globale de l'entreprise et assure aussi le paiement des missions.

Document n°2

Les objectifs et exigences du nouveau système d'information

Les objectifs sont :

- Reprendre les données existantes.
- Réaliser un portail internet "WebInterMissionPlus" permettant aux différents candidats de postuler en ligne sur les différentes offres de mission et de communiquer les différents éléments (curriculum vitae, ...etc.).
- Mettre en place une application web "WebIntraMissionPlus" capable de gérer l'ensemble des profils sélectionnés. Cette application devra être accessible en interne et via internet pour les commerciaux en déplacement chez les clients.
- Gérer la prise de rendez-vous entre les candidats dont le profil correspond et un spécialiste en ressources humaines de la société "MissionPlus".

Les exigences sont :

- Une haute disponibilité du système pour l'ensemble du système durant les heures d'ouverture de la société soit de 7H00 à 20H00.
- Un accès sécurisé pour le portail internet "WebInterMissionPlus" et pour l'accès via internet à l'application "WebIntraMissionPlus" par les commerciaux.
- L'architecture devra reposer essentiellement sur des piles logicielles libres afin de limiter les coûts des licences du projet.
- L'ensemble du parc informatique de la société a été récemment renouvelé et fonctionne sous Windows vista.
- Au niveau des accès concurrents, un pic de 500 utilisateurs pour l'application "WebInterMissionPlus" et 200 utilisateurs pour l'application "WebIntraMissionPlus".
- Aucune perte de données ne sera tolérée et la reprise d'activité en cas d'incident ne devra pas dépasser 4 heures.

VPN - Réseaux Privés Virtuels (RPV)

http://www.commentcamarche.net/contents/initiation/vpn.php3

Le concept de réseau privé virtuel

Les réseaux locaux d'entreprise (LAN ou RLE) sont des réseaux internes à une organisation, c'est-à-dire que les liaisons entre machines appartiennent à l'organisation. Ces réseaux sont de plus en plus souvent reliés à Internet par l'intermédiaire d'équipements d'interconnexion. Il arrive ainsi souvent que des entreprises éprouvent le besoin de communiquer avec des filiales, des clients ou même du personnel géographiquement éloignés via internet.

Pour autant, les données transmises sur Internet sont beaucoup plus vulnérables que lorsqu'elles circulent sur un réseau interne à une organisation car le chemin emprunté n'est pas défini à l'avance, ce qui signifie que les données empruntent une infrastructure réseau publique appartenant à différents opérateurs. Ainsi, il n'est pas impossible que sur le chemin parcouru, le réseau soit écouté par un utilisateur indiscret ou même détourné. Il n'est donc pas concevable de transmettre dans de telles conditions des informations sensibles pour l'organisation ou l'entreprise.

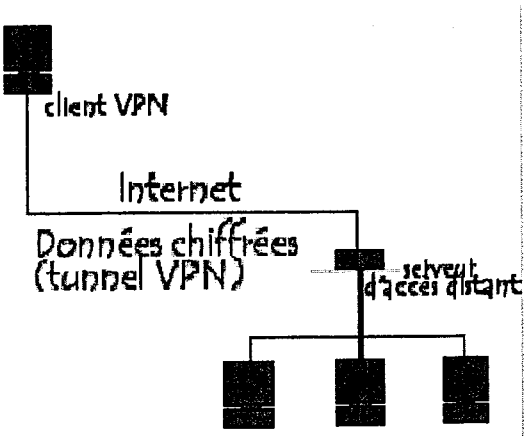
La première solution pour répondre à ce besoin de communication sécurisé consiste à relier les réseaux distants à l'aide de liaisons spécialisées. Toutefois la plupart des entreprises ne peuvent pas se permettre de relier deux réseaux locaux distants par une ligne spécialisée, il est parfois nécessaire d'utiliser Internet comme support de transmission.

Un bon compromis consiste à utiliser Internet comme support de transmission en utilisant un protocole d'"encapsulation" (en anglais *tunneling*, d'où l'utilisation impropre parfois du terme "tunnelisation"), c'est-à-dire encapsulant les données à transmettre de façon chiffrée. On parle alors de **réseau privé virtuel** (noté *RPV* ou **VPN**, acronyme de *Virtual Private Network*) pour désigner le réseau ainsi artificiellement créé.
Ce réseau est dit *virtuel* car il relie deux réseaux "physiques" (réseaux locaux) par une liaison non fiable (Internet), et *privé* car seuls les ordinateurs des réseaux locaux de part et d'autre du VPN peuvent "voir" les données.

Le système de *VPN* permet donc d'obtenir une liaison sécurisée à moindre coût, si ce n'est la mise en œuvre des équipements terminaux. En contrepartie, il ne permet pas d'assurer une qualité de service comparable à une ligne louée dans la mesure où le réseau physique est public et donc non garanti.

Fonctionnement d'un VPN

Un réseau privé virtuel repose sur un protocole, appelé **protocole de tunnelisation** (*tunneling*), c'est-à-dire un protocole permettant aux données passant d'une extrémité du VPN à l'autre d'être sécurisées par des algorithmes de cryptographie.



Le terme de "tunnel" est utilisé pour symboliser le fait qu'entre l'entrée et la sortie du VPN les données sont chiffrées (cryptées) et donc incompréhensible pour toute personne située entre les deux extrémités du VPN, comme si les données passaient dans un tunnel. Dans le cas d'un *VPN* établi entre deux machines, on appelle *client VPN* l'élément permettant de chiffrer et de déchiffrer les données du côté utilisateur (client) et *serveur VPN* (ou plus généralement **serveur d'accès distant**) l'élément chiffrant et déchiffrant les données du côté de l'organisation.

De cette façon, lorsqu'un utilisateur nécessite d'accéder au réseau privé virtuel, sa requête va être transmise en clair au système passerelle, qui va se connecter au réseau distant par l'intermédiaire d'une infrastructure de réseau public, puis va transmettre la requête de façon chiffrée. L'ordinateur distant va alors fournir les données au serveur VPN de son réseau local qui va transmettre la réponse de façon chiffrée. A réception sur le client VPN de l'utilisateur, les données seront déchiffrées, puis transmises à l'utilisateur ...

Les protocoles de tunnelisation

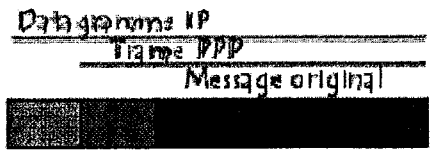
Les principaux protocoles de tunneling sont les suivants :

- PPTP (*Point-to-Point Tunneling Protocol*) est un protocole de niveau 2 développé par Microsoft, 3Com, Ascend, US Robotics et ECI Telematics.
- L2F (*Layer Two Forwarding*) est un protocole de niveau 2 développé par Cisco, Northern Telecom et Shiva. Il est désormais quasi-obsolète
- L2TP (*Layer Two Tunneling Protocol*) est l'aboutissement des travaux de l'*IETF* (RFC 2661) pour faire converger les fonctionnalités de *PPTP* et *L2F*. Il s'agit ainsi d'un protocole de niveau 2 s'appuyant sur PPP.
- **IPSec** est un protocole de niveau 3, issu des travaux de l'IETF, permettant de transporter des données chiffrées pour les réseaux IP.

Le protocole PPTP

Le principe du protocole PPTP (*Point To Point Tunneling Protocol*) est de créer des trames sous le protocole PPP et de les encapsuler dans un datagramme IP.

Ainsi, dans ce mode de connexion, les machines distantes des deux réseaux locaux sont connectées par une connexion point à point (comprenant un système de chiffrement et d'authentification, et le paquet transite au sein d'un datagramme IP.



De cette façon, les données du réseau local (ainsi que les adresses des machines présentes dans l'en-tête du message) sont encapsulées dans un message PPP, qui est lui-même encapsulé dans un message IP.

Le protocole L2TP

Le protocole L2TP est un protocole standard de tunnelisation (standardisé dans un RFC) très proche de PPTP. Ainsi le protocole L2TP encapsule des trames protocole PPP, encapsulant elles-mêmes d'autres protocoles (tels que IP, IPX ou encore NetBIOS).

Le protocole IPSec

IPSec est un protocole défini par l'IETF permettant de sécuriser les échanges au niveau de la couche réseau. Il s'agit en fait d'un protocole apportant des améliorations au niveau de la sécurité au protocole IP afin de garantir la confidentialité, l'intégrité et l'authentification des échanges.

Le protocole IPSec est basé sur trois modules :

- *IP Authentification Header* (**AH**) concernant l'intégrité, l'authentification et la protection contre le rejeu. des paquets à encapsuler
- *Encapsulating Security Payload* (**ESP**) définissant le chiffrement de paquets. ESP fournit la confidentialité, l'intégrité, l'authentification et la protection contre le rejeu.
- *Security Assocation* (**SA**) définissant l'échange des clés et des paramètres de sécurité. Les SA rassemblent ainsi l'ensemble des informations sur le traitement à appliquer aux paquets IP (les protocoles AH et/ou ESP, mode tunnel ou transport, les algo de sécurité utilisés par les protocoles, les clés utilisées,...). L'échange des clés se fait soit de manière manuelle soit avec le protocole d'échange IKE (la plupart du temps), qui permet aux deux parties de s'entendre sur les SA.

Comprendre l'architecture des sites Web Questions-clés, définitions, liens utiles, acteurs, chiffres et citations : l'essentiel de ce qu'il faut savoir en un coup d'oeil.

:: 5 questions-clés ::

De quels éléments se compose un site Web ?
La première des démarches, lors du lancement d'un site Web, consiste à s'attribuer un nom de domaine ou DNS, soit une adresse logique qui redirige vers une ou plusieurs machines connectées au réseau. On appelle ces machines des serveurs frontaux. Ils jouent un rôle clé dans l'infrastructure d'un site Web en interprétant les actions de l'utilisateur en demandes de services.

Un serveur frontal - ou serveur Web - va assurer la navigation dans le site en transformant les actions des visiteurs en pages HTML lisibles à partir d'un navigateur. Il interprète pour cela les requêtes du protocole HTTP en vigueur sur Internet. Les serveurs Web les plus populaires sont IIS ou Apache. Derrière ces serveurs frontaux se trouvent les serveurs d'applications et serveurs de bases de données.

Les premiers servent à pré-calculer le résultat d'applications Web et à assurer le lien entre des environnements a priori non-compatibles (par exemple, une application mainframe portée sur Internet). Dans ce dernier cas, les serveurs d'applications évitent de réécrire en langage Web une application déjà disponible.

Les serveurs de base de données représentent le dernier élément clé d'une infrastructure Web. Ils stockent les informations produits ou clients et les remontent en fonction des besoins applicatifs. Les serveurs de base de données sont au cœur des services Web évolués (forums, gestion de comptes en ligne, création de pages dynamiques...).

Quels langages de développement Web se distinguent ?
Sur Internet, de multiples langages cohabitent selon les tâches à remplir. Lorsqu'il s'agit de développement d'applications attaquant une base de données, les langages les plus utilisés sont les langages de script - comme PHP ou ASP. Si l'entreprise qui développe un site Web souhaite s'appuyer sur une partie de son existant, elle se tournera plutôt vers les langages Java ou C#, généralement synonymes de serveurs d'applications.

Pour la capture d'événements clavier ou souris, ou la gestion de paramètres du navigateur, le langage Javascript dispose de fonctions adaptées. Concernant la génération et la diffusion d'informations, le standard du Web est XML avec ses dérivés (VoiceXML, RSS...). Enfin, une foule de langages moins connus accompagne les besoins spécifiques des entreprises tels ColdFusion, PL/SQL, JSP...

Y a-t-il une relation de dépendance entre les composants de l'infrastructure ?
Oui, le marché parle même souvent de plate-forme, par exemple LAMP qui regroupe Linux, Apache, MySQL et PHP. Il s'agit de regrouper des composants prévus pour fonctionner ensemble, afin d'optimiser le fonctionnement global de l'infrastructure.

Concurrent de LAMP, l'environnement Microsoft propose le couple Windows / IIS / SQL Server / ASP .Net. Moins populaire, Sun dispose de son propre environnement Web à travers Solaris / SunOne / Java. Dès lors, le choix d'une brique peut déterminer tout ou partie du reste de l'infrastructure, l'ensemble devant aboutir à un bon compromis entre services, performances et évolutivité.

Quels autres composants interviennent dans la transmission de l'information sur Internet ?
Deux autres composants majeurs peuvent intervenir dans l'infrastructure globale d'un site Internet : un réseau de stockage et une solution d'optimisation des flux. Dans le premier cas, le site Web déporte la gestion de l'information sur un réseau dédié au stockage (SAN, NAS) afin d'en simplifier l'administration et d'accélérer la rapidité de traitement. La mise en place d'un tel système n'est toutefois valable que pour des gros volumes, à partir d'un teraoctet par exemple.

L'optimisation de flux regroupe quant à elle les solutions de compression de flux, mise en cache de pages Web ou répartition de charge. Ces solutions se placent entre les serveurs frontaux et la connexion au réseau Internet. Toutes ont pour but d'améliorer la disponibilité des pages mais chacune s'y prend différemment. Les solutions de mise en cache pré-chargent les pages Web sur des serveurs secondaires, les solutions de compression IP minimisent la taille des paquets à transporter tandis que les solutions de répartitions de charge redirigent le trafic en fonction de la disponibilité et de la charge des serveurs.

Quels impacts provoque la sécurité sur l'architecture d'un site ?
Une des évolutions les plus courantes pour rendre une architecture Web sécurisée consiste à multiplier les serveurs des différents composantes du site (serveur Web, serveur d'application ou serveur de bases de données). On parle alors de fermes de serveur et d'architecture en fail over. Concrètement, il s'agit de basculer les traitements d'un serveur à un autre dans le cas d'indisponibilité ou de dysfonctionnement.

Ce cas est différent d'un système de répartition de charge (load balancing) où plusieurs serveurs travaillent en parallèle à l'exécution d'une même tâche. Hormis le travail sur la disponibilité, la sécurité engendre aussi des conséquences sur l'organisation des différentes machines. La mise en place de pare-feu, très courante dans le cas d'un site Web professionnel, permet d'isoler les frontaux Web et les machines connectées derrière, du réseau Internet. Cette séparation filtre les connexions entrantes et sortantes afin de limiter les intrusions ou au contraire la fuite d'informations.

ARCHITECTURE TROIS TIERS

http://fr.wikipedia.org/wiki/Architecture_trois_tiers (dernière modification le 6 octobre 2009)

L'architecture 3-tiers (« 3-tier » en anglais) ou architecture à trois niveaux est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- couche présentation
- couche métier
- couche accès aux données

C'est une extension du modèle client/serveur.

L'architecture 3-tier (de l'anglais tier signifiant étage ou niveau) est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches, étages, niveaux ou strates dont le rôle est clairement défini :

- la présentation des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- le traitement métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- et enfin l'accès aux données persistantes (persistance en anglais) : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Dans cette approche, les couches communiquent entre elles au travers d'un « modèle d'échange », et chacune d'entre elles propose un ensemble de services rendus. Les services d'une couche sont mis à disposition de la couche supérieure. On s'interdit par conséquent qu'une couche invoque les services d'une couche plus basse que la couche immédiatement inférieure ou plus haute que la couche immédiatement supérieure (chaque niveau ne communique qu'avec ses voisins immédiats).

Le rôle de chacune des couches et leur interface de communication étant bien définis, les fonctionnalités de chacune d'entre elles peuvent évoluer sans induire de changement dans les autres couches. Cependant, une nouvelle fonctionnalité de l'application peut avoir des répercussions dans plusieurs d'entre elles. Il est donc essentiel de définir un modèle d'échange assez souple, pour permettre une maintenance aisée de l'application.

Ce modèle d'architecture 3-tiers a pour objectif de répondre aux préoccupations suivantes :

- allègement du poste de travail client (notamment vis-à-vis des architectures classiques client-serveur de données-typiques des applications dans un contexte Oracle/Unix) ;
- prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.) ;
- introduction de clients dits « légers » (plus liée aux technologies Intranet/HTML qu'au 3-tiers proprement dit) ;
- amélioration de la sécurité des données, en supprimant le lien entre le client et les données. Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer dans la couche de données ;

- et enfin, meilleure répartition de la charge entre différents serveurs d'application.

Précédemment, dans les architectures client-serveur classiques, les couches présentation et traitement étaient trop souvent imbriquées. Ce qui posait des problèmes à chaque fois que l'on voulait modifier l'Interface Homme-Machine (IHM) du système.

L'activation à distance (entre la station et le serveur d'application) des objets et de leurs méthodes (on parle d'invocation) peut se faire au travers d'un Object Request Broker (ORB) (avec le protocole Internet Inter ORB Protocol (IIOP) ou au moyen des technologies Components Objects Model/Distributed Component Object Model (COM/DCOM) de Microsoft ou encore avec Remote Method Invocation (RMI) en technologie de Java2 Enterpise Edition (J2EE)). Cette architecture ouverte permet également de répartir les objets sur différents serveurs d'application (soit pour prendre en compte un existant hétérogène, soit pour optimiser la charge).

Il s'agit d'une architecture logique qui se répartit ensuite selon une architecture technique sur différentes machines physiques, bien souvent au nombre de 3, 4 ou plus. Une répartition de la charge doit dans ce cas être mise en place.

Les trois couches

Couche Présentation (premier niveau)

Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'Interface Homme Machine. En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur web ou en WML pour être utilisée par un téléphone portable.

On conçoit facilement que cette interface peut prendre de multiples facettes sans changer la finalité de l'application. Dans le cas d'un système de distributeurs de billets, l'automate peut être différent d'une banque à l'autre, mais les fonctionnalités offertes sont similaires et les services identiques (fournir des billets, donner un extrait de compte, etc.).

Toujours dans le secteur bancaire, une même fonctionnalité métier (par exemple, la commande d'un nouveau chéquier) pourra prendre différentes formes de présentation selon qu'elle se déroule sur Internet, sur un distributeur automatique de billets ou sur l'écran d'un chargé de clientèle en agence.

La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.

Couche Métier / Business (second niveau)

Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.

Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés.

Couche Accès aux données (troisième niveau)
Elle consiste en la partie gérant l'accès aux gisements de données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme (couplage faible).

Données propres au système
Ces données sont pérennes, car destinées à durer dans le temps, de manière plus ou moins longue, voire définitive.

Les données peuvent être stockées indifféremment dans de simples fichiers texte, ou eXtensible Markup Language (XML), ou encore dans une base de données. Quel que soit le support de stockage choisi, l'accès aux données doit être le même. Cette abstraction améliore la maintenance du système.

Les services sont mis à disposition de la couche métier. Les données renvoyées sont issues du/des gisements de données du système.

Pour une implémentation « native », le motif de conception (en anglais design pattern) à implémenter dans cette couche est le Data Access Object (DAO). Ce dernier consiste à représenter les données du système sous la forme d'un modèle objet. Par exemple un objet pourrait représenter un contact ou un rendez-vous.

La représentation du modèle de données objet en base de données (appelée persistance) peut s'effectuer à l'aide d'outils tels que Hibernate.

Données gérées par un autre système
Les données peuvent aussi être gérées de manière externe. Elles ne sont pas stockées par le système considéré, il s'appuie sur la capacité d'un autre système à fournir ces informations.

Par exemple, une application de pilotage de l'entreprise peut ne pas sauvegarder des données comptables de haut niveau dont elle a besoin, mais les demander à une application de comptabilité. Celle-ci est indépendante et préexistante, et on ne se préoccupe pas de savoir comment elle les obtient ou si elle les sauvegarde, on utilise simplement sa capacité à fournir des données à jour.

La Haute disponibilité

http://fr.wikipedia.org/w/index.php?title=Haute_disponibilité

La **haute disponibilité** est un terme souvent utilisé en informatique, à propos d'architecture de système ou d'un service pour désigner le fait que cette architecture ou ce service a un taux de disponibilité convenable.

La disponibilité est aujourd'hui un enjeu important des infrastructures informatiques. On estime aujourd'hui que la non-disponibilité d'un service informatique peut avoir des coûts se chiffrant en millions, particulièrement dans le domaine de l'industrie où l'arrêt d'une chaîne de production peut avoir un effet dévastateur.

Deux moyens complémentaires sont utilisés pour améliorer la haute disponibilité :

- La mise en place d'une infrastructure matérielle dédiée, généralement en se basant sur de la redondance matérielle. Est alors créé un cluster de haute-disponibilité (par opposition à un cluster de calcul) : une grappe d'ordinateurs dont le but est d'assurer un service en évitant au maximum les indisponibilités.
- La mise en place de processus adaptés permettant de réduire les erreurs, et d'accélérer la reprise en cas d'erreur. ITIL contient de nombreux processus de ce type.

Pour mesurer la disponibilité, on utilise souvent un pourcentage essentiellement composé de '9' :

- 99% désigne le fait que le service est **indisponible** moins de 3,65 jours par an
- 99,9%, moins de 8,75 heures par an
- 99,99%, moins de 52 minutes par an
- 99,999%, moins de 5,2 minutes par an
- 99,9999%, moins de 54,8 secondes par an
- 99,99999%, moins de 3,1 secondes par an

etc.

L'amalgame est souvent fait, à tort, entre la haute disponibilité et le plan de reprise d'activité. Il s'agit de deux tâches différentes, complémentaires pour atteindre la disponibilité continue.

Sommaire

1- Techniques améliorant la disponibilité

De nombreuses techniques sont utilisées pour améliorer la disponibilité :

- la redondance des matériels et la mise en cluster ;
- la sécurisation des données : RAID (Recherche Assistance Intervention et Dissuasion) , snapshots, Oracle Data Guard , BCV (Business Copy Volume), SRDF (Symmetrix Remote Data Facility), DRBD (Distributed Replicated Block Device) ;
- la possibilité de reconfigurer le serveur « à chaud » (c'est-à-dire lorsque celui-ci fonctionne) ;
- mode dégradé ou un mode panique ;
- plan de secours ;
- et sécurisation des sauvegardes : externalisation, centralisation sur site tiers.

La haute disponibilité exige le plus souvent un local adapté : alimentation stabilisée, climatisation sur plancher, avec filtre à particules, service de maintenance, service de gardiennage et de sécurité contre la malveillance et le vol. Attention aussi au risque d'incendie et de dégâts des eaux. Les câbles d'alimentation et de communication doivent être multiples et enterrés. Ils ne doivent pas être saillants dans le parking souterrain de l'immeuble, ce qui est trop souvent vu dans les immeubles parisiens.
Ces critères sont les premiers à entrer en compte lors du choix d'un prestataire d'hébergement (cas de la location d'un local à haute disponibilité).

Pour chaque niveau de l'architecture, pour chaque composant, chaque liaison entre composants, il faut établir :

- Comment détecter une panne. Exemples : Tests de vie TCP Health Check implémenté par un boîtier Alteon , programme de test invoqué périodiquement (« heartbeat »), interface de type « diagnostic » sur les composants...
- Comment le composant est sécurisé, redondé, secouru...

Exemples : serveur de secours, cluster système, clustering Websphere, stockage RAID, sauvegardes, double attachement SAN, mode dégradé, matériel non-utilisé libre (spare) prêt à être réinstallé..

- Comment on désire enclencher la bascule en mode secours / dégradé. Manuellement après analyse ? Automatiquement ?
- Comment on assure que le système de secours reparte sur un état stable et connu.

Exemples : on repart d'une copie de la base et on réapplique les archives logs, relancement des batchs depuis un état connu, commit à 2 phases pour les transactions mettant à jour plusieurs gisements de données...

- Comment l'application redémarre sur le mécanisme de secours.

Exemples : redémarrage de l'application, redémarrage des batches interrompus, activation d'un mode dégradé, reprise de l'adresse IP du serveur défaillant par le serveur de secours...

- Comment reprendre éventuellement les transactions ou sessions en cours. Exemples : persistance de session sur le serveur applicatif, mécanisme pour assurer une réponse à un client pour une transaction qui s'est bien effectuée avant défaillance mais pour laquelle le client n'a pas eu de réponse...

- Comment revenir à la situation nominale. Exemples :

. si un mode dégradé permet en cas de défaillance d'une base de données de stocker des transactions en attente dans un fichier, comment les transactions sont-elles réappliquées quand la base de données redevient active ?

. si un composant défaillant a été inactivé, comment s'effectue sa réintroduction en service actif (nécessité par exemple de resynchroniser des données, de retester le composant...)

2- Dépendance vis-à-vis des autres applications

Pour une application qui sollicite d'autres applications avec des middlewares en mode synchrone (service webs en http, Tuxedo, Corba, EJB) le taux de disponibilité de l'application sera fortement lié à la disponibilité des applications dont elle dépend. La sensibilité des applications dont on dépend doit donc être équivalente ou supérieure à la sensibilité de l'application elle-même.

Sinon, il faut envisager :

- l'utilisation d'un middleware asynchrone : MQ Series, JMS, SonicMQ, CFT
- la mise en œuvre d'un mode dégradé quand une application dont on dépend est défaillante.

Pour cette raison on privilégiera l'utilisation de middlewares asynchrones pour privilégier une bonne disponibilité quand c'est possible.

3- Répartition de charge et sensibilité

La sensibilité est souvent gérée en redondant les éléments avec un mécanisme de répartition de charge (un cluster websphere avec un load-balancing Alteon par exemple). Pour que ce système apporte un réel gain en termes de fiabilité, il faut vérifier que si un des éléments est défaillant, les éléments restants disposent d'une puissance suffisante pour assurer le service.

Autrement dit, dans le cas de deux serveurs actifs avec répartition de charge, la puissance d'un seul serveur doit permettre d'assurer la totalité de la charge. Avec trois serveurs, la puissance d'un seul serveur doit permettre d'assurer 50% de la charge (en supposant que la probabilité d'avoir un incident sur deux serveurs en même temps est négligeable). Pour assurer une bonne fiabilité, il est inutile de mettre un grand nombre de serveurs se secourant mutuellement. Par exemple, un élément fiable à 99% redondé une fois donne une fiabilité de 99.99% (probabilité pour laquelle les deux éléments soient défaillants au même moment = 1/100x1/100 = 1/10.000)

4- Redondance différentielle

La redondance d'un élément est généralement effectuée en choisissant de redonder avec plusieurs composants identiques. Ceci suppose, pour être efficace, qu'une défaillance d'un des composants est aléatoire et indépendante d'une défaillance d'un des autres composants. C'est par exemple le cas des pannes matérielles.

Ce n'est pas le cas de toutes les défaillances : par exemple, une faille du système d'exploitation ou une anomalie d'un composant logiciel peuvent survenir, quand les conditions sont favorables, sur l'ensemble des composants à la fois. Pour cette raison, quand l'application est extrêmement sensible, on considèrera de redonder les éléments avec des composants de natures différentes mais assurant les mêmes fonctions.
Ceci peut conduire à :

- choisir des serveurs de natures différentes, avec des OS (Operatine Système) différents, des produits logiciels d'infrastructure différents,
- développer le même composant deux fois en respectant à chaque fois les contrats d'interface qui s'appliquent au composant.

5- Redondance avec système de vote

Dans ce mode, différents composants traitent les mêmes entrées et produisent donc (en principe) les mêmes sorties.

Les résultats produits par tous les composants sont collectés, puis un algorithme est mis en œuvre pour produire le résultat final. L'algorithme peut être simple (vote à la majorité) ou complexe (moyenne, moyenne pondérée, médiane...), l'objectif étant d'éliminer les résultats erronés imputables à un dysfonctionnement sur l'un des composants et/ou de fiabiliser un résultat en combinant plusieurs résultats légèrement différents.

Ce procédé :

- ne permet pas de répartition de charge ;
- introduit le problème de fiabilisation du composant gérant l'algorithme de vote.

Ce procédé est utilisé généralement dans les cas suivants :

- des systèmes reposant sur des capteurs (exemple : capteurs de température) pour lesquels les capteurs sont redondés
- des systèmes où plusieurs composants différents assurant la même fonction sont utilisés (cf. redondance différentielle) et pour lesquels un meilleur résultat final peut être obtenu en combinant les résultats produits par les composants (exemple : système de reconnaissance de formes utilisant plusieurs algorithmes pour obtenir un meilleur taux de reconnaissance).

6- « Shadow operations »

Lors du dysfonctionnement d'un composant redondé et après l'avoir réparé, on peut souhaiter le réintroduire en service actif, vérifier son bon fonctionnement effectif, mais sans que les résultats soient utilisés. Dans ce cas, les entrées sont traitées par un (ou plusieurs) composant(s) réputé(s) fiable(s). Ceux-ci produisent le résultat exploité par le reste du système. Les mêmes entrées sont également traitées par le composant réintroduit qui est dit en mode « shadow ». On peut vérifier le bon fonctionnement du composant en comparant les résultats produits avec ceux des composants fiables. Ce procédé est souvent utilisé dans les systèmes à base de vote car il suffit d'exclure le composant en mode « shadow » du vote final.

7- Les processus qui permettent d'améliorer la disponibilité

On peut distinguer deux rôles dans ces processus.

7-1 Les processus qui réduisent le nombre de pannes

En se basant sur le fait que **mieux vaut prévenir que guérir**, mettre en place des processus de contrôle qui permettront de réduire le nombre d'incidents sur le système permet d'améliorer la disponibilité.

Deux processus permettent de jouer ce rôle :

- le processus de gestion des changements : 60% des erreurs sont liées à un changement récent. En mettant en place un processus formalisé, accompagné de tests suffisants (et réalisés dans un environnement de pré-production correct), de nombreux incidents peuvent être éliminés.
- un processus de gestion proactive des erreurs : les incidents peuvent bien souvent être détectés avant de survenir : les temps de réponse augmentent. Un processus dédié à cette tâche, et muni des outils adéquats (système de mesure, de reporting) pourra intervenir avant même que l'incident n'arrive.

En mettant en place ces deux processus, de nombreux incidents peuvent être évités.

7-2 Les processus qui réduisent la durée des pannes

Les pannes finissent toujours par arriver. A ce moment-là, le processus de reprise en cas d'erreur est primordial pour que le service soit restauré au plus vite. Ce processus doit avoir un objectif : permettre à l'utilisateur d'utiliser un service le plus rapidement possible. La réparation définitive doit donc être évitée car elle prend beaucoup plus de temps. Ce processus devra donc mettre en place une solution de contournement du problème.

8- Cluster haute disponibilité

Un cluster haute disponibilité (par opposition à un cluster de calcul) est une grappe d'ordinateurs dont le but est d'assurer un service en évitant au maximum les indisponibilités.

Exemples d'application de clustering dans le monde UNIX (fonctionnant sous AIX ou Linux) :

- IBM HACMP
- Bull Application Roll-over Facility

Document n°7

http://fr.wikipedia.org/wiki/Certificat_%C3%A9lectronique
(dernière modification : 11 janvier 2010)

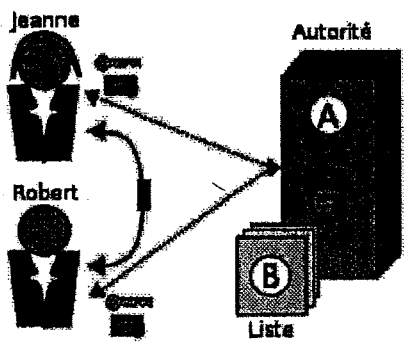
Certificat électronique

Un **certificat électronique** est une carte d'identité numérique dont l'objet est d'identifier une entité physique ou non-physique. Le certificat numérique ou électronique est un lien entre l'entité physique et l'entité numérique (Virtuel). L'autorité de certification fait foi de tiers de confiance et atteste du lien entre l'identité physique et l'entité numérique. Le standard le plus utilisé pour la création des certificats numériques est le X.509.

Sommaire

1 Fonctionnement

L'autorité d'enregistrement et de certification (repère A) recense et contrôle l'utilisation des certificats. Il possède une liste (repère B) des certificats révoqués.



Le principe de fonctionnement est le suivant:

- l'utilisatrice Jeanne télécharge un certificat donné par l'utilisateur Robert;
- l'ordinateur de Jeanne va contrôler la validité et les paramètres de codage en se référant à une autorité de certification;
- la clé publique est envoyée à l'utilisateur Robert;
- à son tour, l'ordinateur de Robert; va contrôler la cohérence entre le certificat et les données reçues en se référant à l'autorité de certification;
- les données reçues peuvent être déchiffrées.

2 _ Description du certificat

2.1 _ Contenu

Il existe 4 types de certificats en fonction du niveau de sécurité:

- classe 1 : adresse électronique du demandeur requise;
- classe 2 : preuve de l'identité requise (photocopie de carte d'identité par exemple);
- classe 3 : présentation physique du demandeur obligatoire.
- classe 3+ : identique à la classe 3, mais le certificat est stocké sur un support physique (clé USB à puce, ou carte à puce; exclut donc les certificats logiciels)

Tel qu'on l'utilise en cryptographie et en sécurité informatique, un certificat électronique est un bloc de données contenant, dans un format spécifié, les parties suivantes :

- un numéro de série;
- l'identification de l'algorithme de signature;
- la désignation de l'autorité de certification émettrice du certificat;
- la période de validité au-delà de laquelle il sera suspendu ou révoqué;
- le nom du titulaire de la clé publique;
- l'identification de l'algorithme de chiffrement et la valeur de la clé publique constitués d'une paire de clés asymétriques (comme par exemple RSA);
- des informations complémentaires optionnelles;
- l'identification de l'algorithme de signature et la valeur de la signature numérique.

Un certificat électronique est géré tout au long de son cycle de vie (Certificate revocation list (CRL), Protocole de vérification en ligne de certificat) avec une infrastructure à clés publiques (PKI pour *Public Key Infrastructure*).

2.2 _ Normes

Les certificats électroniques respectent des standards spécifiant leur contenu de façon rigoureuse. En particulier, la norme X.509 en version 1, 2 et 3 définit à travers plusieurs RFC (*requests for comments*) les propriétés et usages de certificats électronique X.509. Les certificats à la norme X.509 se distinguent des solutions de simple cryptographie (signature et chiffrement) comme OpenPGP (normalisé dans le RFC 2440) par le fait qu'ils contiennent des champs d'information certifiés par des autorités de certification reconnues. Ces champs sont visibles en cliquant sur le petit cadenas d'un navigateur web. Par exemple : un certificat contient le nom de l'entité juridique vérifiée dans le champ Organisation du certificat (société ou personne physique), on y trouve aussi des informations comme le pays et la ville. Le processus d'audit et de vérification de ces informations par les autorités de certification est normalisé dans la norme "webtrust" et à pour but principal d'introduire de la confiance dans les échanges numériques. Les normes sont en perpétuelle évolution, il faut se tenir informé de leurs avancées.

3 _ Utilisation des certificats

3.1 _ Sécurité des systèmes d'information

Les certificats électroniques peuvent être utilisés dans différentes applications informatiques dans le cadre de la sécurité des systèmes d'information pour garantir :

- la non-répudiation et l'intégrité des données avec la signature numérique ou signature électronique (avancée);
- la confidentialité des données grâce au chiffrement des données;
- l'authentification ou l'authentification forte d'un individu ou d'une identité non-physique (Web Serveur - SSL, Poste de travail - 802.1x, VPN IPSEC - SSH - SSL, Code Mobile, documents électroniques).

introduction et installation à la Sécurité informatique

http://www.securiteinfo.com/conseils/introsecu.shtml

Introduction à la sécurité informatique

Le but de cette fiche est de familiariser les lecteurs avec les notions de base de la sécurité informatique, telles que définies dans la norme ISO 7498-2 par exemple.

Les objectifs de la sécurité informatique

La sécurité informatique a plusieurs objectifs, bien sûr liés aux types de menaces ainsi qu'aux types de ressources, etc... Néanmoins, les points principaux points sont les suivants :

- empêcher la divulgation non-autorisée de données
- empêcher la modification non-autorisée de données
- empêcher l'utilisation non-autorisée de ressources réseau ou informatiques de façon générale

Les champs d'application de la sécurité informatique

Ces objectifs s'appliquent dans différents domaines ou champs d'applications, chacun faisant appel à des techniques différentes pour atteindre le ou les mêmes objectifs; ces champs sont:
la sécurité physique

- la sécurité personnelle
- la sécurité procédurale (audit de sécurité, procédures informatiques...)
- la sécurité des émissions physiques (écrans, câbles d'alimentation, courbes de consommation de courant...)
- la sécurité des systèmes d'exploitation
- la sécurité des communications

Terminologie de la sécurité informatique

La sécurité informatique utilise un vocabulaire bien défini (...)

Les vulnérabilités : ce sont les failles de sécurité dans un ou plusieurs systèmes. Tout système vu dans sa globalité présente des vulnérabilités, qui peuvent être exploitables ou non.
Les attaques (exploits): elles représentent les moyens d'exploiter une vulnérabilité. Il peut y avoir plusieurs attaques pour une même vulnérabilité mais toutes les vulnérabilités ne sont pas exploitables.
Les contre-mesures : ce sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique (auquel cas il peut exister d'autres attaques sur la même vulnérabilité).
Les menaces : ce sont des adversaires déterminés capables de monter une attaque exploitant une vulnérabilité.

(...)

Types d'attaques

Les attaques peuvent à première vue être classées en 2 grandes catégories :

- les attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.
- les attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active

peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

Profils et capacités des attaquants

Les attaquants peuvent être classés non-seulement par leur connaissances (newbies, experts, etc...) mais également suivant leurs capacités d'attaques dan sune situation bien définie. Ainsi, on dénombrera les capacités suivantes :

- transmission de messages sans capacité d'écoute (IP spoofing...)
- écoute et transmission de messages
- écoute et perturbation des communications (blocage de paquets, DoS et DDoS...)
- écoute, perturbation et transmissions de messages
- écoute et relai de messages (attaques type man-in-the-middle)

Une autre caractéristique des attaquants va être leur emprise uni-directionnelle ou bi-directionnelle sur les communications, du fait de la nature asymétrique de celles-ci. En effet, la plupart des canaux de transmissions sur Internet ou sur tout autre réseau hétérogène sont uni-directionnels et empruntent des chemins différents suivant les règles de routage. Ainsi, de nombreux protocoles de sécurité sont également unidirectionnels et il faut établir plusieurs canaux pour permettre un échange en "duplex". Ces canaux qui sont au nombre de 2 minimum, sont la plupart du temps gérés de façon totalement indépendante par les protocole de sécurité. C'est le cas pour SSL/TLS mais également pour IPSec dont les associations de sécurité (SA) sont unidirectionnelles et indépendantes, chacune définissant sont propre jeu de clés, algorithmes, etc...

Services principaux de la sécurité informatique

Pour remédier aux failles et pour contrer les attaques, la sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace. A ce niveau, aucune technique n'est encore envisagée; il ne s'agit que d'un niveau d'abstraction visant à obtenir une granularité minimale pour déployer une politique de sécurité de façon optimale (les aspects pratiques tels qu'analyses de risques, solutions technologiques et coûts viendront par la suite). Les principaux services de sécurité sont :

- confidentialité
- authentification (entité, origine des données)
- intégrité
- machines (tamper-résistance, tamper-proofness, exécution sécurisée...)
- données (avec possibilité de récupération)
- flux :
  - mode non-connecté, au niveau paquet (échanges de type requête-réponse, comme UDP)
  - mode orienté-connexion (ensemble de l'échange, comme TCP)
- intégrité de séquences partielles (VoIP, applications, etc... permet d'éviter les DoS par exemple)
- contrôle d'accès (= autorisation, à différentier de l'authentification)
- non-répudiation (avec preuve d'émission ou avec preuve de réception)

Notez que le chiffrement, les signatures digitales et autres techniques correspondent au niveau d'abstraction inférieur, décrit comme l'ensemble des mécanismes de sécurité permettant de réaliser les services décrits ci-dessus. Plusieurs mécanismes peuvent par exemple réaliser le service d'authentification (schémas d'authentification, chiffrement, signatures digitales...). Néanmoins, ces mécanismes de sécurité ne correspondent pas encore aux solutions finales qui seront réellement implémentées. Il faudra pour cela effectuer un dernier raffinement, consistant à choisir les algorithmes symétriques, les algorithmes asymétriques, la tailles des clés, etc...

Enfin, il existe d'autres notions qui ne peuvent être classées directement dans ces listes; la confiance (trust) est un bon exemple. En effet, bien qu'elle soit très coûteuse, la confiance est obligatoire pour garantir l'efficacité des mécanismes de sécurité mis en place. Citons l'exemple d'un protocole d'encapsulation chiffrée (tunneling), développé en soft, permettant d'échanger des données tout en préservant leur confidentialité. Or, si seules les données sont protégées, il est plus simple pour un attaquant de s'introduire dans l'une des machines aux extrémités (PC ou autres), de modifier les librairies correspondantes de façon à fausser le mécanisme de sécurité (nombres aléatoires forcés à une valeur constante, valeurs de clés prédéfinies, algorithmes NULL) et enfin de pouvoir accéder à loisir aux données transmises. D'où la nécessité de mettre en place un schéma de confiance visant à interdire ce type d'attaque; il est nécessaire de pouvoir faire confiance aux équipements de sécurité car dans le cas contraire, l'utilité même des mécanismes de sécurité est remise en question.