**UE D – EXPERTISE PROFESSIONNELLE**

*Mastère européen Manager de projets informatiques*

*UC D51.1 - Environnement informatique*

Corrigé

Type d’épreuve : QCM + écrit

Durée : 4 heures

Session : 2020

*1/ QCM (1h30)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 |  | x |  |  |
| 2 |  | x |  |  |
| 3 |  |  | x |  |
| 4 |  |  |  | x |
| 5 | x |  |  |  |
| 6 |  |  | x |  |
| 7 |  |  | x |  |
| 8 |  | x |  |  |
| 9 |  |  |  | x |
| 10 |  |  |  | x |
| 11 |  |  |  | x |
| 12 | x |  |  |  |
| 13 | x |  |  |  |
| 14 |  |  | x |  |
| 15 |  |  | x |  |
| 16 |  |  | x |  |
| 17 |  |  | x |  |
| 18 |  |  | x |  |
| 19 |  |  |  | x |
| 20 |  | x |  |  |
| 21 |  |  | x |  |
| 22 |  |  |  | x |
| 23 |  | x |  |  |
| 24 |  | x |  |  |
| 25 |  |  |  | x |
| 26 |  |  | x |  |
| 27 |  | x |  |  |
| 28 |  | x |  |  |
| 29 |  |  | x |  |
| 30 |  | x |  |  |
| 31 |  | x |  |  |
| 32 | x |  |  |  |
| 33 |  | x |  |  |
| 34 |  |  |  | x |
| 35 |  |  | x |  |
| 36 |  | x |  |  |
| 37 |  |  |  | x |
| 38 |  |  |  | x |
| 39 | x |  |  |  |
| 40 |  |  | x |  |
| 41 |  |  |  | x |
| 42 |  |  |  | x |
| 43 |  |  |  | x |
| 44 |  |  | x |  |
| 45 |  |  | x |  |
| 46 |  |  | x |  |
| 47 |  |  | x |  |
| 48 |  |  | x |  |
| 49 | x |  |  |  |
| 50 |  |  |  | x |

*2/ Epreuve écrite (2h30)*

* Dossier 1 - Firewalling

**Question 1  : Expliquez le rôle du script suivant :**

iptables -N blockchain

iptables -t filter -A blockchain --jump LOG --log-prefix Blocked

iptables -t filter -A blockchain --jump DROP

iptables -t filter -A INPUT --jump blockchain

iptables -t filter -A OUTPUT --jump blockchain

Solution : 15 points

Ce script Netfilter permet de garder une trace des paquets refusés en créant une chaîne utilisateur appelée « blockchain » ou tous les paquets seront enregistré avec le prefix « Blocked ».

**Question 2 : La commande suivante permet le « masquerading » mais quel en est l’intérêt ?**

iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

Solution : 15 points

Le masquerading permet quant à lui aux machines d'un réseau interne de pouvoir sortir sur un réseau externe en utilisant une seule adresse IP.

**Question 3 : Quelle différence voyez-vous entre le « Masquerading » et les différents types de NAT, expliquez.**

Solution : 30 points

Rappels : NAT est la traduction d'adresse et le PAT la traduction des ports

NAT est une fonction du routeur qui va traduire des adresses IP, son but est de réaliser, au niveau de la passerelle (routeur), une translation (littéralement une « traduction ») des paquets provenant du réseau interne vers le réseau externe. Ce mécanisme de translation d'adresses permet d'assurer une fonction de sécurisation, car pour un utilisateur externe au réseau, toutes les requêtes semblent provenir de l'adresse IP de la passerelle puisque la passerelle camoufle complètement l'adressage interne d'un réseau.

Mais il y a 2 sortes de NAT:

* Le Nat statique qui consiste à associer une adresse IP publique à une adresse IP privée interne au réseau de manière statique
* Le Nat dynamique qui lui permet de partager une adresse IP routable entre plusieurs machines en adressage privé. de là, toutes les machines du réseau interne possèdent virtuellement, vu de l'extérieur, la même adresse IP. C'est la raison pour laquelle le terme de « mascarade IP »ou (IP masquerading) est parfois utilisé pour désigner le mécanisme de translation d'adresse dynamique.

Afin de pouvoir partager les différentes adresses IP sur une ou plusieurs adresses IP routables le NAT dynamique utilise le mécanisme de translation de port (PAT - Port Address Translation),c'est-à-dire affecter un port source différent à chaque requête de telle manière à pouvoir maintenir une correspondance entre les requêtes provenant du réseau interne et les réponses des machines sur Internet, toutes adressées à l'adresse IP du routeur.

IL y a aussi le NAT source et NAT destination:

* Le NAT source change l’adresse **source** dans l'en- [tête](http://en.wikipedia.org/wiki/IPv4#Header) IP d'un paquet. Cela peut également changer le port **source** dans les en-têtes TCP / UDP. L'utilisation typique consiste à changer l'adresse / port privé (rfc1918) en une adresse / port public pour les paquets quittant votre réseau.
* Le NAT de destination change l’adresse de **destination** dans l'en-tête IP d'un paquet. Cela peut également changer le port de **destination** dans les en-têtes TCP / UDP. Son utilisation typique consiste à rediriger les paquets entrants avec une destination d'adresse / de port public vers une adresse / port IP privée à l'intérieur de votre réseau.

**En conclusion**

**Le masquage est une forme spéciale de NAT source, où l’adresse source est inconnue au moment où la règle est ajoutée aux tables du noyau. Si vous souhaitez autoriser des hôtes avec une adresse privée derrière votre pare-feu à accéder à Internet et que l’adresse externe est variable (DHCP), c’est ce que vous devez utiliser. Le masquage modifie l'adresse IP source et le port du paquet en tant qu'adresse IP principale attribuée à l'interface sortante**.

Si votre interface sortante a une adresse statique, vous n'avez pas besoin d'utiliser MASQ, mais vous pouvez utiliser SNAT, ce qui sera un peu plus rapide, car il n'est pas nécessaire de déterminer quelle est l'adresse IP externe à chaque fois.

* Dossier  2- Routage

**Question 1 : Expliquez le script suivant, et notament les termes  : #router-id, ospf 1, area 0.**

R1(config)#router ospf 1

R1(config-router)#router-id 1.1.1.1

R1(config-router)#network 172.16.1.1 0.0.0.0 area 0

R1(config-router)#network 172.16.2.1 0.0.0.0 area 0

R2(config)#router ospf 1

R2(config-router)#router-id 2.2.2.2

R2(config-router)#network 172.16.1.2 0.0.0.0 area 0

R2(config-router)#network 172.16.3.2 0.0.0.0 area 0

R2(config-router)#network 10.0.0.0 0.0.3.255 area 0

R3(config)#router ospf 1

R3(config-router)#router-id 3.3.3.3

R3(config-router)#network 172.16.3.3 0.0.0.0 area 0

R3(config-router)#network 172.16.2.3 0.0.0.0 area 0

Solution : 15 points

C’est un script de configuration de routage OSPF

Le numéro « 1 » après la commande « routeur OSPF », correspond au processus ID.

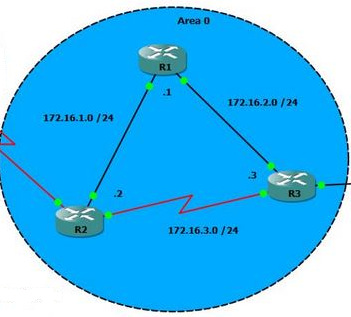
Il a une signification locale, cela permet d’activer plusieurs processus OSPF sur le routeur.

Area 0 indique le numéro de la zone.

#router-id : identifie le Neighbor id

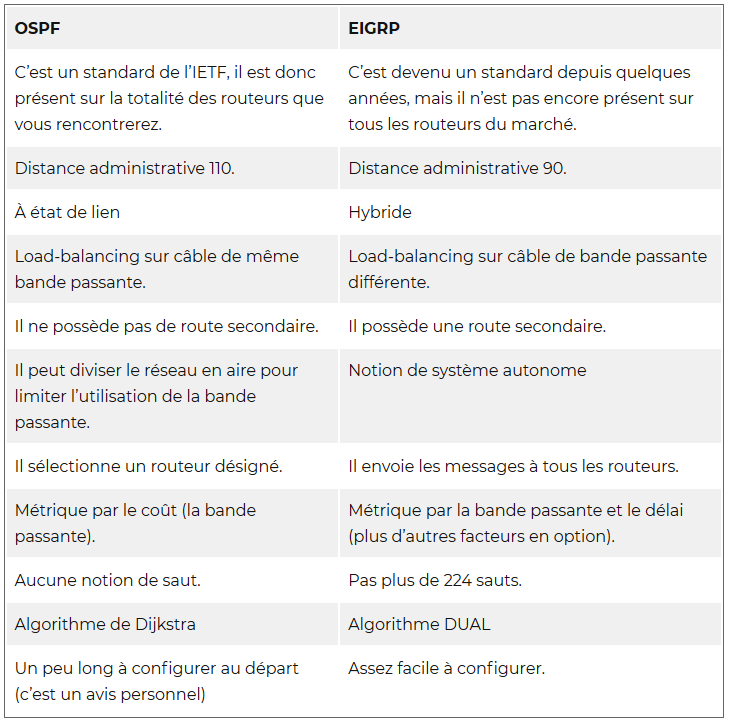
**Question 2 : Représentez le réseau définit ci-dessus par un schéma.**

Solution : 15 points



**Question 3** : **Réalisez un tableau comparatif entre les protocoles OSPF et EIGRP.**

Solution : 30 points



* Dossier 3 – Base de Données Relationnel-Objet

**Soit le Type T\_Point définit comme suit en langage SQL3 d’Oracle:**

**CREATE TYPE T\_Point AS OBJECT (  
 x number,  
 y number  
);**

**Question 1 : Définissez un type d’objet T\_Ligne caractérisée par deux points définis précédemment.**

Solution : 10 points

* + - CREATE TYPE T\_Ligne AS OBJECT (  
       point1 T\_Point,  
       point2 T\_Point  
      );

**Question 2 : Créer une ligne de votre choix.**

Solution : 20 points

* + - CREATE TABLE Lignes OF T\_Ligne;
    - INSERT INTO Lignes VALUES (T\_Point(5,5),T\_Point(10,10));

**Question 3 : Afficher les points de cette ligne définit ci-dessus.**

Solution : 10 points

* + - SELECT L.point1.x, L.point1.y, L.point2.x, L.point2.y FROM Lignes L;

**Question 4 : La base de données Relationnel-Objet Oracle propose des types de collections comme les NESTED TABLE et les VARRAY, définissez les.**

Solution : 10 points

* NESTED TABLE (Les tables en tant qu’attributs de tables)

Définit un ensemble non ordonné

De taille variable et non limité

Les éléments peuvent être accédés singulièrement

* VARRAY (Un VARRAY contient (0 ou n) éléments du même type)

Définit un ensemble ordonné

De taille variable mais limité (à définir au début)

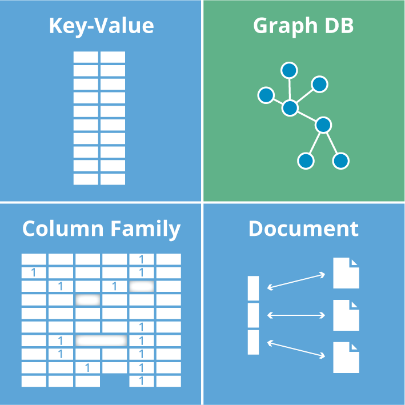
Les éléments ne peuvent pas être accédés singulièrement

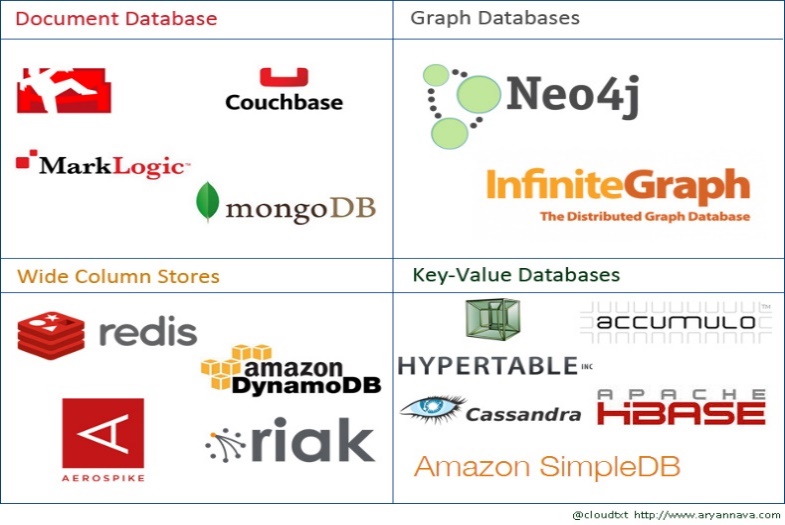
* Dossier 4 - Big Datas

**Question 1 : Il existe 4 familles de bases « Big datas - NoSQL », citez les, expliquez succinctement chaque principe et donnez 1 exemple de base que vous connaissez pour chacune de ces familles.**

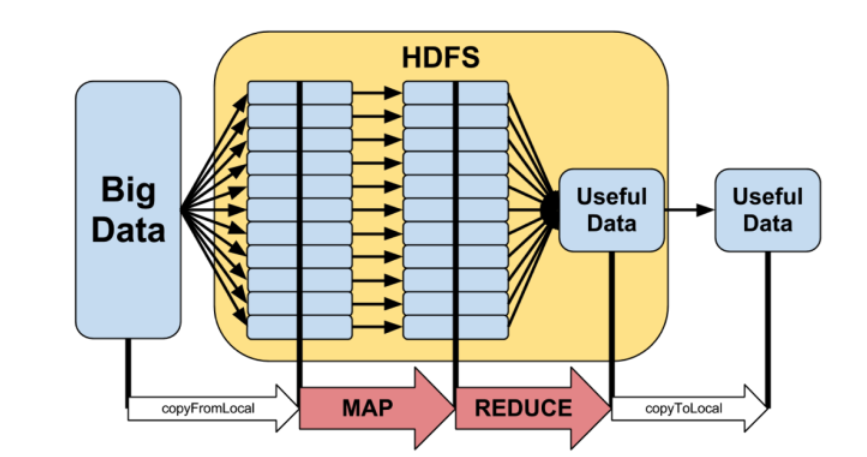
Solution : 20 points

* Key-values Stores
  + Hash table where there is a unique key and a pointer to a particular item of data.
  + Focus on scaling to huge amounts of data
  + E.g. cassandra, redis
* Column Family Stores
  + To store and process very large amounts of data distributed over many machines
  + E.g. Cassandra, HBase
* Document Databases
  + Collections of Key-Value collections
  + The next level of Key/value, allowing nested values associated with each key.
  + Appropriate for Web apps.
  + E.g. CouchDB, MongoDb
* Graph Databases
  + Bases on property-graph model
  + Appropriate for Social networking, Recommendations
  + E.g. Neo4J, Infinite Graph





**Question 2 : Expliquez à partir du schéma ci-dessous ce que sont HDFS et MAP REDUCE.**



Solution : 20 points

HDFS

HDFS est le système de fichier distribué de  Hadoop Apache. Il s’agit d’un composant central du Framework de Apache, et plus précisément de son système de stockage. Découvrez le fonctionnement, les avantages et les inconvénients de HDFS.

HDFS (Hadoop Distributed File System) est un système de fichier distribué permettant de stocker et de récupérer des fichiers en un temps record. Il s’agit de l’un des [composants basiques du framework Hadoop](https://www.lebigdata.fr/hadoop) Apache, et plus précisément de son système de stockage. HDFS Hadoop compte parmi les projets « top level » de Apache

De par sa capacité massive et sa fiabilité, HDFS est un système de stockage très adapté au Big Data. En combinaison avec YARN, ce système augmente les [possibilités de gestion de données](https://www.lebigdata.fr/data-management-top-des-meilleurs-fournisseurs) du cluster HDFS Hadoop et permet donc de traiter le Big Data efficacement. Parmi ses principales fonctionnalités, on compte la possibilité de stocker des terabytes, voire des petabytes de données.

Le système est capable de gérer des milliers de nœuds sans intervention d’un opérateur. Il permet de bénéficier simultanément des avantages du computing parallèle et du computing distribué. Après une modification, il permet de restaurer facilement la précédente version d’une donnée.

HDFS peut être lancé sur commodity hardware, ce qui le rend très tolérant aux erreurs. Chaque donnée est stockée à plusieurs endroits, et peut donc être récupérée en toutes circonstances. De même, cette réplication permet de lutter contre la corruption potentielle des données.

Les serveurs sont connectés et communiquent par le biais de protocoles TCP. Même s’il est conçu pour les bases de données massives, les systèmes de fichiers normaux comme FAT et NTFS sont également compatibles. Enfin, une fonctionnalité CheckpointNode permet de vérifier le statut des nœuds en temps réel.

MAP REDUCE

MapReduce est un Framework de traitement de données en clusters. Composé des fonctions Map et Reduce, il permet de répartir les tâches de traitement de données entre différents ordinateurs, pour ensuite réduire les résultats en une seule synthèse. Découvrez ses origines, son fonctionnement, ses avantages, et les critiques émises à son égard.

MapReduce est un modèle de programmation créé par Google pour le traitement et la génération de larges ensembles de données sur des clusters d’ordinateurs. Il s’agit d’un composant central du Framework logiciel [Apache Hadoop](https://www.lebigdata.fr/hadoop), qui permet le traitement résilient et distribué d’ensembles de données non structurées massifs sur des clusters d’ordinateurs, au sein desquels chaque nœud possède son propre espace de stockage. Concrètement, le framework propose deux fonctionnalités principales. Il répartit le labeur sur les différents nœuds du cluster (map), puis les organise et réduit les résultats fournis par chaque nœud en une seule réponse cohérente à une requête. Cela est rendu possible grâce à [son système de fichiers distribués HDFS.](https://www.lebigdata.fr/hdfs-fonctionnement-avantages)

MapReduce est un patron d'architecture de développement informatique, inventé par Google1, dans lequel sont effectués des calculs parallèles, et souvent distribués, de données potentiellement très volumineuses, typiquement supérieures en taille à 1 téraoctet.

Les termes « map » et « reduce », et les concepts sous-jacents, sont empruntés aux langages de programmation fonctionnelle utilisés pour leur construction (map et réduction de la programmation fonctionnelle et des langages de programmation tableau).

MapReduce permet de manipuler de grandes quantités de données en les distribuant dans un cluster de machines pour être traitées. Ce modèle connaît un vif succès auprès de sociétés possédant d'importants centres de traitement de données telles Amazon.com ou Facebook. Il commence aussi à être utilisé au sein du Cloud computing. De nombreux frameworks ont vu le jour afin d'implémenter le MapReduce. Le plus connu est Hadoop qui a été développé par Apache Software Foundation. Mais ce framework possède des inconvénients qui réduisent considérablement ses performances notamment en milieu hétérogène. Des frameworks permettant d'améliorer les performances de Hadoop ou les performances globales du MapReduce, tant en termes de vitesse de traitement qu'en consommation électrique, commencent à voir le jour.

**Question 3 : Expliquez ce qu’est SPARK et Pourquoi est-il souvent utilisé conjointement à HADOOP ?**

Solution (20 points)

Spark (ou Apache Spark[2](https://fr.wikipedia.org/wiki/Apache_Spark#cite_note-2)) est un [framework](https://fr.wikipedia.org/wiki/Framework) [open source](https://fr.wikipedia.org/wiki/Open_source) de [calcul distribué](https://fr.wikipedia.org/wiki/Calcul_distribu%C3%A9). Il s'agit d'un ensemble d'outils et de composants logiciels structurés selon une architecture définie. Développé à [l'université de Californie à Berkeley](https://fr.wikipedia.org/wiki/Universit%C3%A9_de_Californie_%C3%A0_Berkeley) par AMPLab , Spark est aujourd'hui un projet de la [fondation Apache](https://fr.wikipedia.org/wiki/Fondation_Apache). Ce produit est un cadre applicatif de traitements [big data](https://fr.wikipedia.org/wiki/Big_data) pour effectuer des analyses complexes à grande échelle

Spark ne dispose pas de [système de gestion de fichier](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_fichiers_distribu%C3%A9) qui lui est propre. Il est nécessaire de lui en fournir un, par exemple [Hadoop Distributed File System](https://fr.wikipedia.org/wiki/Hadoop#Hadoop_Distributed_File_System), [Informix](https://fr.wikipedia.org/wiki/Informix), [Cassandra](https://fr.wikipedia.org/wiki/Cassandra_(base_de_donn%C3%A9es)), [OpenStack Swift](https://fr.wikipedia.org/wiki/OpenStack) ou [Amazon S3](https://fr.wikipedia.org/wiki/Amazon_S3) . Il est conseillé de l'utiliser avec Hadoop qui reste actuellement la meilleure solution globale de stockage grâce à ses outils d'administration, de sécurité et de [monitoring](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_fichiers) plus avancés.

**SPARK il plus rapide que HADOOP MAP REDUCE**

Spark réalise une lecture des données au niveau du [cluster](https://fr.wikipedia.org/wiki/Grappe_de_serveurs) (grappe de serveurs sur un réseau),effectue toutes les opérations d'analyse nécessaires, puis écrit les résultats à ce même niveau. Malgré le fait que les tâches s'écrivent avec les langages [Scala](https://fr.wikipedia.org/wiki/Scala_(langage)), [Java](https://fr.wikipedia.org/wiki/Java_(langage)) et [Python](https://fr.wikipedia.org/wiki/Python_(langage)), il utilise au mieux ses capacités avec son langage natif, Scala .

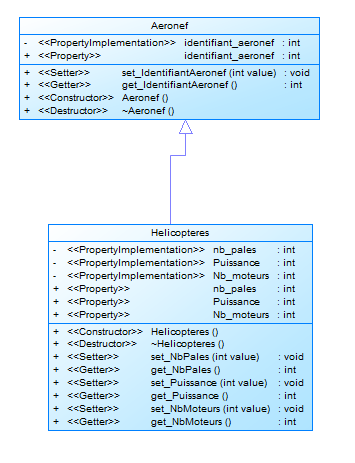
De ce fait, là où le MapReduce de Hadoop travaille par étape, Spark peut travailler sur la totalité des données en même temps. Il est donc jusqu'à dix fois plus rapide pour le traitement en lots et jusqu'à cent fois plus rapide pour effectuer l'analyse en mémoire.

Spark exécute la totalité des opérations d'analyse de données en mémoire et en temps réel. Il s'appuie sur des disques seulement lorsque sa mémoire n'est plus suffisante. À l'inverse, avec Hadoop les données sont écrites sur le disque après chacune des opérations . Ce travail en mémoire permet de réduire les temps de latence entre les traitements, ce qui explique une telle rapidité.

Annexe 2

* Dossier 5 - Langage C# et XAMARIN

**Question 1 : A partir du diagramme de classes ci-dessous donnez le code C# correspondant à la classe hélicoptères.**



Solution (20 points) (Notamment pour l’héritage et les Property Implementation)

// File: Helicopteres.cs

// Author: wilfrid MOREL

// Created: mardi 14 avril 2020 15:24:34

// Purpose: Definition of Class Helicopteres

using System;

public class Helicopteres : Aeronef

{

private int nbPales;

private int puissance;

private int nbMoteurs;

public int NbPales

{

get

{

return nbPales;

}

set

{

this.nbPales = value;

}

}

public int Puissance

{

get

{

return puissance;

}

set

{

this.puissance = value;

}

}

public int NbMoteurs

{

get

{

return nbMoteurs;

}

set

{

this.nbMoteurs = value;

}

}

public Helicopteres()

{

throw new NotImplementedException();

}

}

**Question 2 : Expliquez ce qu’est Xamarin.**

Solution (20 points)

Xamarin est une plateforme open source qui permet de générer des applications modernes et performantes pour iOS, Android et Windows avec .NET. Xamarin est une couche d’abstraction qui gère la communication entre le code partagé et le code de plateforme sous-jacent. Xamarin s’exécute dans un environnement managé qui offre des commodités telles que l’allocation de mémoire et le garbage collection.

Xamarin permet aux développeurs de partager en moyenne 90 % de leur application entre les plateformes. Ce modèle permet aux développeurs d’écrire toute leur logique métier dans un seul langage (ou de réutiliser du code d’application existant) tout en obtenant des performances, une apparence et une convivialité natives sur chaque plateforme.

Les applications Xamarin peuvent être écrites sur PC ou Mac et être compilées dans des paquets d’application natifs, par exemple un fichier .apk sur Android ou un fichier .ipa sur iOS.

**Question 3 : Xamarin s’appuie sur « Mono », mais qu’est-ce que Mono ?**

Solution (10 points)

Xamarin s’appuie sur Mono, une version open source du .NET Framework basée sur les standards ECMA de .NET. Mono existe depuis presque aussi longtemps que le .NET Framework. Il s’exécute sur la plupart des plateformes, notamment Linux, Unix, FreeBSD et macOS. L’environnement d’exécution de Mono gère automatiquement les tâches telles que l’allocation de mémoire, le garbage collection et l’interopérabilité des plateformes sous-jacente

**Question 4 : Définissez ce qu’est « IL »dans Visual studio.net ?**

Solution (10 points)

Les applications Xamarin.Android sont compilées du C# en IL (langage intermédiaire), lequel fait ensuite l’objet d’une compilation JIT (juste-à-temps) en assembly natif au moment du lancement de l’application. Les applications Xamarin.Android s’exécutent dans l’environnement d’exécution Mono, côte à côte avec la machine virtuelle ART (Android Runtime). Xamarin fournit des liaisons .NET aux espaces de noms Android.\* et Java.\*. L’environnement d’exécution Mono fait appel à ces espaces de noms via les wrappers MCW (Managed Callable Wrappers), et fournit les wrappers ACW (Android Callable Wrappers) à la machine virtuelle ART, ce qui permet aux deux environnements d’appeler du code

**GRILLE DE NOTATION ECRIT**

**UC D51.1 - Environnement informatique**

NOM ET PRENOM DU CORRECTEUR\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

N° de candidat\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Dossier | Note attribuée | Observations obligatoires |
| Dossier 1 - Firewalling | /60 |  |
| Dossier 2 - Routage | /60 |  |
| Dossier 3 - Base de Données Relationnel-Objet | /50 |  |
| Dossier 4 - Big Datas | /60 |  |
| Dossier 5 - Langage C# et XAMARIN | /60 |  |
| Présentation et orthographe | /10 |  |
| **TOTAL** | /300 |  |

Appréciation générale :

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fait à \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ le \_\_\_\_\_\_\_\_\_\_\_\_

Signature :