CONCOURS DE L'AGRÉGATION INTERNE

« ÉCONOMIE ET GESTION »

SESSION 2015

SECONDE ÉPREUVE

Épreuve de cas pratique dans la spécialité correspondant à l'option choisie par le candidat

Option D

Durée de préparation : 4 heures

Durée totale de l'épreuve : une heure (exposé : quarante minutes maximum ; entretien : vingt minutes maximum) ; coefficient 1.

AVERTISSEMENT

Si le texte du sujet, de ses questions ou de ses annexes, vous conduit à formuler une ou plusieurs hypothèses, il vous est demandé de la ou de les mentionner **explicitement** lors de votre exposé

Présentation de la société[1]

Dans les organisations le patrimoine applicatif est crucial : il représente, automatise et contrôle le cœur des activités de l'entreprise. Or à mesure que la stratégie et l'activité changent, les applications évoluent. Ce patrimoine devient alors de plus en plus complexe au gré de l'arrivée de nouvelles applications, des évolutions de l'existant, des nouveaux développements, des nouvelles technologies. En parallèle les collaborateurs possédant la connaissance au sein de la DSI ou coté maîtrise d'ouvrage, changent de fonction ou quittent l'entreprise. La connaissance du patrimoine applicatif est alors perdue.

IT4Control se propose de fournir aux Directions des Systèmes d'Information les moyens de réussir l'alignement de l'IT sur les objectifs de l'entreprise, de simplifier sa gestion, et d'instaurer un dialogue stratégique avec la Direction Générale et les Directions Métiers. La solution IT4 vise à gagner de la visibilité et une compréhension plus fine du fonctionnement et des enjeux business, techniques et économiques du système d'information.

Pour cela IT4Control propose une plateforme pour la gestion du SI et de sa gouvernance articulée autour de plusieurs produits :

- **4PPM «** *Project Portfolio Management* » : pour la gestion de portefeuille de projets
- **4APM «** *Application Portfolio Management* » : pour la Gestion de Patrimoine Applicatif
- **4DRM »***Demand & Request Management* » : pour la Gestion des demandes, bogues, anomalies et problèmes
- **4ISM «** *IT Service Management* » : pour le Pilotage des processus et services de la DSI, disponible paramétré pour les processus ITIL.
- **4UDT »** *Universal Data Tracking* » : pour le suivi et la traçabilité des données au sein du SI. Ce module est en cours de développement et traitera en particulier la problématique des données à caractère personnel (DCP). .

Ce cas s'appuie sur la plateforme d'IT4Control et propose d'étudier ses différents aspects à la fois dans son utilisation, sa conception et son architecture.

Les annexes décrivent une partie de la solution IT4 :

- **Annexe 1 : Fonctionnalités, cartographie des applications et infrastructure**
- **Annexe 2 : Architecture technologique**
- **Annexe 3 : Méta modèle**

[1] (http://www.it4control.eu)

Partie 1 : SI et gouvernance

Le produit IT4 se veut un produit permettant d'améliorer la gouvernance du SI ainsi que l'alignement de celui-ci sur la stratégie.

1. *Quels sont les liens entre stratégie et SI ?*
2. *En quoi ce type de produit améliore-t-il la gouvernance du SI ?*

Vous illustrerez vos propos par des exemples pertinents.

Partie 2 : Architecture applicative

Le produit IT4 est construit autour d'un référentiel dont l'architecture est décrite en annexe 2.

Ce produit est construit selon une architecture en couches

1. *Quel est l'intérêt d'avoir une architecture dans une application et plus particulièrement une architecture en couches ?*

L'annexe 2 propose un extrait de code d'une application permettant d'afficher les éléments de l'infrastructure. On souhaite faire évoluer cette partie d'application pour qu'elle permette aussi de :

- obtenir la liste des infrastructures par ordre alphabétique
- afficher le renseignement sur le type de l'infrastructure
- présenter l'affichage du résultat sous forme de liste à puces

2. *Sur quelles couches faut-il intervenir pour cela ?*
3. *Proposer les modifications de code nécessaires.*

Le référentiel IT4 est un méta modèle dont la structure principale est fournie en annexe3. Ce méta-modèle est présenté aux clients d'IT4Control comme une force.

4. *Quels sont les intérêts et limites de l'utilisation d'un méta-modèle ?*

Partie 3 : Architecture réseau

Ici, nous nous intéresserons en particulier à la partie architecture matérielle du produit d'IT4Control et nous appuierons donc sur l'annexe 1 en restant focalisé sur l'identification des matériels et l'interconnexion de ceux-ci. L'exemple 2 donne un aperçu d'un élément de l'infrastructure de façon plus détaillée.

La solution d'IT4Control permet de tracer une cartographie complète de l'architecture matérielle.

1. *Quelles sont d'après vous les solutions techniques permettant de réaliser automatiquement cette opération ? Vous éclairerez votre propos en donnant des solutions avec les protocoles utilisés et les avantages/inconvénients de chacune d'elles.*
2. *Quels sont les différents types d'éléments qu'il vous parait important de prendre en compte dans cette cartographie ? Pour répondre à cette question, vous pouvez vous appuyer sur le modèle en couche de type OSI ou tout autre modèle pertinent.*

La solution d'IT4Control du fait de l'intégration de la cartographie à sa solution ne s'appuie pas sur des applications externes. Les ingénieurs ont développé la solution afin de l'intégrer de façon native à leur application en s'appuyant sur le protocole SNMP. L'application permettant la cartographie d'architecture est hébergée sur le serveur Server—01.

3. *Expliquez de façon détaillée en quoi consiste le protocole SNMP et quels sont les éléments importants à prendre en compte afin de collecter les informations des produits supervisés. Vous pouvez étayer votre développement en vous appuyant sur un ou plusieurs exemples.*
4. *Afin que tous les éléments d'architecture, quels que soit le site d'implantation et leurs détails puissent être analysés via le protocole SNMP et remontés au serveur, quelles solutions techniques proposez-vous de mettre en œuvre ?*

Partie 4 : Modélisation

Un des éléments important du métier d'IT4Control repose sur l'architecture distribuée. Dans ce cadre, il est important de cartographier les applications en s'intéressant notamment à leur localisation et à la gestion des accès.

- Une application représente une entité métier (Vente, RH, Production) qui est découpée en modules. Chaque module peut communiquer avec son module père ou fils suivant des méthodes normalisées.
- Chacun des modules peut être installé au plus près de son utilisation sur un des serveurs de l'organisation.
- De même les modules sont accessibles en fonction d'une grille des droits répertoriant les autorisations d'accès au module pour les utilisateurs (exécution, lecture seule, modification …).
- La définition des droits se fait à travers des groupes de l'annuaire fédérateur.
- Par une simple authentification, l'utilisateur accède à des modules de l'application.

1. *Proposer une modélisation de cette cartographie.*
2. *Les applications doivent pouvoir communiquer entre elles. De quelle façon cela vous parait-il réalisable en respectant les règles de communication inter-couches ?*

Une des forces de cette organisation est la modularité et la possibilité d'évolution en méthode agile. Il est donc indispensable de faire évoluer les modules de façon permanente. Un module peut donc être présent dans l'organisation en plusieurs versions (test, production…). Il est donc important de savoir à tout moment quelles sont les versions utilisées ainsi que leur localisation. Certaines versions ne sont accessibles qu'à des groupes bien identifiés.

3. *Adaptez votre modèle en permettant de gérer cette modularité*
4. *En quoi les méthodes agiles sont-elles un avantage pour l'organisation ?*

Annexe1 : Fonctionnalités, cartographie des applications et infrastructure

Le logiciel IT4 permet une cartographie des infrastructures et applicative générée automatiquement sans effort de dessin. Pour cela il dispose notamment de :

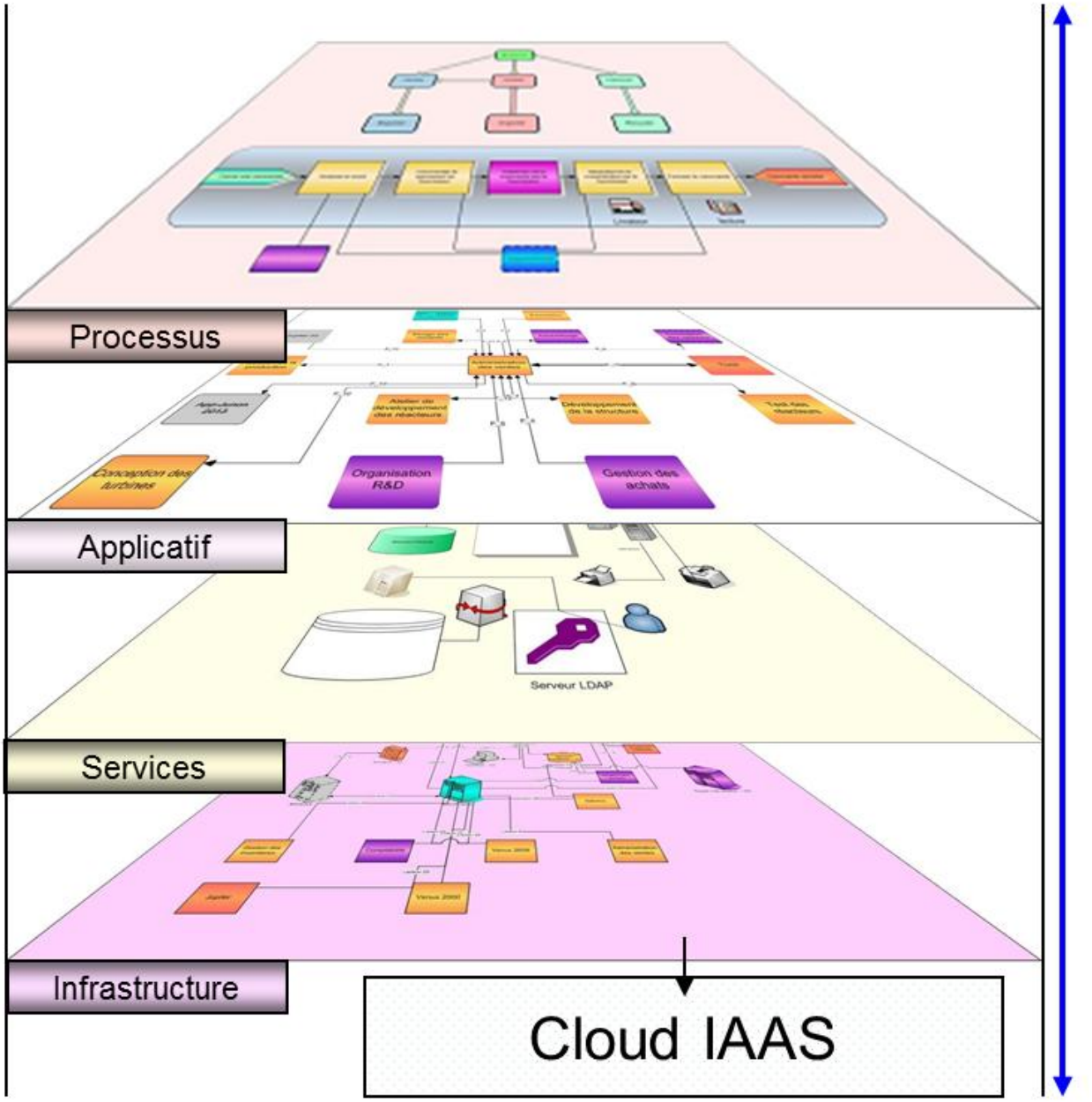
- Un référentiel des infrastructures serveurs, bases de données, applications, (CI ou Configuration Items), incluant les configurations, le nombre d'utilisateurs, etc....
- Le détail des relations qui unissent tous ces objets.
- Une description précise de chaque objet (serveurs, bases de données, relation etc...) comportant tous les paramètres nécessaires à sa définition, son identification et son fonctionnement. Cet ensemble de paramètres peut être enrichi au cours du temps.

Pour l'urbanisation du SI il permet la définition de domaines, blocs, quartier, ilots, relations et de générer des rapports de composition.

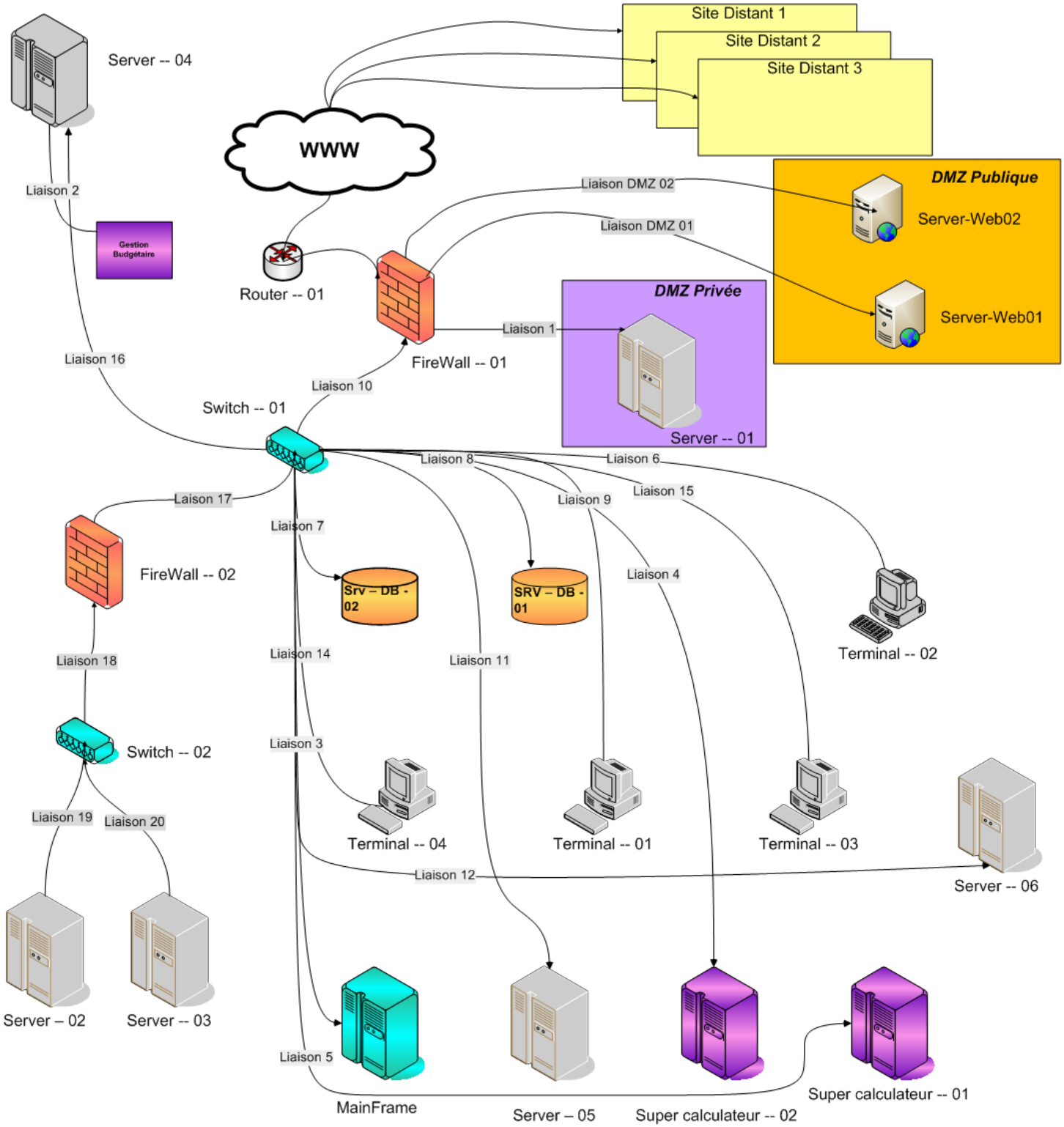
Il permet une visibilité intéressante sur le patrimoine applicatif ainsi que sur ses constituants, les applications ou mailles applicatives, en proposant les fonctionnalités suivantes :

- Référentiel applicatif : technique, métier, urbanisation, financier, compétences associées
- Fiches applicatives, « carte d'identité » de chaque application
- Gestion du cycle de vie des applications
- Cartographie applicative
- Intégration des composantes projets des applications (évolutions, nouvelles fonctionnalités, etc.) grâce au module 4PPM ou par connexion avec des solutions tierces
- Intégration des demandes d'évolution et des anomalies grâce au module 4DRM ou par connexion avec des solutions tierces
- Prise en compte de la qualité fournie par les applications en termes de performance et disponibilité
- Suivi des indicateurs, qui donnent l'état de santé des applications et du patrimoine, notamment le nombre de points de fonction, la maintenabilité, le niveau de risque, etc…, par collecte des informations issues des solutions Cast Software d'Application Intelligence ou autres solutions du marché
- Restitution et rapports ad hoc et sur demande

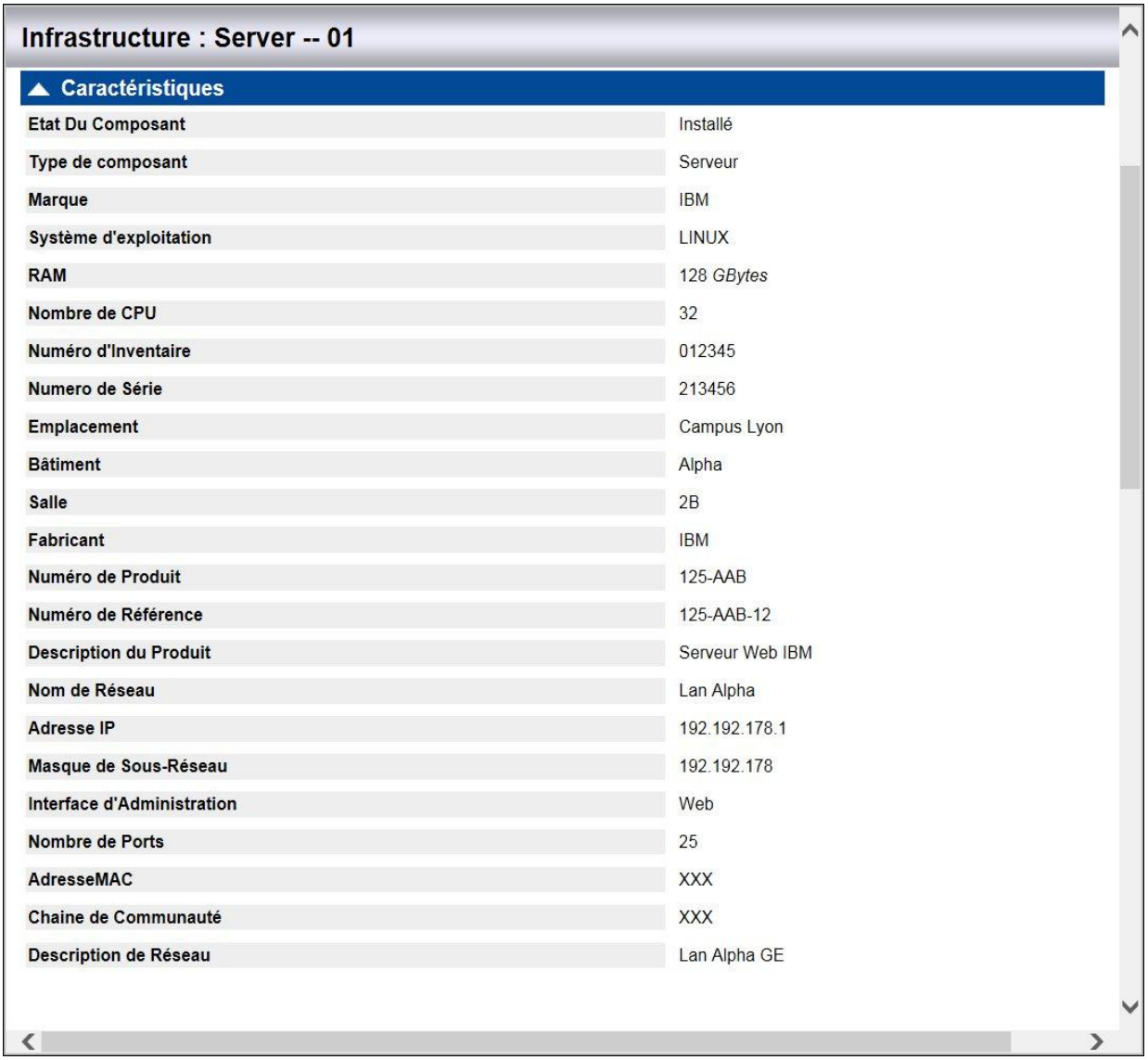
Exemple 1 : Cartographie applicative et description du modèle en couche



Exemple 2 : Cartographie d'une infrastructure



Exemple 3: Description d'un objet (ici un serveur)



Infrastructure : Server -- 01

Caractéristiques

Etat Du Composant	Installé
Type de composant	Serveur
Marque	IBM
Système d'exploitation	LINUX
RAM	128 *GBytes*
Nombre de CPU	32
Numéro d'Inventaire	012345
Numero de Série	213456
Emplacement	Campus Lyon
Bâtiment	Alpha
Salle	2B
Fabricant	IBM
Numéro de Produit	125-AAB
Numéro de Référence	125-AAB-12
Description du Produit	Serveur Web IBM
Nom de Réseau	Lan Alpha
Adresse IP	192.192.178.1
Masque de Sous-Réseau	192.192.178
Interface d'Administration	Web
Nombre de Ports	25
AdresseMAC	XXX
Chaine de Communauté	XXX
Description de Réseau	Lan Alpha GE

Annexe2 : Architecture technologique

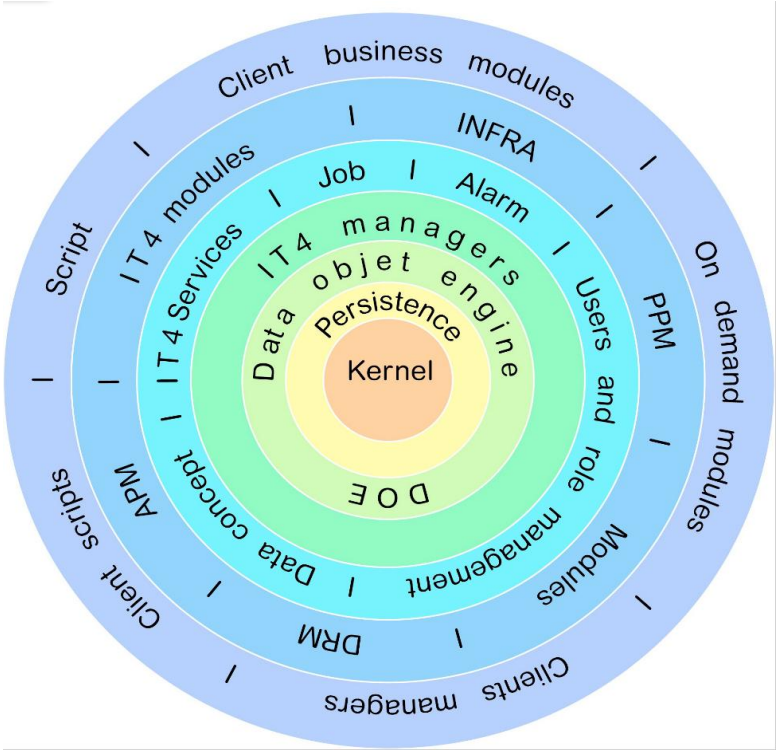
IT4 est une infrastructure technologique web avec accès à un référentiel

Le référentiel est le véritable cœur de la solution et datamart du SI, qui classe, consolide, historise toutes les informations et données du système d'information. Ce référentiel intègre ainsi toutes les données liées aux applications, projets, technologies utilisées, responsables internes, contacts externes, indicateurs métiers, économiques, techniques, etc. Le référentiel peut également être utilisé comme CMDB (base de données de gestion de configuration) et intégrer des informations nécessaires à l'exploitation. Il est possible de paramétrer les produits IT4 pour établir des règles de relation et d'impact entre éléments du SI

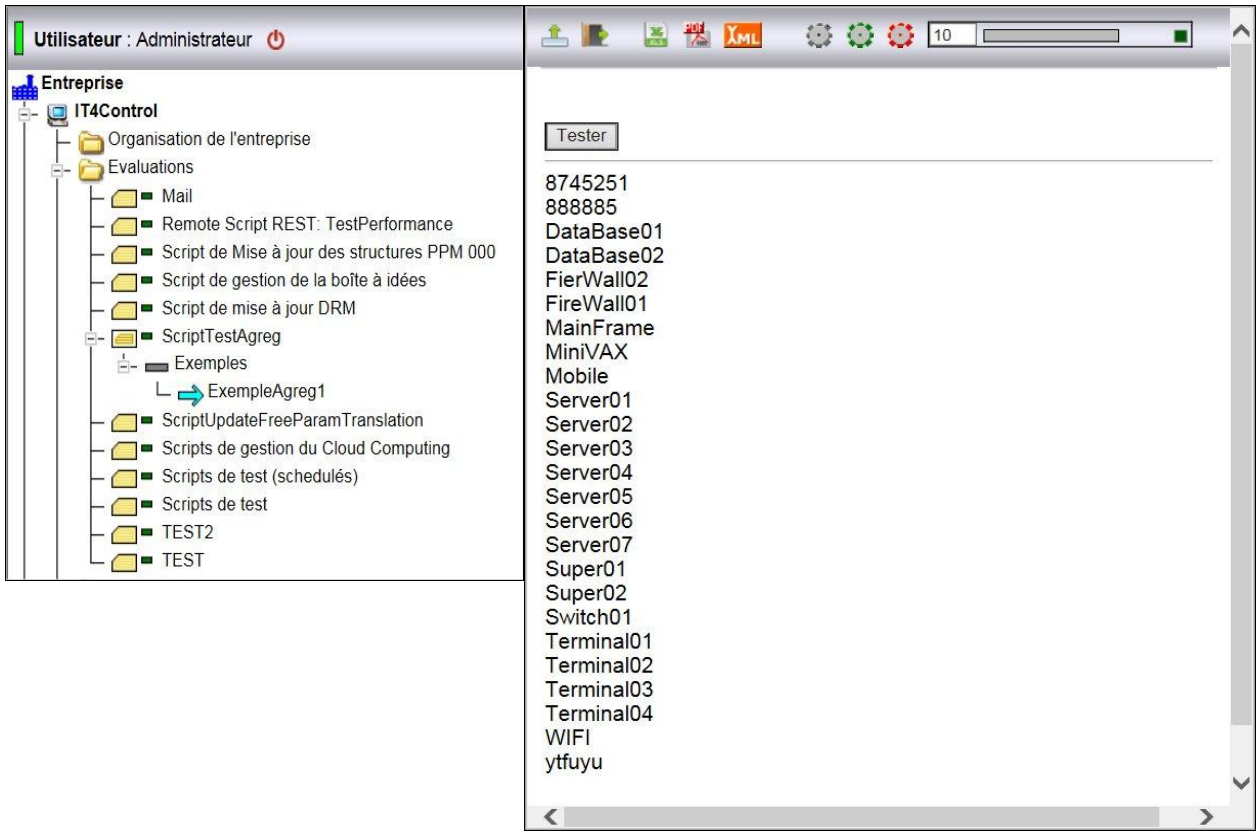
La gestion des données

Les données sont mémorisées dans une base de données relationnelle qui est le véritable référentiel de la DSI. Elles seront ainsi exploitées à tout moment par IT4 pour fournir le reporting et les tableaux de bord

Structure interne du produit



Exemple d'application (ici affichage des éléments d'infrastructure)



La sélection de « Exemples.exempleAgreg1 » dans le menu de gauche affiche l'écran à droite et après l'activation du bouton détaille les éléments d'infrastructure dans la même fenêtre.

Extrait du code :

Remarque : Seules les méthodes essentielles sont indiquées

-Couche 1 « SCRIPT » : voici le code du bouton permettant d'afficher l'infrastructure:

```
<input type="button" onclick=" var res =
runServerScript('ScriptTestAgreg.Exemples.ExempleAgreg1',");
$$('ZoneResultat').innerHTML = res; " value="Tester">


…..
<hr>


<div id="ZoneResultat">
```

*sur activation déclenche la fonction runServerScript (non décrite ici) appelant une méthode de la couche IT4Modules (non décrite ici) qui renvoie une liste html qui sera intégrée dans la zone *

-Couche 2 « IT4 modules »: non décrite ici

appel de la méthode execute d'une classe de service, retourne le résultat récupéré de la classe service

-Couche 3 « IT4 Services » :

```
...
 public String execute {
      try {
            // Récupèrel'identifiant des objets « infrastructure »
        ViewStructSTD viewStructSTD =
structSTDManager.getStructSTDByName("Infrastructure");
            // Obtient la collection des objets infrastructure
        ArrayList<ViewGenericObject> infraList =
genericObjectManager.getGenericObjectList(viewStructSTD.getId());
            // Liste les instances          ViewGenericObject GOB = null;
        String result = "";
        // construit une liste (au sens HTML avec les noms des composants de l'infrastructure )
        for(int i = 0; i < infraList.size(); i++) {
           GOB = infraList.get(i);
            result += GOB.getName() + "<br/>";
        }
        return result
       } catch …..
```

-Couche 4 « IT4 managers » :

```
public ArrayList<ViewGenericObject> getGenericObjectList(long structSTDId) {
    ArrayList<ViewGenericObject> list =
this.dBAccessGenericObject.getGenericObjectListFromDataBase(null, structSTDId);
    return list;
}
// appelle la couche DOE
```

-Couche 5 « Data Object Engine » DOE

```
public ArrayList<ViewGenericObject> getGenericObjectListFromDataBase (Connection
connection, long structSTDId) {

    Connection c  = null;
    java.sql.Statement stmt = null;
    ResultSet  r = null;
    ViewGenericObject viewGenericObject = null;
    ArrayList<ViewGenericObject> list = new ArrayList<ViewGenericObject>();

    // ouverture d'une connection
    if (connection == null)
      c = dBConnection.openConnection();
    else c = connection;


    if (c != null) {
       try {
          stmt = c.createStatement(ResultSet.TYPE_FORWARD_ONLY,
ResultSet.CONCUR_READ_ONLY);

          String request =
             "SELECT * FROM plt_GenericObject " +
             " WHERE structSTDId = " + structSTDId ;

          r = stmt.executeQuery (request);
         // lecture du jeu d'enregistrements
          while(r.next()) {
             viewGenericObject = setViewGenericObject(r);
             list.add(viewGenericObject);
          }

       } catch (Exception exp) {
          …....
     }

    return list;
}
```

Annexe 3 : Meta modèle

Adaptive Object-Models provide an alternative to traditional object-oriented design. Traditional object-oriented design generates classes for the different types of business entities and associates attributes and methods with them. The classes model the business, so a change in the business causes a change to the code and leads to a new version of the application. An Adaptive Object-Model does not model these business entities as classes. Rather, they are modeled by descriptions that are interpreted at run-time. Thus, whenever a business change is needed, these descriptions are changed which are then immediately reflected in the running application.

Adaptive Object-Model architectures are made up of several smaller patterns. *TypeObject* [6] separates an Entity from an EntityType. Entities have Attributes, which are implemented with *Properties* [1], and the *TypeObject* pattern is used a second time to separate Attributes from AttributeTypes. The *Strategy* pattern [4] is often used to define the behavior of an EntityType. As is common in Entity-Relationship modeling, an Adaptive Object-Model usually separates attributes from relationships. Finally, there is usually an interface for non-programmers to define new EntityTypes. Applying the *TypeObject* pattern twice for both Entities and Properties as shown in Figure 1 (*TypeSquare*) is a common theme in many Adaptive Object-Models architectures.

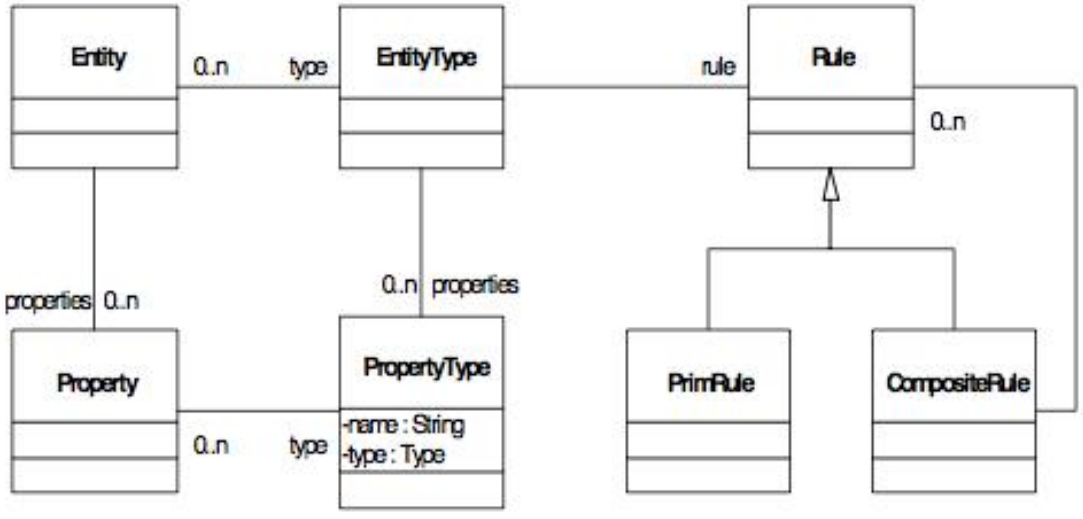


Figure 1 - Type Square