

Synthèse Architecture et Outils pour les applications SIVAL Dot Net

AUTEURS

FRANCEAGRIMER/SI/U_SIGA/Pôle aides nationales

IDENTIFICATION DU DOCUMENT

CI_SYNTHESE_ARCHITECTURE_OUTILS_100.doc – Version 1.00

ETAT DU DOCUMENT

En cours	Vérifié	Validé	Périmé
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

CONFIDENTIALITE DU DOCUMENT

Public	Interne	Sensible	Restreint
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>


OBJET DU DOCUMENT

Document décrivant les différentes architectures physiques et techniques mises en place dans les applications SivalNet. Ce document recense également la liste des outils, des composants et des documents d'aide au développement.

Précision importante : les documents référencés sous forme de lien seront mis à disposition lors de la prestation de prise de connaissance technique au lancement du marché.

Sommaire

1	Introduction	3
2	Les architectures	4
2.1	Architecture multi-niveaux basée sur les services web	4
2.2	Architecture multi-niveaux basée sur le Remoting.Net	6
2.3	Cas particuliers	8
2.3.1	WebAdmin	8
2.3.2	Serveur de fichier	9
2.4	Documentation sur les différents environnements	9
3	Gestion des sources	10
4	Charte graphique	11
5	Synthèse des normes de nommage	12
5.1	Définition des style de nommage	12
5.2	Convention de nommage	12
5.3	Abréviation	12
6	Sécurité et authentification de l'utilisateur	14
6.1	Cryptage SSL	14
6.2	Authentification	14
6.2.1	Cas général	14
6.2.2	Cas Particuliers	16
7	Centralisation des erreurs	17
8	Outils et composants	18
8.1	Composants web	18
8.2	Assistants pour Visual Studio	18

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

1 INTRODUCTION

Ce document a pour but de synthétiser les différentes architectures physiques ou techniques mises en place dans les applications SivalNet. Cette architecture a évolué avec les applications, ce document précise donc l'architecture employée pour chaque application.

De plus, un certain nombre d'outils (SecuWeb, WebAdmin, assistants pour Visual Studio, générateur d'entité métier, ETP de génération d'application...), de composants (composants graphiques pour la partie présentation, composants d'authentification, entité métier de base...) et de normes de développement ont été mis en place afin de faciliter et d'encadrer le travail des développeurs. Ces différents éléments seront décrits et un lien vers leur documentation sera fourni.

2 LES ARCHITECTURES

Les architectures mises en place dans les applications SivalNet sont des architectures multi-niveaux qui s'appuient sur 3 couches de référence :

- La couche de présentation : c'est le module qui reçoit la requête, authentifie l'utilisateur, valide le format des données, gère la navigation et met en forme la réponse.
- La couche métier : c'est le module qui reçoit les données en provenance de la couche présentation, valide leur cohérence sémantique, calcule la réponse ou met en forme les requêtes SQL et les procédures stockées destinées à la couche de données.
- La couche de données : elle est généralement implémentée par la base de données et exécute les requêtes SQL et les procédures stockées.

La différence entre les applications se situe au niveau de la couche métier. Certaines applications ont une couche métier exposée sous la forme de services web (ancienne architecture), d'autres exposent leurs objets métiers via la technologie Remoting.Net en TCP/IP.

2.1 Architecture multi-niveaux basée sur les services web

La Figure 1 illustre l'architecture multi-niveaux de référence :

- Le premier niveau, la couche de présentation, est hébergé par un serveur Windows 20XX. La couche de présentation s'appuie sur la technologie des WebForms de .NET et est hébergée derrière un serveur Internet IIS. La machine hôte peut être située physiquement dans la DMZ (zone démilitarisée) et effectuer elle-même l'authentification et le cryptage des données. Alternativement, elle peut être située en aval de la DMZ et recevoir les informations d'authentification en provenance d'un serveur dédié.
- Le deuxième niveau, la couche métier, est également hébergé par un serveur Windows 20XX, généralement distinct du précédent. La couche métier s'appuie sur les technologies des WebServices et ADO.NET de .NET et est hébergée derrière un deuxième serveur Internet IIS. Le serveur IIS assure deux fonctions : il expose les WebServices à la couche de présentation et autorise des niveaux de protection supplémentaires utiles pour la mise en œuvre de la stratégie de défense en profondeur. La couche métier est généralement scindée en 2 modules :
 - Un premier module, dit de « logique métier », reçoit les requêtes et met en forme les réponses en s'appuyant sur les protocoles standards des WebServices tel que SOAP. Ces protocoles permettent d'assurer un niveau raisonnable d'intégrité des messages en rejetant automatiquement les messages mal formés. Ce module traite également les requêtes ne nécessitant pas d'accès à la base de données
 - Un deuxième module, dit « d'accès aux données » met en forme les requêtes SQL et les appels aux procédures stockées destinés à la couche de données.

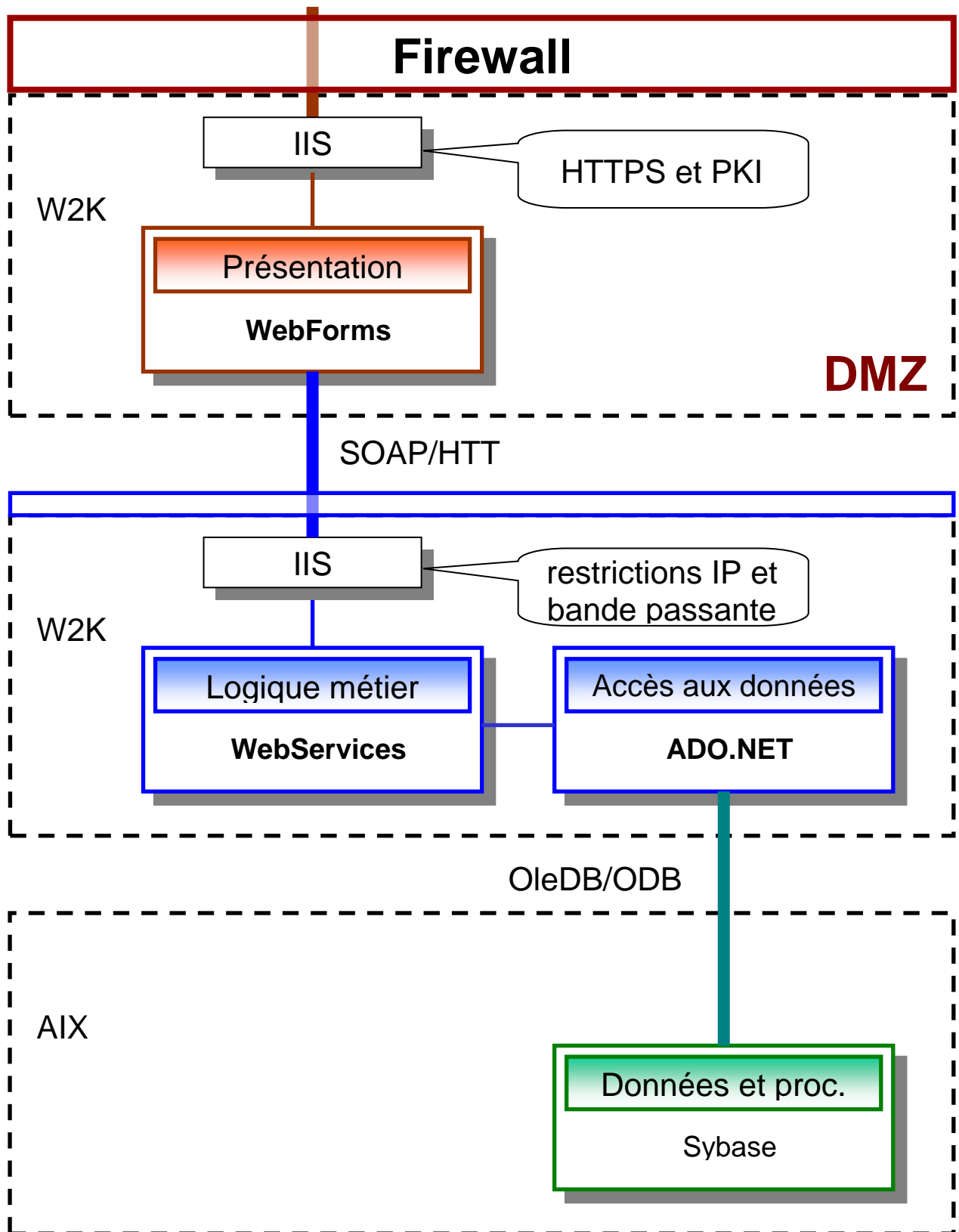



Figure 1 : Architecture multi-niveaux de référence

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

- Le troisième niveau, la couche de données, est implémenté par un SGBD Sybase hébergé par un serveur AIX. Ce niveau comporte les tables de données et les procédures stockées. La documentation pour installer le client Sybase se trouve ici : [Déploiement du pilote Sybase ADO.NET](#)

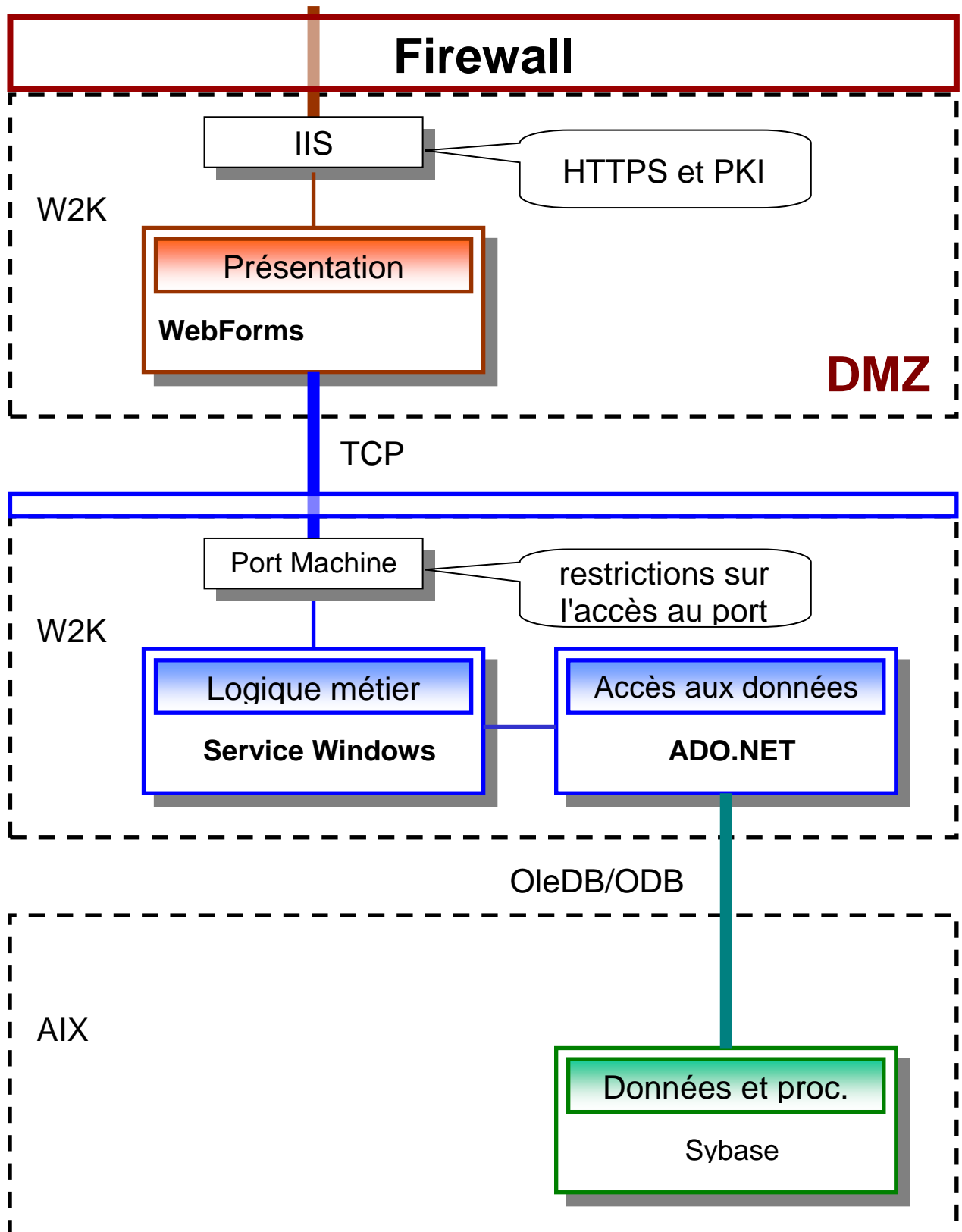
Cette architecture concerne les applications suivantes :


- Expadon (en phase de décommissionnement)
- Impadon (en phase de décommissionnement)
- Secuweb

Pour ces applications, la partie métier est fortement basée sur l'utilisation de Dataset.

2.2 Architecture multi-niveaux basée sur le Remoting.Net

Cette architecture est également basée sur trois niveaux. Les niveaux 1 et 3 sont semblables à l'architecture précédente, seul le deuxième niveau est modifié. Les services web sont remplacés par des services Windows qui exposent par Remoting.Net des objets métier. Le protocole utilisé pour le Remoting est le protocole TCP.



	Synthèse architecture des téléprocédures SIVAL	SI / U_SIGA
	Architecture et Outils pour les applications .Net	

Cette architecture concerne les applications suivantes :

- KPPA
- OASIS
- OASAR
- CERTIVETO
- INVOCM

La partie métier est basée sur :

- des classes d'objet métier,
- des datasets
- une couche d'accès aux données intermédiaire
- des manager/factory exposés par Remoting et contenant une méthode par cas d'utilisation

Une présentation de ce type de développement est disponible au format Powerpoint :



[Présentation développement .net à l'Office \(ancien modèle\)](#)

Pour les autres applications (les plus récentes), la partie métier est composée

- de classes d'entités métier. Les datasets ne sont plus utilisés.
- des manager/factory exposés par Remoting et contenant une méthode par cas d'utilisation

Une présentation de ce type de développement est disponible au format Powerpoint :



[Présentation développement .net orienté objet](#)

2.3 Cas particuliers

Deux applications se différencient des architectures à cause des fonctionnalités qu'elles doivent apporter : il s'agit de l'application WebAdmin qui recense les erreurs sur les applications et du serveur de fichier qui permet aux applications d'accéder depuis la couche de présentation à des fichiers se trouve sur le serveur métier.


2.3.1 WebAdmin

L'architecture de l'application est également en trois niveaux :

- une couche de présentation pour la partie administration. Cette partie n'est accessible qu'en Intranet.
- une couche métier qui offre des services web.
- Une couche de données dont la particularité est que l'ensemble des données nécessaires au fonctionnement de l'application sont stockées dans des fichiers XML afin de pallier à une éventuelle déficience du serveur de base de données. En effet, le but de cette application est de collecter les erreurs provenant des autres applications, elle doit avoir une disponibilité maximale même en cas de problème sur les serveurs de base de données.

Pour plus de détails sur l'application, les documents suivants sont disponibles :

Guide de déploiement :

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------



[WebAdmin déploiement](#)

Présentation fonctionnelle de l'application :



[WebAdmin Présentation](#)

Guide d'utilisation :



[WebAdmin Utilisation](#)

2.3.2 Serveur de fichier

Le serveur de fichier est constitué uniquement d'un service web qui se trouve sur le serveur dont on veut exposer les fichiers (en l'occurrence sur le serveur métier).

Pour plus de détails sur l'application, les documents suivants sont disponibles :



[Dossier installation Serveur Fichier](#)




[Dossier utilisation Serveur Fichier](#)

2.4 Documentation sur les différents environnements

Le document suivant synthétise l'ensemble des environnements et des architectures pour chaque application SivalNet :




[Suivi Applications](#)

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

3 GESTION DES SOURCES

Afin de gérer l'historique des sources des différentes applications SivalNet, les sources sont gérés sous GIT.

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

4 CHARTE GRAPHIQUE

La charte graphique des applications SivalNet est la charte définie par le ministère à la création du socle.

Un ensemble d'images et de feuilles de style css ont été créées et sont utilisées par l'ensemble des applications.

De plus, afin de faciliter la création de nouvelle page aspx répondant aux exigences de la charte, des assistants pour Visual Studio.Net ont été créés et intégrés dans Visual Studio. Des composants tels que l'entête des pages ou le menu ont également été développés pour faciliter la tâche des développeurs (voir la partie 8).

5 SYNTHÈSE DES NORMES DE NOMMAGE

5.1 Définition des style de nommage

- *Casse Pascal* : La première lettre de l'identificateur et la première lettre de chaque mot concaténé suivant sont en majuscules. Vous pouvez utiliser la casse Pascal pour les identificateurs comportant trois caractères ou plus. Par exemple : **BackColor**
- *Casse mixte* : La première lettre de l'identificateur est en minuscules et la première lettre de chaque mot concaténé suivant est en majuscules. Par exemple : **backColor**
- *Majuscules* : Toutes les lettres de l'identificateur sont en majuscules. Utilisez uniquement cette convention pour les identificateurs comportant deux lettres ou moins. Par exemple : `System.IO`, `System.Web.UI`


5.2 Convention de nommage

Identificateur	Style	Exemple
Classe	Pascal	AppDomain
Type énuméré	Pascal	ErrorLevel
Valeur énumérée	Pascal	FatalError
Événement	Pascal	ValueChange
Class d'exception	Pascal	WebException Note : Se termine toujours par le suffixe Exception .
Champ statique en lecture seule	Pascal	RedValue
Interface	Pascal	IDisposable Note : Commence toujours par le préfixe I .
Méthode	Pascal	ToString
Namespace	Pascal	System.Drawing
Paramètre	Mixte	typeName
Propriété	Pascal	BackColor
Champ d'instance protégé	Mixte *	_RedValue * préfixé par un underscore
Champ d'instance public	Pascal	RedValue Note : Rarement utilisé. L'utilisation d'une propriété est préférable à celle d'un champ d'instance protégé.

5.3 Abréviation

N'utilisez pas d'abréviations ou de contractions comme parties des noms d'identificateurs. Par exemple, utilisez `GetWindow` au lieu de `GetWin`.

N'utilisez pas d'acronyme qui ne soit communément accepté dans le domaine informatique.


	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

Lorsqu'il est approprié, utilisez un acronyme connu pour remplacer des termes longs. Par exemple, utilisez UI pour « User Interface » et OLAP pour « On-line Analytical Processing ».

Si vous utilisez des acronymes, choisissez les casses Pascal ou mixte pour les acronymes comportant plus de deux caractères. Par exemple, utilisez HtmlButton ou htmlButton.

En revanche, mettez en majuscules les acronymes comportant uniquement deux caractères, tels que System.IO au lieu de System.io.

N'utilisez pas d'abréviations dans les identificateurs ou les noms de paramètres. Si vous devez utiliser une abréviation, choisissez la **casse mixte** pour les abréviations comportant deux caractères ou plus, même si cela va à l'encontre de l'abréviation standard de ce mot.

	Synthèse architecture des téléprocédures SIVAL	SI / U_SIGA
	Architecture et Outils pour les applications .Net	

6 SECURITE ET AUTHENTIFICATION DE L'UTILISATEUR

6.1 Cryptage SSL

L'ensemble des applications extranet et des sites internet de l'Office sont accessibles via le protocole SSL (Secured Socket Layer) qui permet de crypter les informations échangées entre le navigateur client et la téléprocédure de l'Office.

6.2 Authentification

6.2.1 Cas général

Les couches de présentation des applications SivalNet sont protégées par une authentification par formulaire. Ainsi les utilisateurs se connectant à une partie protégée d'une application sont redirigés vers un formulaire d'identification (formulaire de login) dans lequel ils saisissent leur identifiant et leur mot de passe. Ensuite, si son identifiant et son mot de passe sont corrects, il pourra choisir son profil (s'il en possède plusieurs), changer son mot de passe (si c'est sa première connexion), renseigner ses informations personnelles (téléphone, e-mail, adresse...).

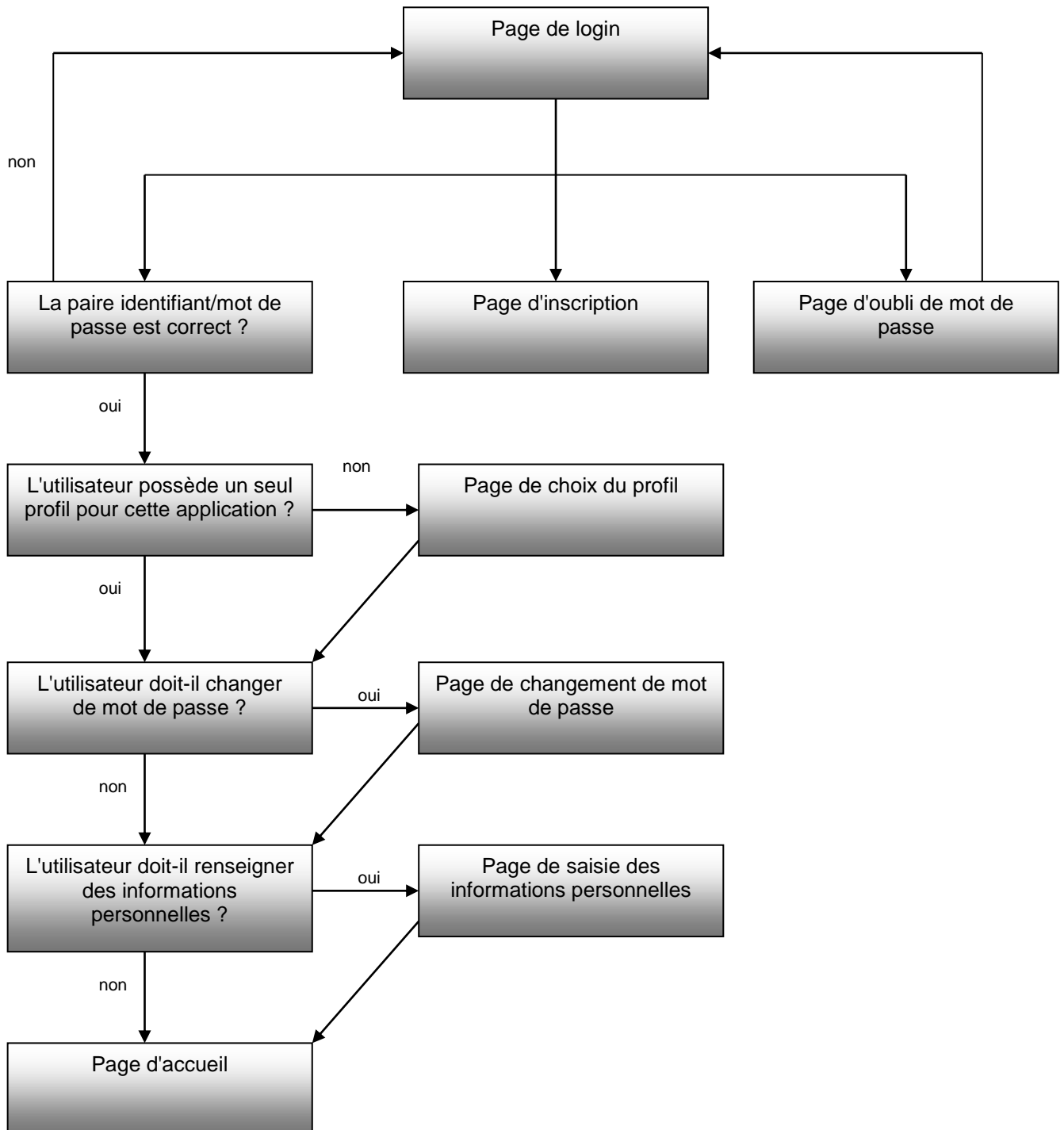
L'ensemble de ses étapes est géré par l'application **SecuWeb**. Des évolutions sont en cours pour migrer l'authentification vers la cassification (dans ce cas l'authentification se base sur l'AD de l'établissement ou l'open LDP du portail PAF pour les gestionnaires hors FranceAgriMer). Secuweb est composé d'un site web d'administration et d'un ensemble de services web permettant d'identifier une personne, de récupérer ses profils, de faire une demande d'inscription, de changer le mot de passe d'un utilisateur... La liste exhaustive des services proposés par Secuweb est décrite dans le document :




[Service commun Secuweb](#)

Pour éviter que le code correspondant à l'identification, au changement de profil et aux autres opérations concernant l'utilisateur soit dupliqué dans chaque application, des composants web ont été créés et réalisent chacun une opération particulière. Par exemple le composant BoiteLogin affiche un formulaire de saisie du login et du mot de passe, valide ce login et de ce mot de passe en faisant appel à Secuweb et renvoie des événements pour communiquer avec l'application hôte (voir la partie 8).

L'enchaînement des écrans relatifs à l'authentification de l'utilisateur est précisé par le diagramme suivant :



Enchaînement des écrans d'authentification de l'utilisateur dans une application SivalNet

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

La procédure de demande d'inscription et la procédure en cas d'oubli de mot de passe sont décrites dans le document :



[Analyse fonctionnelle Secuweb](#)

Point important :

Depuis plusieurs années un chantier de cassification a été initié. A ce stade l'application SECUWEB a été modifiée pour se connecter avec une authentification CAS et intégrer les évolutions nécessaires à la cassification de l'ensemble des applications du périmètre.

Pour les applications qui seront cassifiées alors l'authentification est portée via le CAS sur l'AD de l'entreprise et l'openLDAP du portail des gestionnaires FranceAgriMer. Les habilitations restent gérées par l'application SECUWEB.

6.2.2 Cas Particuliers

L'application WEBADMIN ne rentre pas dans le cadre général décrit précédemment.

6.2.2.1 Console d'administration de WebAdmin

Comme on l'a précisé précédemment, l'application WebAdmin doit être disponible même en cas de problème sur la base de données ou sur d'autres applications telles que SecuWeb. L'authentification sur la console d'administration de WebAdmin n'utilise donc pas Secuweb; elle est basée sur l'authentification Windows. Les utilisateurs et leurs rôles sont gérés dans la console d'administration (voir le paragraphe 2.2 du document).

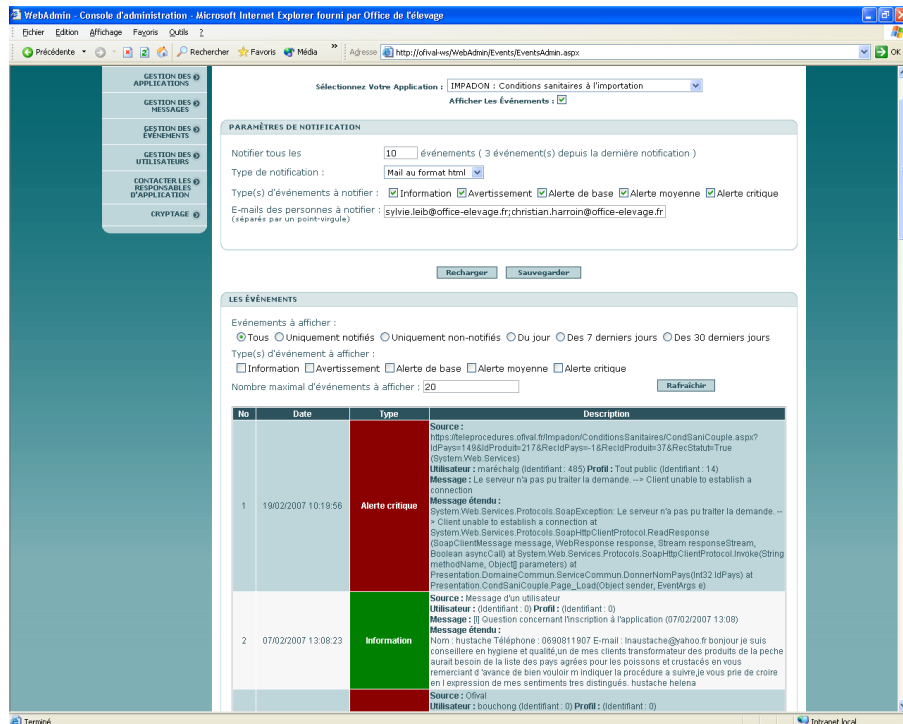
7 CENTRALISATION DES ERREURS

Afin de recenser les erreurs survenues sur les applications et afin d'avoir une trace de ces erreurs, un service web a été développé dans WebAdmin. Ce service web permet aux applications d'envoyer à WebAdmin un rapport d'événement. Un rapport d'événement regroupe les informations suivantes :

- son type (alerte, information, erreur critique...),
- l'application émettrice,
- la date et l'heure de l'événement,
- la source de l'événement (par exemple la page aspx)
- le login et l'identifiant de l'utilisateur courant lorsque l'événement s'est produit,
- le libellé et l'identifiant du profil de l'utilisateur courant lorsque l'événement s'est produit,
- un message décrivant l'événement (par exemple le contenu d'une exception).

Le code permettant l'envoi d'un rapport vers WebAdmin doit être placé dans l'événement `Application_Error` du fichier `global.asax` de chaque application. Ainsi, dès qu'une exception ne sera pas traitée par l'application un rapport d'événement sera envoyé vers WebAdmin. Pour plus d'informations, voir le paragraphe 3.2 de la documentation Webadmin.

La console d'administration de WebAdmin permet au responsable de chaque application de voir les erreurs survenues au sein de l'application dont il est le responsable. Il peut également être prévenu par mail en cas d'erreur grâce à un système d'alerte automatique et paramétrable.



The screenshot shows the WebAdmin console interface. On the left is a navigation menu with options like 'GESTION DES APPLICATIONS', 'GESTION DES MESSAGES', 'GESTION DES EVENEMENTS', 'GESTION DES UTILISATEURS', 'CONTACTER LES RESPONSABLES D'APPLICATION', and 'CRYPTAGE'. The main area is titled 'Sélectionnez Votre Application : IMPADON : Conditions sanitaires à l'importation'. Below this, there are 'PARAMETRES DE NOTIFICATION' and 'LES EVENEMENTS' sections.

PARAMETRES DE NOTIFICATION

- Notifier tous les: 10 événements (3 événement(s) depuis la dernière notification)
- Type de notification: Mail au format html
- Type(s) d'événements à notifier: ☒ Information ☒ Avertissement ☒ Alerte de base ☒ Alerte moyenne ☒ Alerte critique
- E-mails des personnes à notifier: sylvie.leib@office-elevage.fr;christian.harroin@office-elevage.fr


LES EVENEMENTS

Evénements à afficher: ☒ Tous ☐ Uniquement notifiés ☐ Uniquement non-notifiés ☐ Du jour ☐ Des 7 derniers jours ☐ Des 30 derniers jours

Type(s) d'événement à afficher: ☐ Information ☐ Avertissement ☐ Alerte de base ☐ Alerte moyenne ☐ Alerte critique

Nombre maximal d'événements à afficher: 20

No	Date	Type	Description
1	19/02/2007 10:19:56	Alerte critique	Source : https://teleprocdures.ofval.fr/impadon/ConditionsSanitaires/CondSanCouple.aspx?IdPays=14&IdProduit=217&RecIdPays=1&RecIdProduit=37&RecStatut=True (System.Web.Services) Utilisateur : maréchal (Identifiant : 495) Profil : Tout public (Identifiant : 14) Message : Le serveur n'a pas pu traiter la demande. -> Client unable to establish a connection Message étendu : System.Web.Services.Protocols.SoapException: Le serveur n'a pas pu traiter la demande. -> Client unable to establish a connection at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse (SoapClientMessage message, WebResponse response, Stream responseStream, Boolean asyncCall) at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodName, Object[] parameters) at Presentation.DomaineCommun.ServiceCommun.DonnerNomPays(Int32 IdPays) at Presentation.CondSanCouple.Page_Load(Object sender, EventArgs e)
2	07/02/2007 13:08:23	Information	Source : Message d'un utilisateur Utilisateur : (Identifiant : 0) Profil : (Identifiant : 0) Message : [1] Question concernant l'inscription à l'application (07/02/2007 13:08) Message étendu : Nom : hustache Téléphone : 0690811907 E-mail : hustache@yahoo.fr bonjour je suis conseiller en hygiène et qualité un de mes clients transformateur des produits de la pêche aurait besoin de la liste des pays agréés pour les poissons et crustacés en vous remerciant d'avance de bien vouloir m'indiquer la procédure à suivre, je vous prie de croire en l'expression de mes sentiments très distingués. hustache helena Source : Ofval Utilisateur : bouchong (Identifiant : 0) Profil : (Identifiant : 0)

	Synthèse architecture des téléprocédures SIVAL Architecture et Outils pour les applications .Net	SI / U_SIGA
---	---	--------------------

8 OUTILS ET COMPOSANTS

Trois types d'outils ont été développés pour faciliter le développement des applications : des composants web, des assistants pour Visual Studio, un ETP pour générer une application d'aide de crise rapidement.

8.1 Composants web

Ces éléments sont principalement de deux types :

- Les éléments communs aux applications qui ne doivent être modifiés que par la cellule technique (composants ascx, images, feuilles de style respectant la charte, classes vb...)
- Les éléments modèles destinés à présenter une manière d'utiliser les éléments communs (pages aspx, composants ascx...)

Ces différents composants, la manière de les utiliser et de les partager sont détaillés dans le document :



[Composants et modèles SivalNet](#)

L'ensemble des modèles sont archivés dans la base Sourcesafe.Net (N : \SOURCES.PBL\SourceSafe.Net.) sous le projet SivalNet_v2/SivalNet_v2.root.

8.2 Assistants pour Visual Studio

Les assistants pour Visual Studio.Net permettent de générer rapidement des entités métier à partir d'une table de base de données. Ils permettent également de générer des pages aspx pour la partie Présentation (page d'accueil, page de login...). Ce document décrit également les différents assistants :



[Wizards SivalNet](#)

Les assistants doivent être installés sur les postes développeur avec le fichier d'installation (.msi) se trouvant dans le répertoire :

[N:\Pwb_Appl\SivalNet\Sources\Wizards VS.Net](#)

