

## Pourquoi avoir créé SOFI ?

Les trois (3) mots clés derrière la création de SOFI sont « simplification », « efficacité » et « synergie ».

**SIMPLIFICATION** parce que SOFI intègre une multitude de cadres complémentaires issus du logiciel libre et permet de simplifier considérablement l'utilisation de ceux-ci, dorénavant orchestrés et imbriqués dans SOFI.

**EFFICACITÉ** parce que les ressources qui utilisent SOFI dans le cadre de leurs projets de développement Web sont beaucoup **plus productifs**, sans compter que SOFI permet une **diminution** considérable du **temps d'appropriation des nouvelles technologies Java** et des façons de faire particulières à un développement Web. SOFI a donc vu le jour en réponse au besoin du marché pour lequel il fallait absolument **accroître l'efficacité des développeurs** et favoriser la **réutilisation** pour livrer des applications de qualité, dans le respect des temps et des budgets.

**SYNERGIE** parce que l'évolution de SOFI est basée sur les concepts du logiciel libre. Son utilisation amène donc les développeurs Java à participer à **une communauté d'intérêt** animée par les créateurs de SOFI. Plusieurs organismes québécois sont aujourd'hui **propriétaires des codes sources** et des droits d'utilisation de SOFI et cette réalité permet le **partage de l'expertise** de développement et l'échange d'une quantité importante de composants communs entre différents projets interministériels.

Nurun entretien aussi un extranet SOFI pour la communauté Java de la région de Québec et une nouvelle version de ce portail sera rendu disponible dans les prochaines semaines.

## Principales fonctionnalités de SOFI-Framework

### Description de SOFI

SOFI est donc un cadre d'applications Java qui **s'appuie sur les patrons de conception les plus reconnus du marché** ainsi que sur **l'architecture orientée services (AOS)**, afin de simplifier le développement d'applications Web.

SOFI est construit sur les technologies de base du Java et sur les cadres d'applications reconnus et offre ainsi une infrastructure de développement robuste.

Afin de faciliter le développement, le cadre d'applications SOFI offre en premier lieu une infrastructure qui fournit les services de base des applications Web :

- 🔗 messages et libellés multilingues;
- 🔗 paramètres de systèmes;
- 🔗 journal des opérations;
- 🔗 aide en ligne;
- 🔗 référentiel de navigation;
- 🔗 sécurité fonctionnelle.



**FIGURE 1 : LES SERVICES D'INFRASTRUCTURE DE SOFI**

Ces composants sont indépendants et peuvent être assemblés selon les besoins du système. Ils sont, de plus, dissociés de l'implémentation physique des données et peuvent donc être alimentés directement à partir des systèmes de mission de l'organisme.

SOFI apporte ensuite une quantité imposante de composants couvrant des besoins très variés touchant tout autant la logique d'affaires, le contrôleur et la couche de présentation de l'application. L'assemblage de ces composants, tous appuyés par l'infrastructure SOFI, est la clé de l'efficacité accrue acquise grâce au cadre d'applications.

La figure présentée en page suivante dresse la liste des différents composants de SOFI, développés en date du 20 novembre 2005.

<p><b>Composants de base</b></p> <ul style="list-style-type: none"> <li>• Messages</li> <li>• Libellés</li> <li>• Menus</li> <li>• Onglets</li> <li>• Domaines de valeurs</li> <li>• Journalisation</li> <li>• Paramètres système</li> <li>• Sécurité</li> <li>• Aide en ligne</li> </ul>	<p><b>Simplification de la présentation</b></p> <ul style="list-style-type: none"> <li>• Gestion des formulaires sur plusieurs pages (étape par étape)</li> <li>• Validations unitaires et de cohérence automatisées</li> <li>• Gestion des documents électroniques</li> <li>• Affichage de données formatées dans les pages</li> <li>• Développement des contrôleurs</li> <li>• Rapports simples en format PDF</li> <li>• Aide contextuelle</li> </ul>
<p><b>Simplification de la logique d'affaires</b></p> <ul style="list-style-type: none"> <li>• Filtres de recherche automatisés</li> <li>• Simplification du développement de la logique d'affaires</li> <li>• Sélections simplifiées d'enregistrements et de listes</li> <li>• Insertions et modifications simplifiées</li> <li>• Gestion de sélections multiples par listes de navigation</li> <li>• Gestion des services communs</li> </ul>	<p><b>Sécurité</b></p> <ul style="list-style-type: none"> <li>• Gestion complète de la sécurité applicative</li> <li>• Accès au référentiel des composants sécurisables</li> <li>• Gestion des utilisateurs, rôles et services</li> <li>• Chiffrement</li> <li>• Accès au répertoire LDAP</li> </ul>
<p><b>Balises personnalisées</b></p> <ul style="list-style-type: none"> <li>• Fonctionnalités Ajax</li> <li>• Formulaires multilignes</li> <li>• Liste de résultats navigable page par page avec tri automatisé</li> <li>• Aide en ligne</li> <li>• Formatage dynamique de champs</li> <li>• Calendrier pour la sélection de date</li> <li>• Listes de valeurs</li> <li>• Balise de gestion de contenu HTML</li> </ul>	<p><b>Utilitaires</b></p> <ul style="list-style-type: none"> <li>• Gestion des dates</li> <li>• Gestion des erreurs</li> <li>• Gestion des chaînes de caractères</li> <li>• Gestion de la génération XML</li> <li>• Gestion du chargement XML</li> <li>• Gestion des documents électroniques</li> <li>• Gestion des listes et tableaux</li> <li>• Gestion des fichiers</li> </ul>

**FIGURE 2 : LES DIFFÉRENTS COMPOSANTS DE SOFI**

### **Apport de SOFI**

Le développement avec SOFI apporte une solution aux problèmes rencontrés dans le marché du développement Java, soit :

- ① en rapprochant le développement Web avec Java d'une approche 4GL;
- ① en faisant abstraction de la complexité des autres cadres d'applications utilisés (Struts, Hibernate, etc.);

- ④ en diminuant grandement la présence de codes redondants;
- ④ en rendant communes les façons de faire entre développeurs;
- ④ et finalement, en offrant une boîte à outils évolutive.

SOFI est maintenant utilisé non seulement par Nurun, mais également par plusieurs organismes partenaires dans les développements de plus d'une vingtaine d'applications transactionnelles avec Java dont la majorité est déjà en production sur le Web.

## Économisez et soyez plus efficace avec SOFI

### SOFI apporte des économies réelles

CRITÈRES	DÉMONSTRATION DES RÉSULTATS																																																																																																									
<p>Économie de temps et d'argent et amélioration de l'efficacité</p>	<p>Pour les projets de développement Web, l'utilisation du canevas d'applications SOFI permet la réutilisation de composants. Cette réalité permet d'économiser un minimum de 20% du temps de développement. Pour des projets où la date de livraison du système est très rapprochée, SOFI représente un avantage certain, à valeur inestimable.</p> <p>Afin de vous démontrer l'économie réelle amenée par l'utilisation de SOFI, voici un exemple concret et bien réel, basé sur un projet en cours de développement par Nurun.</p> <p><b>EXEMPLE RÉEL D'ÉCONOMIE DE TEMPS ET D'ARGENT</b></p> <p>Voici tout d'abord les barèmes utilisés aujourd'hui par Nurun pour un projet de développement Java. Le tableau de la page suivante vous présente les efforts estimés AVEC et SANS l'utilisation de SOFI.</p> <table border="1" data-bbox="368 853 1469 1290"> <thead> <tr> <th rowspan="2"></th> <th colspan="4">Traitement Java <b>SANS</b> SOFI</th> <th colspan="4">Traitement Java <b>AVEC</b> SOFI</th> </tr> <tr> <th>Analyse</th> <th>Réalisation</th> <th>Essais</th> <th>Total</th> <th>Analyse</th> <th>Réalisation</th> <th>Essais</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td></td> <td colspan="4">Barèmes d'efforts en JOUR</td> <td colspan="4">Barèmes d'efforts en JOUR</td> </tr> <tr> <td>Très simple</td> <td>3</td> <td>5</td> <td>1</td> <td>13</td> <td>3</td> <td>2</td> <td>1</td> <td>6</td> </tr> <tr> <td>Simple</td> <td>5</td> <td>8</td> <td>2</td> <td>20,5</td> <td>5</td> <td>3</td> <td>2</td> <td>10</td> </tr> <tr> <td>Simple - moyen</td> <td>10</td> <td>16</td> <td>3</td> <td>32</td> <td>10</td> <td>8</td> <td>3</td> <td>21</td> </tr> <tr> <td>Moyen</td> <td>12,5</td> <td>20</td> <td>6,5</td> <td>40</td> <td>12,5</td> <td>12</td> <td>6,5</td> <td>31</td> </tr> <tr> <td>Moyen - complexe</td> <td>14,5</td> <td>30</td> <td>10</td> <td>54,5</td> <td>14,5</td> <td>18,5</td> <td>10</td> <td>43</td> </tr> <tr> <td>Complexe</td> <td>17,5</td> <td>40</td> <td>13,5</td> <td>71</td> <td>17,5</td> <td>25</td> <td>13,5</td> <td>56</td> </tr> <tr> <td>Très - complexe</td> <td>35</td> <td>60</td> <td>15</td> <td>110</td> <td>21</td> <td>35</td> <td>15</td> <td>71</td> </tr> </tbody> </table> <p>La comparaison des chiffres présents dans les deux (2) colonnes <b>Total</b> vous permet de constater que SOFI amène des économies concrètes de temps lors des phases d'analyse fonctionnelle, de réalisation et d'essais.</p> <p>Ces économies, par unité de traitement, ont des répercussions importantes lors d'un développement. Dans le cadre de notre projet, les unités de traitement demandées par le client sont les suivantes :</p> <table border="1" data-bbox="421 1547 1422 1827"> <thead> <tr> <th></th> <th>Nombre</th> <th>Complexité</th> </tr> </thead> <tbody> <tr> <td rowspan="6">Unités de traitement</td> <td>23</td> <td>Très simple</td> </tr> <tr> <td>42</td> <td>Simple</td> </tr> <tr> <td>38</td> <td>Simple à moyen</td> </tr> <tr> <td>18</td> <td>Moyen</td> </tr> <tr> <td>10</td> <td>Moyen à complexe</td> </tr> <tr> <td>2</td> <td>Complexe</td> </tr> </tbody> </table> <p>Si nous appliquons les micro-barèmes, c'est-à-dire le nombre de jours pour réaliser le système pour les activités d'analyse fonctionnelle, de réalisation et d'essais fonctionnels, nous obtenons les totaux suivants :</p> <ul style="list-style-type: none"> <li>■ Total des efforts en micro-barèmes SANS SOFI : 3 783 jours/personnes</li> <li>■ Total des efforts en micro-barèmes AVEC SOFI : 2 456 jours/personnes</li> </ul>		Traitement Java <b>SANS</b> SOFI				Traitement Java <b>AVEC</b> SOFI				Analyse	Réalisation	Essais	Total	Analyse	Réalisation	Essais	Total		Barèmes d'efforts en JOUR				Barèmes d'efforts en JOUR				Très simple	3	5	1	13	3	2	1	6	Simple	5	8	2	20,5	5	3	2	10	Simple - moyen	10	16	3	32	10	8	3	21	Moyen	12,5	20	6,5	40	12,5	12	6,5	31	Moyen - complexe	14,5	30	10	54,5	14,5	18,5	10	43	Complexe	17,5	40	13,5	71	17,5	25	13,5	56	Très - complexe	35	60	15	110	21	35	15	71		Nombre	Complexité	Unités de traitement	23	Très simple	42	Simple	38	Simple à moyen	18	Moyen	10	Moyen à complexe	2	Complexe
	Traitement Java <b>SANS</b> SOFI				Traitement Java <b>AVEC</b> SOFI																																																																																																					
	Analyse	Réalisation	Essais	Total	Analyse	Réalisation	Essais	Total																																																																																																		
	Barèmes d'efforts en JOUR				Barèmes d'efforts en JOUR																																																																																																					
Très simple	3	5	1	13	3	2	1	6																																																																																																		
Simple	5	8	2	20,5	5	3	2	10																																																																																																		
Simple - moyen	10	16	3	32	10	8	3	21																																																																																																		
Moyen	12,5	20	6,5	40	12,5	12	6,5	31																																																																																																		
Moyen - complexe	14,5	30	10	54,5	14,5	18,5	10	43																																																																																																		
Complexe	17,5	40	13,5	71	17,5	25	13,5	56																																																																																																		
Très - complexe	35	60	15	110	21	35	15	71																																																																																																		
	Nombre	Complexité																																																																																																								
Unités de traitement	23	Très simple																																																																																																								
	42	Simple																																																																																																								
	38	Simple à moyen																																																																																																								
	18	Moyen																																																																																																								
	10	Moyen à complexe																																																																																																								
	2	Complexe																																																																																																								

CRITÈRES	DÉMONSTRATION DES RÉSULTATS
	<p>Lors des projets de développement, nous devons ajouter un pourcentage moyen de 75 % de « macro-barèmes », c'est-à-dire le nombre de jours nécessaires à la gestion du projet et à l'architecture organique et de données. En l'ajoutant aux micro-barèmes, on obtient le total suivant pour l'ensemble du projet :</p> <ul style="list-style-type: none"> <li>■ Total du projet SANS SOFI : 6 620 jours/personnes</li> <li>■ Total du projet AVEC SOFI : 4 298 jours/personnes</li> </ul> <p>L'économie de temps est donc facilement visible avec une diminution de 2 322 jours/personnes, soit 35 % du temps de réalisation du projet.</p> <p>Il s'agit ainsi, sur un projet d'une envergure moyenne comme celui-ci, avec un taux moyen de 500\$ par jour, d'une <b>économie réelle de 1,16 millions de dollars pour le client.</b></p>
Amélioration de l'efficacité organisationnelle	<p>Pour une équipe de développement de systèmes informatiques, le virage technologique qu'entraîne le développement de systèmes Web est considérable, car il implique le développement de nouvelles compétences.</p> <p>En effet, les nouvelles approches de développement sont très différentes des approches traditionnelles, ce qui entraîne souvent une augmentation temporaire du délai de livraison des projets.</p> <p>Heureusement, SOFI permet une diminution considérable du temps d'appropriation des nouvelles technologies Java et des façons de faire particulières à un développement Web. L'arrivée de SOFI permet aussi de diminuer le degré de résistance au changement, car les équipes de développement ont tôt fait de constater que la réutilisation leur évite de tout réinventer et leur permet de livrer des applications de qualité, dans les temps et dans le respect des budgets.</p> <p>L'amélioration de l'efficacité est à la base de la création de SOFI.</p> <p>Un fait important à noter, la standardisation des façons de faire qu'entraîne l'utilisation de SOFI, donne l'assurance aux gestionnaires de projets que tous les développeurs peuvent facilement supporter les modifications et les corrections de l'application, <b>peu importe qui</b> a écrit le code initial. Cette uniformité dans le développement a pour conséquence d'amener une plus grande efficacité en projet.</p>
Interface utilisateur	<p><b>SOFI permet le développement d'une interface utilisateur riche</b></p> <p>L'utilisation de plusieurs fonctionnalités, dont l'aide contextuelle de SOFI ainsi que la mise en place des composants d'interfaces riches de SOFI basés sur la norme AJAX, ont permis la mise en ligne d'une application conviviale et simple d'utilisation. SOFI offre aussi des facilités de navigation simplifiée, la gestion intelligente de l'utilisation du bouton « précédent », de l'aide en ligne interactive et plusieurs autres fonctionnalités. L'expérience de l'utilisateur a donc été prise en compte lors du développement de SOFI.</p>

CRITÈRES	DÉMONSTRATION DES RÉSULTATS
<p>Amélioration des aspects liés à la sécurité de l'information</p>	<p>L'utilisation des composants de sécurité fonctionnelle de SOFI permet d'effectuer une sécurité applicative efficace afin de bien protéger l'information centrale à l'application. Les composants de sécurité fonctionnelle permettent d'effectuer de façon très simple les opérations suivantes :</p> <ul style="list-style-type: none"> <li>■ contrôler l'accès aux différents modules de l'application selon les rôles des utilisateurs et ce, non seulement par un menu dynamique, mais aussi par une sécurité directe au niveau des adresses de chaque module;</li> <li>■ contrôler l'accès aux champs contenant de l'information nominative selon les rôles des utilisateurs, soit en rendant un champ protégé contre la modification ou encore en le cachant complètement;</li> <li>■ contrôler l'accès aux différentes opérations d'un écran (consultation, modification, suppression, ajout) selon les rôles des utilisateurs.</li> </ul> <p>Bien que ces fonctionnalités soient régulièrement utilisées dans le marché, ce qui les rend unique à SOFI dans le monde des applications Web en Java, c'est <b>qu'elles sont effectuées de façon indépendante au code de l'application</b>. Ce n'est donc pas le développeur qui écrit directement des lignes de code pour effectuer cette sécurité, mais bien le cadre SOFI qui gère toutes ces fonctionnalités. L'analyste fonctionnel, ou une autre ressource, peut ainsi inscrire sa sécurité applicative par l'entremise d'une interface Web conviviale offerte par SOFI et ce, sans impacter le développement.</p> <p>Cette caractéristique, unique à SOFI, permet ainsi d'ajouter facilement de la sécurité sur l'information autant pendant la phase de développement qu'après celle-ci. Il s'agit de la façon idéale de créer une application dans le respect des principes de la Loi sur la protection des renseignements personnels.</p>

## SOFI pour les plus techniques

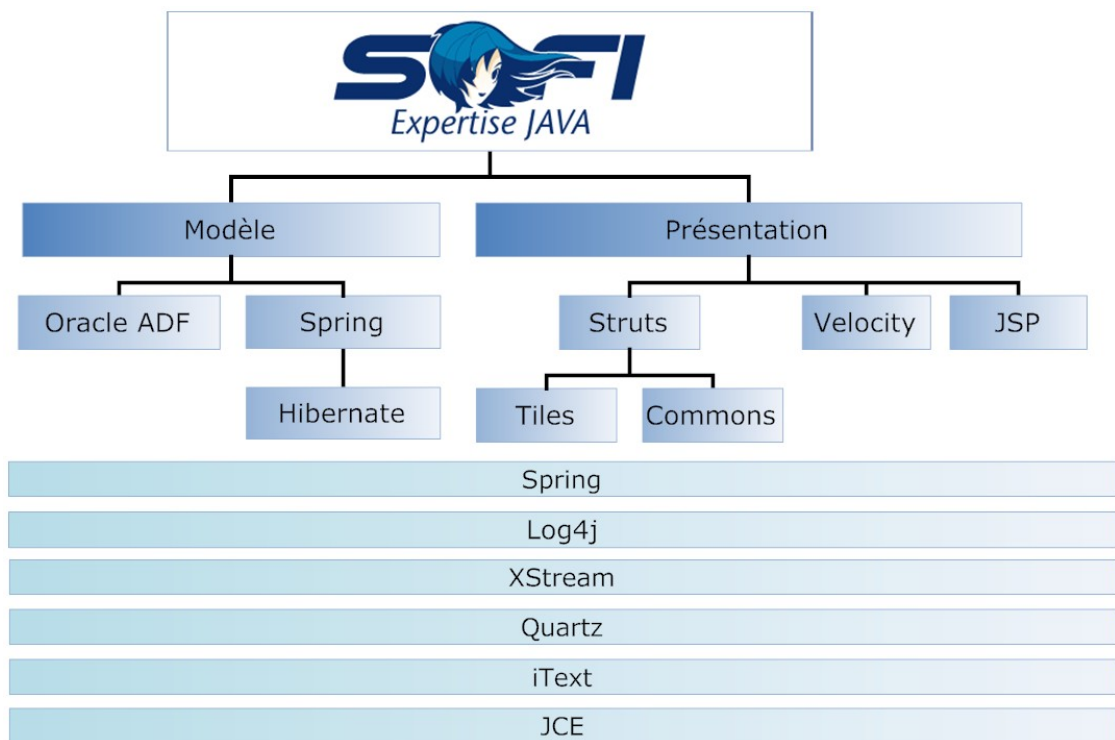
### Volet technique de la réalisation

SOFI est un produit d'intégration, c'est-à-dire un cadre d'applications simplifiant l'utilisation de plusieurs autres cadres. SOFI offre ainsi une couche de simplification sur plusieurs logiciels libres dont les principaux sont les suivants :

- ① **Jakarta Struts**, le standard international dans la gestion du patron Modèle – Vue – Contrôleur en Java;
- ① **Tiles**, une technologie de découpage des pages Web en sections réutilisables;
- ① **Velocity**, un générateur de contenu HTML à partir d'objets Java.

SOFI offre également une grande simplification du développement de la logique d'affaires grâce à une abstraction des cadres d'applications spécialisés dans l'accès à la base de données. Les cadres supportés par SOFI sont Oracle ADF Business Component ou une combinaison de Spring et Hibernate pour une solution basée 100% logiciels libres, souvent préconisée lors des développements sous la plate-forme IBM.

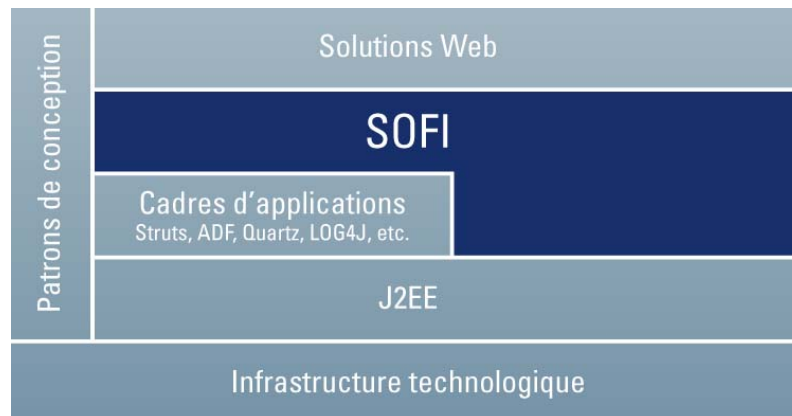
La figure suivante représente cette intégration offerte par SOFI.



Un développement de système Web avec SOFI se base tout d'abord sur une **infrastructure technologique**. Des applications développées avec SOFI existent présentement en production sur des serveurs Windows, Linux et AIX. Cette possibilité assure, aux clients de SOFI, une indépendance du fournisseur technologique.

Le développement est également basé sur les technologies Java 2 de base, c'est-à-dire un serveur d'applications tel que Tomcat, Oracle Application Server ou IBM Websphere. Encore

une fois, les applications SOFI sont indépendantes de la couche serveur utilisée, pouvant ainsi être déployées sur plusieurs plates-formes différentes.



Les applications SOFI s'appuient ensuite sur les cadres d'applications décrits précédemment. Il s'agit de logiciels libres considérés comme les meilleurs du marché.

La couche SOFI apparaît ensuite, présentant tel que mentionné, l'abstraction des cadres d'applications, mais aussi une quantité importante de composants uniques et exclusifs, basés directement sur l'infrastructure Java. Les solutions Web utilisent la couche SOFI afin d'offrir leurs services.

Notons finalement la présence, dans toute l'architecture, des patrons de conception, des bonnes pratiques du marché, qui demeurent toujours au cœur des préoccupations des développeurs d'applications qui utilisent SOFI.

### **Les avantages de SOFI pour la console de gestion des interventions environnementales**

Parmi les multiples avantages de baser son développement sur une infrastructure complète comme celle de SOFI, en voici quelques-uns qui sont rattachés au projet Console de gestion des interventions environnementales.

#### **Interfaces riches AJAX et Javascript**

SOFI comprend une librairie importante constituée de plusieurs composants servant à la création d'interfaces riches basées sur la technologie AJAX et sur le Javascript. Ces composants serviront entre autres à la rencontre des éléments suivants du projet :

- ⌚ rafraichissement automatisé des données selon une période de temps;
- ⌚ recherche avec complétion automatique à la « Google Suggest »;
- ⌚ affichage d'historique sur un lien dans une aide interactive (« tooltip »);
- ⌚ affichage de sous-formulaires sans affecter l'affichage du premier élément;
- ⌚ gestion des compressions ou développement de l'arbre SGI;
- ⌚ déplacement (« drag-and-drop ») des ressources dans l'arbre SGI;

- 🔒 saisie de date avec un calendrier de façon conviviale;
- 🔒 saisie de période de temps de façon conviviale.

### **Sécurité applicative**

L'infrastructure de SOFI comprend une librairie importante de fonctionnalités axées sur la sécurité, tant technologique qu'applicative. Cette librairie servira à supporter de façon efficace les éléments suivants :

- 🔒 accès protégé à l'application par mot de passe;
- 🔒 sécurisation de l'accès à chacune des pages selon les rôles;
- 🔒 modification et envoi de mot de passe;
- 🔒 sécurité d'accès aux champs et aux fonctionnalités.

### **Aide en ligne**

Le système d'aide en ligne livré avec SOFI est complet et peut-être alimenté de façon complètement indépendante du code, de façon à déléguer cette responsabilité aux pilotes de système. Il sera donc possible, suite au développement, d'ajouter de l'aide en ligne contextuelle sur les éléments suivants, selon le niveau de détail désiré :

- 🔒 sur la console de gestion et sur les formulaires de façon générale;
- 🔒 sur un des onglets des écrans;
- 🔒 sur un des champs dans chacune des pages.

L'aide contextuelle peut être alimentée via un contrôle de texte riche directement dans l'infrastructure de SOFI, être développée dans un système de gestion de contenu ou encore être stockée de façon simple au format html sur un serveur Web. Cette gestion est efficace et permettra de facilement améliorer l'aide en ligne applicative dans le temps.

### **Gestion multilingue des applications**

Toutes les applications développées avec SOFI sont compatibles avec l'affichage multilingue des libellés, messages et autres éléments. La gestion de ces libellés et messages est décentralisée et peut donc être effectuée par un pilote technologique ou applicatif via une console Web livrée avec le canevas. De cette façon, la Garde côtière pourra traduire de façon très simple et efficace le système.

### **Gestion des fichiers électroniques**

Le canevas SOFI contient une librairie importante qui gère l'envoi, la réception et l'affichage de documents électroniques à partir d'applications Web. Cet ensemble de composantes pourra donc être utilisé pour développer les éléments suivants :

- 🔒 consultation des fichiers déposés;
- 🔒 dépôt de fichiers.

### **Module de persistance et d'accès aux données**

L'utilisation de SOFI et de sa couche de simplification logicielle pour le canevas d'application Hibernate permettra une architecture robuste et efficace pour l'accès à la base de données,

tant en mode lecture que mise à jour. Cette architecture permettra une utilisation efficace de la base de données MySQL (tables InnoDB) compte tenu qu'elle est pleinement compatible avec Hibernate.