

## BDCS : un outil innovant à l'appui de la création et sélection

F. Boyer

eRcane BP 315 – 97494 Sainte Clotilde Cedex – La Réunion  
[boyer@ercane.re](mailto:boyer@ercane.re)

### Résumé

De nos jours, l'accès à l'information est une des principales préoccupations des entreprises. "La solution consiste souvent à mettre en place un système d'information permettant de centraliser les données afin de les rendre accessibles à tous" (Zermatten, 2008). Confrontées à ces problèmes d'accès aux données mais également à des difficultés de synthèse et de redondance importante, les équipes de la création variétale à eRcane ont été conduites à faire ce choix. Le système d'information BDCS (Base de Données Canne à Sucre) a permis de restructurer la gestion de ces données composée alors de différents fichiers Excel. Grâce notamment à l'émergence de nouvelles technologies et l'usage de bonnes pratiques de programmation, elles disposent aujourd'hui d'un outil d'aide à la décision performant et efficace.

*Mots clés* : Canne à sucre, création variétale, sélection variétale, SGBDR (Système de Gestion de Base de Données Relationnel), application web, SI (Système d'information), PDA (Personal Digital Assistant), base de données.

### Introduction

Le cœur de métier d'eRcane est la création de variétés de canne à sucre apportant un gain de productivité par rapport aux variétés leaders cultivées. Le processus de sélection permettant d'identifier, et de libérer, ces variétés plus performantes que les précédentes, dure environ 15 ans. Il comprend une phase d'hybridation d'un an et cinq stades de sélection d'une totalité de 14 ans (Figure 1).

#### Schéma de sélection

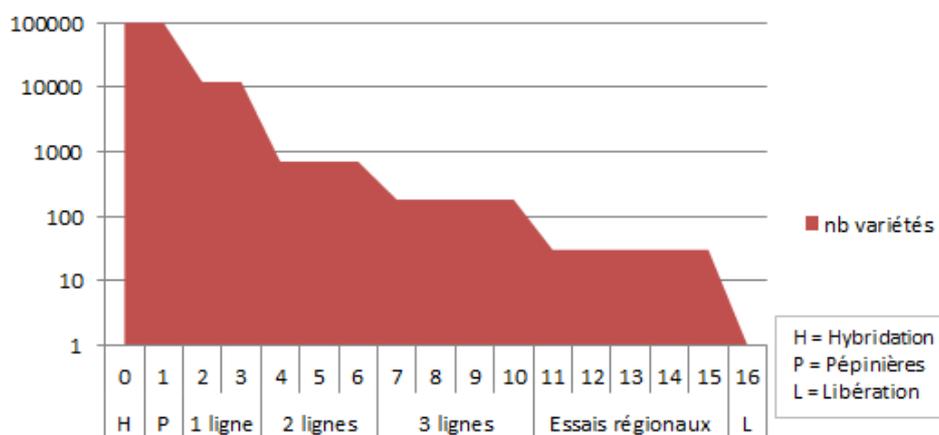


Figure 1. Les étapes de la création et sélection variétale à eRcane

Pour chacun des stades (H=Hybridation, P=Pépinières, 1 ligne, 2 lignes, 3 lignes, Essais Régionaux), les équipes d'hybridation et de sélection utilisaient un, voire plusieurs fichiers

Excel pour gérer leurs données, soit, un fichier par année de création multiplié par le nombre de stations de sélection (cinq initialement).

A partir du stade "2 lignes", à ces fichiers Excel étaient ajoutés des fichiers basiques d'EREG. EREG est une application "maison" créée en langage basic pour calculer les indices économiques de différentes variétés de canne à sucre. Ces fichiers étant indépendants les uns des autres, il était difficile de suivre les informations au fil des stades de sélection.

C'est donc pour pallier aux problèmes inhérents à un tel système (redondance importante, synthèse des données longue et fastidieuse pouvant engendrer des risques d'erreurs) qu'il fut décidé de mettre en place un système d'information informatisé, composé d'une véritable base de données relationnelle et d'une interface utilisateur pour le traitement des informations.

La mise en place d'un nouveau système d'information avait deux objectifs principaux :

- assurer les mêmes fonctionnalités qu'offrait l'ancien système ;
- harmoniser les différentes ressources, centraliser les données pour en faciliter l'accès et la synthèse, libérant du temps aux équipes de la création variétale pour se consacrer à leurs activités principales.

Le présent article présente les solutions techniques choisies pour concevoir et implémenter cet outil ainsi que les perspectives qu'il laisse envisager dans un futur proche.

## **Matériel et Méthode**

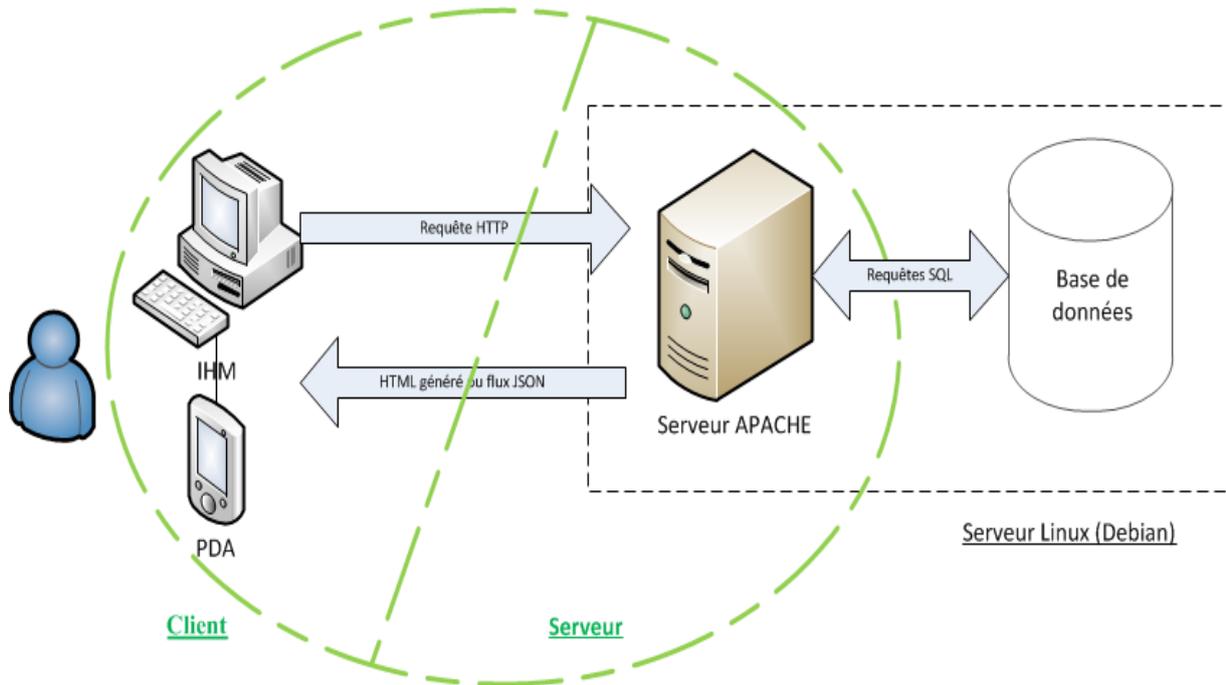
Comme l'avait préconisé Stéphane SAUREL (Service Informatique CERF) dans son rapport d'étude préalable sur ce qui fut les prémisses du projet BDCS (Base de Données Canne à Sucre) en 1996, c'est bien la solution basée sur le modèle CLIENT/SERVEUR qui fut choisie et mise en place par Eric ZERMATTEN (Ingénieur Agronome ENITAB spécialisé en Informatique) en 2007.

Le principal intérêt de l'architecture CLIENT/SERVEUR est de pouvoir centraliser les données en un seul endroit (le serveur) pour y accéder de n'importe quel PC se trouvant sur le même réseau (le client) :

- la base de données est hébergée sur un serveur puissant ;
- les utilisateurs interrogent et font travailler le système depuis un simple PC

On bénéficie alors à la fois de la puissance et la sécurité des serveurs et des interfaces riches des PC.

Cette solution considérée comme la meilleure (redondance faible, sécurité accrue, données centralisées), mais aussi la plus coûteuse à l'époque, ne fut envisageable qu'une dizaine d'années plus tard (Logiciels OPEN source, forte baisse des prix dans l'informatique, client léger : une application informatique dite légère car s'exécutant dans un navigateur web, etc.).



**Figure 2. Organisation du système d'information actuel**

L'utilisateur va par l'intermédiaire de son navigateur web (ou depuis un PDA) envoyer des requêtes http au serveur web APACHE, qui grâce au couplage avec PHP (acronyme récursif pour "PHP: Hypertext Preprocessor") pourra lui retourner du contenu généré dynamiquement (Figure 2).

On peut voir sur ce schéma que l'application web et la base de données résident sur la même machine physique, un serveur Linux avec une distribution "Debian". Pour des raisons de performances, il aurait pourtant été mieux de séparer le serveur contenant le SGBDR (Système de Gestion de Base de Données Relationnel) du serveur Apache contenant l'application PHP. Le volume de la base de données grossissant, il faudra probablement d'ici quelques années songer à faire cette optimisation en achetant une machine supplémentaire dédiée.

Pour des raisons de sécurité et de disponibilité (éventuelles coupures internet), le serveur est accessible uniquement via le réseau local d'eRcane (Intranet).

### *La base de données*

La modélisation de la base de données est une étape incontournable de la conception d'un système d'information. Elle a été réalisée en respectant la méthode MERISE (MERISE est une méthode d'analyse et de conception de système informatique), qui est la référence en la matière. Cette méthode a aidé à mettre en évidence les différentes entités et relations caractérisant le processus de création variétale et d'en définir ainsi le modèle conceptuel de données (Figure 3).

Même si la partie "traitement" (les modèles conceptuels de traitement) avait été réalisée en 1996, il n'y a que la partie "données" (les modèles conceptuels de données) qui fut par la suite réutilisée. En effet, MERISE est surtout utilisée pour la conception de base de données et il

est en général préférable d'opter pour une approche "objet" pour modéliser les applications informatiques (cf. "Application web").

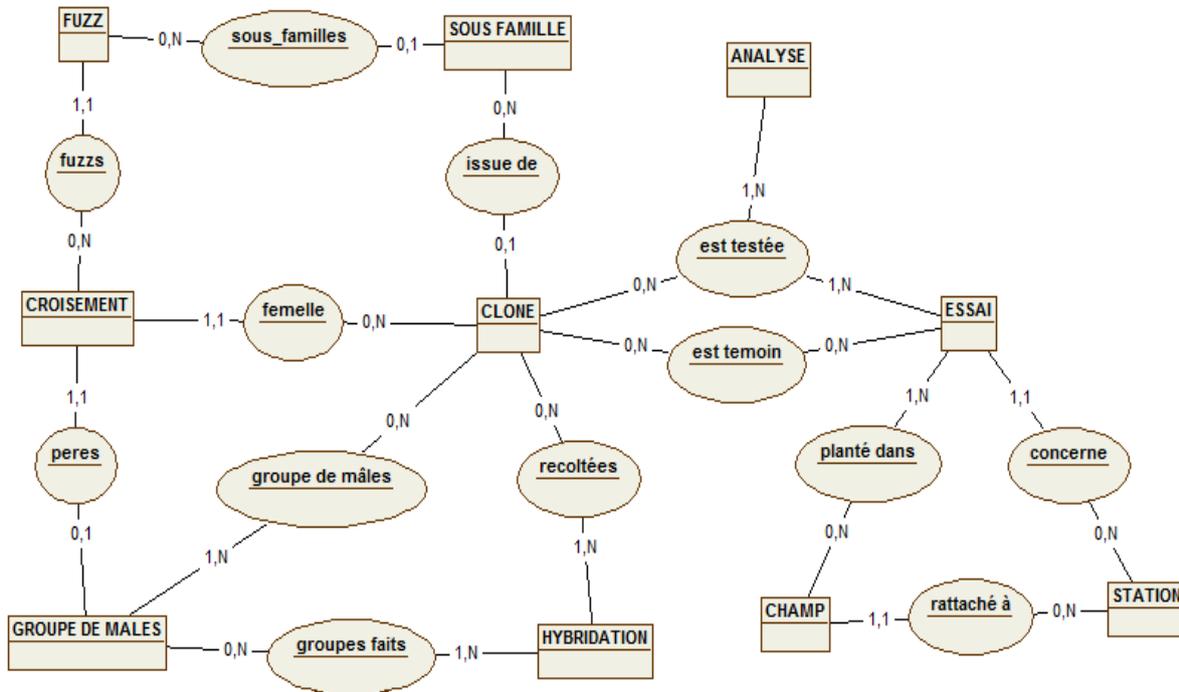


Figure 3. Modèle conceptuel de données « simplifié » du système (Les valeurs 0,1 et N sont les cardinalités des relations entre les entités : elles expriment le nombre d'occurrence minimale et maximale qu'une entité peut avoir avec la relation).

C'est le SGBDR PostgreSQL qui a été choisi (au détriment de MySQL) pour héberger la base de données car il possède un module pour le support d'objets géographiques (objets possédant des coordonnées GPS par exemple, pour la réalisation de cartes) : l'extension POSTGIS.

La possibilité de développer un aspect géographique de l'application est bien envisagé depuis le début du projet mais cette fonctionnalité revêtant une importance secondaire n'a pour le moment pas encore été développée. A l'avenir, la génération de cartes des parcelles d'expérimentations à partir d'informations contenues dans la base pourrait s'avérer utile dans le processus de sélection variétale, car cela automatisera un travail qui peut être fait par la machine et donc libérer du temps pour optimiser la sélection.

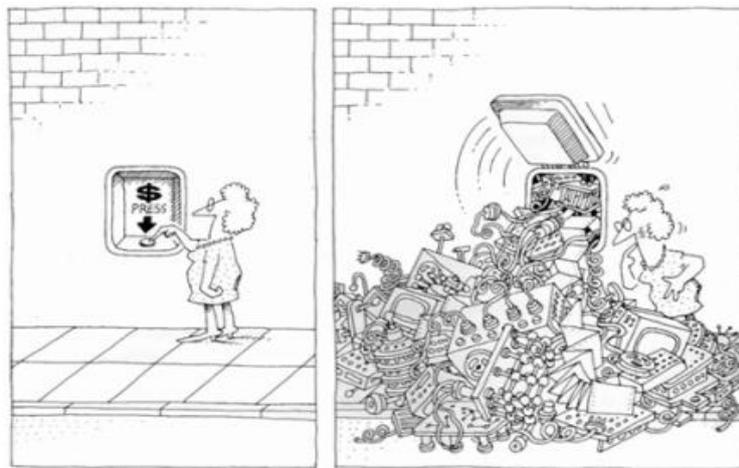
L'utilisation d'un SGBDR va permettre de garantir et préserver l'intégrité, l'atomicité et la consistance des données.

Dans un souci d'optimisation des temps d'exécution, des procédures stockées ont également été créées. Certaines d'entre elles ont même remplacé des parties du programme devenues trop lourdes. Une procédure stockée est une fonction, un ensemble d'instructions précompilées, stockée directement dans la base de données, ce qui rend son exécution plus performante (cf. Indices Economiques).

### L'application web

L'accès aux données de la base directement depuis le SGBDR est tout à fait possible mais nécessite une maîtrise du langage SQL et plus généralement d'avoir quelques notions en informatique (être capable de se connecter à la base par exemple). D'autre part, la présentation des résultats des requêtes faites par ce biais ne sera pas aussi parlante qu'un graphique ou qu'un tableau de bord par exemple.

C'est pourquoi les utilisateurs du système ont besoin d'une IHM (Interface Homme Machine) pour pouvoir gérer leurs données de manière plus conviviale. C'est en quelque sorte la partie visible de l'application avec lequel l'utilisateur interagit (Figure 4).



**Figure 4.** "La tâche de l'équipe de développement est de donner l'illusion de la simplicité" [BOOCH, 94]

Afin de maîtriser la complexité inhérente au logiciel, plusieurs techniques ont dû être mises en œuvre pendant la conception et le développement de l'application web créée pour BDCS : l'application « Création Variétale ».

Outre une décomposition en modules et une répartition plus logique des fonctionnalités de l'application sur les couches CLIENT et SERVEUR, une réorganisation complète du code source initial a été achevée fin 2010 en suivant le modèle MVC (Modèle Vue Contrôleur). L'architecture MVC (Modèle Vue Contrôleur) a pour but d'organiser une application en séparant : les données (modèle), l'interface utilisateur (vue) et le comportement de l'application (contrôleur). Le but de toute cette démarche est d'arriver à diviser l'application en sous-systèmes plus simples à appréhender (*Diviser pour « mieux » régner. E. DIJKSTRA*).

Un autre moyen de maîtriser cette complexité est d'utiliser le paradigme « objet » lors de la modélisation de l'application. Le paradigme « objet » est une représentation du monde (que l'on souhaite modéliser) avec des objets ayant des caractéristiques propres (des données) et un comportement propre (du code travaillant sur ses données). Un programme informatique orienté objet est composé d'objets qui communiquent entre eux. La modélisation objet de l'application « Création variétale » a été faite lors du premier semestre 2008 en utilisant le langage de modélisation UML.

### *Le PDA (Personal Digital Assistant)*

En utilisant l'application « Création Variétale » sur poste fixe, il s'est vite révélé qu'une solution nomade serait un outil précieux étant donné que de nombreuses données sont récoltées au champ ou dans les serres d'hybridation ou de semis.



Ainsi, en 2008 le choix d'un PDA "durci" (PDA répondant aux normes IP64\*) s'est imposé (Figure 5).

\*Le code IP, pour Ingress Protection (indice de protection en français), est une norme qui permet de définir le degré de protection des enveloppes des matériels électriques. Un matériel IP64 est ainsi totalement protégé contre la poussière et contre les éclaboussures. Compte tenu des conditions climatiques spécifiques à l'île de La Réunion (humidité dans l'air importante) et des conditions au champ et dans les serres il s'est avéré nécessaire d'opter pour un tel matériel.

**Figure 5. Exemple de PDA**

Deux applications ont été développées pour l'acquisition de données sur le terrain avec le PDA (en langage C#). L'une d'entre elle a été abandonnée car elle n'était pas adaptée à ce travail de terrain. L'autre apporte un réel plus à l'équipe d'hybridation pour gérer les croisements annuels grâce à l'utilisation de code-barres identifiant chaque caisse de semis, le PDA étant pourvu d'un lecteur de code-barres.

### *Emergence de nouvelles technologies*

L'application "Création variétale", ou du moins certaines parties qui la composent, a pu être réalisée grâce à l'émergence des nouvelles technologies en programmation. Voici brièvement certaines d'entre elles.

#### *- AJAX (Asynchronous Javascript And XML)*

AJAX est une technologie récente (2005), ou pour être plus exact un ensemble de technologies (JavaScript, XML, CSS, DOM et surtout XMLHttpRequest) utilisées conjointement. Elle a révolutionné le fonctionnement des sites et des applications web. Lors d'une navigation classique, chaque action de l'utilisateur (clic sur un lien, sur un bouton, etc...) se traduisait par une requête envoyée au serveur, et une fois la réponse de ce dernier reçue, par l'affichage d'une nouvelle page. Avec AJAX, il n'y a pas de rechargement de la page et les échanges client-serveur se font de manière asynchrone, c'est-à-dire sans bloquer l'utilisateur.

#### *- JSON (JavaScript Object Notation)*

JSON est un format d'échange de données « léger » de type texte. Il est par conséquent facile à lire ou à écrire pour un humain et peut-être facilement analysé et généré par une machine. Dans l'application "Création variétale", les réponses des appels faits en AJAX sont presque toujours au format JSON. Le serveur ne s'occupe que de récupérer les données à afficher et

les retourner au format JSON. Quand la réponse est reçue par le client, le code JavaScript (partie du code qui s'exécute sur le PC de l'utilisateur pour traduire la réponse du serveur) va pouvoir analyser le flux JSON et mettre en forme les données pour l'utilisateur. Ainsi, on peut plus efficacement répartir la charge entre le code devant s'exécuter sur le serveur (la récupération des données) et celui qui peut s'exécuter sur le poste client (l'affichage).

Lors du développement des applications sur le PDA, il a fallu dès le début penser aux ressources disponibles car celles-ci étaient restreintes (64MB RAM à partager entre le stockage, la mémoire pour les programmes et le système lui-même).

Le PDA ne disposant pas d'interface WIFI ou GPRS, il a fallu utiliser un fichier « tampon » de petite taille pour le chargement et la sauvegarde des données. Le format JSON semblait idéal pour ce fichier temporaire (~1MB).

- JQuery

JQuery est une bibliothèque de code JavaScript gratuite pour réaliser des interfaces graphiques riches (animations, effets, menu déroulant) avec très peu de lignes de code. Grâce à son support en natif, elle permet également d'utiliser les technologies AJAX et JSON plus simplement.

## Quelques résultats

### Carte d'identité variétale

La figure 6 montre le formulaire pour la consultation et la mise à jour des variétés de canne à sucre du système BDCS. Grâce à l'utilisation des requêtes AJAX, les échanges d'informations entre le client et le serveur se font très rapidement et l'utilisateur peut consulter et mettre à jour plusieurs variétés à la volée sans rechargement de la page.

Figure 6. Formulaire de consultation et de modification des variétés de canne à sucre

## Indices Economiques

A partir du stade « 2 lignes », le classement des variétés au sein d'un même essai se fait grâce à des indices calculés à partir des données brutes recueillies. Ces indices sont les suivants :

- le tonnage par hectare ;
- le taux de fibre ;
- le sucre extractible (SE% C) ;
- l'indice économique (SE-4) ;

Au départ, ces calculs se faisaient depuis l'application « Création variétale » grâce au module de statistiques « stats ». Avec cette procédure, le temps de réponse du système était beaucoup trop long (environ 20 minutes pour afficher tous les indices calculés pour 1 essai donné). Grâce à l'utilisation des procédures stockées, les calculs sont maintenant directement réalisés dans le SGBDR et le calcul et l'affichage des indices économiques ne prennent plus que quelques secondes (Figure 7).

Fonction	Arguments	Valeur de sortie	Langage
calculer_indicesessai_semoins4	integer	setof type_retour_indice_economique	sql
<b>Définition</b>			
<pre> 1 2 select (TONNAGE*(SUCRE_EXTRACTIBLE-4)/100) as SE_moins_4,VARIETE_tonnage,BLOC_tonnage from 3 (select cast(valeur_indice as float) as TONNAGE,id_variete as VARIETE_tonnage,id_bloc as BLOC_tonnage from 4 Calculer_indicesessai_tonnage(#1) as subquery_tonnage) as sous_requetel, 5 6 (select cast(valeur_indice as float) as SUCRE_EXTRACTIBLE,id_variete as VARIETE_sucreeextractible,id_bloc as BLOC_sucreeextractible from 7 Calculer_IndicesEssai_SucreeExtractible2(#1) as subquery_sucreeextractible) as subquery_semoins4 8 WHERE VARIETE_tonnage=VARIETE_sucreeextractible AND BLOC_tonnage=BLOC_sucreeextractible; 9 10 11 </pre>			

Figure 7. Exemple de procédure stockée (appelant 2 autres) utilisée pour le calcul de l'indice économique

## Tableau d'hybridation

Tous les ans, pendant la période des hybridations, l'équipe de la création variétale doit réaliser des croisements à partir des fleurs de canne présentes dans différentes parcelles de floraison. Le choix des croisements à effectuer est facilité par le système d'information. C'est par exemple ici que JQuery, associé avec AJAX, offre au responsable de la création variétale des effets graphiques qui entraînent une bonne interactivité et convivialité.

## Conclusion et Perspectives

BDCS est donc une refonte d'un système présentant trop de lacunes (redondance d'informations, synthèse des données trop longue...) dont le but est de permettre une gestion des données aussi "simple, intuitive, et efficace" (Saurel, 1996) que possible. Bien que toujours perfectible sur plusieurs points, ce système constitue déjà aujourd'hui un véritable outil pour les équipes de la création variétale qui l'utilisent quotidiennement, et conformément à leur souhait et à la stratégie de sélection d'eRcane, reste un outil au service de, et piloté par les sélectionneurs. Par exemple, ce n'est pas le système d'information qui va automatiquement choisir les croisements « prometteurs » à hybrider ou les variétés « élites » à sélectionner. Les perspectives envisagées pour faire évoluer ce projet sont nombreuses (cartographie des parcelles d'expérimentations avec le module PostGIS, gestion du stock de fuzz avec le PDA et les codes-barres, tableau de bord de la sélection, valorisation des données de l'hybridation, nouvelles requêtes, etc...).

### **Références bibliographiques**

**Saurel, S.** (1996) Dossier d'analyse B.D.C.S.

**Zermatten, E.** (2008) Application C.E.R.F. « Création variétale ».

**Booch, G.** (1994) Object-Oriented Analysis and Design with Applications.