

**CONVENTION DE PARTENARIAT POUR LE DEVELOPPEMENT DE L'OUTIL INFORMATIQUE CAPDEMAT**

**Entre :**

**Le Conseil Général du Val d'Oise**, représenté par Didier ARNAL, Président du Conseil Général du Val d'Oise agissant au nom et pour le compte du Département, en vertu d'une délibération de la Commission Permanente, en date du ....., élisant domicile à l'Hôtel du Département, 2 avenue du Parc 95032 CERGY PONTOISE CEDEX.

**Le Conseil Général de Seine St Denis**, représenté par Claude BARTOLONE, Président du Conseil Général de Seine St Denis agissant au nom et pour le compte du Département, en vertu d'une délibération de la Commission Permanente, en date du ....., élisant domicile à l'Hôtel du Département, 93006 BOBIGNY CEDEX.

**Le Conseil Général de Seine et Marne**, représenté par Vincent EBLE, Président du Conseil Général de Seine et Marne agissant au nom et pour le compte du Département, en vertu d'une délibération du Conseil général, en date du ....., élisant domicile à l'Hôtel du Département, 77000 MELUN CEDEX.

**Le Conseil Général de la Gironde**, représenté par Philippe MADRELLE, Président du Conseil Général de la Gironde agissant au nom et pour le compte du Département, en vertu d'une délibération de la Commission Permanente, en date du ....., élisant domicile à l'Hôtel du Département, Esplanade Charles de Gaulle - 33074 BORDEAUX CEDEX.

**La Commune de Limoges**, représentée par Alain RODET, Maire de la Commune de Limoges agissant au nom et pour le compte de la commune, en vertu d'une délibération du Conseil Municipal en date du ....., élisant domicile à l'Hôtel de Ville, place Léon Betoulle 87000 LIMOGES CEDEX.

**La Commune de Poitiers**, représentée par Alain CLAEYS, Maire de la Commune de Poitiers agissant au nom et pour le compte de la commune, en vertu d'une délibération du Conseil Municipal en date du ....., élisant domicile à l'Hôtel de Ville, BP 569 - 86021 POITIERS CEDEX

**La Commune de Vincennes**, représentée par Laurent LAFON, Maire de la Commune de Vincennes agissant au nom et pour le compte de la commune, en vertu d'une délibération du Conseil Municipal en date du ....., élisant domicile à l'Hôtel de Ville, 53 bis rue Fontenay 94300 VINCENNES

**La Commune de Pessac**, représentée par Jean-Jacques BENOÎT Maire de la Commune de Pessac agissant au nom et pour le compte de la commune, en vertu d'une délibération du Conseil Municipal en date du ....., élisant domicile à l'Hôtel de Ville, place de la 5ème République 33604 PESSAC CEDEX

**Il a été convenu ce qui suit :**

## Sommaire

PREAMBULE.....	3
Article 1 – Objet de la convention .....	3
Article 2 – Objectifs de la Communauté.....	3
Article 3 – Outils informatiques concernés par la convention.....	4
Article 4 - Orientations stratégiques partagées .....	4
Article 5 - Engagements des partenaires .....	5
Article 6 - Vie et gestion de la Communauté et création d’une structure pour assurer la gestion de la communauté et mettre en œuvre ses actions .....	5
6.1- Pilote de La Communauté .....	6
6.2- Groupement de plusieurs membres au sein de la Communauté.....	6
6.3- Actions à réaliser sous l’égide de la Communauté et création d’une structure ad hoc pour assurer la gestion de la communauté et mettre en œuvre ses actions.....	6
6.4- Modalités de partage des frais entre partenaires .....	8
6.5- Droit de vote dans le cadre de la future structure.....	8
6.6- Mode de gouvernance .....	9
Article 7 - Représentants des partenaires dans la Communauté .....	9
Article 8 - Développements informatiques externes et documentations afférentes .....	9
Article 9 - Propriété des développements à venir et documents afférents .....	9
Article 10 - Acquisition du statut de membre fondateur.....	10
Article 11 : Modifications de la convention.....	10
Article 12 : Durée de la convention .....	10
Article 13 : Résiliation de la convention.....	10
Article 14 : Règlement des litiges .....	11
ANNEXES .....	13
Annexe 1 : Historique .....	13
Annexe 2 : Catalogue des outils concernés par la convention .....	15
Description fonctionnelle et technique de CapDémat :.....	16
Description fonctionnelle et technique d’Angélus .....	20
Description fonctionnelle et technique de Sematic .....	24
Annexe 3 : Règles de développement applicables à toutes contributions.....	27
Fin des annexes .....	35

## **PREAMBULE**

Le Département du Val d'Oise a développé pour le compte des collectivités valdoisiennes et son propre compte un ensemble d'outils open source réunis sous la dénomination de CapWebCT dont il est propriétaire et dont l'historique est résumé en annexe 1.

Le Département du Val d'Oise propose cette offre technologique open source à toutes les collectivités territoriales (départements, régions, communes et communautés de communes francophones) qui souhaitent soit réduire la fracture numérique sur leur territoire soit les utiliser pour leurs propres besoins, en bénéficiant d'outils d'administration électronique open source rodés et performants, d'une méthode et d'une expérience de 6 ans.

Les signataires de la présente convention, dénommés « partenaires fondateurs », ont décidé d'utiliser la plateforme open source CapDémat provenant de la suite CapWebCT pour répondre à leurs besoins propres ou pour le compte des collectivités territoriales de leur territoire. Par ailleurs, les partenaires fondateurs souhaitent faire évoluer la plateforme CapDémat en coordonnant leurs efforts et en mutualisant ceux-ci chaque fois que cela est possible. Les partenaires fondateurs forment ainsi une communauté d'intérêts dénommée ci-après « la Communauté »

Les partenaires souhaitent maîtriser leur système d'information de relation avec leurs usagers et leurs interlocuteurs (fournisseurs, associations, autres collectivités, services de l'État, ...). CapDémat apparaît comme un outil central pour participer à l'élaboration du plan d'évolution du système d'information, contribuer à son évolution et éviter toute création de développements dérivés qui pénaliseraient la mutualisation et les intérêts des utilisateurs.

### **Article 1 – Objet de la convention**

La convention a pour objet:

- de définir les objectifs communs et les règles auxquelles les partenaires acceptent de se soumettre pour atteindre ces objectifs communs
- de définir le périmètre des outils informatiques concernés par la convention
- de régir les liens entre les partenaires pour assurer la coordination des travaux sur ces outils, mettre en place et faire vivre la gouvernance de la Communauté et favoriser toute mutualisation des efforts humains et financiers pour faire évoluer les outils concernés par cette convention et assurer leur pérennité.

### **Article 2 – Objectifs de la Communauté**

Les objectifs de la Communauté sont :

- Optimiser et partager les coûts de maintenance et d'investissements des collectivités sur CapDémat et tous les outils connexes listés en annexe 2 de la présente convention,
- Permettre d'avancer plus rapidement à plusieurs partenaires vers la production et l'usage d'outils pérennes et éprouvés d'administration électronique locale (communale, intercommunale, départementale et régionale) inter opérables avec ceux de l'administration centrale et déconcentrée de l'Etat.
- Structurer les outils pour accompagner aisément toutes modifications à venir de responsabilité entre les différents types de collectivités

- Eviter de créer des souches logicielles divergentes devenant rapidement incompatibles entre-elles et rendant impossible la mutualisation attendue.
- Mutualiser les expériences des partenaires en matière d'administration électronique
- Favoriser l'utilisation des outils par d'autres collectivités
- Valoriser la Communauté et ses partenaires par l'exemplarité de la démarche de mutualisation et par l'avancée dans le domaine de la relation avec les usagers et les interlocuteurs de la collectivité,
- Construire une communauté utilisatrice autour de la communauté fondatrice
- Constituer une liste de société ayant une bonne connaissance du socle technique et fonctionnel leur permettant d'accompagner les communautés fondatrice et utilisatrice.

### **Article 3 – Outils informatiques concernés par la convention**

Les outils open source concernés par la convention sont définis et décrits en annexe 2 de la présente convention.

CapDémat est l'outil principal de cette liste initialement composée de 3 outils. Cette liste des outils informatiques fixée en annexe 2 de la présente convention n'est pas exhaustive. D'autres outils informatiques open source pourront être proposés par l'un ou l'autre des partenaires fondateurs. Ils devront relever strictement du périmètre décrit à l'article 3 de la présente convention. L'inscription d'un nouvel outil open source relèvera d'une décision unanime des partenaires.

Le périmètre cible des outils open source concernés par la convention qui peuvent être inscrits en annexe 2 de la présente convention est défini par la liste des type d'outils suivants:

1. Outils d'informations des usagers et correspondants : outil de gestion de sites web (CMS)
2. Outils d'interaction numérique avec l'utilisateur et les correspondants sans authentification forte
3. Outils de transaction numérique avec l'utilisateur et les correspondants avec authentification forte et traitement des demandes de bout en bout
4. Gestion des identités des usagers et des correspondants en mode référentiel du système d'information
5. Plateforme de débats publics
6. Gestion multi canal des flux d'informations entre les usagers, les correspondants et les collectivités
7. Réseaux sociaux territoriaux
8. Intra/extranet collaboratif et bureau virtuel
9. Réseau social d'administration

L'ensemble de ce périmètre concourt à l'amélioration de la relation aux usagers et aux correspondants des collectivités.

La liste des outils doit pouvoir évoluer ; elle devra être établie et validée par la Communauté et détenu par le pilote.

### **Article 4 - Orientations stratégiques partagées**

Tous les partenaires de la convention ont adopté les orientations stratégiques suivantes:

- Disposer d'un ensemble d'outils open source cohérents et maîtrisés de relation numérique avec ses usagers et ses interlocuteurs, sans dépendance forte avec les applications métiers qui composent chaque système d'information,
- Partager ces outils avec le plus grand nombre de collectivités francophones,
- Assurer ensemble l'adéquation aux besoins évolutifs des collectivités et la pérennité,
- Eviter, autant que faire se peut, que voient le jour des évolutions séparées des outils en fonction de la cible (Région, Département, Communauté de communes ou d'agglomération et Communes),
- Publier officiellement toutes les spécifications d'interfaces avec les applications métiers pour favoriser la réalisation, par les éditeurs de ces applicatifs, de leurs interfaces avec les outils de l'annexe 2 de la présente convention.
- Développer la Communauté pour devenir un interlocuteur incontournable des éditeurs et de l'Etat en matière d'administration électronique territoriale,
- Assurer la communication et la promotion autour du produit.

## **Article 5 - Engagements des partenaires**

- La concertation des membres de la Communauté sera faite sous forme de réunions auxquelles la présence de tous les membres est recommandée pour un meilleur pilotage de projet.
- Chaque partenaire s'engage à rechercher conjointement toute solution pour éviter la divergence de souches logicielles des outils mentionnés en annexe 2 de la présente convention et à se concerter régulièrement sur les développements en cours, les plans d'évolution de chacun des membres et les changements potentiels, notamment en participant aux instances prévues de la structure.
- Chaque partenaire s'engage à fournir aux autres parties toute information ou document utile à la compréhension des développements prévus par lui-même, en cours de réalisation et réalisés et à assurer l'intégration dans le gestionnaire de sources unique par outil.
- Chaque partenaire cherchera à éviter, tout développement individuel et spécifique pour sa collectivité sur l'un ou l'autre des outils mentionnés à l'annexe 2 de la présente convention.
- Chaque partenaire développant ou faisant développer des fonctionnalités nouvelles ou des évolutions de fonctionnalités des outils mentionnés à l'annexe 2 de la présente convention, est désigné comme partenaire contributeur. Il s'engage à ce que les développements soient faits dans les règles de l'art avec la meilleure qualité de développement, conforme aux exigences de l'annexe 3 de la présente convention. Des procédures d'intégration des contributions et outils de vérification qualitative du code source seront précisés ultérieurement afin de faciliter l'intégration des développements dans le tronc commun et la maintenance ultérieure du code source. La maintenabilité et l'évolutivité du code source est une préoccupation permanente de la Communauté car elle sait que toute dégradation de qualité amènera ultérieurement un surcoût de maintenance et de frais de développement.

## **Article 6 - Vie et gestion de la Communauté et création d'une structure pour assurer la gestion de la communauté et mettre en œuvre ses actions**

La vie et la gestion de la Communauté représentant une charge de travail importante, la Communauté s'engage à créer rapidement une structure **dont la forme reste à définir** (association, ou autre forme de groupement) permettant de pouvoir acheter des prestations en recourant aux marchés publics et d'élargir au fur et à mesure l'adhésion à ce groupement à d'autres collectivités souhaitant participer aux actions et partager les efforts humains et financiers

de celui-ci. Cette structure prendra les décisions pour l'ensemble des membres et agira en leur nom.

### **6.1- Pilote de La Communauté**

La Communauté confie à un des partenaires fondateurs la responsabilité du pilotage de la Communauté.

Ce partenaire est dénommé le Pilote.

Le Pilote sera nommé à l'unanimité des membres fondateurs pour 1 an. Ce mandat fera l'objet de reconduction expresse d'année en année dès lors qu'aucun des partenaires fondateurs ne le dénonce 3 mois au plus tard avant la fin du mandat du Pilote. Le changement de Pilote pourrait intervenir durant les quatre ans.

La désignation du Pilote ne nécessitera pas de modification de la présente convention.

Le Pilote s'engage à assumer l'animation de la Communauté et d'en garantir la bonne gouvernance et accepte la charge et le rôle qui lui sont attribués et les règles et conditions de fonctionnement décrites par la présente convention.

Les missions du Pilote sont les suivantes :

Pendant la période aboutissant à la création de la structure,

- Organiser la démarche de création de la structure à venir pour la mener à terme dans les meilleurs délais,
- Maintenir le même niveau d'information de tous les membres fondateurs
- Piloter des opérations engagées sur décision unanime des fondateurs qui ne peuvent attendre la création de la structure

Après la création de la structure,

- Participer ou piloter le conseil de surveillance de la structure en fonction des décisions qui seront arrêtées d'un commun accord pour la structure.

### **6.2- Groupement de plusieurs membres au sein de la Communauté**

Tant que de besoin et pour favoriser la mutualisation, un ou plusieurs partenaires peuvent constituer un groupement au sein de la Communauté, dénommé groupe de partenaires, pour engager une action commune dès lors que cette action n'intéresse pas la totalité de la Communauté.

Le groupe de partenaires est constitué après que les représentants des partenaires du groupement ont signifié par écrit au Pilote la constitution du groupe et les objectifs partagés de ces partenaires. Ces objectifs devront être conformes à ceux définis par la présente convention et notamment à ses articles 2, 3, et 4 de la convention.

Le Pilote informera alors les membres de la Communauté de la création du groupement.

En dehors de ces groupements, chacun pourra financer ses propres projets qui seront, par la suite, intégrés au projet initial et tous les autres membres pourront en bénéficier.

### **6.3- Actions à réaliser sous l'égide de la Communauté et création d'une structure ad hoc pour assurer la gestion de la communauté et mettre en œuvre ses actions**

Les actions à mener nécessaires à la vie, à la gestion de la Communauté et à son extension sont les suivantes :

- Définition, validation collective et création de la structure chargée de porter les actions de mutualisation humaine et financière nécessaires à la vie, à la gestion de la communauté et son agrandissement
- Mise en place de la gouvernance et des règles de répartitions des frais,
- Mise à jour régulière des normes applicables (article 4 de la convention)
- Constitution et mise à jour de la road map, analyse des impacts sur le SI
- Surveillance et contrôle de la structure qui sera créée.

Le Pilote s'engagera à faire un reporting diffusé électroniquement de toutes ses actions à l'ensemble de la Communauté. Il sera notamment chargé de diffuser tous les comptes-rendus élaborés à tour de rôle par les partenaires de la présente convention.

Les partenaires ont déterminé les actions qui seront transférées à la structure gestionnaire du projet. De manière non-exhaustive ces actions sont les suivantes :

Actions de pilotage, communication, animation et formation :

- Animation du club utilisateurs des outils
- Mise en place, hébergement et gestion d'un extranet collaboratif de la Communauté
- Liens d'informations avec les éditeurs métier
- Organisation d'un colloque annuel d'administration électronique
- Actions de communications (site web, newsletter, préparation de présentations, interventions pour le compte de la communauté,...)
- Formations
- Organisation de formations d'éditeurs et de sociétés intégratrices des outils (la réalisation étant confiée à un prestataire qualifié qui sera payé directement pas ces sociétés privés)
- Gestion des réunions mensuelles de la Communauté

Actions de Pilotage de la forge :

- Expression de besoin et spécifications communes
- Définition des procédures d'intégration des contributions et outils de vérifications qualitative du code source
- Gestion de la forge et des merges sur version stable de la forge
- Intégrations des contributions validées
- Développement des évolutions technologiques des outils
- Maintenance corrective et évolutive des outils sur la Forge
- Gestion documentaire et juridique avec publication dans l'extranet collaboratif de la communauté.
- Pilote les audits sécurité et la conformité au **Référentiel Général d'Accessibilité des Administrations** selon le décret n°2009-546 du 14 mai 2009 d'accessibilité.

Actions de validation :

- Vérification de qualité du code soumis par les partenaires ou par des contributeurs externes à La Communauté et intégration des livrables
- Tests de labellisation des intégrateurs et proposition aux partenaires fondateurs à la labellisation
- Tests de labellisation des connecteurs métier et proposition aux partenaires fondateurs à la labellisation
- Vérification des règles de l'annexe 3
- Validation de la documentation

Actions de consulting :

Cadrage stratégique et opérationnel, méthodologie, organisation, accompagnement, pilotage, formation

#### **6.4- Modalités de partage des frais entre partenaires**

Le montant prévisionnel des actions à réaliser, si elles sont confiées à un prestataire extérieur, est évalué la première année à la somme de 450 000 € HT et les années suivantes à 400 000 € HT.

La prise en charge de ces montants sera partagée entre les partenaires de la Communauté dont l'intérêt est de s'élargir rapidement.

Les statuts de la future structure détermineront les modalités de partage des frais déduits des recettes perçues. Pour définir ces modalités, il est nécessaire de prendre en compte les capacités financières des uns et des autres partenaires sachant qu'elles ne sont pas équivalentes. Il est retenu que le budget DSI décompté des frais de téléphonie et d'imprimerie, et augmenté de la masse salariale chargée sera une des clefs de répartition.

Il est acté qu'une collectivité peut proposer du temps x homme pour la communauté valorisé en équivalent masse salariale chargée.

La charge du Pilote sera valorisée ainsi et inscrite dans les dépenses engagées par le Pilote.

Une pondération sur la clef précédente sera utile pour faire coïncider les capacités d'investissement des uns et des autres avec l'application d'une simple règle mathématique.

Il est entendu que le montage juridique de la structure doit permettre de favoriser la mutualisation financière d'opérations conjointes de plusieurs collectivités souhaitant développer ensemble telles ou telles fonctionnalités (la répartition des frais afférents sera alors fonction des engagements réciproques des partenaires de ce groupe).

Un certain nombre de recettes pourront être mobilisées par la structure pour diminuer le coût global des actions à réaliser.

Les recettes potentielles à mobiliser sont les suivantes :

- Cotisations d'adhésions des nouveaux membres (non fondateurs)
- Contributions financières des nouveaux adhérents
- Contributions de développements externes
- Missions d'accompagnement AMOA de collectivités
- Formations payantes dispensées par la structure ou ses prestataires
- Cotisations demandées aux Intégrateurs labellisés pour profiter des informations nécessaires à leurs prestations auprès des collectivités francophones
- Subventions diverses sollicitées par la Communauté ou ses membres

#### **6.5- Droit de vote dans le cadre de la future structure**

Le nombre de voix par membre de la Communauté sera défini ultérieurement dans le cadre des statuts de la structure à créer.

Les statuts de la structure détermineront deux types de collectivités adhérentes non-fondatrices :

- Les collectivités actives qui auront cotisé et contribué humainement et/ou financièrement à l'évolution des outils
- Les collectivités passives qui auront uniquement acquitté leurs cotisations

Un droit de vote sera attribué aux collectivités actives au prorata des contributions de l'ensemble de ces collectivités. Ce droit de vote cumulé sera plafonné pour maintenir le rôle prépondérant des partenaires fondateurs.

Une typologie des décisions, tant fonctionnelles que techniques, soumises au vote de la Communauté sera rédigée et validée par l'ensemble de la Communauté à l'occasion de la construction de la structure.

#### **6.6- Mode de gouvernance**

La Communauté se réunit deux fois par an pour réaliser un bilan intermédiaire et un bilan annuel des actions entreprises en commun et par groupe. Ces bilans sont préparés et présentés par le Pilote. Dans le même temps le Pilote procédera à la mise à jour de la road map. Le Pilote s'appuiera sur la structure pour la préparation de ces réunions dès que celle-ci sera en fonctionnement. La planification de ces deux instances devra anticiper les étapes budgétaires des collectivités partenaires.

A la création de la structure, la gouvernance de reversement, le mode de présentation des évolutions fonctionnelles et leur inscription à road map seront précisés ainsi que leur fréquence.

### **Article 7 - Représentants des partenaires dans la Communauté**

Les partenaires sont représentés dans la Communauté par le signataire à la présente convention ou par une personne ayant reçu délégation.

### **Article 8 - Développements informatiques externes et documentations afférentes**

Chaque partenaire dispose soit d'équipes internes de développement informatique soit de marché actif présent ou à venir de sous-traitance de développement informatique. Pour garder une liberté et la souplesse nécessaire pour atteindre ses propres objectifs de couverture fonctionnelle et de plannings, chaque partenaire peut développer ou faire développer des évolutions aux outils de l'annexe 2 de la présente convention en dehors de la Communauté. S'il souhaite intégrer ses développements dans la Communauté, ils devront être conformes avec les engagements techniques et de qualité définis dans le cadre de la présente convention, en respect de la Charte Qualité « CapDémat » qui sera une des premières actions mise en place par la structure.

Chaque développement proposé en contribution par un partenaire ou un groupe de partenaires, sera audité par la structure. Les corrections à apporter, suite à l'audit, seront à la charge du partenaire ou du groupe l'ayant développé ou fait développer, ainsi que l'intégration au tronc commun avec les tests nécessaires. Le dépôt des sources s'accompagnera à chaque fois du cahier de tests, des résultats et de la documentation fonctionnelle et technique complémentaire ou la mise à jour de la documentation en place. La définition de ces différents livrables sera effectuée lors de la constitution de la structure et fera l'objet d'une modification de l'annexe 3 de la présente convention sans nécessiter un avenant à cette convention.

### **Article 9 - Propriété des développements à venir et documents afférents**

La propriété des développements est régie, outil par outil de l'annexe 2 de la présente convention, par les termes de la licence open source voté par délibérations de l'organe délibérant de la collectivité propriétaire des sources.

## **Article 10 - Acquisition du statut de membre fondateur**

La Communauté souhaite s'élargir au mieux afin de mutualiser au maximum ses frais de vie décrits à l'article 6 de la présente convention, ainsi que les dépenses de développement et de maintenance des outils de l'annexe 2 de cette même convention. Les partenaires fondateurs signataires de la présente convention sont au nombre maximum de 10 collectivités. Cependant une collectivité adhérente à la structure à créer qui apporte une forte contribution peut souhaiter devenir un des partenaires fondateurs de la Communauté. Pour ce faire, les partenaires fondateurs décident à l'unanimité de conférer à ce membre la qualité de membre fondateur avec engagement de ce dernier de contribuer financièrement et humainement à la gestion de la communauté et de se soumettre aux mêmes règles et engagements pris par les partenaires fondateurs.

Pour obtenir la qualité de membre fondateur, la collectivité demandeuse doit faire une demande écrite au Pilote qui se chargera de recueillir les avis écrits de tous les partenaires fondateurs. En cas d'avis unanime, le Pilote informera la collectivité demandeuse de son acceptation en qualité de membre fondateur, des frais de cotisation, des frais supplémentaires engagés auxquels elle sera assujettie et lui transmettra la convention qui devra être votée par son assemblée délibérante.

Par ailleurs, la présente convention fera l'objet d'un avenant modifiant les parties à la présente convention en ajoutant le nouveau membre devenu membre fondateur.

La collectivité deviendra alors partenaire fondateur de la Communauté et jouira à ce titre dans la structure des mêmes prérogatives dont jouit chaque membre fondateur.

## **Article 11 : Modifications de la convention**

Toute modification du contenu de la présente convention fera l'objet d'un avenant, à celle-ci, soumis à l'approbation de l'assemblée délibérante de chacun des membres.

## **Article 12 : Durée de la convention**

La présente convention est conclue pour une durée de quatre ans.

La présente convention prendra effet au jour de sa notification à chacune des parties à la convention après signature des parties et transmission au représentant de l'État le concernant.

A l'issue de la convention, un bilan général sera présenté par les parties et une nouvelle convention sera proposée.

## **Article 13 : Résiliation de la convention**

- 1- Si l'une des parties souhaite mettre fin à la présente convention avant son terme, elle devra avertir tous les autres membres fondateurs par lettre recommandée avec accusé de réception en respectant un délai de préavis de 3 mois.
- 2- En cas de non respect des engagements réciproques inscrits dans la présente convention, celle-ci pourra être résiliée de plein droit par l'une des parties, à l'expiration d'un délai de quinze

jours suivant l'envoi d'une lettre recommandée avec accusé de réception valant mise en demeure.

#### **Article 14 : Règlement des litiges**

En cas de litige, né de l'application ou de l'interprétation de la présente convention, les parties s'engagent à épuiser toutes les voies de conciliation possible, avant de saisir le tribunal compétent.

Fait en 10 exemplaires.

A Cergy, le	<b>Monsieur Didier ARNAL</b> <b>Président du Conseil Général du Val d'Oise</b>
A Bobigny, le	<b>Monsieur Claude BARTOLONE</b> <b>Président du Conseil Général de Seine Saint Denis</b>
A Bordeaux, le	<b>Monsieur Philippe MADRELLE</b> <b>Président du Conseil Général de la Gironde</b>
A Melun, le	<b>Monsieur Vincent EBLE</b> <b>Président du Conseil Général de Seine et Marne</b>
A Limoges, le	<b>Monsieur Alain RODET</b> <b>Maire de la Commune de Limoges</b>
A Poitiers, le	<b>Monsieur Alain CLAEYS</b> <b>Maire de la Commune de Poitiers</b>
A Vincennes, le	

	<b>Monsieur Laurent LAFON</b> <b>Maire de la Commune de Vincennes</b>
--	--

A Pessac, le	<b>Monsieur Jean-Jacques BENOÎT</b> <b>Maire de la Commune de Pessac</b>
--------------	---

# ANNEXES

## Annexe 1 : Historique

Le Département du Val d'Oise, pendant 12 ans, a aidé les communes de moins de 10 000 habitants à s'informatiser dans les domaines budgétaires, comptables et ressources humaines.

En septembre 2000, L'Assemblée départementale du Val d'Oise a souhaité remplacer ce dispositif pour aider toutes les collectivités du Val d'Oise à pénétrer peu à peu et à leur rythme dans l'ère de l'administration électronique.

En mars 2002, un plan cadre, issu d'un travail en commun avec des élus locaux et des directeurs généraux, a été voté par l'Assemblée Départementale. Composé de 7 projets cohérents, ce plan s'est mis en place peu à peu avec une ligne de force: proposer aux collectivités une offre d'infrastructures technologiques dites "sans soucis" et les aides correspondantes pour leur mise en oeuvre. Cette offre a aujourd'hui 4 composantes:

- technique: une offre logicielle proposée gratuitement (open source) par le Département qui en assure le fonctionnement, l'évolution et la pérennité
- l'hébergement en mode « software as a service »
- une assistance à la mise en oeuvre par collectivité par des sociétés labellisées par le département
- un co-financement de cette assistance avec la région Ile de France

Dans la progression de l'informatif vers l'interactif puis le transactif avec l'utilisateur, l'offre open source dénommé CapwebCT se déploie en Val d'Oise et hors Val d'Oise aujourd'hui avec plusieurs modules:

- Un générateur de sites web orientés relation à l'utilisateur avec gestion de contenu, workflow de publication, géolocalisation et outils d'interactivité avec l'utilisateur dénommé CapInfo
- Un portail intranet/extranet collaboratif avec bureau virtuel dénommé CapAgent
- Un portail de télé-services issu de l'expérimentation "Cartevaloise" dans le cadre des projets « carte de vie quotidienne » retenus par l'ADAE et son atelier de création de télé-services dénommé CapDémat.

Le générateur de sites web et le portail intranet sont en cours de fusion dans un même outil dénommé CapXnet développé à partir des outils open source de la société ExoPlatform,

Par ailleurs, ces outils s'étendent aux besoins propres des départements en matière de relation à l'utilisateur (Handicap, personnes âgées, bourses,...) et complètent ainsi une offre territoriale multi-niveaux et interopérable avec l'offre "Monservicepublique.fr" de la DGME.

Le Département du Val d'Oise est propriétaire des outils recouverts par la dénomination CapWebCT, de la documentation technique et d'un ensemble de documents utiles pour la compréhension des outils et des prestations à réaliser pour déployer ces outils (documentation d'installation et d'exploitation, exemples de délibération prise par le Val d'Oise, kit de déploiement des outils pour les collectivités, déclaration à la CNIL-type, etc....). L'intégralité des outils logiciels développés en java, J2EE sur des plates-formes Linux, Apache, et Tomcat sont sous licence logiciel libre de type GPL3 tel que l'a décidé l'assemblée départementale du Val d'Oise par délibérations en date du 19/09/2003, 15/04/2005 et 07/07/2008.

Le Département du Val d'Oise gère, depuis leur création, l'évolution des outils de CapWebCT par un marché de tierce-maintenance applicative dans le cadre d'un marché européen.

Cette offre technologique libre est aujourd'hui disponible pour toutes les collectivités (départements, régions, communes et communautés de communes de France) qui souhaitent soit réduire la fracture numérique sur leur territoire en bénéficiant d'outils d'administration électronique open source rodés et performants, une méthode et une expérience de 6 ans soit les utiliser pour leurs propres besoins.

## Annexe 2 : Catalogue des outils concernés par la convention

Nom de l'outil	Type d'outil (article 3)	Propriétaire	Licence open source	Site de la forge	Date d'entrée dans la liste
CapDémat	2 et 3	Département du Val d'Oise	GPL3	<a href="http://www.capwebct.fr">www.capwebct.fr</a>	DSC *
Angelus	4	Commune de Limoges	GPL	<a href="http://www.addulact.fr">www.addulact.fr</a>	DSC *
Sematic	1	Département de Seine et Marne	GPL3	<a href="http://www.sematic.fr">http://www.sematic.fr</a>	DSC *

\*DSC= Date de Signature de la Convention

## Description fonctionnelle et technique de CapDémat :

### Présentation

Le Département du Val d'Oise, avec la maîtrise d'œuvre de la DSI, s'est inscrit dans un vaste projet de Développement de l'Administration Electronique. Les objectifs de ce projet sont d'offrir aux usagers des collectivités un accès simplifié aux services administratifs, d'établir de nouvelles relations de partenariat entre les responsables publics et privés chargés des services de proximité et d'offrir une meilleure visibilité aux acteurs engagés dans les projets de dématérialisation.

La conséquence la plus concrète de ce projet est la plate-forme de télé-services, le module «CapDémat» de la suite CapWebCT. Utilisable dans la vie de tous les jours pour des services aussi divers que l'inscription des enfants à l'école, à la cantine, aux centres de loisirs, l'accès aux équipements sportifs et culturels, mais aussi pour les transports ou le stationnement... L'objectif est d'offrir aux usagers particuliers sur un territoire (ville, département, région) un bouquet de services publics locaux facilement accessibles à partir de l'Internet ou via l'utilisation d'une borne interactive.

Une nouvelle version V4.2 est maintenant disponible, elle agrège les évolutions technologiques, les retours d'expérience de l'utilisation des versions antérieurs et l'évolution des besoins de gestion de la relation avec les citoyens.

Quatre axes majeurs déterminent les axes de changement de cette nouvelle version :

- **De nouvelles interfaces usager plus lisibles et accessibles**
  - Une vision plus homogène
  - Une simplification des formulaires de demande
  - Une vision plus moderne web2.0 intégrant les contraintes d'accessibilité
  - Un mode brouillon permettant à l'utilisateur de réaliser ces demandes en plusieurs fois sans perte d'information
  - De l'aide fonctionnelle à chaque étape d'une demande
  - De l'aide générale et des "Faq" pour mieux comprendre l'utilisation de la plate-forme
  - ...





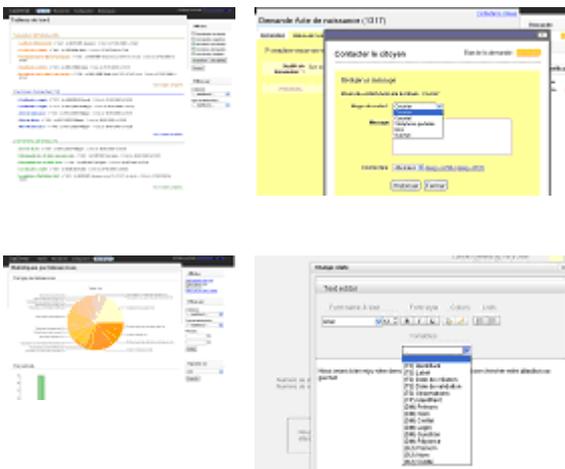
- **Un bouquet de service en constante évolution**

- De nouveaux services comme les déchets verts et les encombrant
- Un bouquet de service qui s'ouvre aux Départements dans le domaine du social et des personnes âgées
- ...



- **Un "Back Office" agent plus complet**

- Un gestionnaire de tâches qui apporte une vue de synthèse des demandes à traiter
- Un nouveau moteur de recherche plus puissant avec une représentation de type liste comme sur le web
- Un espace centralisé pour le traitement des demandes apportant une vision plus globale du dossier
- La possibilité de créer des annotations internes ou à destination de l'utilisateur pour améliorer le traitement des demandes
- Une gestion des contacts avec le citoyen améliorée
- ...



- **Une administration simplifiée de la plate-forme pour chaque collectivité**

- Des fonctions en ligne permettant de fournir l'ensemble des éléments graphiques du site

- Une meilleure gestion de l'organisation et des utilisateurs
- Des fonctions de gestion des courriers types avec un éditeur simplifié
- ...



## **Les services mis en oeuvre**

Le système CapDémat est une plate-forme de télé-services, qui offre un bouquet de services en ligne accessible par le citoyen.

Les services déjà proposés sont:

### **ETAT CIVIL**

- Demande d'extrait d'acte de mariage, décès, naissance
- Inscription sur listes électorales
- Recensement militaire

### **ENVIRONNEMENT**

- Enlèvement des encombrants
- Enlèvement des déchets verts

### **CULTUREL**

- Inscription de la bibliothèque
- Achat de place de spectacle

### **URBANISME / SERVICE TECHNIQUE**

- Demande d'ouverture de tranchée pour raccordement sur l'égout public
- Demande d'un certificat d'alignement
- Demande d'intervention technique

### **SCOLAIRES / PERISCOLAIRES / RESTAURATION**

- Inscription annuelle
- Inscription activité pré-post scolaire
- Relevé d'activité
- Paiement en ligne
- Bourse mobile étude

### **SECURITE**

- Inscription tranquillité vacances

### **SERVICES TECHNIQUES**

- Demande d'intervention technique

### **SOCIAL**

- Téléassistance
- Aide ménagère

20 autres téléservices sont en préparation

## **Les nouveaux services disponibles**

La plate-forme CapDémat, propose également de nouveaux télé-services dédiés aux Départements dans les domaines des aides aux personnes âgées , du social et du scolaire qui concernent les :

- ☞ Demande de repas et d'aide ménagère
- ☞ Demande de compensation du handicap adulte
- ☞ Demande de compensation du handicap enfant
- ☞ Demande de bourse
- ☞ Demande d'APA
- ☞ ...

### **Les technologies employées**

Le Département du Val d'Oise a mis en œuvre CapWebCT avec des technologies Open Source dans le respect des recommandations de la DGME, de l'ARTESI, de la CNIL et celle en matière d'accessibilité aux malvoyants.

Les technologies utilisées :

- Linux
- Apache
- Tomcat
- PostgreSql
- Ldap
- Java/J2EE
- Hibernate/Spring
- ...

### **Les contacts**

Département du Val d'Oise  
2 av du Parc 95032 Cergy-Pontoise Cedex

Bruno PERRIN

Directeur des Systèmes d'Information

tél: 01 34 25 32 70

fax: 01 34 25 32 79

E-mail: [bruno.perrin@valdoise.fr](mailto:bruno.perrin@valdoise.fr)

Philippe USCLADE

Chef de projet NTIC

tél: 01 34 25 33 35

fax: 01 34 25 32 79

E-mail: [philippe.usclade@valdoise.fr](mailto:philippe.usclade@valdoise.fr)

Liens Utiles : Site open source de l'offre CapWebCT <http://www.capwebct.fr>

## Description fonctionnelle et technique d'Angelus



- UN ANNUAIRE DES USAGERS DE LA COLLECTIVITÉ •
- UNE SOLUTION GÉNÉRIQUE ET LIBRE •

Bien souvent, dans nos collectivités, les usagers doivent justifier de leur identité et de leur domicile à chaque service auquel ils s'inscrivent. Il en résulte de multiples doublons au sein des différentes applications métiers.

La solution consiste à mettre en place un référentiel commun et transversal. Les données mises en commun concernent :

- l'identité :
  - nom patronymique,
  - nom d'usage,
  - prénom(s),
  - date de naissance (pour les contrôles d'unicité) ;
- le domicile :
  - numéro,
  - complément de numéro,
  - type de voie,
  - libellé de voie,
  - complément d'adresse,
  - code postal,
  - commune.

D'autres informations sont stockées. Elles concernent des aspects techniques du fonctionnement de l'annuaire lui-même :

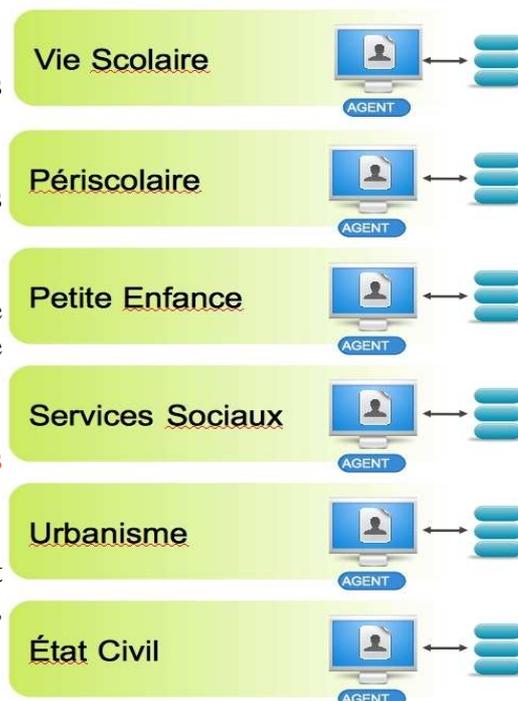
- référence application métier,
- identifiant dans l'application métier,
- dates de création ou modifications,
- opérateurs étant intervenus,
- ...

Parmi les applications concernées, il y a également les applications transversales comme la gestion financière en ce qui concerne la gestion des tiers qui, pour être efficace, doit être intégrée au dispositif de façon à mettre en cohérence les données des redevables avec les applications métiers.

Nous vous présentons ci-après les grands principes de fonctionnement de ce référentiel baptisé « Angelus » (ANnuaire GENéralisé Libre des Usagers).

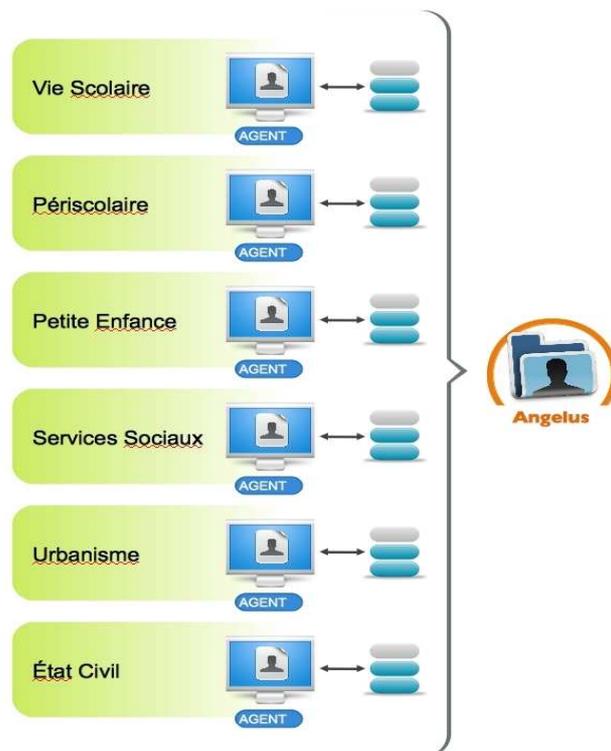
## Le point de départ : une problématique récurrente...

- Les services d'une collectivité ont chacun leurs applications «métier»
- Ces applications gèrent des usagers, au sens «citoyens en relation avec la ville»
- Exemples: abonnements bibliothèque, service périscolaire, petite enfance, facturation école de musique, etc.
- **Il existe autant de bases de données utilisateurs que d'applications !**
- Ainsi, parmi les inconvénients, un usager doit présenter à chaque fois des justificatifs d'identité, d'un guichet à l'autre.



## L'objectif : la solution Angelus

- Créer un système centralisé pour les données principales des usagers
- Les partager au mieux avec les différentes applications métier en place, pour éviter les multiples saisies et les données incohérentes
- **Prévoir un déploiement progressif** n'imposant pas une refonte complète du Système d'Information de la ville
- Prévoir des mécanismes pour gérer le problème des doublons d'usagers, une même personne existant plusieurs fois dans des bases de données



## Que gère Angelus ?

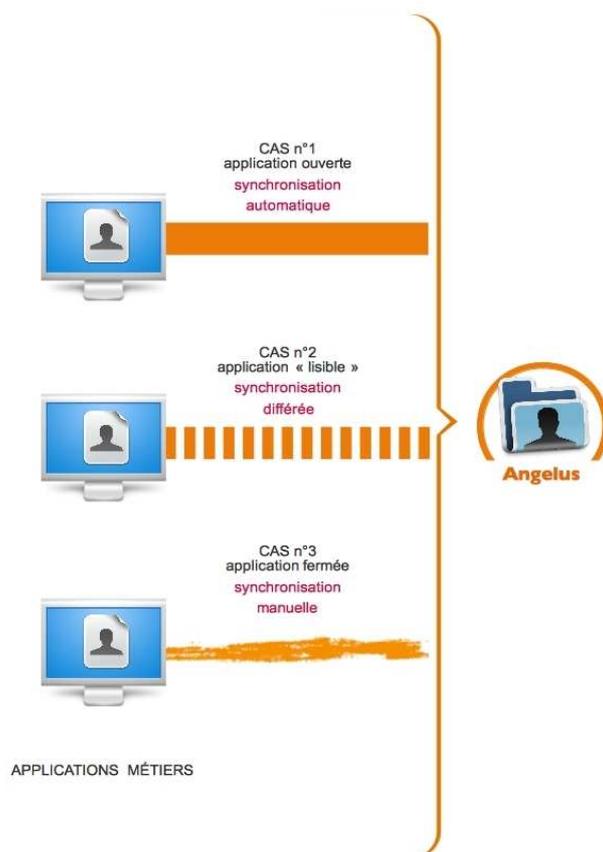
- Informations stockées pour chaque usager dans Angelus:
  - état civil (nom, coordonnées)
  - numérisation de justificatifs d'identité et de domicile
- Les autres informations sont considérées comme «métier» et laissées à chaque application
  - exemple: relations familiales, quotient familial, inscriptions, factures, etc.
- Conservation de l'historique des changements sur chaque usager
- Les applications métier doivent continuer à fonctionner même en cas de coupure d'Angelus



## Les liens entre les applications métier et Angelus

Angelus prévoit plusieurs niveaux d'intégration:

- cas «idéal»  
l'application métier et Angelus échangent en direct les informations, via l'API d'Angelus
- cas «différé»  
synchronisation périodique de la base des usagers de l'application métier avec Angelus, par scripts
- cas «manuel»  
pas d'interfaçage automatique, l'agent utilise en parallèle Angelus et son application métier et fonctionne par comparaison et copier/coller



## Vers une simplification de la relation avec le citoyen



## Exemple d'écran Angelus

Angelus - Annuaire générique et libre des usagers admin | déconnexion

Info usager **Outils de nettoyage** Applications tierces Administration

**Comparaison**  
Doublons potentiels

Fiches récentes :

GALOP Roger

MOORE Roger

**Comparer**

	Roger GALOP #69		Roger MOORE #77
ID unique	#69		#77
Type	Particulier		Particulier
Décédé ?	Non		Non
Civilité	M.		M.
Sexe	Homme		Homme
Nom de naissance	GALOP		MOORE
Nom d'usage			
Prénom 1	Roger		Roger
Prénom 2			
Prénom 3			
Date de naissance	15/12/2001	Personnes différentes	01/01/1960
Adresse		Doublons potentiels	2 RUE DES FRERES LUMIERE 87280 Limoges
Tél.			
Tél. portable			
E-mail			
Tél. professionnel			
Numéro de poste			
Fax			
E-mail professionnel			
PCS			Professions libérales et assimilés
Profession en clair			Intermittant du spectacle

Cette fenêtre montre comment l'administrateur de la base centrale Angelus peut comparer deux enregistrements

**Les contacts : Ville de Limoges**

Alexandre Garcia  
Ingénieur principal  
DGST Antenne informatique Bfm et Portail des écoles  
E-mail: [agarcia@bm-limoges.fr](mailto:agarcia@bm-limoges.fr)

Claude Boissou  
Chef de Service DGST Informatique  
Tel : 05 55 45 61 74  
E-mail: [claud.boissou@ville-limoges.fr](mailto:claud.boissou@ville-limoges.fr)

Site de la forge Angelus <http://www.addulact.fr>

## Description fonctionnelle et technique de Sematic

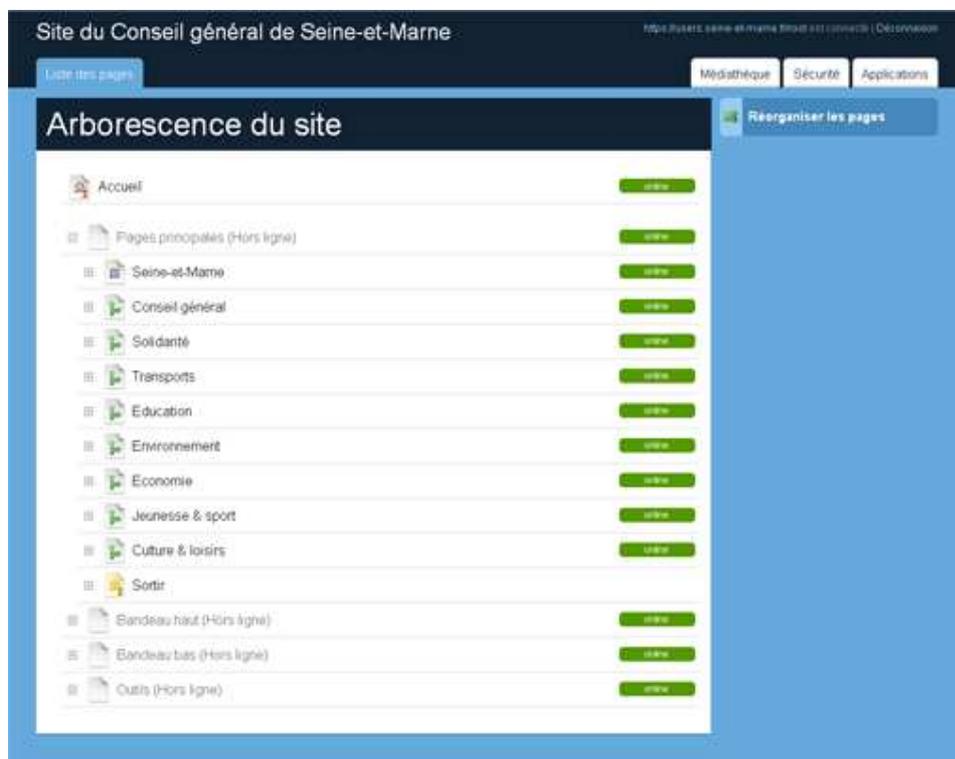
### Présentation

Le Département de Seine-et-Marne, avec la maîtrise d'œuvre de la Direction de l'innovation et de l'e-administration (DIE), a décidé de remplacer son gestionnaire de contenu Internet vieillissant par un nouvel outil, adapté à ses besoins. Après un tour d'horizon des produits disponibles sur le marché, il est ressorti que les systèmes de gestion de contenus sont souvent de complexité fonctionnelle élevée : ils sont lourds à utiliser et de stabilité relative. Le Département a conçu Sematic dans une optique de simplicité et de facilité d'utilisation, tout en garantissant des performances optimales.

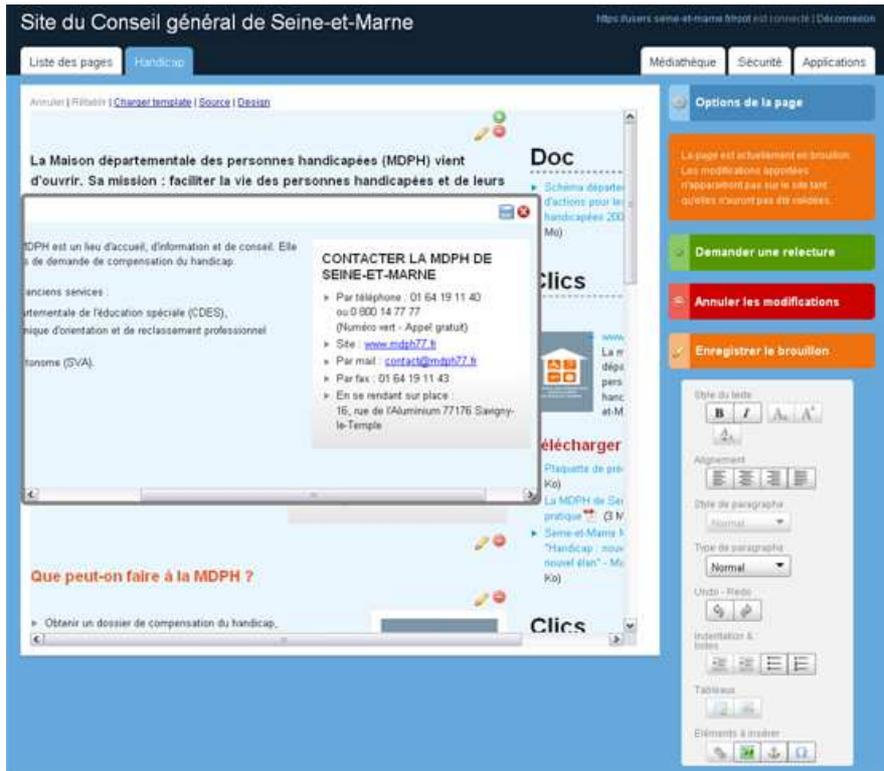
L'objectif de ce projet était d'apporter une solution simple à une problématique fonctionnelle complexe :

- Evolution constante des besoins fonctionnels.
- Grande simplicité d'utilisation induisant de faibles coûts pour les formations utilisateur.
- Grande réactivité et rapidité pour la création de nouveaux sites.
- Accessibilité du back office aux non-voyants.
- Faibles coûts de maintenance évolutive et corrective.

La plate-forme Sematic est aujourd'hui publiée sous licence Open Source (GPL version 3.0) dans une forge logicielle. Véritable site concentrant le savoir et l'expertise autour de cette plate-forme ouverte, ce site se veut aussi une véritable plate-forme d'échange de code source où collectivités, entreprises et particuliers peuvent récupérer et reverser de nouveaux modules suivant le principe des licences Open Source.



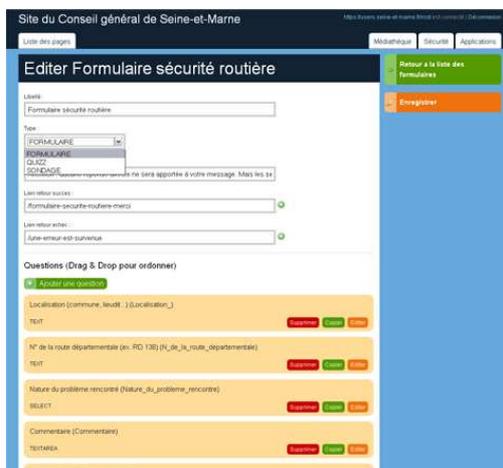
Arborescence du site



*Edition d'une page de contenu*

## Les modules disponibles

- Module d'actualité
- Module d'agenda
- Module de formulaire (formulaire classique, quizz, sondage)
- Module d'offres d'emploi
- Module de gestion des MAPA
- Module de recherche des délibérations
- Module d'albums photo et vidéo
- Visionneuse multimédia
- Annuaire géo localisés



*Edition d'un formulaire*



*Module « offres d'emploi »*

## Les technologies employées

Sematic est basé sur des technologies Open Source :

- Linux
- Apache
- PostgreSQL
- Java/J2EE
- Play
- Swarm (serveur Openid)
- ...

## **Les contacts**

Département de Seine-et Marne  
Rue des St Pères  
77000 Melun Cedex

Christine BERTRAND

Directrice de l'innovation et de l'e-administration

tél: 01 64 14 72 32

fax: 01 64 14 75 59

E-mail: [christine.bertrand@cg77.fr](mailto:christine.bertrand@cg77.fr)

Sébastien ROUSSEAU

Directeur adjoint de l'innovation et de l'e-administration

tél: 01 64 14 72 95

fax: 01 64 14 75 59

E-mail: [sebastien.rousseau@cg77.fr](mailto:sebastien.rousseau@cg77.fr)

## **Liens Utiles**

Site de la forge Sematic

<http://www.sematic.fr>

## **Annexe 3 : Règles de développement applicables à toutes contributions**

### **Respect du CRC – qualité du code source produit**

Le partenaire contributeur s'engage à ce que les livrables faits à la Communauté soient parfaitement en phase (C=R=C)

- Conçu : fonctionnel, techniques, infrastructure (sous format de documents formalisés et validés...)
- Réalisé : Code source qui traduit les réalisations de ce qui est conçu, spécifié et réalisé (code source, exécutables, script de build et de déploiement...)
- Communiqué : les documents communiqués à la Communauté doivent correspondre parfaitement à ce qui est conçu et réalisé.

Le partenaire contributeur s'engage à exiger de ses développeurs ou de ses sous-traitants un respect scrupuleux des règles de développement et des contraintes architecturales énoncées ci-après.

En cas de non-respect de ces règles, Le partenaire aura l'obligation de reprise du code, tests et mise en ordre de marche avec VA et VSR, avant toute nouvelle soumission à acceptation du code par la Communauté et intégration dans la forge commune.

Contrôle : tout livrable ayant eu un impact sur le code source (modification, suppression et ajout), pour être recevable, devra comporter le code source modifié, supprimé et ajouté et celui-ci devra être repérable très facilement .

### **Normes et outils de développement**

Les préconisations énoncées dans ce chapitre visent à fédérer et organiser les développements pour améliorer la qualité logicielle. Dans un premier temps, sont décrits un jeu de recommandations permettant d'appliquer les principes fondamentaux d'architecture. La même démarche sera appliquée pour le code.

### **Conception Objet**

Le logiciel, objet des développements (nouveau module, nouvelles fonctionnalités ou modification de fonctionnalité, upgrade de composant, etc....) doit respecter les principes fondamentaux de la conception objet que sont l'encapsulation et la mise en œuvre judicieuse de l'héritage et du polymorphisme. L'encapsulation est respectée à partir du moment où il n'est pas possible d'accéder directement à une propriété d'un objet autrement que par l'intermédiaire de ses méthodes d'accès. L'utilisation de l'héritage et du polymorphisme doit être conforme aux modèles de conception (« design patterns ») identifiés et reconnus par les praticiens de la communauté objet.

Le respect de ces principes fondamentaux ne suffit pas à garantir une bonne conception, parce qu'ils ne sont pas suffisamment explicites. Plus précisément, ils sont nécessaires mais pas suffisants. Ils ne permettent pas à eux seuls de se garantir contre la dégénérescence de l'application, qui se manifeste généralement par :

- La rigidité du logiciel. Le logiciel est difficile à modifier, car un changement entraîne des modifications en cascade dans les modules dépendants. Ce qui a un impact sur de nombreux composants de l'application et a pour conséquence une augmentation des coûts de développement et de maintenance.
- La fragilité du logiciel. La probabilité qu'une modification entraîne l'apparition d'erreurs dans une autre partie du logiciel est élevée.
- L'immobilité du logiciel. L'extraction d'une partie de l'application en vue d'une réutilisation est difficile.

Pour contrôler les risques énoncés ci-dessus, il faut contrôler rigoureusement les dépendances entre les modules de l'application en mettant en œuvre les principes de conception éprouvés (1) suivants :

- **Principe d'ouverture/fermeture** : un module doit être ouvert aux extensions mais fermé aux modifications. En d'autres termes, on doit pouvoir étendre les fonctionnalités d'un module (ouvert aux extensions), sans modifier le code existant (fermé aux modifications). En Java, l'utilisation d'interfaces exprimant l'abstraction est la clé pour la mise en œuvre de ce principe. Ce principe doit être appliqué à bon escient pour éviter une augmentation trop importante de la complexité. Généralement, il est utilisé aux points de variations identifiés dans le plan d'urbanisation ou dans les modèles de classes (anticipation des changements).
- **Principe de substitution de Liskov** : les méthodes utilisant des objets d'une classe, doivent pouvoir utiliser des objets dérivés de celle-ci sans modification. Autrement dit, une sous-classe doit respecter le contrat de sa super-classe.
- **Principe d'inversion des dépendances** : les modules de haut niveau ne doivent pas dépendre de modules de bas niveau. Tous deux doivent dépendre d'abstractions, en pratique les modules de haut niveau utilisent des interfaces mises en œuvre par les modules de bas niveau.
- **Principe de séparation des interfaces** : les clients ne doivent pas être forcés de dépendre d'interfaces qu'ils n'utilisent pas. Le respect de ce principe a pour conséquence la création de petites interfaces, regroupant de façon cohérente des méthodes fonctionnellement connexes.
- **Principe d'équivalence livraison/réutilisation** : la granularité en termes de réutilisation est le paquetage. Seuls les paquetages livrés sont susceptibles d'être réutilisés.
- **Principe de réutilisation commune** : la réutilisation d'une classe d'un paquetage revient à réutiliser le paquetage entier.
- **Principe de fermeture commune** : les classes impactées par une même modification doivent être placées dans un même paquetage.
- **Principe de dépendances acycliques** : les dépendances entre paquetages doivent former un graphe acyclique.

(1) **Agile Software Development, Principles, Patterns, and Practices** - Robert C Martin - Addison Wesley - 2002.

## Conventions de codage

Une convention de codage (coding convention) est établie par outil. Ces conventions de codage seront vérifiées de façon automatique et systématique par l'environnement de développement. Toutes violations significatives entraîneront un rejet du code lors des phases de recette.

## Sémantique de nommage

La sémantique de nommage des identificateurs devra rester homogène à l'intérieur d'une unité de travail. Par exemple, il n'est pas souhaitable d'utiliser simultanément les termes « livre », « book » et « ensemble de pages » pour parler d'un livre dans un même composant.

## JavaBeans

La conception des objets doit respecter les idiomes communément considérés comme de bonnes pratiques pour la programmation orientée objet Java. Plus précisément, les conventions JavaBean devront être suivies.

## JSP/Présentation

Les pages JSP et autres systèmes de rendus ne doivent pas contenir de code réalisant des fonctionnalités métiers. C'est le travail des « contrôleurs » que de préparer les données à destination des « vues ». L'utilisation de « taglibs » (bibliothèques de tags JSP) est fortement recommandée.

## Qualité du code

### Simplicité et lisibilité

Il est demandé de produire un code source simple et lisible. La première étape étant de respecter les normes de codage et de nommage. Le code doit être découpé de façon à être facilement lisible et maintenable :

- Découpage en paquets en respectant les principes de couplage exposés dans la première partie de cette étude.
- Principe de séparation des préoccupations (SOC) appliqué au sein des objets. Une méthode doit généralement traiter d'un aspect à la fois. La complexité d'une méthode doit rester faible et maîtrisée.
- Tout « code mort » est à proscrire.
- Tout « code bloat » est à proscrire.

## Commentaires et Javadoc

Les commentaires permettant de générer la documentation Java (Javadoc) doivent être utilisés judicieusement :

- Les interfaces de service doivent être commentées de façon exhaustive.
- Les accesseurs ne doivent être commentés qu'en cas de besoin. Par exemple:

```
/**
 * Set the name
 * @param name the name to set
 */
```

```
public void setName (String name) {
    ...
}
```

est un commentaire qui n'apporte que peu de valeur ajoutées. A l'inverse, commenter le comportement effectif aux cas limites apporte de la valeur.

- Les commentaires doivent être synchrones avec l'état du code source. Un commentaire se rapportant à un élément qui n'existe pas ou plus sous sa forme originelle n'est d'aucune utilité et contribue à alourdir d'autant le code.

Bien souvent, il est préférable de refactorer un code anormalement complexe que de le commenter pour en expliquer sa complexité.

## Complexité du code

Il est demandé un code source maintenable. Par code source maintenable, il faut comprendre code pour lequel les métriques définies dans la partie théorique restent dans les intervalles définis.

## Refactoring

Durant la vie d'un logiciel, le code source est amené à évoluer avec le temps :

- Correction d'anomalies.
- Fonctionnalités maquette.
- Fonctionnalités abandonnées.
- Évolutions fonctionnelles.
- Algorithmes implémentés « à la va vite ».
- Ajout sauvage de fonctionnalités.
- etc

Le refactoring consiste à retravailler le code pour l'améliorer et le perfectionner. Par exemple, il est souhaitable de ré-écrire des parties du logiciel pour :

- Diminuer la complexité d'une classe ou d'une méthode.
- Rationaliser les paquets, les noms...
- Simplifier un algorithme.
- Optimiser un code.
- Modifier sa présentation et sa lisibilité.
- Refondre la conception pour suivre les besoins induits par le métier.
- Supprimer du « code mort » (qui n'est plus utilisé).
- Factoriser du code dupliqué.
- Améliorer sa « testabilité ».

Les phases de refactoring sont **indispensables** pour produire un logiciel de qualité.

## Déclenchement du refactoring

Les opérations de refactoring peuvent être suggérées par :

- Les équipes du partenaire.
- Le prestataire du partenaire
- Le résultat des outils d'assurance qualité.

Au démarrage du projet, des outils d'assurance qualité mesurant les critères de qualité logicielle (architecture, état du code, etc) décrits dans la première partie de ce document devront être mis en oeuvre.

## **Self-Building Code**

Toute unité de code (projet, etc) doit être auto-constructible. C'est à dire qu'il doit disposer d'un script automatisant la création des artefacts suivants :

- Code compilé.
- Archive de déploiement (fichier .jar pour les bibliothèques, fichier .war, etc).
- Documentation (Javadoc).
- Tests.

## **Self-Testing Code**

La qualité souhaitée du code source ainsi que le respect de l'architecture (verticale, horizontale) imposent de concevoir un code modulaire. Chaque ensemble de code doit être testable unitairement. De plus, chaque service doit aussi disposer d'un ensemble de tests démontrant que ce dernier est bien à même d'exécuter le contrat de service.

Ces tests ont un deuxième objectif : ce sont aussi des exemples concrets de l'utilisation qu'un développeur peut faire d'un code livré, en complément des Javadocs et autres documents.

## **Documentation**

Chaque service doit disposer des documents suivants :

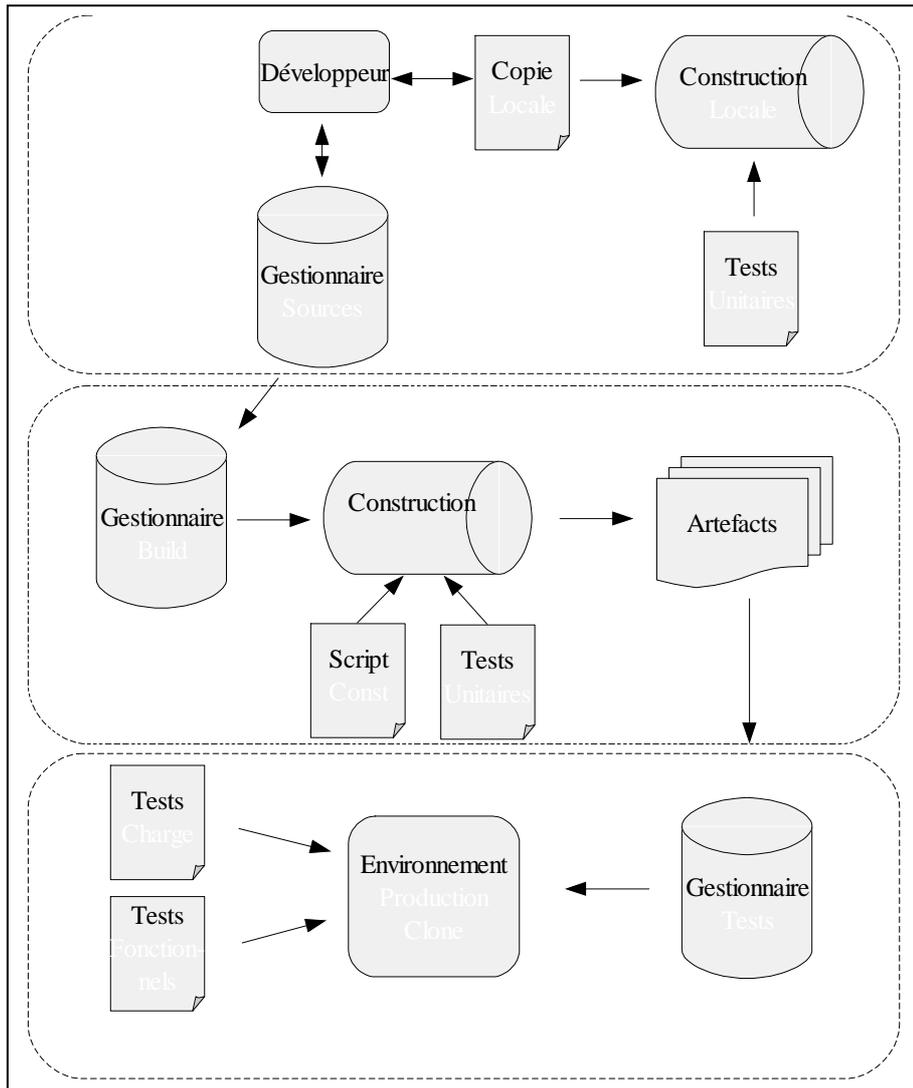
- Un diagramme de classes UML du modèle objet.
- Un document d'architecture.
- Un document décrivant les services et leurs contrats.
- Un document de présentation générale.
- Un document explicitant les flux et interactions avec les autres éléments constitutifs de la plate-forme ou les services externes.

Une documentation au format Wiki est demandée. Elle n'a pas forcément besoin d'être volumineuse pour être efficace.

## Pratiques et Outils

### Intégration continue

L'intégration continue apporte une augmentation de la qualité logicielle et permet de réduire et de mieux prédire les coûts afférents à l'intégration des modifications apportées au code. Il s'agit de mettre en place un ensemble de pratiques et d'outils permettant de construire et tester automatiquement les modifications apportées au code source. Les principes proposés ici sont ceux formalisés par Martin Fowler [1], référence en la matière, et ceux issus d'un ensemble de pratiques communément mises en œuvre et reconnues.



Architecture d'une plateforme d'intégration continue

## Gestionnaire de sources

Le gestionnaire de sources est le premier maillon de la chaîne d'intégration continue. Il doit :

- Être centralisé. Il n'existe qu'un seul dépôt des sources. Il est accessible facilement (protocole ouvert et disponible à tous les acteurs nécessaires).
- Accessible à tous. Tous les acteurs qui travaillent sur le projet ont au minimum un accès en lecture au dépôt. Tous les développeurs ont un accès en lecture-écriture.
- Être traçable. Il doit être possible de tracer, commit après commit (opération d'écriture sur le dépôt), chacune des modifications. La notion de « changeset », opération de modification atomique sur n ressources gérées par le dépôt, est indispensable car elle permet d'associer une unité de travail (ticket : évolution, bug report, etc.) à un jeu de modifications. Le dépôt ne doit pas être destructif et garder les traces des ressources supprimées.
- Gérer de multiples versions / branches. Au cours du cycle de vie d'un logiciel, un développeur ou une équipe peut être amenée à appliquer beaucoup de modifications « tests » qui peuvent rendre la branche principale relativement instable. Dans ce cas, le dépôt devra permettre de créer une branche autonome supportant ce type de modifications, puis proposer des capacités de « merge » (fusion) de ces branches au sein de la branche principale.
- Être non intrusif. Tout dépôt de source imposant une structure ou un mode de fonctionnement particulier à ce dépôt est à proscrire. CVS, par exemple, est à proscrire car il impose de conserver un ensemble de répertoires vides après des opérations de refactoring, sous peine de perdre l'historique des modifications des répertoires.

Le dépôt des sources est un outil de travail collaboratif entre les développeurs qui travaillent sur le projet. **Chaque contributeur (commiter, développeur qui enregistre une opération de modification des sources) est responsable des impacts de ses modifications** sur le projet. Il doit s'assurer de la consistance de ses modifications avec le reste des modifications apportées par les autres intervenants.

Un intervenant ne respectant pas ses règles élémentaires de « bienséance » pourra voir ses modifications annulées jusqu'à ce qu'il s'assure de la bonne intégration de ses travaux.

De plus, chaque opération de modifications (commit) devra être accompagnée d'un commentaire expliquant les modifications ou fournissant un lien vers le ticket correspondant.

## Gestionnaire de projets

Le gestionnaire de projets, du point de vue développeur, est un outil permettant de gérer le cycle de vie logiciel du projet. On y trouve notamment :

- Un gestionnaire de tickets. Le ticket est une unité de travail. Il peut s'agir d'une demande d'évolution, d'une demande de correction d'un dysfonctionnement, ou d'une tâche, etc. Le périmètre d'un ticket doit être clair, précis et très limité. De plus, comme il s'agit d'un outil orienté équipe de développement, il ne devra pas contenir d'autres types de demandes. Chaque ticket devra être qualifié (demande d'évolution, anomalie, etc) et lié à une étape du projet (voir les jalons ci-dessous). Enfin, les tickets doivent pouvoir être associés à une personne de l'équipe.
- Un gestionnaire de jalons (milestones). Un jalon est une étape importante d'un projet. Le gestionnaire de projet doit les connaître. Un jalon est un ensemble de tickets associés à une

date limite prévue d'exécution. Un jalon n'est pas forcément associée à une release, il peut s'agir d'une étape intermédiaire.

- Un outil de traçabilité du code source. Cet outil permet de visualiser les changesets appliqués au code source. Il permet aussi, lorsqu'un changeset est sélectionné, de présenter une différence (diff) du code source entre les versions que le changeset a impacté. De plus, il est souhaitable de pouvoir associer un ticket à un changeset et inversement.
- Un outil de documentation pragmatique. L'approche la plus pragmatique à ce jour est un Wiki. Il va permettre une édition rapide et collaborative de la documentation. Étant donné que le logiciel évolue, la documentation doit aussi évoluer. Un Wiki permet généralement de tracer et versionner les pages qui le composent. Le Wiki permet aussi à des acteurs « externes » d'enrichir la documentation et d'en faire bénéficier la communauté.

### **Gestionnaire de vie (Forge)**

La partie visible d'un projet Open Source est sa forge. Il s'agit de coupler des pages informatives et les artefacts de construction (téléchargements) au gestionnaire de projets. Ce rôle peut d'ailleurs être dévolu au gestionnaire de projets.

Chaque outil de l'annexe 1 à sa propre forge.

### **Gestionnaire de constructions automatiques (builds )**

La plate-forme de construction est architecturée autour d'un programme chargé des opérations suivantes :

- Récupération (checkout) des sources. Le gestionnaire de constructions est connecté au dépôt de sources. Lorsque les sources sont modifiées, à intervalles réguliers ou à la demande, le gestionnaire de constructions démarre un environnement de construction et y dépose les dernières sources disponibles.
- Construction. Le gestionnaire de constructions va reposer sur la capacité de self-building du code pour déclencher son exécution : compilation, génération des Javadocs, packaging, etc. Les résultantes de cette opérations sont appelés « artefacts ».
- Rapport et présentation des artefacts. Finalement, le gestionnaire présente sous forme d'un rapport synthétique les constructions réussies et échouées.

Il est aussi souhaitable, étant donné la modularité visée du projet CapWebCT, que ce gestionnaire soit aussi capable de gérer les dépendances entre modules.

Avec ce système, l'équipe de développement est libérée d'une partie du travail de construction. Les constructions au fil de l'eau sont communément appelées « Nightly Builds ». Lorsque les phases de tests et recette s'achèvent sur une branche, il suffit de marquer la version courante des sources comme release, puis de relancer un processus de construction pour obtenir de façon simple et transparente les paquets « release ».

### **Exécution de selfs-tests automatiques (Unitaire, Style, Architecture, etc...)**

Si le code est conçu en intégrant de nombreux tests unitaires, ayant servi de base aux développements (« self-testing code »), il est alors possible d'automatiser la phase de tests unitaires. Il peut s'agir d'un programme dédié, qui va récupérer les artefacts du gestionnaire de constructions à intervalles réguliers, puis déployer et exécuter ces tests. Il est aussi possible d'intégrer directement ces tests en tant que phase de construction, au sein du gestionnaire de tests.

D'autres tests automatiques, non unitaires, sont souhaitables. Par exemple, on peut réaliser des tests de conformité des codes sources à une norme de codage (coding convention), de qualité intrinsèque du code, ou encore de respect de l'architecture établie.

### **Gestionnaire de tests fonctionnels (« in situ »)**

Il s'agit de décharger non seulement l'équipe de développement mais aussi l'équipe de recette d'une partie du travail répétitif de validation du fonctionnement global. En effet, si les tests unitaires et qualitatifs sont adaptés pour tester le fonctionnement du coeur métier, ils ne sont en aucun cas capables de tester les interfaces homme-machine présentées aux utilisateurs finaux. De plus, ils peuvent difficilement détecter les problèmes d'intégration des composants (services, projets) entre eux. Des tests « grandeur nature », in situ, sont donc nécessaires.

Un test « in situ » consiste à installer les artefacts produits par le gestionnaire de constructions au sein d'un environnement le plus proche possible de l'environnement de production cible. Le gestionnaire de tests in-situ est donc composé des fonctionnalités suivantes :

- Environnement d'exécution. L'environnement d'exécution test est une réplique au plus proche de l'environnement cible : système d'exploitation, logiciels installés, configuration, etc. Ce système doit être réinitialisable à volonté (machine virtuelle, scripts de nettoyage sur un serveur, etc).
- Configuration automatique. Le gestionnaire de tests in-situ se substitue à un administrateur système. Il installe les artefacts dans l'environnement d'exécution hôte, configure les artefacts, démarre les programmes et vérifie leur bon démarrage.
- Scénario et vérifications. Une fois l'environnement installé, le gestionnaire démarre une batterie de tests déroulant les scénarios fonctionnels standards et vérifie, au sein des écrans utilisateur, la présence d'éléments ou de résultats attendus.

Les scénarios de tests fonctionnels peuvent coûter relativement cher à concevoir, implémenter et maintenir en fonction de la granularité souhaitée. S'ils épargnent et automatisent une partie du travail de validation et de recette, ils ne se substituent pas au travail d'une personne.

### **Fin des annexes**