

L'Adoption SOA

POUR
LES NULS[®]

Edition Spéciale Software AG

**Une référence
pour nous
tous !**

eTips gratuits sur dummies.com

**Miko Matsumura
Bjoern Brauel
Jignesh Shah**

**Découvrez
comment convertir
votre entreprise à
l'architecture SOA**



À PROPOS DE SOFTWARE AG

Nos 4,000 clients dans le monde utilisent nos solutions logicielles pour améliorer leurs processus métier et piloter une infrastructure informatique souple. Leur objectif est de réduire leurs coûts et de gagner en flexibilité et en efficacité. Nous les y aidons en optimisant et en gouvernant leurs opérations et en alignant leur informatique sur les objectifs de leur entreprise.

Notre portefeuille de logiciels d'infrastructure métier favorise l'intégration et la modernisation des systèmes et des données. Il permet de bénéficier d'une souplesse incomparable grâce à l'architecture SOA, et de créer rapidement de nouveaux processus métier avec le BPM.

Nos quarante années de succès garantissent à nos clients une plate-forme fiable pour obtenir plus vite de meilleurs résultats.

Software AG - *Get There Faster*

L'Adoption
SOA
POUR
LES NULS®

ÉDITION SPÉCIALE SOFTWARE AG

**Par Miko Matsumura, Bjoern Brauel
et Jignesh Shah**



WILEY

Wiley Publishing, Inc.

L'Adoption SOA pour les Nuls®, Édition spéciale Software AG

Publié par

Wiley Publishing, Inc.

111 River Street Hoboken, NJ 07030-5774, États-Unis

Copyright © 2009 Wiley Publishing, Inc., Indianapolis, Indiana Publié par Wiley Publishing, Inc., Indianapolis, Indiana

Aucun extrait de cette publication ne peut être reproduit, stocké dans une base de données ni transmis, sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, photocopie, enregistrement, numérisation ou autre), sauf aux conditions autorisées aux alinéas 107 et 108 du United States Copyright Act de 1976, en l'absence d'autorisation écrite préalable de l'Éditeur. Les demandes d'autorisation doivent être adressées par courrier à Permissions Dept., 111 River St., Hoboken, NJ, 07030, USA, (201) 748-6011, fax (201) 748-6008, ou en ligne sur www.wiley.com/go/permissions.

Marques : Wiley, le logo Wiley Publishing, pour les Nuls, le logo Dummies Man, et les appellations commerciales afférentes sont des marques ou des marques déposées de John Wiley & Sons, Inc. et/ou de ses sociétés affiliées aux États-Unis et dans d'autres pays, dont l'utilisation est interdite en l'absence d'autorisation écrite. Software AG et le logo Software AG sont des marques ou des marques déposées de Software AG, Inc. aux États-Unis et dans d'autres pays. Toutes les autres marques citées sont la propriété de leurs détenteurs respectifs. Wiley Publishing, Inc. n'est lié à aucun des produits ou fournisseurs cités dans cet ouvrage.

LIMITATION DE RESPONSABILITÉ/DÉNI DE GARANTIE : L'ÉDITEUR ET L'AUTEUR S'ABSTIENNENT DE TOUTE DÉCLARATION OU GARANTIE S'AGISSANT DE L'EXACTITUDE OU DE L'EXHAUSTIVITÉ DU CONTENU DE CET OUVRAGE, ET REJETTENT EN PARTICULIER TOUTE GARANTIE, Y COMPRIS, NON LIMITATIVEMENT, TOUTE GARANTIE D'APTITUDE À UN USAGE PARTICULIER. AUCUNE GARANTIE NE PEUT ÊTRE CONSENTIE OU ÉTENDUE AU TITRE D'UN DOCUMENT COMMERCIAL OU PROMOTIONNEL. LES CONSEILS ET STRATÉGIES PRÉSENTÉS ICI RISQUENT DE NE PAS CONVENIR À TOUTES LES SITUATIONS. CET OUVRAGE EST COMMERCIALISÉ, SACHANT QUE L'ÉDITEUR NE DISPENSE AUCUN SERVICE JURIDIQUE, COMPTABLE OU PROFESSIONNEL AUTRE. SI UNE ASSISTANCE PROFESSIONNELLE EST REQUISE, LES SERVICES D'UN PROFESSIONNEL COMPÉTENT DEVRONT ÊTRE SOLlicitÉS. NI L'ÉDITEUR, NI L'AUTEUR NE POURRONT ÊTRE TENUS RESPONSABLES DES DOMMAGES DÉCOULANT DES PRÉSENTES. SI UN ÉTABLISSEMENT OU SITE WEB EST RÉFÉRENCÉ DANS UNE CITATION ET/OU COMME SOURCE POTENTIELLE D'INFORMATIONS COMPLÉMENTAIRES DANS CET OUVRAGE, CELA NE SIGNIFIE AUCUNEMENT QUE L'AUTEUR OU L'ÉDITEUR AVAISE LES INFORMATIONS SUSCEPTIBLES D'ÊTRE COMMUNIQUÉES PAR CET ÉTABLISSEMENT OU CE SITE WEB OU SES RECOMMANDATIONS. PAR AILLEURS, LE LECTEUR DOIT AVOIR CONSCIENCE QUE LES SITES WEB CITÉS DANS CET OUVRAGE PEUVENT AVOIR ÉVOLUÉ OU DISPARU ENTRE LE MOMENT OÙ CE LIVRE A ÉTÉ ÉCRIT ET CELUI OÙ IL EST LU.

Pour toute information d'ordre général sur nos autres produits et services, veuillez contacter notre service clientèle par téléphone au 877-762-2974 (depuis les États-Unis) ou au +1-317-572-3993 (en dehors des États-Unis), ou par télécopie au +1-317-572-4002.

ISBN : 978-0-470-59358-5

Fabrique aux États-Unis

10 9 8 7 6 5 4 3 2 1



À propos des auteurs

Miko Matsumura est vice-président et directeur technique adjoint de Software AG. Il a été président fondateur du Comité Technique Oasis « SOA Adoption Blueprints » et organisateur de l'initiative SOA Link Interoperability. Il intervient régulièrement dans le monde entier sur les questions ayant trait à l'architecture SOA, et s'exprime également sur ce sujet dans des blogs à l'adresse www.SOACenter.com.

Avant le rachat d'Infravio, Inc. par webMethods, Miko Matsumura a officié en qualité de vice-président du marketing et des standards technologiques chez Infravio, où il a piloté les opérations marketing et la planification stratégique. Il s'est distingué comme leader d'opinion au sein de The Middleware Company où il a co-créé le programme de partenariat pour les « SOA Blueprints », première spécification véritablement indépendante d'un ensemble d'applications SOA, défendue par BEA, Borland, HP, Microsoft, Oracle, Sun Microsystems et Veritas, entre autres. Chez Systinet, Miko Matsumura a coopéré avec l'équipe dirigeante et le centre de développement offshore sur le développement et la stratégie produits ainsi que le marketing traditionnel, y compris en représentant la société lors de manifestations. Chez Sun Microsystems, en sa qualité d'expert Java en chef, il a été désigné porte-parole pour les technologies Java et a servi la communauté des développeurs en travaillant en étroite collaboration avec les éditeurs de logiciels indépendants et licenciés Java. Miko Matsumura est titulaire d'une maîtrise en neurosciences délivrée par l'Université de Yale et d'un MBA décerné par SFSU (San Francisco State University).

Bjoern Brauel est vice-président et directeur technique adjoint de Software AG, premier fournisseur mondial indépendant de logiciels d'infrastructure métier.

À ce poste, il expose la stratégie SOA de Software AG dans des conférences et sur des salons professionnels ainsi que dans le cadre de manifestations clients et de séminaires aux quatre coins du monde. À l'écoute des clients, communautés et analystes, Bjoern Brauel n'a pas son pareil pour aiguiller les orientations technologiques, concrétiser les projets des clients et cerner leurs besoins informatiques futurs. Disposant d'une solide formation technologique, il a travaillé dans la recherche et le développement (R&D) ainsi qu'en avant-ventes ; il a également occupé des postes stratégiques dans les domaines marketing et produits. Cet intervenant est réputé pour sa capacité à véhiculer des messages à caractère technologique auprès d'un public de professionnels, tout en étant précis et

objectif. Au cours des cinq dernières années, il s'est surtout focalisé sur les Architectures Orientées Services, la gestion des processus métier (BPM) et les technologies d'intégration articulées autour du langage XML et des services Web.

Au moment de rejoindre Software AG, il avait déjà plus de dix années d'expérience au sein de la communauté Open Source.

Jignesh Shah est vice-président en charge du marketing et de la gestion des produits SOA chez Software AG. Il accompagne des clients du monde entier dans la mise en œuvre de leurs projets SOA. Avant de rejoindre Software AG, Jignesh Shah comptait parmi les membres fondateurs d'OpsPlanner, solution d'administration des situations d'urgence en mode SaaS. Auparavant, il était architecte en solutions chez BearingPoint. Il a piloté la conception et la mise en œuvre de plusieurs solutions informatiques de grande envergure pour des clients du classement Fortune 500 présents dans divers secteurs d'activité (technologies de pointe, biens de consommation, industrie pharmaceutique, santé, industrie et administration).

Dédicaces

Miko Matsumura : À Lis et à Jackson, qui est aussi passionné d'astronautique que moi.

Bjoern Brauel : À mon père qui a su faire de moi un libre penseur.

Jignesh Shah : À Aarti, pour sa présence à mes côtés, comme amie et guide. Et à Maanav : vois-tu, papa adore les jeux de construction à base de briques, tout comme toi !

Remerciements

Les auteurs tiennent à remercier Jim Fowler pour l'éclairage apporté sur le rôle de l'architecture SOA dans la modernisation d'un patrimoine applicatif existant.

Toute leur gratitude va à Claas Wallrodt pour le formidable travail de synthèse accompli.

Et une ovation pour Jim Bole et Garry Clarkson, deux « astronautes de la SOA » hors pair, sans oublier Ivo Totev et Kevin Iaquinto, qui ont brillé par leurs encouragements et leur soutien sans faille à cet ouvrage et à sa mission.

Sommaire



Introduction1

À propos de ce livre.....	1
Icônes utilisées	2

Chapitre 1 : Créer une entreprise agile3

Comprendre le concept de la SOA.....	3
Analyse d'un service	3
Décryptage de l'architecture.....	4
SOA, au cœur de l'entreprise	4
Cromprendre le plan directeur SOA.....	5
Décryptage du plan directeur SOA.....	6
Lecture d'un plan d'organisation.....	7
Réalisation du plan directeur, projet par projet	8

Chapitre 2 : Obstacle à la mission : le bric-à-brac informatique9

Définir la notion de bric-à-brac	9
Comprendre la prolifération anarchique des systèmes d'information.....	10
L'asphyxie par les strates.....	10
L'exclusion par les silos.....	11
La paralysie par l'effet spaghetti	12
Endiguer l'anarchie organisationnelle en informatique	12
Examen des éléments qui mènent au bric-à-brac.....	13
Les guerres tribales au sein des structures informatiques	15
Ranger le bric-à-brac avec la gouvernance SOA.....	15
Intégrer gouvernance système et gouvernance organisationnelle	16

Chapitre 3 : Réalisation du plan d'architecture SOA . . 17

Convenir de politiques et procédures	17
Le centre de compétence SOA	18
Automatiser l'application des politiques et procédures	18
Politiques et procédures	19
Politiques et procédures de conception.....	19

Politiques et procédures d'exécution	20
Instaurer des points de contrôle vérifiant la mise en application des politiques	22
Chapitre 4 : Infrastructure de services	23
Comprendre l'activation de services	23
Stratégie de superposition à l'existant	24
Bus de services d'entreprise et activation de services	25
Encapsuleurs d'applications	27
Comprendre la médiation de services	28
Réaliser la virtualisation de services.....	29
Couplage faible.....	29
Indépendance de l'emplacement.....	30
Indépendance du transport	30
Indépendance du message.....	30
Spécifications opérationnelles.....	31
Bus de services d'entreprise et médiation de services	32
Intermédiaires/passerelles de services	32
Appliances SOA.....	32
Chapitre 5 : Infrastructure de gouvernance	35
Manipuler le registre/référentiel.....	35
Gestion des politiques	37
Utilisation du registre/référentiel comme point de contrôle de conception	38
Comprendre le cycle de vie	39
Faire appel à la gestion d'exécution	41
« Embarquer » les consommateurs de services	43
Boucler la boucle	44
Chapitre 6 : Applications composites	45
Comprendre les applications composites	46
Utiliser la gestion des processus métier (BPM).....	46
Développer des applications composites.....	47
Chapitre 7 : Agilité organisationnelle	49
Combattre les guerres tribales.....	50
Vivre le cycle de vie SOA	51
Identifier vos cycles de vie SOA.....	52
Définition des intervenants	52
Exécution des validations.....	54
Instauration des contrats	54
Gérer l'évolution SOA.....	55

Analyser un exemple de structure informatique	56
Ressentiment au sein d'un pôle informatique centralisé	57
Tensions entre la division A et le pôle informatique centralisé.....	58
Défiance entre la division A et la division B.....	59
Tensions entre tribus au cours du cycle de service.....	59

Chapitre 8 : Qui finance la SOA ?61

Comment financer votre SOA.....	61
Approche tactique.....	62
Réflexion stratégique	63
Démarche pratique : le BPM.....	64
Proposer des incitations organisationnelles.....	65

Chapitre 9 : Votre premier projet SOA67

Lancer un projet SOA	67
Choix des premiers services	68
Choix des alliés SOA.....	69
Conserver sa trajectoire	69
Mesure de la conformité informatique	69
Mesure du retour sur investissement.....	70
Automatiser les règles et procédures	71
Démarche graduelle	71
À quel moment instaurer l'infrastructure de gouvernance.....	71

Chapitre 10 : La fusée SOA73

Examiner la fusée SOA.....	73
Du projet SOA au programme SOA.....	73
La phase critique de la SOA	75
Orienter la fusée dans la bonne direction	75
Force d'accélération et indicateurs de plus-value informatique.....	75
Force d'accélération et indicateurs de plus-value métier	76
Systèmes de guidage organisationnel.....	77
Systèmes de guidage architectural	78
Motivation de vos collaborateurs	78
Mettre la fusée SOA en orbite.....	81
Votre destination.....	81
Autre destination possible	82

Chapitre 11 : En route vers les étoiles83

Analyser la phase critique	83
Erreurs SOA	83
La longue trajectoire de mise en orbite	84
Tirer parti de la mise en orbite	85
Vers l'infini et au-delà... ..	86

Introduction

SOA, acronyme de *Service Oriented Architecture*, désigne une *architecture orientée services*. Celle-ci permet aux architectes d'entreprise de créer des *plans directeurs* SOA (en anglais, « blueprints ») qui régissent la transformation des systèmes d'information et des organisations. La réalisation de ces plans suppose la mise en œuvre d'un processus baptisé « adoption de la SOA ».

Le présent ouvrage expose notre approche de l'adoption de la SOA, inspirée de l'*astronautique (science de la navigation spatiale)*. À l'instar d'une fusée, l'adoption de la SOA connaît une phase critique entre le décollage et la mise en orbite. Au terme de sa trajectoire, elle se révèle à même de transformer votre activité. Mais, tant que cet objectif n'est pas atteint, vous risquez de voir vos rêves d'architecture SOA se briser.

Pour franchir la phase critique dans la mise en œuvre de la SOA, observez ces quelques principes fondamentaux :

- ✓ Orientez la fusée SOA dans la bonne direction, en mesurant votre progression et en procédant à des corrections de trajectoire au fur et à mesure.
- ✓ Maintenez la force de poussée en motivant les équipes et les acteurs de votre adoption de la SOA.
- ✓ Ne relâchez pas vos efforts jusqu'à la mise en orbite, en automatisant les processus jusqu'à ce que l'implémentation de la SOA devienne une seconde nature, déconcertante de facilité.

Cet ouvrage a pour objectif de mener à bien votre adoption de la SOA en surmontant sa phase critique.

À propos de ce livre

Ce livre n'est pas un ouvrage sur l'architecture. De nombreux écrits consacrés à l'architecture SOA existent déjà sur le marché. Cet ouvrage traite de l'*adoption* de la SOA, autrement dit de méthodes concrètes et pratiques pour donner corps à un projet SOA.

L'Adoption SOA pour les Nuls met en exergue les aspects essentiels d'une architecture SOA et explique comment ne pas les perdre de vue. Vous trouverez toutes les informations se rapportant à un thème donné au paragraphe qui lui est consacré dans ce livre. Si vous êtes néophyte dans le domaine, nous ne saurions trop vous recommander de lire cet ouvrage dans son intégralité. En revanche, si l'architecture SOA n'a aucun secret pour vous, passez directement au chapitre qui vous intéresse. À supposer, par exemple, que le financement de votre programme SOA soit à l'ordre du jour, inutile de parcourir la totalité de ce livre pour accéder aux précisions requises ; il vous suffit de consulter le chapitre consacré à ce thème (Chapitre 8) pour obtenir toutes informations utiles.

À propos, pour prononcer SOA, dissociez chacune des lettres de cet acronyme : S-O-A.

Icônes utilisées

Les icônes ci-dessous sont utilisées pour mettre en évidence certaines informations tout au long de ce livre :



Ces précieux conseils facilitent votre adoption de la SOA. Conformez-vous aux informations figurant dans ces paragraphes pour que vos efforts soient payés en retour.



« Attention » signale les pièges fréquemment associés à l'adoption SOA.



« Note technique » apporte d'intéressants éclaircissements techniques ; le contenu est ici un peu plus détaillé que dans le reste de l'ouvrage. Si le temps vous manque, ignorez ces paragraphes : vous aurez tout le loisir d'y revenir ultérieurement.



« Ne pas oublier » met en évidence des informations importantes, à retenir absolument.

Chapitre 1

Créer une entreprise agile

Dans ce chapitre

- Analyse du concept SOA.
- Résolution des problèmes de l'entreprise grâce à l'architecture SOA.
- Examen du plan directeur SOA.

L'architecture orientée services (SOA) est une vision structurée du monde, qui passe par la création d'un *plan directeur SOA*.

Mais, pour atteindre cet objectif il faut bien plus qu'une simple vision et qu'un plan directeur. Dans ce chapitre, nous appliquons les principes de l'architecture SOA aux problèmes métier et exposons un mode d'adoption pragmatique de votre plan directeur SOA, consistant à mettre en œuvre un projet à la fois.

Comprendre le concept de la SOA

L'architecture SOA est une vision du monde.

Dans une *vue orientée services*, chaque composant se présente comme un service. Brique élémentaire de l'architecture SOA, le *service* permet d'accéder à des fonctionnalités métier réutilisables.

Analyse d'un service

Un service SOA se définit, au minimum, par :

- ✓ **Sa finalité.** Un service est une action exécutée au profit d'un *consommateur*. Il peut s'agir, par exemple, du traitement d'un changement d'adresse pour un client d'un établissement bancaire.
- ✓ **Ses modalités d'utilisation.** L'utilisation d'un service met en œuvre une méthode spécifique appelée *invocation*. L'accès aux caractéristiques du service s'effectue via une *interface* bien définie.

Ne sont pas explicitement définis dans un service SOA :

- ✓ **Sa localisation.** Le service est *activable à distance*, ce qui signifie qu'il est invocable depuis tout point du réseau.
- ✓ **Son fonctionnement.** L'*opacité* du service est telle que personne ne sait comment celui-ci s'acquitte de sa mission.

Les services SOA peuvent être couplés pour créer d'autres services, et assemblés en séquences pour constituer des processus.

Décryptage de l'architecture

Si les services sont les briques de l'architecture SOA, alors celle-ci s'apparente au coffret Lego Star Wars du Millenium Falcon, édition Ultimate Collector (vaisseau comptant plus de 5 000 pièces fourni avec toutes les figurines, y compris Chewbacca). Autant dire qu'elle a plus d'une brique dans son sac !

L'architecture SOA définit :

- ✓ La localisation d'un service.
- ✓ L'interopérabilité entre les différents services.
- ✓ L'intégration de chaque service au sein de ce système de services.

Avec un jeu de construction, vous identifiez les briques dans la boîte, les assemblez au moyen des petits ergots, et les intégrez à une réalisation d'ensemble suivant le schéma de montage fourni.

Au sein de l'architecture SOA, vous identifiez les services dans un catalogue appelé *annuaire de services*, les imbriquez au moyen d'*applications composites*, et montez l'ensemble suivant un *plan directeur SOA*.

SOA, au cœur de l'entreprise

Si l'architecture SOA n'était utile qu'aux fanas d'informatique, elle ne présenterait que peu d'intérêt. En réalité, cette architecture performante exprime son potentiel technique en termes métier, en donnant aux entreprises les moyens de traduire sa force en solutions nouvelles.

Si vous fréquentez des architectes d'entreprise, vous les entendrez sans doute invoquer des expressions aussi obscures que *couplage faible* et *granularité large*. Décryptons ce jargon SOA et précisons l'importance de ces notions pour l'entreprise.

- ✓ La **granularité** décrit la taille des composants d'un système. L'architecture SOA préfère les composants volumineux (à macro granularité large), ou services métier. Ils reposent généralement sur des services techniques plus compacts (à micro granularité fine) préexistants.

Cette notion est importante, car des composants volumineux facilitent la compréhension, la réutilisation et la gestion des services SOA par les professionnels métier (responsables fonctionnel ou opérationnel).

- ✓ L'**interface vs implémentation** distingue la *finalité* du service mis en œuvre de la *façon* dont il est exécuté : ce que le service fait de la manière dont il le fait.

Notion importante qui simplifie la vision de l'utilisateur métier sur l'architecture SOA : la finalité du service est privilégiée par rapport à ses rouages technologiques.

- ✓ Les **contrats** définissent les obligations entre le *fournisseur* et le *consommateur* du service. Ils peuvent formaliser certaines attentes (disponibilité, fiabilité, indicateurs clés de performance, coût et support).

Notion importante contribuant à l'objectivité des décisions prises par les utilisateurs métier s'agissant de la fiabilité des services.

- ✓ Le **couplage faible** est une technique permettant de concevoir des services plus souples et moins dépendants les uns des autres. Il favorise la réutilisation d'un service sans devoir reprendre les services qui lui sont liés.

Notion importante car il est effectivement plus rapide d'assembler des solutions métier à partir de briques préfabriquées que de créer entièrement chacune d'elles.

Cromprendre le plan directeur SOA

Cet ouvrage traite de l'adoption SOA pour les maîtres d'œuvre SOA, et non de la conception SOA pour les architectes SOA. Pour autant, les maîtres d'œuvre SOA doivent être au fait de ce que recouvre un plan directeur et savoir le lire.

Voici ce que vous devez retenir sur les plans directeurs SOA :

- ✓ Ils présentent l'objectif réalisé en totalité.
- ✓ Ils sont réajustés en permanence.

Tout au long de votre adoption SOA, veillez à réorienter en permanence votre « fusée » de sorte qu'elle ne s'écarte pas de sa trajectoire – mais si des corrections sont apportées au plan directeur, préparez-vous à réviser votre objectif ! En effet, à chacune des étapes ponctuant votre démarche SOA, faites le point sur vos succès et vos échecs. Or, ces nouvelles informations ne vous seront profitables qu'à condition de rectifier votre plan directeur pour en tenir compte.

Décryptage du plan directeur SOA

Le plan directeur SOA doit spécifier la *cible*. Concrètement, il doit présenter une image très complète de ce à quoi ressemblera l'implémentation SOA une fois celle-ci menée à bien. Sur ce plan directeur doivent en principe figurer les éléments suivants :

- ✓ Services métier
- ✓ Description des services
- ✓ Indicateurs de performance des services
- ✓ Standards d'interopérabilité
- ✓ Schémas de données
- ✓ Règles
- ✓ Découverte de services et classification

Ces notions seront plus explicites à mesure que vous progresserez dans la lecture de cet ouvrage.

Vous devez également prendre en compte ce qui suit :

- ✓ **Schéma d'infrastructure SOA** : Une cartographie de l'ensemble des composants SOA (logiciels et matériels) requis. Ces composants sont détaillés aux Chapitres 4 à 6.
- ✓ **Plan de développement** : Une feuille de route détaillée pour la réalisation complète du plan directeur qui est, en règle générale, appelée à évoluer.
- ✓ **Plan d'organisation** : Le plan d'organisation esquisse la forme de l'organisation SOA finale. Il fait l'objet d'une analyse plus détaillée au paragraphe suivant.

Lecture d'un plan d'organisation

Tout comme le plan d'architecture facilite la refonte de votre système d'information, le plan d'organisation simplifie celle de votre organisation informatique. La méthode SOA inspirée de l'astronautique attache autant d'importance à l'une qu'à l'autre. Un plan d'organisation s'articulera autour des questions suivantes :

- ✓ **Évaluation des compétences** : Avez-vous les connaissances SOA nécessaires pour réussir ?
- ✓ **Structure organisationnelle** : Comment optimiser la traçabilité entre fournisseurs et consommateurs ?
- ✓ **Instance dirigeante** : Qui définit les règles et procédures mises en œuvre pour l'adoption de la SOA ? Quels sont les groupes qui se doivent d'être représentés par cette instance ?
- ✓ **Incitations comportementales** : Comment les évaluations, rémunérations et promotions professionnelles servent-elles la réalisation des objectifs de la SOA ?
- ✓ **Rôles et responsabilités** : Comment ajuster les fonctions, descriptifs de poste et responsabilités professionnelles à l'architecture SOA ?
- ✓ **Modèle de financement applicable à une infrastructure mutualisée (refacturation interne et fiscalité)** : Qui paie pour un service donné et pour les changements apportés à ce service ?
- ✓ **Indicateurs partagés** : Quelles seront les mesures effectuées pour évaluer le statut pris en compte pour documenter votre architecture SOA et guider l'entreprise ?
- ✓ **Cycle de vie** : Quelles sont les étapes indispensables à la conception, au déploiement, à la maintenance et au retrait de services ?
- ✓ **Plan de développement organisationnel** : Comment évoluer vers un plan d'organisation étape par étape ?



Même s'il est souhaitable de promouvoir votre *plan directeur* SOA autour de vous, n'oubliez pas que votre *plan d'organisation* doit par contre être manié avec précaution car il est susceptible de contenir des informations sur les postes et rôles de vos collaborateurs.

Réalisation du plan directeur, projet par projet

Notre approche SOA, inspirée de l'astronautique, réalise les plans d'architecture et d'organisation projet par projet. Pour plus d'informations sur cette approche, voir le Chapitre 10.



Evitez toute approche de type « big bang » qui consisterait à réaliser l'intégralité du plan directeur SOA dans le cadre d'un projet gigantesque et onéreux, susceptible de s'éterniser. Sélectionnez et enchaînez plutôt des projets SOA de petite envergure, procurant chacun des avantages quantifiables.

Chaque projet doit dégager un retour sur investissement, mais aussi jouer un rôle « moteur » pour les projets futurs. À mesure que vous implémentez des projets SOA, peaufinez et automatisez les processus correspondants jusqu'à la « mise en orbite ».

Chapitre 2

Obstacle à la mission : le bric-à-brac informatique

Dans ce chapitre

- Définir la notion de bric-à-brac.
- Comprendre la prolifération anarchique des systèmes d'information.
- Endiguer l'anarchie organisationnelle en informatique.
- Ranger le bric-à-brac SOA.

La formule « plan directeur » sous-entend que vous avez toute latitude pour créer de A à Z une nouvelle architecture SOA. C'est oublier les structures et systèmes qui se dressent en travers de votre chemin. Il serait certes tentant de faire table rase de l'existant. Mais puisque ces systèmes sont toujours en service, leur mise au rebut ne saurait être envisagée. En d'autres termes, des bâtiments sont déjà érigés sur vos terres – et des collaborateurs y résident : il va falloir faire avec.

Les urbanistes qualifient d'*anarchique* une urbanisation désordonnée et effrénée. Ce chapitre est consacré à la prolifération anarchique du parc et de l'organisation d'une structure informatique, laquelle ne fait que s'accroître au fil du temps, et au rôle joué par la gouvernance SOA pour inverser cette tendance.

Définir la notion de bric-à-brac

Imaginez des systèmes empilés et juxtaposés à l'intérieur de silos inaccessibles. Le tout sous un enchevêtrement de spaghettis.

Imaginez des structures entortillées comme de la barbe à papa à force d'expansion géographique, de fusions-acquisitions, de regroupement centralisé et d'externalisation organisationnelle.

Imaginez des antagonismes politiques et des luttes pour le pouvoir, des ressentiments et de l'hostilité. Imaginez un pôle informatique ne cessant de décevoir les décideurs par ses fiascos

et ses retards, jouant avec leurs nerfs du fait des obligations de mise en conformité et des contraintes d'infrastructure.

Inutile de forcer le trait. Ce tableau est monnaie courante dans la plupart des entreprises.



Si nous évoquons ces problèmes informatiques, c'est en partie parce que l'architecture SOA s'efforce de les résoudre, mais avant tout parce qu'ils représentent l'obstacle le plus marquant à son adoption. Ce casse-tête ne pourra être résolu tant que vous ne cernerez pas parfaitement les systèmes et structures informatiques en place.

Comprendre la prolifération anarchique des systèmes d'information

La prolifération anarchique d'un système d'information revêt trois formes distinctes :

- ✓ **Strates** : Couches de systèmes d'information, empilées au fil du temps.
- ✓ **Silos** : Systèmes redondants isolés, mutuellement inaccessibles.
- ✓ **Effet spaghetti** : Intégrations point-à-point désordonnées.

Chacune d'elles est détaillée dans les paragraphes suivants :

L'asphyxie par les strates

Le système d'information se compose d'une multiplicité de couches informatiques, parfois hétéroclites : applications personnalisées, grands systèmes (« mainframes »), applications client-serveur et progiciels de gestion intégrés, sans oublier des systèmes plus modernes de type serveurs d'applications Java.

La superposition de couches informatiques à l'infini n'est pas sans conséquences :

- ✓ Toute transformation s'annonce fastidieuse et risquée.
- ✓ Personne n'est capable de maîtriser la totalité des systèmes.
- ✓ La logique n'est pas clairement découpée par couche.
- ✓ La maintenance des systèmes est onéreuse.
- ✓ Les systèmes différents ne communiquent pas forcément entre eux.

À ces systèmes correspondent des langages de programmation, des caractéristiques de performances et des conceptions architecturales distincts. Du mainframe au Web en passant par l'architecture client-serveur ou à trois niveaux (« 3 tiers »), l'ensemble forme un patchwork complexe.

La SOA vient en aide en exposant sous forme de services métier les vieilles fonctions préexistantes, piégées dans ces couches. Pour être certain que ces nouveaux services fonctionnent en parfaite harmonie avec les anciens, la SOA réorganise l'élaboration des projets au sein du cycle de vie de l'informatique interne.



Certains sont ennuyés du fait que l'architecture SOA vient rajouter une couche supplémentaire au-dessus de toutes les autres. Effectivement, qui voudrait d'une couche en plus ? Sachez pourtant que cette approche facilite considérablement le regroupement et la rationalisation des systèmes sous-jacents.

L'exclusion par les silos

Autre caractéristique des systèmes informatiques existants : la prédominance de *silos*. Les silos sont des systèmes de données intégrés verticalement ; autant dire que leur interopérabilité est exclue. Leur redondance est généralement mise en évidence à l'occasion de fusions et d'acquisitions. Ils sont également monnaie courante dès lors que les différentes divisions d'une entreprise se voient confiées la responsabilité de leur propre budget informatique sans qu'on se soucie du gaspillage que l'absence de mutualisation des efforts peut engendrer.

Les silos sont souvent le produit d'un comportement organisationnel. Par exemple, chacune des divisions de l'entreprise peut posséder sa propre base de données clients. Dans une telle organisation, modifier des données clients suppose de déterminer au préalable leur emplacement et celui de la logique applicative, et d'identifier les propriétaires de l'une et de l'autre.

La SOA permet de gérer le problème posé par les silos en créant des accords d'interopérabilité qui harmonisent les modes de communication entre les systèmes et les formats de données utilisés, et qui lèvent les obstacles organisationnels à la coopération. Si parvenir techniquement à l'interopérabilité des systèmes et des données s'avère complexe, le premier frein à l'adoption de la SOA est la passation d'accords entre les différentes structures qu'elles conviennent de partager. Cet aspect sera détaillé au Chapitre 3.

La paralysie par l'effet spaghetti

L'histoire de l'informatique correspond à un enchevêtrement d'intégrations, d'applications et de processus point à point formant un *effet spaghetti*, aux interdépendances multiples. Le problème des interdépendances est particulièrement pernicieux en informatique car il risque de provoquer des pannes en cascade en cas de défaillance de systèmes interdépendants.

Imaginez, à votre domicile, une installation électrique dont vous ne verriez que les fils dénudés sortant des murs, des plafonds et du plancher. Chaque fois que vous souhaiteriez allumer une ampoule ou écouter la radio, il faudrait trouver les bons fils électriques et faire des essais de branchement jusqu'à trouver la combinaison adéquate. Les dangers d'un système de ce type sont évidents.

Avec un tel entrelacs de fils et de câbles, le fournisseur d'électricité ne peut ni quantifier ni *a fortiori* facturer avec exactitude votre consommation électrique – sans compter qu'avec une installation de ce type, votre famille et vous-même n'êtes pas à l'abri d'une électrocution ! Une fausse manipulation, et c'est votre chambre, votre maison, vos voisins, voire le quartier tout entier, qui risquent de se retrouver dans le noir. D'ailleurs, si vous êtes à l'origine d'une panne de courant, faisons le pari que vous serez incapable de repérer le branchement défaillant !

Aussi effarant que cela puisse paraître, cette situation est loin d'être inhabituelle au sein des structures informatiques actuelles. Tout simplement parce que l'histoire de l'informatique est celle d'une longue chaîne de projets égocentriques. Chacun d'eux n'ayant d'autre ambition que de recueillir des données de la manière la plus économique et la plus rapide qui soit, il en résulte une architecture médiocre, inélégante et dégradée.

En présence d'un effet spaghetti, la SOA crée un système uniformisé et ordonné applicable à la recherche, la connexion et l'utilisation de services informatiques ; ce faisant, les projets disposent des fonctionnalités requises sans qu'il soit nécessaire de recourir à du « bricolage », qui ne ferait qu'aggraver la situation.

Endiquer l'anarchie organisationnelle en informatique

Comme l'amibe, la structure informatique prolifère et se restructure sous l'effet de contraintes diverses. Elle s'étend et

se divise, se multipliant dans des environnements distribués, spécialisés ou géographiques créés pour mieux se replier, périodiquement, sur une forme de contrôle centralisé.

Examen des éléments qui mènent au bric-à-brac

Les organisations informatiques se sont développées comme les systèmes matériels et les logiciels – organiquement. L'adoption de la SOA nécessite donc tout autant de surmonter la « confusion » organisationnelle que de maîtriser les systèmes technologiques. Passons en revue les éléments qui mènent à l'anarchie organisationnelle en informatique et les difficultés créées en termes d'adoption de la SOA.

✔ **Découpage par fonction** : À mesure que les structures informatiques gagnent en maturité, des spécialités viennent s'ajouter au *cycle de développement des systèmes*. Ce cycle s'apparente à une chaîne de montage dédiée à la création de nouveaux logiciels ou services. Des équipes différentes seront chargées de la conception, de la programmation, du déploiement, de l'assistance, de la maintenance et de l'évolution des systèmes.

✔ **Découpage par plate-forme** : Les structures informatiques sont généralement divisées en équipes, par éditeur ou plate-forme. Il s'ensuit une sorte de « guerre tribale ».

Les développeurs Java ne fréquentent pas les développeurs Microsoft .NET. Les spécialistes SAP ne portent pas dans leur cœur les spécialistes Oracle. Tout serait parfait si l'entreprise adoptait une plate-forme unique (et congédiait tous les autres collaborateurs). Et encore ! Une fusion-acquisition risquerait d'amener de nouvelles tribus adoratrices d'autres éditeurs.

Principale difficulté en terme d'adoption de la SOA : donner à ces multiples tribus les moyens de coexister via la mise en œuvre d'accords d'interopérabilité.

✔ **Découpage par patrimoine applicatif** : Le *patrimoine applicatif existant*, tel que les mainframes, peut être assimilé à une autre catégorie de plate-forme. Mais il mérite une mention particulière car les équipes qui en assurent le support appartiennent, en règle générale, à une ancienne génération de responsables informatiques.

Principale difficulté en terme d'adoption de la SOA : rentabiliser au maximum ces patrimoines applicatifs existants

sans vous départir des compétences organisationnelles acquises lors du départ en retraite de vos collaborateurs.

- ✓ **Découpage par secteur géographique** : Lorsqu'une entité s'implante sur un nouveau territoire, de nouveaux centres de données voient le jour. Pour réduire leurs coûts, les structures informatiques misent sur l'externalisation « offshore » ou bien sur les pays à faible coût de main-d'œuvre informatique ou à jeux de compétences spécialisés.

Principale difficulté en terme d'adoption de la SOA : coordonner des équipes géographiquement dispersées et les faire travailler sur différentes parties d'un même plan directeur de la SOA.

- ✓ **Découpage par fusions-acquisitions** : Lorsqu'une entreprise prend le contrôle d'une autre, dans le cadre d'un rachat par exemple, elle récupère généralement une structure informatique opérationnelle à 100 %, exploitant tout un ensemble de progiciels et plates-formes (y compris ceux et celles qui risquent de ne pas être du goût de l'acheteur).

Principale difficulté en terme d'adoption de la SOA : continuer à assurer un support aux utilisateurs des systèmes en place et combattre l'isolationnisme et l'antagonisme entre groupes tout en évoluant vers une architecture SOA dont les services mutualisés contribuent à réduire les redondances et à améliorer l'agilité.

- ✓ **Invasion des intégrateurs systèmes** : À mesure que les structures informatiques se développent, les sous-traitants assument un nombre croissant de fonctions.

Principale difficulté en terme d'adoption de la SOA : garder la main. Ces groupes ont tout intérêt à vous rendre de plus en plus dépendants à l'égard de leurs consultants.

- ✓ **Découpage par division** : Nombre d'entreprises et d'organisations sont segmentées en filiales, agences, administrations, départements, services ou divisions. Cette segmentation s'opère suivant les fonctions métier. En règle générale, chaque division possède sa propre structure informatique.

Principale difficulté en terme d'adoption de la SOA : les divisions se disputent les financements et n'apprécient guère de partager les ressources. De même, elles détestent être corsetées par des règles et des services « taille unique ». Elles sont coutumières des jeux politiques, laissant à autrui le soin de financer l'infrastructure tandis qu'elles en récoltent les bénéfices.

✓ **Centralisation** : Pour gérer leur croissance, les structures informatiques ont une technique : miser sur la centralisation. La rentabilité les y pousse, au même titre que le désir d'uniformisation. Les divisions, par exemple, n'ont pas toutes besoin ou ne peuvent pas toutes s'offrir les services de spécialistes en architecture de sécurité, ou en architecture d'entreprise.

Principale difficulté en terme d'adoption de la SOA : parvenir à un juste équilibre entre le coût de l'informatique centralisée et la maîtrise conférée par celle-ci, d'une part, et la liberté et la souplesse de l'informatique distribuée, d'autre part. Pour atteindre cet objectif, la SOA recourt à une structure appelée *fédération*.

Les guerres tribales au sein des structures informatiques

Ces multiples découpages au sein de la structure informatique d'une entreprise donnent naissance à des « tribus ». Chaque tribu représente un éditeur, un secteur géographique, une division, un cabinet de consultants ou toute autre entité symboliquement distinctive au sein de la structure informatique.

Chacune de ces tribus a pour ambition de réussir en imposant sa domination aux autres, un penchant naturel qui est aussi une triste réalité au sein des structures informatiques de grande envergure. Faute de comprendre et de dompter les élans organisationnels qui créent et entretiennent silos, strates et effet spaghetti, votre approche technologique, quelle qu'elle soit, sera vouée à l'échec. Cette notion de tribu sera détaillée au Chapitre 7.

Ranger le bric-à-brac avec la gouvernance SOA

Le terme *gouvernance* évoque généralement une bureaucratie étouffante et des règles aveugles émanant d'en haut, qui annihile créativité et souplesse. Il est bien sûr *possible* de mettre en œuvre votre gouvernance SOA sur un mode dictatorial et anti créatif mais vous vous heurteriez à une résistance à l'encontre de l'adoption de la SOA, qui aurait tôt fait d'anéantir les progrès déjà accomplis.



L'approche de la SOA, inspirée de l'astronautique, que nous recommandons en matière de gouvernance s'attache à définir et à mettre en œuvre des accords entre tribus informatiques, et à calculer et corriger votre trajectoire au fur et à mesure.

Aux antipodes d'une gouvernance de type « big bang » qui imposerait tout un ensemble de règles nouvelles, tel Moïse et ses Dix Commandements, nous préférons ajouter et appliquer de nouvelles règles au coup par coup, sans jamais cesser de recalculer la trajectoire et de la rectifier au besoin.



Gouvernance et agilité ne sont pas antinomiques. C'est l'absence de gouvernance, c'est-à-dire l'*anarchie*, qui s'oppose à l'agilité. La gouvernance SOA est un ensemble de pratiques recommandées contribuant à lutter contre l'anarchie. Une urbanisation pleine de bon sens contribuera dans une large mesure au bon fonctionnement de la « ville » SOA.

Intégrer gouvernance système et gouvernance organisationnelle

Le plan d'architecture SOA lutte contre la prolifération anarchique des systèmes d'information tandis que le plan d'organisation combat l'anarchie organisationnelle de la structure informatique. Leurs trajectoires sont-elles pour autant divergentes ?

Certainement pas ! La prolifération anarchique d'un système d'information et l'anarchie organisationnelle de sa structure sont interdépendantes. Une tribu informatique chaotique engendre des systèmes informatiques chaotiques. Une tribu vouée au culte d'un éditeur défend les systèmes propriétaires de ce dernier et entretient des relations incestueuses avec lui. Une tribu liée à une division de l'entreprise protège les applications en silos. Une tribu d'ancienne génération s'arc-boute sur des systèmes mal connus pour sauvegarder ses emplois. Une tribu de consultants externes cultive des relations de dépendance afin de mieux vous asservir à sa horde.

Les informaticiens sont loin d'être stupides. Si c'est l'impression que vous donnent leurs systèmes, réfléchissez-y à deux fois. Il est évident qu'une décision stupide dont votre système d'information aurait fait les frais a été profitable à quelqu'un – le plus souvent aux dépens du reste de l'entreprise. Vous ne parviendrez à éradiquer la prolifération anarchique des systèmes informatiques qu'en triomphant des forces organisationnelles à l'origine de celle-ci.

Même si le prochain chapitre se focalise sur la réalisation du plan d'architecture SOA comme moyen de remédier à la prolifération anarchique des systèmes d'information, un traitement doit être administré simultanément aux systèmes et aux structures informatiques. C'est là l'objet de la méthode SOA inspirée de l'astronautique, détaillée au Chapitre 10.

Chapitre 3

Réalisation du plan d'architecture SOA

Dans ce chapitre

- Choix des politiques et procédures.
- Mise en place d'un centre de compétence.
- Automatisation des règles et procédures.
- Instauration de points de contrôle vérifiant la mise en application des politiques.

Dans ce chapitre, nous nous intéressons à l'application automatisée de règles et procédures comme moyen de lutte contre la prolifération anarchique des systèmes d'information.

En instaurant des points de contrôle, vous avez la certitude que chaque étape est axée sur la réalisation de votre plan directeur SOA.

Convenir de politiques et procédures

Le terme « politique » revêt plusieurs significations mais nous l'emploierons ici pour décrire les préceptes guidant nos décisions et actions futures. Politiques et procédures, en tant que telles, sont à même d'aiguiller l'implémentation SOA vers la réalisation du plan directeur.

En général, les politiques imposent des contraintes à une tribu dans l'intérêt d'une autre (ou bien dans l'intérêt général). Même si les contraintes ne sont guère appréciées (surtout chez les développeurs), un minimum de discipline permet d'enrayer la prolifération anarchique du système d'information et de dégager des bénéfices pour chacun.



Si des politiques réforment les activités d'une tribu au profit d'une autre, il est important que les deux parties appréhendent ces règles et y souscrivent afin de parer à toute résistance passive ou rébellion ouverte contre celles-ci.

Le centre de compétence SOA

L'instance dirigeante qui définit et applique les politiques SOA est appelée *centre d'excellence* ou *centre de compétence SOA*. Qui participe à un centre de compétence SOA ? Les représentants de chacune des tribus concernées par vos projets SOA.

Pour la quasi-intégralité de votre plan directeur SOA (y compris la nature des services à proposer, la manière dont ils seront définis et leur mode d'interopérabilité), des politiques applicables à votre entreprise sont définies implicitement. Eu égard à l'abondance de ces règles tacites, il est essentiel que la première démarche du centre de compétence SOA consiste à entériner ce plan directeur comme objectif commun.



Il est important que chaque groupe concerné comprenne et accepte les implications du plan directeur SOA au quotidien. Par conséquent, la validation de ce plan ne doit pas être faite à la légère. Invitez toutes les parties prenantes à réfléchir aux implications de cette vision.

Automatiser l'application des politiques et procédures

Certains associeront la mise en œuvre automatisée de politiques à une atteinte à leur liberté et à leur créativité. Dans une société civile, les individus sont libres d'agir à leur guise, mais les règles instaurées leur évitent de nuire à autrui, intentionnellement ou non. La gouvernance s'apparente, en quelque sorte, au « Code de la route » :

- ✓ Quelques réglementations suffisent à rendre la route plus sûre et plus agréable pour chacun. Un poste de péage occasionnel permet de financer la réparation des nids-de-poule, et des feux de régulation de réduire les embouteillages.
- ✓ Les procédures automatisées sont préférables aux procédures manuelles. En équipant votre véhicule d'un badge de télépéage, vous vous dispensez de marquer l'arrêt au poste de péage et de faire l'appoint.

Les politiques ne peuvent pas toutes être automatisées. À vous de juger et, le cas échéant, d'intervenir.

Politiques et procédures

Une gouvernance SOA adéquate couvre plusieurs points d'application tout au long du cycle de vie du service. Néanmoins, simplifions. Sachez que les politiques et procédures SOA se répartissent en deux catégories pour ce qui est de leur automatisation :

- ✓ **Règles de gouvernance de conception** : Elles permettent de s'assurer que les objets SOA sont conformes aux impératifs de conception exposés dans le plan directeur SOA.
- ✓ **Règles de gouvernance d'exécution** : Elles garantissent la conformité des services SOA aux impératifs d'exécution négociés entre le fournisseur et le consommateur du service.

Les types de politiques relevant de l'une et l'autre catégorie sont détaillés dans les deux sections suivantes.

Politiques et procédures de conception

Les *politiques de conception* permettent de s'assurer que les services sont conformes aux spécifications exposées dans le plan directeur SOA. Elles imposent notamment des contraintes à leurs concepteurs et développeurs, dans l'intérêt général :

- ✓ **Interopérabilité** : Un plan directeur SOA uniformise l'interopérabilité entre services, généralement en entérinant un ensemble de normes.
- ✓ **Découvrabilité** : Certains attributs peuvent être exigés, tels un descriptif métier ou des détails sur la classification du service à l'intérieur du *catalogue (registre)*. Éléments favorisant la découverte de services, ils peuvent être définis à l'aide de règles.
- ✓ **Sécurité** : Le plan directeur SOA doit uniformiser la sécurisation des services SOA. Des politiques peuvent être établies pour définir le mode et les paramètres de sécurité.
- ✓ **Unicité** : Il convient de distinguer les nouveaux services de ceux préexistants en les nommant différemment. Un mécanisme baptisé *espace de noms* (« namespace ») s'acquitte généralement de cette tâche. L'établissement de règles esquivait ce problème.

Défaut d'interopérabilité

La perte de la sonde Mars Climate Orbiter illustre les conséquences effroyables d'un défaut d'interopérabilité. Le contact avec ce module spatial a été perdu car les données transmises par les ingénieurs de Lockheed Martin, fabricant de la sonde, étaient exprimées

dans des unités du système de mesure anglo-saxon (livres-secondes) alors que leurs confrères de la NASA utilisaient le système métrique (newtons-secondes). À cause de cette bévue, quelque 125 millions de dollars ont été réduits en cendres dans l'atmosphère martienne.

- ✓ **Respect de l'interface** : L'utilisation (*invocation*) des services se doit d'être uniformisée. Des politiques peuvent rendre obligatoire la standardisation de l'interface.
- ✓ **Respect du format des données** : Pour garantir le caractère réutilisable d'un service, il convient d'instaurer des formats de données communs, ou *schémas*. Ce faisant, vous avez la certitude qu'un champ d'adresse pour un service donné pourra être réutilisé par un autre, même si les deux stockent les données différemment. L'adoption de schémas communs peut être rendue obligatoire par des règles.
- ✓ **Indicateurs** : Statistiques et comptes-rendus sur les problèmes de conception de services peuvent également faire l'objet de politiques.

Les *procédures de conception* relèvent du cycle de développement des systèmes et de sa transformation en cycle de développement des services. Ce point est abordé en détail au Chapitre 7.

Politiques et procédures d'exécution

En matière de gouvernance, les politiques d'exécution génèrent moins de conflits organisationnels puisque les contraintes sont surtout imposées aux systèmes informatiques dans l'intérêt des consommateurs de services SOA.



La plupart du temps, des règles d'exécution existent afin de s'assurer que le service est dispensé conformément aux attentes (de son consommateur). En voici un échantillon :

- ✓ **Engagements de qualité de service (SLA : Service Level Agreement)** : Fournisseurs et consommateurs conviennent

d'objectifs de performances et de critères d'évaluation des services.

- ✓ **Authentification** : Fournisseurs et consommateurs doivent s'entendre sur les réponses à apporter aux questions suivantes : comment s'identifier en tant que consommateur de service ? Quels systèmes d'identité utiliser ? Le cas échéant, quels identifiants de sécurité exploiter ? Autant de questions à traiter au moyen de règles d'exécution.
- ✓ **Autorisation** : Quelle méthode employer pour déterminer si un fournisseur est autorisé à invoquer un service ?
- ✓ **Cryptage** : Comment crypter les messages pour qu'ils ne puissent être lus, si ce n'est par leurs destinataires ?

Intérêt du langage XML

XML est l'acronyme d'eXtensible Markup Language. Les documents et messages XML se composent d'un ensemble de balises entre crochets triangulaires (< >) rappelant le HTML, le langage du Web.

Même s'il n'y a aucune obligation à utiliser XML au sein d'une architecture SOA, ce langage s'avère idéal dans ce contexte compte-tenu de ses propriétés :

- ✓ **Interopérabilité** : XML facilite les communications entre des systèmes différents. La prédiction s'accomplit presque d'elle-même car nombre de fournisseurs de plates-formes logicielles et matérielles ont décidé d'adopter XML comme standard de communication.
- ✓ **Lisibilité machine** : XML est lisible à la fois par l'homme et par la machine. Cette caractéristique facilite considérablement les échanges entre fournisseurs de services, consommateurs de

services et points d'application, au même titre que la mise en œuvre des règles.

Un dialecte XML, appelé *services Web*, est très répandu pour l'implémentation de la SOA. Les services Web offrent une structure standard, SOAP (Simple Object Access Protocol), pour la transmission de messages. Ils proposent également WSDL (Web Services Description Language), mécanisme standard pour la description des interfaces de services, ainsi qu'un protocole UDDI (Universal Description, Discovery and Integration) assurant la découverte de services à l'intérieur d'un registre.

Les services Web intègrent un mécanisme pour adresser des instructions au destinataire du document ou du message, mais également aux points de contrôle intermédiaires. Ils contribuent à structurer règles et procédures selon une méthode standard.

- ✓ **Signatures** : Comment être certain que les messages émanent de fournisseurs et de consommateurs légitimes et qu'ils n'ont pas été dénaturés en cours de route ?
- ✓ **Alertes et notifications** : Quelles sont les conditions à réunir pour déclencher des alertes ? À qui l'alerte est-elle adressée ? Les alertes peuvent signaler à la fois des problèmes métier et techniques.
- ✓ **Indicateurs** : Indicateurs clés de performance (KPI : Key Performance Indicator) et critères d'évaluation étayant les décisions sont fixés par des politiques. Ces critères sont un aspect fondamental, sur lequel nous reviendrons au Chapitre 9.

Les politiques d'exécution imposent des contraintes aux équipes opérationnelles et aux systèmes informatiques dans l'intérêt du consommateur de services. Les *procédures d'exécution* peuvent inclure des demandes de support et des réponses à des notifications et à des alertes en temps réel. Il est important, dans un contexte SOA, de favoriser la réactivité face à l'évolution des problèmes d'exécution.

Instaurer des points de contrôle vérifiant la mise en application des politiques

À l'instar d'un poste frontalier chargé de vérifier votre passeport et vos bagages, la gouvernance SOA instaure des points de contrôle s'assurant que les accords entre entreprises sont effectivement appliqués.

Ces points de contrôle incluent les éléments suivants :

- ✓ **Registre/référentiel SOA** : Sert de point d'application aux politiques et procédures de conception.
- ✓ **Système de gestion d'exécution SOA** : Sert de point d'application aux politiques et procédures d'exécution.

Au Chapitre 5, nous analyserons en détail la manière dont ces deux composants clés de la gouvernance SOA peuvent servir à automatiser politiques et procédures.

Chapitre 4

Infrastructure de services

Dans ce chapitre

- Création de nouveaux services par activation.
- Services à couplage faible avec médiation.
- Recherche de souplesse avec la virtualisation de services.

Les services sont le nerf de l'architecture SOA. La plus-value qu'ils dégagent, suivant notre approche de la SOA inspirée de l'astronautique, représente l'énergie de propulsion de votre vol dans l'espace. En règle générale, plus les services à disposition sont réutilisables, plus vous créez d'énergie cinétique. Si vous canalisez correctement cette énergie, elle produit la force de propulsion de votre organisation.

Comme nous l'avons indiqué aux chapitres précédents, un service doit être considéré sous deux angles différents : celui de son interface et celui de son implémentation. L'interface d'un service définit les fonctionnalités offertes par ce service, et son implémentation, les modalités d'accès à ces dernières. L'architecture SOA tire sa performance et sa souplesse de la distinction opérée entre ces deux éléments. Pour obtenir, vous aussi, un maximum de rendement, dissociez implémentation et interface au sein de votre infrastructure de services.

Comprendre l'activation de services

L'*activation de services* au sein de votre infrastructure est liée à leur implémentation. La technologie et l'outillage utilisés à cette fin autorisent la création de nouveaux services. Mais à partir de quoi ? Faut-il mobiliser votre environnement de développement intégré (IDE : Integrated Development Environment) Java préféré et engager des travaux de programmation dès qu'un nouvel ensemble de services vous est réclamé ? Si certains mordus d'informatique risquent de trouver le défi amusant, une stratégie de ce type se révèle extrêmement fastidieuse et onéreuse.

Stratégie de superposition à l'existant

Regardez autour de vous : votre entreprise possède et exploite déjà plusieurs dizaines, voire plusieurs centaines d'applications et de systèmes. Ces applications représentent une véritable manne de services pour votre SOA :

- ✓ L'un des systèmes en place possède déjà très certainement les fonctionnalités requises par vos services.
- ✓ Ces applications, déjà rodées, ont fait leurs preuves en production. Les services qui leur seront adossés auront tôt fait de gagner la confiance des consommateurs potentiels.
- ✓ Il est plus rapide et plus économique de créer des services dérivés d'applications existantes plutôt que de réécrire des applications de A à Z dans une optique orientée services.

Plutôt que de faire table rase de l'existant en le remplaçant par des applications orientées services, conservez-le et superposez-y une *couche* de services autorisant la participation des systèmes en place à votre architecture SOA. À l'aide des outils et compétences adaptés, vous exposez rapidement ces applications sous forme de services et assurez la montée en puissance de votre SOA. Ce faisant, vous protégez les investissements considérables déjà réalisés par votre entreprise dans le parc actuel. D'ailleurs, si vous exposez la logique de cette proposition à vos supérieurs hiérarchiques, ils ne tariront pas d'éloges sur votre discernement et vous voueront une reconnaissance éternelle.



Les applications existantes regorgent de fonctionnalités. Consacrez votre énergie à y superposer vos services.

Mais comment s'y prendre pour exposer ces fonctionnalités via des services ? En règle générale, les applications en place appartiennent à l'une ou l'autre de ces catégories :

- ✓ **Applications maison** : Aux temps reculés de l'informatique (années 1950 à 1970), la mise à disposition de nouvelles fonctionnalités passait par une méthode simple : retroussage de manches et déploiement d'applications logicielles « sur mesure ». La majeure partie du portefeuille applicatif de votre entreprise se compose d'ailleurs probablement d'applications maison, mises au point par le service informatique pour répondre à la spécificité des besoins de votre établissement. Nombre de ces systèmes épousent le cœur de métier de votre entreprise et s'acquittent de tâches centrales comme le traitement des commandes et les expéditions.

- ✓ **Progiciels du marché** : Si les services informatiques continuent à confectionner des applications sur mesure, depuis une vingtaine d'années, les entreprises font l'acquisition de progiciels d'éditeurs qu'elles personnalisent. Résultat : la plupart des grandes structures multiplient aujourd'hui les investissements dans le déploiement de progiciels de gestion intégrés (ERP), de gestion de la relation client (CRM) et autres systèmes du marché.

Or, il est difficile de transformer ces applications en services car la plupart ont été conçues bien avant l'avènement de l'architecture SOA et de l'interopérabilité. Si une application est antérieure à la fin des années 1990, elle ne possède probablement pas d'interface XML. Peut-être intègre-t-elle, en revanche, des API (*Application Programming Interfaces*) faisant appel à l'un des nombreux standards et protocoles de l'ère pré-XML tels que RMI (Remote Method Invocation), CORBA (Common Object Request Broker Architecture), COM (Common Object Model), DCOM (Distributed Component Object Model) ou RPC (Remote Procedure Call). Aïe ! Faute d'API adéquate, sans doute devrez-vous accéder directement au magasin de données sous-jacent. Ouille ! Il existe, fort heureusement, certains outils astucieux qui vous permettront de forcer l'ouverture desdits systèmes pour les exposer sous forme de services.

Bus de services d'entreprise et activation de services

Le bus de services d'entreprise (ESB, *Enterprise Service Bus*) est un excellent choix pour l'activation de services si l'application à transformer en service est dotée d'une interface autorisant sa connexion à d'autres systèmes. L'ESB procure tous les outils nécessaires à l'élaboration de services XML tirant parti de ces API :

- ✓ **Compatibilité avec une multiplicité de protocoles** : Les ESB reconnaissent un large éventail de protocoles, y compris les plus désuets comme RPC. Si l'ESB est valable, la gestion d'un protocole devient quasiment transparente.
- ✓ **Compatibilité avec une multiplicité de formes de communication** : Pour communiquer avec une application, l'ESB fera le plus souvent appel au mécanisme requête/réponse. Il transmettra sa requête à l'application via un protocole et l'application lui renverra aussitôt sa réponse. Cependant, nombre de systèmes stratégiques utilisent des formes de communication élaborées, à base de

messages, comme la publication-abonnement ou la méthode « fire-and-forget ». Un ESB efficace facilite la connexion à un système de ce type et peut combiner différentes formes de communication.

- ✓ **Compatibilité avec une multiplicité de formats de messages** : L'ESB n'a pas son pareil pour convertir des messages au format XML dans un langage intelligible par votre application, qu'il s'agisse de MIME, de texte brut ou de Klingon. L'ESB réalise les conversions et transformations nécessaires à partir du format XML.
- ✓ **Adaptateurs** : L'ESB s'acquitte donc des travaux de « plomberie » indispensables à la connexion à des applications existantes. Il reste néanmoins à démêler l'écheveau interne des applications et des interfaces, et l'entreprise est délicate. Rassurez-vous. Les ESB les plus performants dissimulent la complexité de la connexion à une application derrière une interface commune et cohérente, baptisée *adaptateur*. Ces adaptateurs facilitent considérablement la tâche des développeurs de services. Plutôt que de s'affronter à la difficulté inhérente à la réalisation de connexions à des systèmes hétérogènes, les développeurs se recentrent sur l'exposition de fonctionnalités existantes sous la forme homogène de services.



Peut-être certains passionnés de la SOA sursauteront-ils à la lecture des sections consacrées à l'ESB. Si vous en faites partie, c'est que vous vouez une croyance quasi-religieuse à la vision classique de l'ESB. Dans cette représentation, l'ESB est un élément décisif de l'infrastructure SOA qui s'intercale entre fournisseurs et consommateurs de services. Les services eux-mêmes ne résident pas sur le bus. Si nous croyons, pour notre part, en la nécessité d'une telle infrastructure, et réitérons cette conviction au paragraphe « Comprendre la médiation de services », en revanche, nous n'admettons pas que seuls les produits estampillés ESB aient le droit d'en faire partie.



Nombre de produits aujourd'hui sur le marché se réclament de l'appellation *ESB*. En réalité, ils diffèrent énormément les uns des autres tant par leurs caractéristiques que par les problèmes qu'ils visent à résoudre. Cette catégorie de produits « fourre-tout » recouvre aussi bien la gestion des données que le workflow humain ou le traitement des événements. De là à ce que certains experts SOA proclament très prochainement que la machine à laver est un composant indispensable pour tout ESB... Toujours est-il que nous ne recommandons pas une analyse minutieuse

de cette catégorie de produits dans le but de séparer les vrais ESB des faux, cette catégorisation n'ayant guère d'intérêt. Ce qui importe, en revanche, c'est de cerner vos besoins et de trouver le produit capable de les satisfaire. Dans une optique d'activation de services, nous préconisons de tester les ESB présentant les caractéristiques décrites dans ce paragraphe.

Encapsuleurs d'applications

Les ESB n'ont pas leur pareil pour faciliter l'accès des développeurs de services aux interfaces applicatives, indépendamment du protocole et des formats de messages utilisés. Malheureusement, toutes les applications ne sont pas dotées d'une interface. Certaines sont plus hermétiques qu'une huître. Pour s'immiscer à l'intérieur, il faut redoubler de créativité technologique en faisant appel à des encapsuleurs (en anglais, « wrappers »).



Les *encapsuleurs* peuvent s'immiscer dans une application en cours d'exécution ou se connecter à des programmes internes ou à des appels de fonctions et les exposer sous la forme de services. Vous souvenez-vous de ces films de science-fiction, peuplés de créatures extraterrestres s'introduisant dans le corps humain, neutralisant son système nerveux et dotant les êtres sous leur emprise de superpouvoirs ? L'encapsuleur leur ressemble quelque peu, à ceci près qu'il est totalement inoffensif. Sa mission relève du tour de magie technique, mais les résultats sont probants. Des encapsuleurs sont disponibles pour un large éventail de plates-formes techniques, des langages C et C++ à COBOL et Natural en passant par les mainframes.

Pour qu'un encapsuleur fonctionne, encore faut-il qu'il soit positionné correctement, ce qui suppose qu'un intervenant soit rompu aux rouages de l'application afin de trouver un « point d'accroche ». Or, il arrive qu'une application soit tellement vétuste que son fonctionnement est ignoré de tous. Les applications mainframe finissent par être à ce point opaques que personne n'ose intervenir. Dans cette situation, vous devrez probablement vous tourner vers votre dernier espoir en la matière : la représentation à l'écran des données. Peut-être n'y a-t-il personne pour comprendre les rouages d'un programme mainframe, mais d'autres s'y connaissent en écrans applicatifs – quelles données ceux-ci attendent en entrée et quels résultats graphique ils produisent en sortie. Fort de ces informations et d'un outil de capture de l'image à l'écran, vous pouvez convertir ces écrans applicatifs en services pour votre SOA.



Si vous créez des services à partir d'applications existantes, la granularité de ces services vous sera quasiment imposée. Souvent, la manière dont le processus est modélisé au sein de l'application détermine la granularité du service qu'il est possible de créer. Tout dépend de la manière dont l'application a été conçue et programmée à l'origine. Sachez qu'il existe des processus intéressants, mais nichés au cœur des applications, si bien qu'il n'est pas possible de les exposer en tant que services. C'est frustrant certes, mais vous n'y pouvez rien.

Comprendre la médiation de services

La distance renforcerait, dit-on, les liens du cœur. Nous affirmons, pour notre part, qu'elle consolide votre architecture SOA. Par *distance*, nous entendons celle mise entre les fournisseurs et les consommateurs de vos services. Il va de soi que nous recommandons vivement une étroite collaboration entre les différents acteurs humains des structures qui fournissent ou consomment des services. Mais, pour ce qui est des systèmes, nous conseillons de respecter une séparation bien nette. En d'autres termes, le couplage faible est de rigueur entre consommateurs et fournisseurs de services de sorte que chacun dispose d'une marge de manœuvre pour évoluer sans remettre en cause le lien qui les unit. C'est là tout le rôle de la couche de médiation au sein de votre infrastructure de services. Cette couche abrite l'interface du service. Et elle permet aux consommateurs de communiquer avec les fournisseurs de services, mais en prenant soin de conserver une certaine distance avec eux.



Les consommateurs ne doivent *jamais* se connecter directement aux implémentations d'un service au niveau de la couche d'activation, mais à l'interface du service, logée dans une couche de médiation distincte afin de préserver un maximum de souplesse au niveau de l'architecture SOA.

La médiation de services constitue de surcroît un excellent levier pour améliorer l'interopérabilité entre fournisseurs et consommateurs. Dès lors que tous les messages doivent traverser des composants de médiation de services, vous avez la possibilité de les modifier, au même titre que le protocole, afin de garantir une interopérabilité entre fournisseurs et consommateurs.

La médiation de services offre également une infrastructure commune et centralisée pour l'implémentation des spécifications opérationnelles en rapport avec la qualité de service (QoS), telles

que la sécurisation et la performance. La séparation entre ces spécifications et la logique de mise en œuvre des services permet aux développeurs de se recentrer sur l'élaboration d'une logique métier et de réduire les coûts de développement ; la réutilisation des services est optimisée puisque les impératifs QoS peuvent être revus et corrigés sans qu'il faille intervenir pour autant au niveau du service.

La médiation de services est, somme toute, inestimable pour l'adoption de la SOA. Elle maximise le retour sur investissement du développement des services et permet de faire évoluer l'architecture SOA avec un minimum de perturbations.

Réaliser la virtualisation de services

Vous réalisez la médiation de services en recourant à des *services virtuels*. Un service virtuel n'est pas un véritable service, mais simplement un intermédiaire appelé proxy pour celui-ci. Ce service proxy se trouve au niveau de la couche de médiation. Il présente l'interface du service souhaité aux consommateurs. Les consommateurs du service invoquent ce service proxy qui transmet à son tour les messages au service réel, c'est-à-dire l'implémentation (voir Figure 4.1).

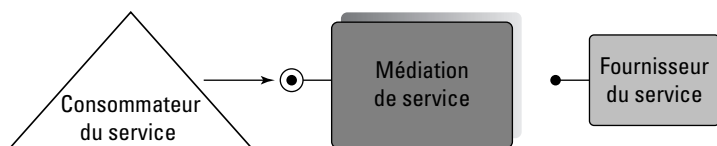


Figure 4.1 : Un service virtuel.

Conséquence de la virtualisation du service, l'interface et la mise en œuvre du service sont séparées en deux couches distinctes. Consommateurs et fournisseurs ne sont jamais en relation directe.

Couplage faible

La médiation de services confère une souplesse inestimable qui vous sera indispensable à mesure que progressera votre adoption SOA. Cette souplesse s'explique par le fait qu'un service virtuel découple le consommateur du fournisseur sur plusieurs plans : l'emplacement, le transport et le message.

Indépendance de l'emplacement

Un service virtuel vous permet de masquer l'emplacement réel d'un service. Vous avez ainsi toute liberté pour déplacer son implémentation sans perturber ses consommateurs. Vous pouvez, par exemple, transférer celle-ci sur des serveurs haute capacité pour faire face à la demande croissante.

Indépendance du transport

La virtualisation de services vous permet d'exposer facilement une implémentation de service sur plusieurs protocoles de transport afin de pallier les différences d'interopérabilité et de proposer d'autres modes de réutilisation. Supposez que vous avez mis en œuvre un service de création de commande (CreateOrder) accessible, au départ, via JMS (Java Message Service). Ce service a rencontré un franc succès, et plusieurs consommateurs ont manifesté leur désir de réutiliser ses fonctionnalités au sein de leurs propres applications. Seul hic : nombre d'entre eux gèrent uniquement le protocole HTTP. Ce qui, concrètement, signifie que ces nouveaux consommateurs ne peuvent interagir avec le protocole géré par le service CreateOrder. Normalement, il vous faudrait créer une autre implémentation de ce service, compatible HTTP, mais grâce à la technologie de médiation de services, vous exposez aisément CreateOrder sous forme de service HTTP virtuel, sans devoir modifier votre implémentation réelle. Voilà qui permet de traiter, en toute transparence, les problèmes d'interopérabilité entre protocoles et de faire en sorte que le service CreateOrder puisse être réutilisé par de nouveaux consommateurs.

Indépendance du message

Parfois, les consommateurs ne sont plus en phase avec les fournisseurs d'un service du fait de la non-conformité des messages XML aux attentes. Peuvent alors s'ensuivre des incohérences de données et des incompatibilités sémantiques. Ces problèmes d'interopérabilité peuvent être engendrés, par exemple, par de nouvelles versions de services ou par des changements dans les schémas XML qui définissent les paramètres des messages. En règle générale, il est préférable que les consommateurs se conforment systématiquement au format de message escompté. Toutefois, en cas de modifications, il est pratiquement impossible de les forcer tous à s'y conformer en même temps. Dans ce cas, la virtualisation de services permet de transformer les messages pour les rendre conformes aux attentes du consommateur et du fournisseur.

Spécifications opérationnelles

Les services virtuels se prêtent à la mise en œuvre de spécifications opérationnelles ou de qualité de service (QoS). Exemples :

- ✓ **Validation de message** : S'assure que le format des messages XML est valide et conforme aux attentes de l'interface du service.
- ✓ **Authentification et autorisation** : Identifie le consommateur et s'assure qu'il est autorisé à invoquer le service.
- ✓ **Cryptage du message et signature** : Décrypte le message et vérifie les signatures.
- ✓ **Basculement automatique et équilibrage de charge** : Garantit une capacité suffisante pour gérer la charge transactionnelle et la disponibilité des services.
- ✓ **Acheminement de message** : Transmet les messages aux différentes implémentations du service en fonction de leur contenu ou de leur contexte.
- ✓ **Supervision et engagements de qualité de service (SLA)** : Surveille la performance des services et s'assure que ces derniers sont dispensés dans le respect des engagements de qualité de service promis aux consommateurs.



Les spécifications énumérées ci-dessus sont plus susceptibles d'évoluer que la logique fonctionnelle d'un service. Par conséquent, en les mettant en œuvre dans une couche séparée, il est possible de les modifier sans toucher à l'implémentation du service, cette dernière intervention étant relativement onéreuse et intrusive. La réutilisation d'un service donné peut également être maximisée en fournissant des versions virtuelles de ce service accompagnées de spécifications QoS différentes. Il est parfaitement envisageable, par exemple, de créer un service virtuel avec authentification HTTP pour les consommateurs à l'intérieur de l'entreprise, et un autre service virtuel avec cryptage XML pour ceux de l'extérieur. Le service réel n'est jamais modifié. En revanche, pour personnaliser le service rendu à diverses catégories de consommateurs, sont créés des services virtuels intégrant des règles QoS différentes.

Autre atout significatif : vous observez une approche unique et cohérente dans la gestion de ces spécifications pour tous les fournisseurs de services. Vous réduisez ainsi la complexité de votre architecture SOA tout en abaissant le coût global de développement et de maintenance des services.

Bus de services d'entreprise et médiation de services

L'ESB peut constituer une solution de médiation de services adéquate. En réalité, au départ, sa conception répondait à cette finalité, en conjuguant souplesse et évolutivité. Malheureusement, l'appellation ESB a ensuite été détournée et ce terme passe-partout renvoie aujourd'hui à une multitude de fonctionnalités. En fait, il existe des ESB qui excellent dans l'activation de services et se révèlent relativement médiocres dans la médiation. Par conséquent, si vous envisagez d'en intégrer un à votre infrastructure de médiation de services, assurez-vous que la virtualisation soit aisée et clé en main. La virtualisation de services doit être pilotée par la configuration et nécessiter peu, voire aucune programmation.



Certains ESB offrent à la fois des fonctions de médiation et d'activation de services. En recourant à un même produit pour ces deux objectifs, vous simplifierez le déploiement de votre infrastructure de services et réduirez son coût de possession. Mais même dans ce cas, veillez à utiliser des instances distinctes du même produit pour chaque couche ; la mise en œuvre et l'interface du service doivent toujours être dissociées.

Intermédiaires/passerelles de services

Nombre de fournisseurs proposent un intermédiaire de services léger, une « passerelle », comme solution de médiation de services efficace. Les passerelles de services se focalisent sur la virtualisation, et se prêtent donc moins à des opérations de montée en capacité et de programmation que les ESB. Leur installation, leur exploitation et leur maintenance sont en revanche facilitées, surtout pour les administrateurs SOA.

Appliances SOA

Les appliances SOA sont une catégorie particulière d'intermédiaire de services, commercialisées sous forme d'équipements autonomes, présentant des caractéristiques et atouts spécifiques :

- ✓ Appliance matérielle garantissant un déploiement clé en main — à mettre sous tension, tout simplement.
- ✓ Processeur XML matériel embarqué pour la manipulation de charges volumineuses et le traitement XML intensif comme la cryptographie.
- ✓ Compatibilité avec un large éventail de normes de sécurité.
- ✓ Protection intégrée contre les menaces et les intrusions XML.
- ✓ Compression/décompression des messages.

Les appliances SOA n'ont pas leur pareil pour garantir sécurité et performance en bordure de réseau. À la périphérie du réseau d'une entreprise se trouve généralement la zone démilitarisée (DMZ) qui abrite des systèmes bastions, accessibles par les clients et partenaires situés à l'extérieur. En périphérie est également délimité le périmètre des centres de données connectés en réseau étendu (WAN). Voici quelques scénarios types susceptibles de bénéficier de l'utilisation d'appliances SOA :

- ✓ Exposition des services à consommer à l'extérieur des pare-feu de l'entreprise (scénarios B2B).
- ✓ Consommateurs et fournisseurs résidant à l'intérieur de frontières de confiance (administrations, par exemple).
- ✓ Déploiements WAN.



L'installation d'appliances SOA peut se concevoir en complément d'une solution de médiation de services à usages multiples. En déléguant les lourdes tâches de traitement XML, comme la validation de schémas et la cryptographie, sur les appliances SOA, vous créez une excellente rampe de lancement vers des ESB et intermédiaires de services.

Chapitre 5

Infrastructure de gouvernance

Dans ce chapitre

- Notion de registre/référentiel.
- Communiquer et appliquer les politiques.
- Gestion des cycles de vie.

Si vous estimez qu'il est plus compliqué de mettre d'accord des mordus de SOA sur une définition de cette architecture que de dresser un chat, demandez-leur la signification de la gouvernance SOA. Les débats vont bon train sur la question de savoir ce que recouvre, ou exclut, la gouvernance SOA. Fort heureusement, son importance dans le succès d'un projet SOA fait l'unanimité. La gouvernance régit les trois principes d'une adoption SOA inspirée de l'astronautique. Dans ce chapitre, nous examinerons comment elle parvient à orienter la fusée dans la bonne direction en dépit des zones de turbulences traversées.

La *gouvernance* est l'ensemble de rôles, politiques et procédures guidant l'adoption d'une architecture SOA. En mettant en œuvre les composantes technologiques de la gouvernance, vous érigez l'infrastructure indispensable à la mise en pratique de ces rôles, politiques et procédures à l'échelle de votre SOA.

Manipuler le registre/référentiel

Parce qu'on ne peut régir que ce que l'on voit, commencez par créer un système de référence où tous les éléments de votre architecture SOA seront visibles par l'ensemble des parties prenantes. En la matière, le registre/référentiel (ou tout simplement *référentiel*) a su s'imposer comme standard.

Les informations consignées dans un référentiel sont certes fonction du style, de l'envergure et de la maturité de l'approche adoptée en matière de gouvernance. Néanmoins, nous recommandons à la plupart des entreprises d'y faire figurer ce qui suit :

- ✓ Services disponibles dans la SOA et totalité des métadonnées correspondantes pour le catalogage, la recherche et la consommation de ces services. Les métadonnées de services incluent des informations d'exécution sur la performance globale de ceux-ci.
- ✓ Ressources SOA liées, tels les schémas XML et les procédures BPEL.
- ✓ Organisations divisions (projets, départements ou division, par exemple) fournissant les services.
- ✓ Applications (ou systèmes) consommant les services.
- ✓ Organisations consommant les services.
- ✓ Politique régissant le comportement des individus et systèmes participant au cycle de vie SOA.
- ✓ Contrats et accords instaurés entre consommateurs et fournisseurs.
- ✓ Dépendances et relations entre l'ensemble des éléments de cette liste.



Les passionnés de SOA argumentent à loisir sur la distinction entre registre et référentiel. Les registres sont censés être orientés exécution, et les référentiels orientés conception. Un distinguo quelque peu arbitraire, découlant de la manière dont les produits ont évolué sur le marché SOA. Un système de référence valable intègre ces deux aspects de manière transparente.

Puisque ce référentiel sera le socle de votre système de gouvernance, il est important de veiller à ce qui suit dans le choix d'une solution :

- ✓ **Prise en charge d'une multiplicité d'intervenants :** La gouvernance est un jeu auquel participent une multitude d'intervenants. Divers acteurs, des développeurs de services aux architectes sécurité en passant par les administrateurs SOA, prennent part aux activités de gouvernance. Il est important que tous trouvent, dans le référentiel, l'authentique plan directeur SOA et ses informations annexes. Le référentiel est conçu pour répondre aux besoins de différents types d'intervenants. Il se veut immédiatement accessible (via un navigateur Web par exemple) et simple d'emploi. Il sera, de préférence, personnalisable afin de permettre à ses utilisateurs d'avoir accès à ce dont ils ont besoin, ni plus, ni moins.

- ✓ **Compatibilité avec des environnements hétérogènes** : Un référentiel digne de ce nom est compatible avec toutes les plates-formes technologiques qui, dans votre entreprise, participeront à la SOA. À défaut, il est pratiquement impossible d'obtenir cette visibilité de bout en bout, indissociable d'une gouvernance efficace.
- ✓ **Personnalisation et évolutivité** : D'expérience, il n'est pas deux entreprises qui appliquent une gouvernance identique. Chaque entreprise a ses propres impératifs en la matière. Résultat : les informations à faire figurer dans le système de référence SOA varient énormément. Ce référentiel doit aussi faciliter la personnalisation des informations afin de répondre à la spécificité des besoins de votre structure.

Gestion des politiques

L'une des principales missions de la gouvernance SOA consiste à assurer un comportement approprié entre individus et systèmes participants. À vous de transmettre clairement vos politiques, puis de les appliquer de manière cohérente tout au long du cycle de vie SOA.

Auparavant, les architectes SOA consacraient volontiers des semaines et des mois entiers à documenter minutieusement ces politiques dans des « pavés » que personne ne se donnait la peine de lire. Et s'il était déjà difficile de sensibiliser les participants à l'existence de politiques, la diffusion des modifications apportées à celles-ci l'était encore davantage. Il fallait instaurer des séances de consultation-validation pour obliger chacun à prendre connaissance des dernières règles en vigueur et à s'y conformer. Les goulets d'étranglement qui n'ont pas manqué de se former ont incité les individus à contourner ces règles, faisant ainsi échouer la mission première de la gouvernance SOA. La mise en œuvre d'une gouvernance peut fort heureusement s'envisager autrement, au moyen de solutions de gestion de politiques SOA. Ces solutions autorisent leurs utilisateurs à :

- ✓ **Exprimer des politiques sous forme déclarative** : Les politiques peuvent être définies, modifiées et supprimées aisément au besoin.
- ✓ **Appliquer activement des politiques** : Les politiques sont appliquées automatiquement tout au long du cycle de vie SOA. Les participants bénéficient d'un retour d'information immédiat et un suivi automatisé de ces politiques est envisageable.



La gestion des politiques SOA est un élément décisif dans une solution de gouvernance SOA. Elle lève les obstacles à la gouvernance SOA, ainsi que les objections formulées à son encontre, en définissant clairement ce qui doit être conforme au plan directeur SOA. Ce faisant, ces solutions améliorent la traçabilité et garantissent des résultats cohérents. En automatisant les processus de gouvernance, vous supprimez les goulets d'étranglement et donnez à celle-ci les moyens de faire face à un nombre croissant de services, de fournisseurs et de consommateurs, à mesure que progressera votre adoption SOA.

Comment s'y prendre pour appliquer automatiquement des politiques ? L'application des politiques s'effectue via des *points de contrôle*. Un point de contrôle évoque un poste de péage. Tout comme le péage taxe les véhicules empruntant un tronçon de route donné, un point de contrôle est, lui aussi, positionné à des endroits stratégiques pour veiller au respect des règles de gouvernance.

Utilisation du registre/référentiel comme point de contrôle de conception

Le registre/référentiel constitue un point de contrôle incontournable en termes de conception de service puisque tous les objets de service doivent y transiter pour pouvoir être mis à disposition des consommateurs.



Dès lors que des ressources SOA sont publiées dans ce registre/référentiel, le système vérifie automatiquement si elles sont conformes aux standards d'architecture spécifiés dans le plan directeur SOA.

L'utilisation d'une norme telle que XML permet de valider automatiquement l'interopérabilité des ressources SOA dès leur publication.

Le registre/référentiel est capable de régir tous les aspects d'un service, y compris sa description. Vous pouvez, par exemple, définir une politique imposant aux éditeurs d'ordonner leurs services suivant une classification hiérarchique appelée taxonomie. Celle-ci facilitera la découverte des services. De même, vous pouvez exiger qu'une documentation soit jointe à chaque service décrivant ses modalités d'utilisation et imposer que cette documentation soit validée par les représentants de la communauté des utilisateurs qui consommeront le service.



Nombre d'objets SOA sont définis dans le plan directeur sous forme de documents XML afin de faciliter leur validation automatique. La lisibilité machine du format XML autorise la validation automatique des ressources XML par le registre/référentiel dès leur publication à l'intérieur de celui-ci.

Certaines politiques, qui ne peuvent être exécutées directement par le registre/référentiel, nécessitent une intervention humaine. Elles relèvent pour la plupart du *workflow de validation*. Il peut s'agir, par exemple, d'une politique spécifiant « l'interdiction de publier des services sans l'aval de Paul, responsable de l'assurance qualité ». La plupart des registres/référentiels gérant la définition et l'exécution de politiques de workflow manuelles, ce style de politique se révèle décisif dans la gestion du cycle de vie SOA, que nous aborderons plus loin.

Comprendre le cycle de vie

Question fondamentale de la gouvernance SOA : à quel moment vérifier la conformité à la politique ? Supposons que Paul publie un service dans le registre/référentiel. Ce service est-il censé être conforme à la totalité des politiques ? Si vous tenez absolument à ce qu'il le soit, Paul s'infligera une pression incroyable pour respecter ce critère préalablement à la publication de son service, qui restera, par conséquent, en dehors du système de gouvernance jusqu'à son déploiement. Supposons à présent que Paul finisse par publier son service, mais que, par malheur, celui-ci déroge à quelques politiques. Paul sera très contrarié d'avoir à le remanier et ne pourra respecter les délais. Il associera alors la gouvernance à un obstacle, doublé d'une contrainte bureaucratique inutile. Et si vous prodiguez des conseils à Paul et lui assuriez un retour d'information systématique, en vérifiant à point nommé les politiques appropriées entre le moment où il propose son service et celui où il s'apprête à le déployer ? Il vous suffit de jongler avec le cycle de vie.

Le *cycle de vie* définit les phases par lesquelles passe un service dès lors qu'il est activé au sein de l'architecture SOA. La Figure 5.1 illustre un cycle de service.

La définition d'un cycle de vie adapté à votre entreprise fait partie des activités de gouvernance prioritaires. Votre registre/référentiel doit vous permettre de modéliser ce cycle de vie et d'en réaliser le suivi, explicitement. La plupart des férus d'informatique maîtrisent ce concept et n'ignorent rien de la

plus-value qu'il confère, tant pour l'orientation que le suivi de la progression. Appliqué à la gouvernance SOA, le cycle de vie fait en outre office de relais, ou de point de contrôle, se prêtant à l'application des politiques de gouvernance. En mettant en œuvre les politiques adaptées à chaque phase du cycle de vie, vous donnez l'orientation adéquate et évitez des confrontations et remaniements en aval.

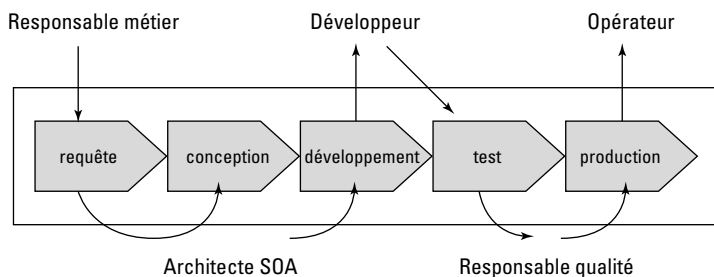


Figure 5.1 : Cycle de service

En conjuguant cycle de vie et politiques, vous instaurez un système de gouvernance souple encourageant la collaboration sous toutes ses formes, critère indispensable pour aller de l'avant. Les points de contrôle fournis permettent de s'assurer de la conformité aux politiques, dans le respect de l'orientation donnée.

Il s'agit d'une méthode simple et logique pour mettre en adéquation la gouvernance SOA avec les processus de cycle de vie existants. Architectes et centre de compétence SOA se servent des processus en place (en matière de gestion de projets et de cycle de vie) pour y incorporer des politiques amorçant l'automatisation du cycle de vie SOA. Voilà qui renforcera la collaboration avec les différents intervenants et consolidera cette solution de gouvernance pour en faire le « centre de commande SOA ».



Le cycle de vie diffère suivant le type de ressource. Ainsi, le cycle de vie d'un schéma XML se distingue de celui d'un service. Il est important que votre solution de gouvernance vous permette de définir des cycles de vie différents pour des ressources différentes.



L'architecture SOA soulève de nouveaux enjeux dans le domaine de l'assurance qualité (QA). Du fait du caractère dynamique du changement dans la SOA et des interdépendances entre ressources et systèmes, le recours à des techniques

traditionnelles d'assurance qualité logicielle risque de ne pas suffire. Chaque fois qu'une ressource est modifiée, ce sont plusieurs dizaines de ressources et systèmes différents qu'il faut tester à nouveau. Les contraintes de réutilisation vous obligent à tester la matrice de dépendances tout entière, plutôt que de vous fier à des tests isolés. Définissez, par conséquent, un point de contrôle sur une phase appropriée du cycle de vie afin de vous assurer que tous les critères de test et de validation SOA sont remplis. Ce point de contrôle sera, dans la mesure du possible, entièrement automatisé : au cours des différentes phases de pré-production du cycle de vie, les résultats pourront ainsi être vérifiés automatiquement dans le système d'assurance qualité et un récapitulatif sera publié dans le registre/référentiel, auquel se reporteront les différentes parties prenantes.

Faire appel à la gestion d'exécution

Même si le suivi et la gestion du cycle de service peuvent s'effectuer depuis le référentiel, l'invocation des services est extérieure à celui-ci. Elle intervient au sein de l'infrastructure de services, lorsque les consommateurs font appel à ceux-ci lors de l'exécution. (Voir les politiques d'exécution au Chapitre 3.) Certaines politiques de gouvernance cruciales doivent être contrôlées à l'exécution, au moment de l'invocation des services. Par ailleurs, les accords instaurés entre fournisseurs et consommateurs doivent également être vérifiés et mis en œuvre à ce moment.

Comment appliquer automatiquement des politiques d'exécution au sein d'une architecture SOA ? Grâce à un point de contrôle d'exécution. Le *point de contrôle d'exécution* s'apparente, là encore, à un poste de péage. Le *point d'application de la règle* (PEP, *Policy Enforcement Point*) (voir Figure 5.2) est installé sur une route à péage très fréquentée au sein de l'architecture SOA. En l'occurrence, c'est un passage obligé pour l'utilisation d'un service : sur le réseau entre le fournisseur et le consommateur. De ce fait, le PEP est en mesure d'appliquer des politiques sur les messages circulant entre le fournisseur et le consommateur.



Le point d'application d'une politique (PEP), dans le cadre d'une gouvernance SOA au stade de l'exécution, intervient sur les messages SOA au cours du processus d'utilisation (invocation) des services (durant leur exécution).

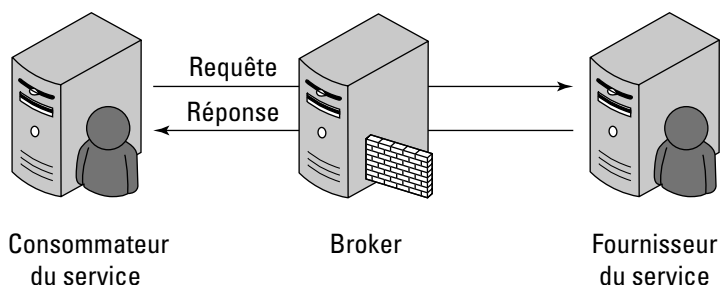


Figure 5.2 : Application d'une règle à l'exécution.

L'expression générique *point d'application d'une politique à l'exécution* est retenue car, en pratique, elle s'applique à nombre de systèmes différents. Dans la couche de médiation de services de votre infrastructure SOA (voir la notion de médiation de services au Chapitre 4) peuvent se nicher quantité de points d'application de ce type. Les composants de médiation s'intercalent entre fournisseurs et consommateurs de services et font office d'infrastructure centralisée pour l'implémentation de spécifications opérationnelles, telles que la sécurité et l'envoi de messages. Il se trouve que nombre de ces spécifications opérationnelles relèvent également de la gouvernance d'exécution et doivent être régies par des politiques. L'application de politiques au niveau de la couche de médiation permet de bénéficier d'une certaine uniformité, quel que soit l'emplacement des consommateurs et fournisseurs de services. Inutile de vous préoccuper des divergences entre les protocoles ou formats de messages employés par les uns et les autres. La couche de médiation, capable de gérer cette diversité, applique des politiques avec uniformité et cohérence à la totalité des messages circulant entre fournisseurs et consommateurs. Ces politiques étant, de surcroît, déclaratives et aisément modifiables, vous réagissez nettement plus rapidement aux demandes de rectification des spécifications opérationnelles.



Intrinsèquement, les composants de médiation de services constituent un point de contrôle pour la mise en œuvre des politiques d'exécution.

« Embarquer » les consommateurs de services

Après avoir découvert, dans le registre/référentiel, un service qu'il souhaite utiliser, le consommateur est mis en relation avec ce service. Ce processus de mise en relation (en anglais, « binding » ou « on-boarding ») constitue un point de contrôle idéal pour négocier les règles d'exécution entre fournisseur et consommateur.



Préalablement à cette mise en relation, le consommateur négocie un engagement de qualité de service (SLA) en vue d'instaurer des politiques d'authentification garantissant que le fournisseur lui ménagera un accès en exclusivité. En règle générale, il négocie des politiques de sécurité qui interdisent à des pirates d'accéder à ce service via ce point de contact.

Ces engagements sont négociés via le registre/référentiel et y sont stockés sous forme de contrat. Celui-ci définit les politiques d'exécution applicables à la mise en relation de chaque consommateur avec chaque service. Lorsque le consommateur entre en relation avec le service à partir du registre/référentiel, il se voit communiquer l'adresse d'une interface virtuelle de ce service qui réside au niveau du point d'application de la politique.



Une relation directe avec un service est appelée *couplage fort*. Dans ce cas, un changement de fournisseur se révèle nettement plus délicat. Il est préférable qu'un utilisateur ne puisse être mis directement en relation avec un service. Mieux vaut qu'il transite par une interface servant d'intermédiaire de gouvernance d'exécution.

Ici encore, la couche de médiation de services est idéale pour mettre en œuvre des engagements et règles de mise en relation à l'exécution. Ses composants facilitent la création de services intermédiaires ou virtuels. Plusieurs services virtuels peuvent être créés pour personnaliser la délivrance d'un même service à des consommateurs différents. La médiation, couplée à la gouvernance d'exécution, favorise la réutilisation des services et accélère la mise en œuvre de changements.

Boucler la boucle

Une solution de gouvernance bien conçue met à profit la visibilité en temps réel procurée par la gestion d'exécution pour enrichir les informations à disposition dans le référentiel. Voici les domaines où l'intégration entre une solution de gestion d'exécution et un référentiel se révèle extrêmement bénéfique :

- ✓ Découverte automatique des services et publication dans le référentiel.
- ✓ Détection des dépendances d'exécution entre services et autres ressources, et renseignement automatique de celles-ci dans le référentiel.
- ✓ Identification des consommateurs de services et renseignement automatique de ceux-ci dans le référentiel.
- ✓ Publication d'informations de haut niveau sur la performance des services afin de faciliter les prises de décision en termes de gouvernance.
- ✓ Détection et signalement des incohérences contenues dans le registre par rapport à l'existant lors de l'exécution (exemple : avertissement de l'administrateur si la définition WSDL du conteneur de service ne correspond pas à celle figurant dans le référentiel).
- ✓ Publication dans le référentiel d'informations concernant des événements en rapport avec la gouvernance (exemple : violation de règles ou d'engagements SLA).

Chapitre 6

Applications composites

Dans ce chapitre

- Tenants et aboutissants des applications composites.
- Avantages de la gestion des processus métier (BPM).
- Travail sur les applications composites.

Avec l'entrée en lice des logiciels dans les années 1960 et 1970, les entreprises ont recruté des programmeurs maîtrisant les environnements C et COBOL ainsi que d'autres langages spécialisés en vue de créer des applications sur mesure. Si elle a mis fin aux « jeux de piste », cette ère « constructive » a donné naissance à une foule de systèmes spécialisés aux coûts de maintenance élevés.

Dans les années 1980, des éditeurs de logiciels d'entreprise, à l'image de SAP, ont compris qu'il était possible de développer des progiciels offrant nombre des fonctionnalités que les entreprises avaient pour habitude d'intégrer à leurs applications. Certes, il était plus rapide et plus facile d'acheter ces logiciels que de les réaliser en interne, ce travail nécessitant toujours de solides connaissances doublées d'une expertise en programmation. Mais encore fallait-il les personnaliser afin de les adapter aux besoins de chaque client.

Au milieu des années 1990, les entreprises exploitaient une multitude de progiciels, tout en conservant en parallèle un très grand nombre de systèmes préexistants. Même si ces systèmes avaient peu de choses en commun, leur mise au rebut était difficilement justifiable. Ils étaient tout simplement trop imbriqués dans les opérations au quotidien et trop onéreux à remplacer ou à reconstruire. Par conséquent, les entreprises se sont employées à intégrer plus étroitement leurs systèmes hétérogènes, en parvenant à les faire communiquer entre eux. D'où l'avènement de la SOA et de l'ère *des applications composites*.

En faisant appel à l'architecture SOA, les entreprises *composent* de nouvelles applications en un temps record et automatisent les processus métier traités par les systèmes informatiques existants. Le recours à la SOA simplifie et optimise les opérations de l'entreprise sans sacrifier les investissements déjà réalisés par cette dernière dans des processus et applications logiciels. Voilà l'avenir de l'ère des applications composites.

Comprendre les applications composites

L'*application composite* est au cœur de l'une des plus formidables promesses de l'architecture SOA : l'agilité. Après la mise en place d'un bon nombre de services, la création de nouvelles fonctionnalités procède simplement de l'interconnexion des services appropriés. Et si les besoins de l'entreprise évoluent, me direz-vous ? Pas de problème : il suffit alors de reconfigurer les services pour être en phase avec les nouveaux impératifs. Excellente idée, en effet !

Concrètement, l'application composite requiert de définir l'ensemble approprié d'opérations pour l'ensemble adéquat de services. De même, la technologie utilisée pour transformer ces services en fonctionnalités doit être pilotée exclusivement sous l'angle des besoins de l'entreprise, et être capable de transformer rapidement les applications métier. Croyez-le ou non, une telle technologie existe.

Utiliser la gestion des processus métier (BPM)

La compétitivité des entreprises les plus innovantes ne repose pas uniquement sur les produits qu'elles vendent : l'efficacité avec laquelle elles livrent leurs produits et services et la réactivité dont elles font preuve pour identifier les problèmes et les opportunités nouvelles entrent également en jeu. En d'autres termes, les entreprises acquièrent un avantage concurrentiel en se recentrant sur les processus métier.

Une solution de *gestion des processus métier* (BPM) dispense des fonctionnalités sous la forme d'implémentations automatisées de processus métier concrets, par exemple le traitement allant

de la commande à l'encaissement ou la gestion des réclamations. Associée à une architecture SOA, cette vision orientée processus de la fourniture de fonctionnalités informatiques présente plusieurs avantages :

- ✓ Un processus métier se décompose en étapes pouvant être mises en œuvre à l'aide de services réutilisables, comme la vérification du crédit (CheckCredit), le contrôle des stocks (CheckInventory) et la passation de commandes (PlaceOrder). De ce fait, le BPM constitue une approche orientée métier efficace pour identifier le type de services à créer.
- ✓ Avec une architecture SOA sous-jacente, les processus peuvent être mis en œuvre rapidement en faisant appel à des services résidant en tout lieu de l'entreprise. À mesure que progresse l'adoption de la SOA, d'autres services sont mis à disposition et les délais nécessaires à l'automatisation de nouveaux processus métier ne cessent de diminuer.
- ✓ La définition d'un processus métier dans un BPM est très graphique, à l'instar d'un ordinogramme. Par conséquent, sa création ou sa modification ne requiert que très peu de programmation, voire aucune.
- ✓ Les services SOA intégrés à un processus BPM peuvent faire office de points de surveillance. Certaines technologies telles que le BAM (*Business Activity Monitoring*, supervision des activités métier) s'en servent pour analyser en temps réel les indicateurs clés de performance métier. Il appartient aux entreprises de mettre à profit les résultats obtenus pour analyser leurs processus et les perfectionner.
- ✓ Le BPM offre un langage commun intelligible pour les informaticiens comme pour les responsables métier. Analystes métier et informaticiens peuvent ainsi travailler sur une vue unifiée du processus.

Développer des applications composites

Les technologies sur lesquelles se fondent les applications composites font passer le développement applicatif à un stade supérieur en mettant à profit l'architecture SOA pour autoriser des applications inédites. Par leur biais, vous créez de nouvelles applications en intégrant simplement des services à une

interface utilisateur riche. Très peu de programmation, voire aucune, n'est requise. À la place, ces applications composites utilisent de nouveaux concepts tels que des *portlets* (minuscules applications Web autonomes) et des *mash-ups* (fusion et mélange d'applications pour postes de travail) afin d'amalgamer des informations issues de sources différentes au sein d'une même application, de manière totalement fluide.



Couplées au BPM, les technologies régissant les applications composites permettent de créer des interfaces utilisateur sophistiquées facilitant les interactions humaines avec un processus.

Chapitre 7

Agilité organisationnelle

Dans ce chapitre

- ▶ Structures organisationnelles et comportements.
- ▶ Votre cycle de vie SOA.
- ▶ Distinction des types de cycles de vie SOA.
- ▶ Gestion de l'évolution SOA.
- ▶ Exemple de structure informatique.

L'adoption de la SOA soulève de nouveaux enjeux pour des entreprises habituées à recourir à des solutions spécifiques pour satisfaire leurs besoins applicatifs. Afin de rompre avec la pensée et l'action cloisonnées, de nouvelles structures et de nouveaux processus, sur lesquels s'appuieront l'agilité organisationnelle et le succès de l'adoption SOA, sont indispensables. Ces processus sont généralement regroupés sous l'appellation *cycle de vie SOA*. Associés à la structure organisationnelle adéquate, ils sont un élément clé pour venir à bout des guerres tribales.

De nos jours, la plupart des services informatiques sont contraints de fournir des solutions rentables en respectant les délais. Pour ce faire, elles utilisent des fonctions et des composants techniques et organisationnels mutualisés, et recourent à des initiatives multiprojets pour renforcer les synergies au sein de leur service. Lorsque ces solutions sont couplées à une philosophie d'offre de services (comme dans toute prestation digne de ce nom, hormis technologique), les entreprises se trouvent alors sur la voie de la SOA.

Pour trouver, vous aussi, votre voie SOA, il vous faut posséder les outils nécessaires à l'adoption SOA :

- ✓ **Philosophie** : Raisonner en termes de chaîne de valeur et comprenez que la raison d'être du service est la satisfaction du client.

- ✓ **Méthodologie** : Rompez avec la réflexion orientée applications en implémentant des processus structurés par-delà les frontières entre projets (cycles de vie).
- ✓ **Individus et organisation** : Tenez vos engagements SOA, mais stimulez également votre organisation en vous efforçant de désamorcer les guerres tribales entre groupes.
- ✓ **Technologie** : Faites coïncider vos impératifs philosophiques, méthodologiques et humains en favorisant l'interopérabilité entre disciplines. Conceptuellement, la technologie n'est qu'une pièce dans le puzzle de l'adoption SOA.



Si philosophie, méthodologie, individus et technologie sont combinés correctement, l'adoption SOA est à même de procurer des avantages considérables à une entreprise en termes d'envergure, d'efficacité et, plus particulièrement, d'agilité.

Combattre les guerres tribales

Dans toute entreprise d'une certaine taille, il existe des *tribus*, autrement dit des groupes d'individus qui privilégient leurs propres intérêts au détriment des autres. Ces tribus luttent constamment pour défendre leurs intérêts. Bien sûr, les individus qui les composent ne vont pas sortir leurs massues, comme les hommes des cavernes, et se pourchasser les uns les autres dans les couloirs en exigeant le remboursement *immédiat* de leurs notes de frais. Du moins, espérons-le. Simplement, parce que ces personnes ont envie que leurs besoins soient satisfaits, elles feront passer leurs intérêts avant ceux d'autrui ou de l'entreprise dans son ensemble. Ce trait est humain, et il se vérifie au sein d'un grand nombre d'entreprises.

En règle générale, les tribus informatiques recourent à plusieurs stratégies élémentaires dans leur affrontement :

- ✓ **Rétention d'informations** : Si un groupe « détient » des données, celles consignées dans une base de données par exemple, il est fort possible qu'il tente de bloquer l'accès à celles-ci en fonction de ses propres priorités politiques.
- ✓ **Monopolisation des compétences** : En veillant à ce que personne, hormis votre groupe, ne connaisse les modalités de fonctionnement d'un système, vous protégez vos emplois. L'administration et la maintenance des systèmes informatiques deviennent alors impossibles.

- ✓ **Guerre entre plates-formes** : Il s'agit ici de « biaiser » le système d'information de l'entreprise, en privilégiant un fournisseur ou une plate-forme de développement par rapport aux autres.
- ✓ **Guerre des règles et procédures** : Il est ici question d'obliger les autres groupes à se conformer à vos règles et procédures.
- ✓ **Guerre budgétaire et organisationnelle** : Les disputes pour des ressources, qu'il s'agisse d'effectifs ou de budgets, sont malheureusement une problématique universelle de l'environnement informatique.

Les tactiques des guerres tribales compliquent la vie des autres groupes. Les groupes informatiques survivent en rejetant les coûts et la complexité sur d'autres structures, telles que les centres offshore. Il est, en règle générale, plus intéressant financièrement de travailler avec des centres offshore mais, au final, cette externalisation est une bataille perdue d'avance. Loin de disparaître, les problèmes ne font en réalité que s'accumuler, et leur coût s'envole.



La SOA foisonne d'idées pour transformer la structure organisationnelle, le comportement, les processus, les règles et les systèmes. Inévitablement, ces transformations conviennent aux uns et non aux autres. Il s'agit de s'assurer, dans le processus d'adoption SOA, du respect d'un principe phare : veiller à contenter tout le monde ; sinon, la SOA ne sera qu'une tactique de plus dans la lutte incessante pour le contrôle de l'environnement informatique.

Vivre le cycle de vie SOA

Le cycle de vie SOA permet aux individus de collaborer plus efficacement. Grâce à lui, vous réalisez les objectifs importants de l'adoption SOA de manière structurée. En parallèle, vous laissez à chacun la liberté nécessaire pour innover et s'approprier sa part du processus. Le cycle de vie SOA est un élément essentiel de l'adoption SOA.



Vous devez implémenter le cycle de vie SOA comme un moyen de partager des responsabilités, et non d'en imposer de nouvelles. Un cycle de vie SOA excessivement centralisé, s'ingéniant à redéfinir la granularité des activités au sein des différents groupes, générera une résistance organisationnelle fatale à l'adoption SOA.

Puisque l'objectif de l'architecture SOA consiste à convertir les solutions délivrées dans un format amalgamant composants, services et processus pour créer une solution unique, le cycle de vie SOA couvre non seulement les services techniques, mais aussi les éléments suivants :

- ✓ **Composants réutilisables** : Interfaces utilisateur graphiques, mais aussi structures de données servant à leur composition (services Web, par exemple).
- ✓ **Processus** : Processus métier spécifié par le BPM.
- ✓ **Interfaces utilisateur** : Services et processus.

Le cycle de vie SOA ne couvre ni le code, ni les ressources opérationnelles comme les serveurs d'applications et bases de données. Pour garantir la cohérence de votre modèle, vous devez mettre en adéquation les cycles de vie de la programmation et des opérations avec les concepts SOA.



Puisque l'adoption SOA exige que vous résistiez à la tentation d'imposer de nouvelles responsabilités centralisées, les cycles de vie matures, notamment ceux applicables au code et aux opérations, doivent demeurer là où réside l'excellence.

Identifier vos cycles de vie SOA

En termes d'architecture SOA, il convient d'opérer une distinction entre les services, que l'on se place du point de vue du consommateur ou du point de vue du fournisseur. Puisque les impératifs, tout comme les responsabilités, de ces deux acteurs divergent, deux types de cycles de vie SOA existent :

- ✓ Cycle côté fournisseur.
- ✓ Cycle côté consommateur.

Ces deux types de cycles de vie exigent des intervenants qu'ils soient responsables de la partie du processus leur incombant et, qu'à l'instar d'un guichet, ils valident les activités menées à bien avant de passer le relais à l'intervenant suivant.

Définition des intervenants

Le cycle de vie met en œuvre un flux d'activités et de points de décision entre les différents intervenants prenant part au processus :

- ✓ **Responsable métier** : Le responsable métier définit les spécifications de la nouvelle fonctionnalité ou solution ou du nouveau processus métier à implémenter. Le meilleur moyen de formaliser ces spécifications est d'utiliser la norme de notation pour la modélisation de processus, BPMN (Business Process Modeling Notation). La modélisation des processus procure un environnement permettant de mieux appréhender les impératifs d'implémentation informatique. Il incombe également au responsable métier de définir les exigences non fonctionnelles (telles que la qualité de service) de la fonctionnalité, de la solution ou du processus.
- ✓ **Architecte SOA** : L'architecte SOA analyse les impératifs métier et les décompose en schémas de services et en schémas de processus. Libre à lui de décider de réutiliser un composant existant plutôt que d'en créer un nouveau, auquel cas il appliquera un cycle côté consommateur. Dès lors qu'il propose une implémentation de service ou de processus nouvelle ou modifiée, l'architecte SOA fournit ses spécifications de conception sous forme de diagrammes d'état, de modèles de processus et de schémas d'interface. Il formalise les exigences non fonctionnelles du composant à implémenter (disponibilité, sécurité, performance, etc.).
- ✓ **Développeur** : Le développeur implémente les composants sur la base des spécifications de conception fournies par l'architecte SOA. Il crée également des plans de tests à partir de celles-ci. Pour faciliter la convergence de la technologie et de la méthodologie, le développeur se sert des éléments produits par l'architecte SOA pour les besoins de l'implémentation (à savoir la génération de code ou l'optimisation du modèle).
- ✓ **Responsable qualité** : Le responsable qualité s'appuie sur les contributions respectives du responsable métier, de l'architecte et du développeur pour juger du service ou du processus mis en œuvre. Il se sert ensuite des plans de tests fournis par le développeur pour tester la solution dans un environnement provisoire et vérifie les indicateurs de qualité, effets indésirables et caractéristiques non fonctionnelles.
- ✓ **Opérateur** : L'opérateur reçoit les solutions testées et validées, et les implémente dans le respect des processus informatiques standard afin de les mettre à la disposition des utilisateurs et des consommateurs. Il se sert des exigences non fonctionnelles (NFR) pour mettre en œuvre une solution virtualisée conforme aux engagements de qualité de service (SLA) requis par les consommateurs. Les solutions de gouvernance SOA, au stade de l'exécution, présentent ces caractéristiques par leur application des NFR et SLA.

Exécution des validations

Les *validations* entérinent des étapes et des activités dès lors que vous passez d'un état du cycle de vie à un autre. Par exemple, pour qu'un schéma de service soit autorisé à passer au stade du développement, un examen est réalisé afin de déterminer si cette proposition de conception est conforme aux normes organisationnelles et techniques ; il s'agit également de vérifier son caractère « réutilisable » et d'éviter les doublons. En règle générale, ces validations passent par un centre d'excellence qui accepte ou refuse le passage d'un état à un autre. (Pour plus d'informations sur les centres d'excellence, voir la section précédente, « Définition des intervenants ».)



Chaque fois que possible, automatisez vos validations. Le centre d'excellence est une nécessité, mais tenez-vous-en au partage des responsabilités plutôt que d'en introduire de nouvelles, centralisées. Le partage de responsabilités compense le possible mécontentement de collaborateurs tenus de renoncer à une charge ou à un pouvoir.



Du fait de la convergence entre technologie et méthodologie, nombre de validations peuvent être automatisées en recourant, au stade de la conception, à des technologies de gouvernance appropriées qui appliquent les décisions (règles) tout au long du cycle de vie SOA.

Instauration des contrats

La différence fondamentale entre le cycle de vie côté fournisseur et le cycle de vie côté consommateur est que, dans le second, le consommateur doit accepter les exigences du fournisseur, et inversement. Comme le cycle de vie SOA privilégie déjà l'extraction des exigences non fonctionnelles (NFR) côté fournisseur, le cycle côté consommateur va plus loin en spécifiant celles nécessaires au fonctionnement approprié de sa solution.

Le compromis entre les exigences non fonctionnelles côté consommateur et celles côté fournisseur constitue un *contrat*. Les technologies modernes applicables au cycle de vie et à la gouvernance SOA gèrent ce concept dans une optique de :

- ✓ Suivi des consommateurs.
- ✓ Notification des événements SLA essentiels.
- ✓ Facturation et prestation.

Dans le contexte d'une implémentation SOA donnée, le suivi des relations avec les consommateurs et les fournisseurs est essentiel à la gestion des versions et des changements.

Gérer l'évolution SOA

Dans un environnement modulaire ultra-distribué tel que la SOA, une gestion des changements s'impose. Rien de bien compliqué, me direz-vous ? N'oubliez pas que les responsabilités sont réparties, et qu'au sein de l'entreprise le désir de contrôle et de cohérence est toujours aussi présent. Pour réussir à gérer l'évolution d'un environnement orienté applications existant (non-SOA vers SOA), une certaine discipline s'impose.

Ces quelques directives vous aideront à gérer efficacement votre évolution SOA :

- ✓ Affectez un processus dédié dans la gestion du cycle de vie pour gérer les changements des services et des processus. Sachant qu'au sein d'un environnement distribué, une simple modification est susceptible d'avoir des répercussions sur une multiplicité de solutions sans que celles-ci soient pour autant évidentes d'emblée, tous les intervenants doivent prendre part à cette évolution.
- ✓ En même temps, examinez les contrats consommateur et fournisseur afin d'évaluer l'impact d'un changement.
- ✓ Avertissez les consommateurs actifs en cas de changement imminent ou lorsqu'ils doivent valider une demande de changement.
- ✓ Précisez un « délai de validité » dans vos contrats consommateur/fournisseur. Voilà qui incite au partage de la responsabilité, côté consommateur et fournisseur, et à l'obtention d'un consensus.
- ✓ Instaurez une gestion des versions afin de parvenir à un compromis entre disponibilité et stabilité des nouvelles versions et mises à jour.

La gestion des versions de services et de processus permet de contourner les obstacles inhérents à la gestion de l'évolution. Ces quelques directives vous aideront à mettre en œuvre une gestion efficace des versions :

- ✓ Évitez les changements de comportement, sauf nécessité absolue ; préférez-leur l'implémentation d'un nouveau service ou processus se différenciant des autres.
- ✓ Les modifications apportées à l'interface d'un service ou d'un processus doivent faire l'objet d'une procédure descendante de gestion de versions, aux fins suivantes :
 - Création d'un nouveau modèle d'interface de service/processus.
 - Possibilité d'exploiter plusieurs versions en parallèle en modifiant, par exemple, l'espace de noms d'une interface de service ou de processus.
 - Possibilité de remplacer une interface de service ou de processus caduque en mettant à profit, par exemple, des technologies d'orchestration pour en simuler la sémantique.
- ✓ Dans le cadre de la gestion de versions des services et des processus, affectez un « délai d'expiration » à vos contrats en vue de sensibiliser consommateurs et fournisseurs au caractère non statique de l'architecture SOA et de les préparer à d'éventuels changements.

Analyser un exemple de structure informatique

Il est impossible de préserver une stabilité organisationnelle lorsque souffle le vent du changement. Tiraillées en quasi-permanence, les structures informatiques sont dans un état de transition perpétuelle. Observez celle figurant dans l'organigramme à la Figure 7.1. À l'évidence, un grand nombre de fonctions informatiques sont centralisées : un groupe s'occupe de l'architecture d'entreprise, un autre des services informatiques et un troisième se consacre exclusivement à la gestion des risques.

L'entreprise représentée à la Figure 7.1 comprend deux divisions, possédant chacune ses propres groupes pour le cycle de développement des systèmes (analyse des besoins, développement, qualité, support et documentation).

L'organigramme ne révèle aucunement les intrigues qui se trament dans l'organisation. Il ne rend pas compte des ressentiments, tensions et hostilités qui se sont formés entre groupes.

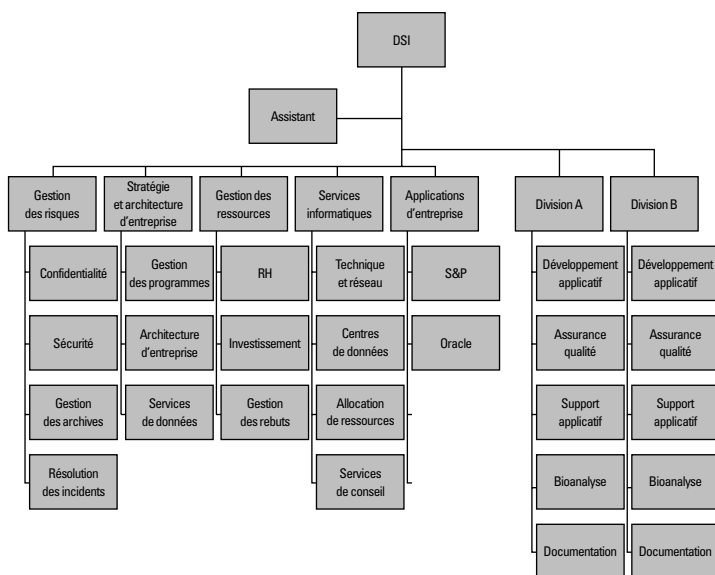


Figure 7.1 : Structure informatique en transition.

Ressentiment au sein d'un pôle informatique centralisé

La croissance de cette entreprise résultant d'une acquisition, au sein de son pôle informatique centralisé coexistent deux systèmes de gestion intégrés (ERP), l'un signé Oracle, l'autre SAP. Chacun des groupes souhaite évincer l'autre car il aspire à ce que l'entreprise adopte *son* progiciel. Ce scénario peut paraître insensé, mais que dire alors des établissements qui ne dénombrent pas moins de 25 systèmes ERP émanant d'éditeurs différents ?

Autre source de tensions au sein de ce pôle informatique centralisé : certaines fonctions sont redondantes au sein des entreprises acquises et acheteuses, et leurs architectes respectifs sont partagés sur la voie à faire suivre à la nouvelle entité.

Par ailleurs, bien que l'organigramme de l'entreprise présente un pôle informatique « centralisé », il s'avère que le DSI a nommé le responsable d'un groupe vice-président de la stratégie d'entreprise et le responsable de l'autre vice-président de l'architecture d'entreprise, dissociant ainsi clairement stratégie et architecture.

L'organigramme ne dit pas non plus que tous deux convoitent le poste de DSI ; ils élaborent ainsi deux feuilles de route distinctes, qui renforcent encore le poids de *leurs* systèmes d'information. Bien entendu, ils ne se communiquent aucune information, et n'attendent qu'une seule chose : que l'autre soit remercié. Tous deux souhaitent que leur vision de l'architecture soit retenue pour :

- ✓ Faire adopter en standard leur plate-forme favorite.
- ✓ Sauver un maximum d'emplois dans leur groupe.
- ✓ Servir leur candidature au poste de DSI.

Chaque vice-président entretient des relations privilégiées avec son éditeur (lequel prend en charge les frais de déplacement vers les lieux exotiques où se déroulent les réunions des clubs d'utilisateurs), et a acquis plusieurs années d'expérience sur ses technologies. Aucun n'abandonnera sa plate-forme de son vivant.

Tensions entre la division A et le pôle informatique centralisé

Même si l'organigramme le place clairement sous l'autorité du DSI, le vice-président de la stratégie d'entreprise fait partie d'une structure « matricielle » où il se retrouve également sous les ordres du vice-président des systèmes d'information pour la division A. Avec deux supérieurs hiérarchiques, tensions et contrariétés sont inéluctables.

Le responsable informatique de la division A, lui, s'est vu donner six mois par son vice-président pour mener à bien un projet d'e-commerce, sous peine d'être licencié. Il a été surpris dans le hall du siège, vociférant sur le responsable de la gestion des risques parce que celui-ci lui impose des contraintes de sécurité et de confidentialité par trop complexes. D'autant qu'en parallèle, l'architecture de l'entreprise lui commande de se conformer au schéma de données standard. Seul hic : ce schéma ne permet de recueillir aucune des informations nécessaires à son projet. Pire, ce format standard utilise bien certains champs de données identiques, mais leur signification est totalement différente. Et, pour couronner le tout, l'un des meilleurs développeurs de l'équipe menace de démissionner pour rejoindre Google.

De son côté, le responsable de la gestion des risques craint qu'une faille de sécurité ne lui coûte son poste. Il est, par conséquent, résolu à faire preuve d'intransigeance dans le domaine de la mise en conformité.

Le responsable informatique de la division A, quant à lui, redoute que le respect de toutes ces contraintes ne retarde son projet, et de porter le chapeau. C'est du moins ce qui s'est passé pour son prédécesseur, remercié durant la fusion.

Défiance entre la division A et la division B

La division B a déjà eu une mauvaise expérience avec la division A. Cette dernière avait sollicité l'équipe en charge des services de données au sein de l'entreprise pour mettre en commun la base de données clients de la division B. Résultat : la division A n'a pas tardé à démarcher les clients de la division B via des campagnes de marketing direct.

Un grand nombre de clients de la division B n'ont pas du tout apprécié ces nouvelles campagnes. Quelques-uns ont mis fin à leurs relations commerciales avec l'établissement, l'un d'eux le menaçant même de poursuites judiciaires et le traitant de spammeur.



Le partage d'informations est une idée formidable. Encore faut-il instaurer des directives régissant la manière de les utiliser. La communication entre tribus est fondamentale !

Tensions entre tribus au cours du cycle de service

Le cycle de service n'est pas si éloigné, dans son domaine, de la construction automobile. Pensez à une chaîne de montages sur laquelle un véhicule passe d'une équipe à l'autre. L'une assemble son moteur, tandis que l'autre travaille sur sa carrosserie, etc. À chaque groupe correspond un rôle précis dans la série de services à dispenser.

Il existe, au sein de cette fabrique de services, des tensions entre certains des spécialistes qui y travaillent. Les développeurs n'ont pas leur pareil pour la programmation, mais ils créent aussi des bugs et infligent des maux de tête aux autres. Voici une liste succincte des tribus informatiques intervenant au cours du cycle de service.

- ✓ Développeurs de logiciels.
- ✓ Architectes d'entreprise.

- ✓ Spécialistes de la sécurité.
- ✓ Opérations informatiques.
- ✓ Assurance qualité.
- ✓ Analystes métier.
- ✓ Développeurs intégrateurs.

Ces groupes ont chacun leurs propres priorités, suivant la manière dont ils conçoivent leur travail. Ils possèdent leurs propres processus informatiques ainsi que leur propre cycle de vie, distinct du cycle de vie SOA. Au départ, ils ne tiennent pas compte du caractère réutilisable des services, de leur interopérabilité, ni d'aucune autre valeur de la SOA.

Dans l'entreprise servant d'exemple, le groupe des développeurs refuse de réutiliser les services de l'annuaire. Ces services n'étant pas correctement documentés, il est, d'après eux, plus rapide d'en programmer de nouveaux. Ces développeurs rechignent également, du point de vue conceptuel, à créer une dépendance externe au niveau de leur code car, en cas d'échec, ils savent qu'ils seront incriminés même si un tiers est à l'origine du bug. Ils ne se fient pas au code d'autrui.

Chapitre 8

Qui finance la SOA ?

Dans ce chapitre

- Financement de votre SOA.
- Mise en place de mesures incitatives au sein de votre entreprise.

Les structures informatiques n'ont cessé d'optimiser l'efficacité et la productivité pour se valoriser au sein de leur entreprise. Auparavant, elles misaient sur de nouvelles technologies, comme les technologies orientées objet et client-serveur, pour parvenir à leurs fins. Malheureusement, il était difficile de quantifier les avantages liés à leur utilisation et encore plus de les justifier. En outre, comme leurs objectifs d'optimisation s'échelonnaient sur une période extrêmement longue, ces équipes informatiques passaient de l'enthousiasme à la désillusion dès lors que leurs solutions ne leur procuraient pas très vite des résultats quantifiables.

En règle générale, il est bien plus aisé de rendre compte de la valeur d'un achat que de celle d'une action. Puisque la SOA se « fabrique », et ne s'achète pas, il s'agit de trouver le moyen d'inciter tous les collaborateurs de votre entreprise à y adhérer sans réserve.

Comment financer votre SOA

La plupart des études de cas évaluent les initiatives informatiques en termes de retour sur investissement. Voici comment calculer ce retour sur investissement :

1. **Appréciez les efforts nécessaires à la mise en place d'une nouvelle fonctionnalité.**
2. **Évaluez les avantages induits en termes d'efficacité, de productivité et de délais de mise sur le marché.**
3. **Calculez sous quel délai les recettes ou économies dégagées par cette fonctionnalité l'emporteront sur le coût de son implémentation.**

Les avantages induits par une initiative SOA vont cependant au-delà de ceux mis en évidence par une étude de cas car ses objectifs fondamentaux sont les suivants :

- ✓ **Continuer, autant que possible, à faire usage des technologies et processus informatiques en place.** Ce faisant, l'entreprise tire le maximum de bénéfices (retour) de ses précédents investissements. Dans le langage de l'entreprise, il s'agit de faire durer le retour sur investissement des initiatives précédentes pour maximiser le retour sur actifs.
- ✓ **Tirer le meilleur parti des synergies informatiques.** La réutilisation de composants à l'échelle de l'entreprise libère des ressources qui peuvent alors se recentrer sur l'automatisation et l'innovation, ainsi que sur la réduction des coûts.
- ✓ **Parvenir à une certaine souplesse afin d'accélérer la composition de solutions nouvelles.** Les applications composites accélèrent la fourniture de solutions et procurent toute liberté pour choisir des technologies d'implémentation de services reposant sur les ressources et compétences actuellement disponibles, sans en imposer aucune.

Il s'agit d'éviter que ces objectifs ne deviennent trop ambitieux et ne soient fixés à trop long terme, au risque de n'être jamais atteints. L'adoption de la SOA propose trois approches que vous pouvez épouser pour choisir une argumentation. Ces approches vous aideront à obtenir l'assentiment de la direction, ainsi qu'un budget.

Approche tactique

Nombre d'environnements informatiques qui ont bâti leur expansion au fil du temps suscitent des doutes quant à leur capacité à assurer la maintenance des applications existantes. Or, ils doivent être capables de gérer les coûts et risques induits par la poursuite de l'exploitation de ces ressources applicatives à l'avenir. Par ailleurs, eu égard à la dynamique du marché et à l'évolution de leurs besoins, il est préférable que ces applications s'intègrent les unes aux autres pour produire des solutions unifiées.

Il est risqué de remplacer ces applications par de nouvelles, tout juste écrites, ou par des progiciels, d'autant que cette initiative occasionnerait un renchérissement des coûts. Un peu comme si vous éteigniez votre vieux système un soir en espérant que le nouveau fonctionne exactement de la même manière le lendemain matin. Le risque encouru est énorme.



Le meilleur moyen de passer d'une application (non-SOA) existante vers une architecture SOA est de la fractionner en composants autonomes, ou *encapsulés*, dotés d'interfaces bien définies (de sorte que, de l'extérieur, vous ne puissiez rien voir de ce qui se passe à l'intérieur). Ce faisant, vous supervisez chacune des phases de la transition et obtenez un système extrêmement souple, où le coût d'ensemble de l'application se révèle moins élevé que la valeur de chacun de ses composants.

À partir du moment où vous décomposez correctement une application, la direction est mieux à même d'apprécier que celle-ci a gagné en souplesse. L'entreprise possède alors toute latitude pour décider de la manière d'aller de l'avant. De plus en plus, elle se sent maître de son destin, et non plus tributaire d'une application donnée. À noter qu'en règle générale il est bien plus aisé d'assurer la maintenance, voire le remplacement, d'une application fractionnée en composants.



Le découplage des composants d'une application améliore la qualité du système car les dépendances entre composants deviennent explicites, l'impact d'un changement devient prévisible et les modifications occasionnent moins d'effets indésirables.

Les avantages tangibles induits par la modernisation de ces applications et leur intégration au sein de l'architecture SOA sont les suivants :

- ✓ Maîtrise des coûts (absence de mauvaises surprises).
- ✓ Rapidité des résultats (intégration).
- ✓ Satisfaction des utilisateurs (applications unifiées).
- ✓ Baisse du coût de l'assurance qualité.

Réflexion stratégique

Si vous entendez implémenter une infrastructure SOA pour pouvoir composer de nouvelles solutions plus rapidement, vous devez mener une réflexion stratégique. Au sein d'un environnement composite, fonctions et composants sont partagés et réutilisés à l'échelle de l'entreprise, dans une optique d'interopérabilité, en vue de fournir un modèle reproductible de solutions. Une implémentation SOA stratégique va de pair avec une gouvernance organisationnelle et technique qui mettra en adéquation planification et exécution informatiques au sein

des projets comme des divisions (voir le Chapitre 7 pour plus d'informations). L'objectif consiste ici à mettre à profit les synergies et à éliminer les redondances, en se focalisant sur l'interopérabilité et la réutilisation.

Une adoption de la SOA stratégique impose que vous vous recentriez sur la mise à disposition de solutions plutôt que sur leur création de toutes pièces. Pour que le succès soit au rendez-vous, la philosophie SOA doit impérativement devenir le credo de chacun des collaborateurs rattachés au pôle informatique. Le meilleur moyen, pour identifier des solutions SOA, consiste à évaluer les impératifs métier à valeur ajoutée couvrant les domaines techniques et fonctionnels. Ne laissez jamais votre stratégie SOA devenir une étude de cas ; obtenez l'adhésion de l'équipe dirigeante. Formez vos collaborateurs et incitez-les à soutenir votre stratégie.

Enfin, vous devez évaluer la réussite de votre stratégie SOA. À vous de prouver que des solutions composites reposant sur votre infrastructure SOA accélèrent la mise sur le marché tout en renforçant la souplesse. Plus tôt vous présenterez ces données, plus vous augmenterez vos chances de succès.

Démarche pratique : le BPM

Une méthode de financement efficace de la SOA consiste à la coupler au BPM, c'est-à-dire à la gestion des processus métier. Le BPM ne se contente pas de dispenser, en toute fiabilité, des solutions à l'entreprise ; il lui procure également une plate-forme prête au changement, se démarquant par sa souplesse. L'extraction de processus en modèle exécutable accélère considérablement l'implémentation de modifications par rapport au développement de code traditionnel. Et, parce que le BPM procure une certaine souplesse, il lui faut une infrastructure souple pour tenir ses engagements. L'association SOA-BPM assure une cohérence sans programmation « en dur ».

Si vous développez des processus au sein d'un environnement BPM, vous identifiez en un clin d'œil les services que doit fournir la plate-forme SOA pour que ces processus puissent être automatisés. Le BPM donne libre champ à l'initiative SOA et ce, de manière telle que la SOA est d'emblée perçue comme une source de plus-value, et non de coûts et de complexité. Parce que la SOA est considérée du point de vue de l'entreprise, son infrastructure sert l'objectif prévu : fournir un service de valeur, et non faire office d'interface.

Proposer des incitations organisationnelles

La réussite de l'adoption de la SOA est indissociable de la promotion des plus grands atouts de son infrastructure à travers des actions de la valorisation de ses bénéfices. Au final, c'est votre capacité à motiver les collaborateurs au sein de l'entreprise et à les persuader d'adopter la SOA qui contribuera à la réussite, ou à l'échec, de votre initiative. Il est deux groupes à ne surtout pas négliger dans votre démarche :

- ✓ **Fournisseurs de services** : Dans une entreprise accoutumée à proposer des fonctionnalités à partir d'applications isolées, les collaborateurs ne sont pas forcément disposés à dispenser et à partager des services. Non pas par malveillance, mais parce qu'ils en cernent mal les avantages et ne souhaitent pas endosser des responsabilités supplémentaires. Plus les consommateurs sont nombreux à utiliser un service, plus un fournisseur doit consacrer de temps à la maintenance de l'application et à la gestion des réclamations. Par ailleurs, à partir du moment où un service a été mis au point et que le coût de son développement a été imputé sur un projet particulier, les autres peuvent-ils tout simplement en profiter gratuitement ?
- ✓ **Consommateurs de services** : Hier encore, le héros d'un projet était le développeur le plus productif, offrant la meilleure qualité, dans les délais les plus courts. La perspective de réutiliser des services d'autres fournisseurs peut être perçue comme une menace pour ceux qui ont l'impression de ne plus produire quelque chose par eux-mêmes.

Les difficultés inhérentes à l'adoption de la SOA exigent des incitations organisationnelles allant bien au-delà des dispositifs traditionnels. En voici quelques-unes qui méritent d'être étudiées :

- ✓ **Partage des services** : Plus les services créés, puis partagés par le fournisseur, sont nombreux, plus l'incitation réservée au développeur et au projet doit être intéressante.
- ✓ **Utilisation des services** : Mettez en avant la personne qui utilise le plus de services émanant d'autrui. Lorsque vous évaluez le projet et l'efficacité de son implémentation, passez au crible les points fonctionnels mis en œuvre et les services réutilisés en les rapportant à la durée d'implémentation, et instaurez des incitations pour les collaborateurs en fonction de ce critère.

- ✓ **Facturation des services** : Les consommateurs capables d'implémenter davantage de points fonctionnels en moins de temps, en réutilisant des services, sont tenus de refacturer certains écarts de coût au fournisseur du service.
- ✓ **Maintenance des services** : Le fournisseur de services doit être récompensé dès que ses services commencent à être utilisés, à la fois sur le plan de la facturation et de la visibilité organisationnelle. Le héros de l'entreprise SOA est celui qui produit le service le plus utilisé par les autres. Les changements imposés par les consommateurs nécessitent la mise en œuvre d'un modèle de partage des coûts.
- ✓ **Valeur des services** : Attachez-vous à la valeur du service et considérez ce dernier comme une chance, et non comme un fardeau. Le consommateur de services doit également se voir comme un fournisseur de services pour véhiculer la notion de chaîne de valeur informatique.



Vous devez parvenir à un juste équilibre, en matière d'incitations, afin de mettre à l'honneur et d'encourager les collaborateurs qui n'hésitent pas à partager et réutiliser des services entre des domaines jusqu'alors isolés. Avec des mesures incitatives adéquates, une entreprise pilotera son adoption de la SOA « de l'intérieur », et non « d'en haut ».

Chapitre 9

Votre premier projet SOA

Dans ce chapitre

- Mise en œuvre d'un projet SOA.
- Choix de la bonne direction.
- Modalités d'introduction de l'automatisation.

L'informatique pâtit d'une suite apparemment aléatoire de projets, qui ignorent le passé et hypothèquent l'avenir. Ce n'est pas simplement le distinguo « bon pour moi, mauvais pour toi » qui nous vaut des ennuis, mais aussi des observations du type « tout semble aller bien pour l'instant, au diable les conséquences ».

Aussi séduisante soit-elle, la version finalisée du plan d'architecture SOA, dites-vous bien qu'il vous sera difficile de n'en faire qu'une bouchée. Par conséquent, il vous faut adopter l'architecture SOA projet par projet.

Mais en vous entourant des partenaires ad hoc et en nouant les alliances appropriées, vous créez une séquence rationnelle de projets prenant appui sur le passé pour construire l'avenir. Ce chapitre explique comment surmonter les difficultés informatiques rencontrées, en entreprenant un projet à la fois.

Lancer un projet SOA

Parce que l'architecture SOA est une vision du monde, un projet SOA, de par sa souplesse, est à même de résoudre un nombre potentiellement illimité de problématiques métier différentes. Vous n'aurez donc aucun mal à le financer et à le justifier.

Si votre premier projet est légitimé par la gestion des processus métier (BPM) par exemple, qu'est-ce qui lui vaut le qualificatif de SOA ?

Un projet SOA obéit aux normes, procédures et règles SOA convenues au sein de votre entreprise et contribue à la réalisation des plans d'architecture et d'organisation SOA. Par conséquent, pratiquement tout projet informatique peut se transformer en projet SOA. Voici quelques exemples concrets :

- ✓ Gestion des processus métier (BPM).
- ✓ Intégration de données mainframe.
- ✓ Intégration à la suite de fusions-acquisitions.
- ✓ Données de référence.
- ✓ Données maître.
- ✓ Qualité des données
- ✓ Modernisation des systèmes existants.
- ✓ Architecture d'entreprise.
- ✓ Rationalisation des applications, schéma de données.
- ✓ Intégration.
- ✓ Gestion des identités.
- ✓ Regroupement des centres de données.
- ✓ Initiatives réglementaires.
- ✓ Méthodologies d'agilité/itération.
- ✓ Normalisation des canaux (terminaux mobiles, agences, Internet, centres d'appels, guichets automatiques).

Choix des premiers services

Les premiers services mutualisés proposés dans un registre/référentiel jouent un rôle décisif, en ce sens qu'ils susciteront ou non l'enthousiasme de l'entreprise et obtiendront ou non sa validation du concept SOA. Prenez soin de sélectionner des services d'intérêt universel, offrant une réelle valeur ajoutée.

Une astuce consiste à retenir un processus ou un service absolument indispensable sur le plan réglementaire ou comportemental. Par exemple, une base de données de contacts centralisée que tous les services doivent tenir à jour car elle sert à l'envoi des bulletins de paie.

Résultat : vous faites de la conformité SOA un passage obligé pour toutes les autres structures.

Choix des alliés SOA

Les projets SOA amalgament nombre de composantes informatiques et métier. La SOA peut ainsi réunir des professionnels du développement applicatif et des spécialistes ERP ou encore des experts en assurance qualité, et rassembler des développeurs intégrateurs, des responsables de portail B2B ou des analystes de processus BPM. Ne soyez pas surpris par ces alliances et mettez à profit ces nouvelles relations.

Conserver sa trajectoire

Pour que la fusée demeure orientée dans la bonne direction, des corrections constantes de trajectoire doivent être effectuées au sein d'un projet, mais aussi entre les projets successifs.

Assurez-vous en permanence de la conformité informatique avec les règles et procédures de votre plan directeur SOA. Si les services créés ne sont ni réutilisables ni interopérables, c'est que vous êtes encore loin de réaliser votre plan d'architecture SOA. Ne perdez pas non plus de vue les résultats de l'entreprise, pas seulement pour trouver des arguments qui justifieront le financement de votre prochain projet SOA, mais aussi pour avoir la certitude que les opérations de recherche, de mise en relation et de réutilisation de vos premiers services s'effectuent correctement côté consommateurs, et que ces derniers en tirent avantage. Procédez sans tarder à des corrections de trajectoire si vous constatez des déficiences au niveau des indicateurs informatiques ou métier.

Mesure de la conformité informatique

Par définition, un projet SOA est un projet informatique qui se conforme à vos règles et procédures SOA et vous rapproche de la réalisation de votre plan directeur. Si vous n'êtes pas en mesure de prouver ce rapprochement, votre projet n'a de SOA que le nom.



Certains qualifient parfois leur premier projet de SOA même s'il déroge aux règles et procédures SOA. Vous ne mènerez jamais à bien votre plan directeur si cette tactique se répète un peu trop, mais elle peut s'avérer payante pour obtenir l'adhésion de la direction ou des fonds supplémentaires destinés à l'infrastructure de gouvernance.

La conformité aux règles et procédures SOA peut se mesurer à l'aide des indicateurs suivants :

- ✓ Nombre de fois où, faute d'utiliser un service existant, un nouveau doit être créé.
- ✓ Nombre de fois où un service doit faire l'objet d'une nouvelle version.
- ✓ Nombre d'utilisateurs d'un service.
- ✓ Délai de création d'un nouveau service.
- ✓ Nombre de services anarchiques.
- ✓ Engagements de qualité de service (SLA).
- ✓ Nombre de demandes de modifications et de nouvelles versions.
- ✓ Nombre de demandes de support.
- ✓ Utilisation de l'infrastructure de services.

L'*indicateur de coût total sur le cycle de vie* permet de mieux appréhender, par exemple, le coût induit par la génération de nouveau code (mais aussi de nouveaux bugs, de nouvelles activations, de nouveaux tests et d'une nouvelle gestion) au cours du cycle de vie comparé à une réutilisation de l'existant. Plutôt que d'évaluer ce coût pour un seul groupe, efforcez-vous de l'évaluer pour l'ensemble des groupes.



Servez-vous de ces indicateurs comme outil de motivation. Associez-les à des incitations financières ou faites-en les objectifs avoués de l'entreprise : il est important de mettre en corrélation le comportement des collaborateurs et les chiffres.

Mesure du retour sur investissement

Du point de vue financier, un projet informatique est un succès s'il dégage un retour sur investissement. Il est important que votre premier projet SOA satisfasse cette condition. Bonne nouvelle : en mettant votre projet SOA au diapason des objectifs métier, vous lui faites quitter le royaume de la « SOA pour la SOA ».

La SOA va bien au-delà de l'informatique : la SOA est l'entreprise, et ses utilisateurs veulent des réponses aux questions suivantes :

- ✓ Combien ce service nous coûtera-t-il ?
- ✓ Quel est son coût de maintenance ?

- ✓ Combien avons-nous gagné/dépensé en utilisant ce service ce mois-ci ?
- ✓ À qui les changements sont-ils facturés ?



Mettez à profit certaines de ces mesures pour proposer des incitations organisationnelles (primes, promotions, objectifs de management).

Automatiser les règles et procédures

Pour réduire au maximum la résistance au sein de l'organisation, introduisez graduellement les règles et procédures.

Démarche graduelle



Dans votre premier projet SOA, pourquoi ne pas introduire un registre pour coordonner le partage de services entre tribus informatiques. Ce faisant, vous instaurerez un nouveau processus de mutualisation de services. Au final, ce registre, qui permettra de superviser les règles de conception, assurera la coordination des processus de cycle de vie SOA.

Cependant, dès le premier jour, l'introduction officielle d'un ensemble de services publiés suffit à entamer le processus d'éviction des services anarchiques mal conçus, à couplage fort et point à point.

N'hésitez pas à faire clairement savoir que quiconque fait appel à des services non répertoriés dans le registre le fait à ses risques et périls, et que votre entreprise ne garantit ni la fiabilité, ni la sécurité, ni la qualité de ces services anarchiques.

À quel moment instaurer l'infrastructure de gouvernance

L'infrastructure de gouvernance se compose du registre/référentiel et du système de gestion d'exécution.

Le registre/référentiel sert de *point de contrôle* pour les règles de conception. Tant que vous n'aurez pas instauré ce composant, il sera difficile d'automatiser leur mise en œuvre.

Il est encore plus urgent de définir, le plus tôt possible, un point de contrôle sur le système de gestion d'exécution pour la mise en œuvre des règles d'exécution. Pourquoi est-ce si important ?

En l'absence de points de contrôle de gouvernance, les consommateurs accèdent directement aux services. Il n'existe aucun système, tel que le registre/référentiel, permettant de savoir qui consomme quel service. Il n'en existe pas non plus qui, comme le système de gestion d'exécution, puisse garantir la fiabilité et la qualité du service. Malheureusement, les services qui ont ainsi été approchés directement deviennent captifs (voir le Chapitre 5).

Si vous autorisez la mise en relation directe avec les services sans aucune infrastructure de gouvernance, vous rencontrez les problèmes suivants :

- ✓ Vous ignorez l'identité de l'utilisateur du service.
- ✓ En cas de modification du service, le lien avec chaque consommateur est rompu.
- ✓ Si un service est défaillant, il peut déséquilibrer tous les autres qui en sont tributaires.
- ✓ Un utilisateur risque de refuser l'accès d'un service à autrui, en consommant la totalité des ressources serveur supportant ce service.

Si vous autorisez un accès direct aux services sans aucune infrastructure de gouvernance, cela revient à interdire tout suivi et à vous éloigner encore un peu plus de vos objectifs SOA.



Raison pour laquelle nous vous recommandons d'instaurer votre infrastructure de gouvernance très en amont, avant même de déployer le premier service de production. Nous préconisons, en revanche, d'intégrer progressivement de nouvelles règles au sein de cet environnement.

Après avoir ajouté une nouvelle règle, examinez systématiquement les indicateurs de conformité informatique et métier afin de vous assurer de son bon fonctionnement.

Chapitre 10

La fusée SOA

Dans ce chapitre

- Une démarche SOA inspirée de l'astronautique, matérialisée par la fusée SOA.
- Choix de la bonne direction.
- La SOA érigée en habitude.

Dans ce chapitre, nous décrivons notre approche SOA inspirée de l'astronautique, et la manière dont elle s'applique à votre premier projet SOA pour coordonner les suivants, potentiellement décousus, au sein d'un programme SOA qui réalise vos plans d'architecture et d'organisation.

Examiner la fusée SOA

Même après un lancement réussi, vous demeurez dans la phase critique de la SOA et risquez de retomber dans les errements du passé. Tant que vous n'avez pas mis votre fusée SOA en orbite, il vous faut continuer à lutter contre la gravité et la propension à revenir à d'anciens comportements défensifs et tribaux.

Du projet SOA au programme SOA

Vous devez entreprendre plusieurs projets SOA et les mener à bien pour que vos plans SOA puissent être réalisés. Voici quelques principes inspirés de l'astronautique qui viendront à votre secours :

1. Orientez toujours la fusée dans la bonne direction.

Pour que la fusée SOA demeure orientée dans la bonne direction, vous devez effectuer des relevés de mesures et recalculer la trajectoire en permanence.

Cela vaut pour un projet (auquel sont apportées de légères retouches, comme l'introduction d'une nouvelle

règle ou d'un système de primes) comme pour des projets successifs (sur lesquels vous tirez les leçons de vos erreurs et rectifiez le tir).

2. Maintenez la force de poussée.

Pour maintenir la force de poussée, motivez l'équipe en charge de l'implémentation, mais n'oubliez pas que les bailleurs de fonds, décideurs et autres intervenants doivent être mobilisés également.

Pour ce faire, nous devons insister sur l'effet « boule de neige » des avantages SOA au fur et à mesure de l'implémentation d'autres projets, et nous servir de ces critères pour justifier le financement d'autres projets, recruter d'autres « tribus » SOA alliées et encourager la mise en conformité de l'architecture.

3. Ne relâchez pas vos efforts jusqu'à la mise en orbite.

Pour atteindre la « vitesse de libération » de votre fusée SOA, vous devez automatiser les règles et procédures utilisées dans votre cycle de service. Vous parvenez à cet état d'apesanteur que nous qualifions de *mise en orbite* dès lors que l'ensemble de votre cycle de vie est orienté service.

Notre approche SOA inspirée de l'astronautique est valable à la fois pour votre plan d'architecture et pour votre plan d'organisation, comme illustré à la Figure 10.1.

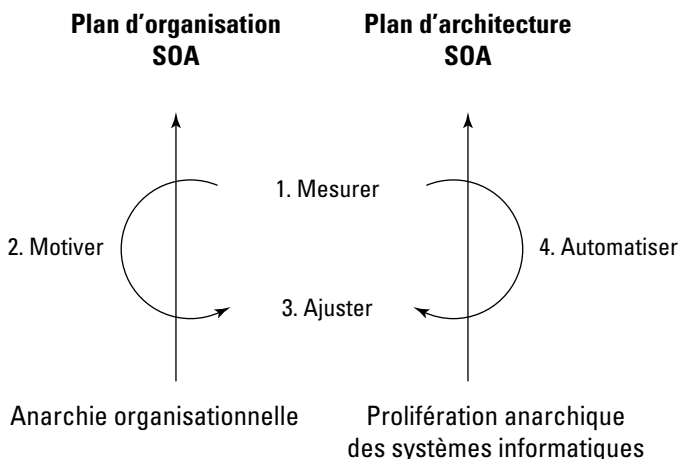


Figure 10.1 : Une approche SOA efficace associe plan d'architecture et plan d'organisation.

La phase critique de la SOA

Jusqu'à la mise en orbite, vous luttez contre la gravité. Durant la phase critique de la SOA, vous n'avez d'autre choix que de dépenser constamment votre énergie, et les occasions sont légion : budget, capital politique, soutien de la direction, investissement dans l'entreprise, attention de l'encadrement et actions en faveur du cycle de vie SOA. À défaut, vous risquez (tout comme une fusée) la désintégration, c'est-à-dire un retour à la situation antérieure à la SOA.

Une fois en orbite, vous progressez dans la réalisation de votre plan directeur sans fournir le moindre effort supplémentaire.

La mise en orbite signifie-t-elle que vous avez mené à bien votre plan directeur SOA ? Pas nécessairement. Elle signifie, en revanche, que vous avez atteint un point de non-retour dans votre programme SOA. Les bailleurs de fonds rêvent dès lors de n'avoir plus à financer que des projets SOA. Le personnel informatique produit des services sans effort. D'autres collaborateurs n'ont qu'une envie : prendre part à cette entreprise.

Orienter la fusée dans la bonne direction

Pour parvenir à destination, une fusée effectue des relevés de mesures et recalcule sa trajectoire en permanence. Pour corriger la vôtre, selon notre méthode inspirée de l'astronautique, il vous faut identifier précisément votre objectif (lequel correspondra, si tout va bien, à votre plan directeur) et maintenir votre poussée (motivation de l'organisation) pour vaincre la gravité.

Cependant, pour être en mesure de rectifier votre trajectoire, encore faut-il que vous disposiez d'indicateurs corrects. Les principaux indicateurs utilisés *au sein* de projets SOA sont exposés au chapitre précédent. Nous nous intéressons ici à ceux qui nous permettront de faire valoir la force d'accélération et les gains dégagés sur des projets SOA *successifs*.

Force d'accélération et indicateurs de plus-value informatique

Vous pouvez contribuer à motiver votre personnel informatique, à tous les niveaux, en lui présentant le potentiel d'accélération induit par certains indicateurs clés, tels que ceux-ci :

- ✓ **Rapidité de déploiement de nouveaux services** : Éprouvez-vous toujours certaines difficultés pour définir un *service* au sens où l'entend votre organisation ? Quel est le degré d'agilité des processus inhérents à votre cycle de vie ?
- ✓ **Temps morts du cycle de service** : Combien de temps vos services se retrouvent-ils bloqués au cours de phases de validation, ou pendant combien de temps leur déploiement est-il rendu impossible en raison d'un problème d'activation, d'autorisation de sécurité ou autre ? Comment automatiser certaines étapes ou créer des rapports et des alertes sur des services bloqués ?
- ✓ **Nombre d'étapes ponctuant le cycle de vie** : Pour optimiser le cycle de vie, mieux vaut moins d'étapes que trop.
- ✓ **Proportion de nouveaux services par rapport à ceux réutilisés** : Une proportion importante de services réutilisés témoigne d'une force d'accélération et d'une plus-value importante.
- ✓ **Coût sur la durée d'un nouveau service** : En améliorant les résultats de cet indicateur au fil du temps, vous appréhendez mieux la force de propulsion que vous appliquez à votre cycle de vie SOA.



Mais en fin de compte, à moins d'associer ces indicateurs à des promotions, rémunérations, primes, émulations entre groupes et bilans de performance, il est difficile de susciter l'intérêt des développeurs pour ces objectifs.

Force d'accélération et indicateurs de plus-value métier

Vous devez, enfin, être en mesure de faire valoir la force d'accélération dégagée à mesure que vous multipliez les projets. Pour ce faire, commencez par cerner la dynamique d'adoption, côté consommateurs de services, afin d'identifier les indicateurs de performance les plus significatifs.

Voici certaines variables qui devraient gagner en importance à mesure de l'installation de votre SOA :

- ✓ Nombre de services réutilisables.
- ✓ Nombre d'applications liées à chaque service.
- ✓ Population d'utilisateurs par application.

- ✓ Volume d'utilisation par utilisateur.
- ✓ Sensibilisation aux services réutilisables.
- ✓ Enthousiasme pour les services.
- ✓ Nombre de modèles d'utilisation.

Si le nombre de services augmente, et qu'à chaque service correspond un nombre croissant d'applications, elles-mêmes exploitées par un nombre croissant d'utilisateurs, sollicitant individuellement de plus en plus chaque application, la plus-value fait alors l'objet d'une force d'accélération peu commune. Le phénomène est encore plus spectaculaire avec la réutilisation de services au sein de processus et d'applications composites.

Systèmes de guidage organisationnel

Sitôt vos indicateurs d'efficacité informatique et métier en place, attellez-vous aux changements organisationnels qui vous permettront de concrétiser votre plan d'organisation. Voici quelques exemples possibles :

- ✓ Restructuration organisationnelle.
- ✓ Modifications apportées à la rémunération des collaborateurs.
- ✓ Corrélation entre réussite organisationnelle et indicateurs SOA.
- ✓ Formation ou recrutement de nouveaux collaborateurs.
- ✓ Promotions professionnelles et licenciements.
- ✓ Modification des rôles et descriptifs de postes.
- ✓ Modification des modèles de financement pour une informatique mutualisée.

L'effet de ces changements peut être déstabilisateur pour votre adoption de la SOA. Supervisez donc l'impact de chacun d'eux afin de déterminer s'il convient d'y renoncer, de l'accentuer ou de l'ajuster.



Ne commettez pas l'erreur de penser que les changements organisationnels ont une incidence exclusivement sur l'entreprise, et les changements architecturaux uniquement sur le personnel informatique. Informaticiens et responsables métier ont besoin d'une motivation suffisante pour changer leurs comportements.

La force de propulsion de votre fusée SOA dépendra d'une variable décisive : l'enthousiasme, l'élan et l'engouement suscités par votre architecture SOA. Vos collaborateurs doivent y voir des perspectives d'évolution professionnelle pour se joindre au mouvement. Il est, par conséquent, indispensable de veiller à conserver un degré élevé de motivation au sein de chaque groupe d'intervenants.

Systèmes de guidage architectural

Tout comme les changements organisationnels concourent à la réalisation du plan d'organisation, les changements architecturaux favorisent celle du plan d'architecture SOA. Voici quelques exemples possibles :

- ✓ Instauration d'une nouvelle règle de gouvernance.
- ✓ Ajout de nouvelles phases ou étapes de processus dans le cycle de vie SOA.
- ✓ Gestion des nouveaux impératifs côté consommateur.
- ✓ Gestion des versions de service.
- ✓ Modification des modalités de description ou de découverte de services.
- ✓ Modification du plan d'architecture SOA.



Grâce à l'automatisation, ces changements se traduisent par une force d'accélération profitable à votre plan d'architecture SOA. À mesure que vous poursuivrez l'automatisation des règles et procédures au sein de votre cycle de vie SOA, vous accélérerez la réalisation de votre plan d'architecture SOA.

Motivation de vos collaborateurs

Lorsqu'une fusée atteint sa vitesse de libération, la force de poussée la propulse vers le haut tandis que la force de gravité s'évertue à l'entraîner vers le bas. Dans cette analogie, la *poussée* représente la motivation, l'enthousiasme déployé par le responsable de l'implémentation, l'adhésion de la direction et le financement associé à la mission SOA.

Au final, les collaborateurs trouvent leur motivation dans l'aspect financier, les objectifs de management et les directives de leur supérieur hiérarchique. Sans ces incitations concrètes, l'adoption de la SOA risque de se solder par un fiasco. Voici quelques réflexions qui précèdent généralement une prise de décision :

- ✓ **Données** : « Parce qu'elles sont exactes. »
- ✓ **Votre supérieur hiérarchique** : « Parce que j'en ai décidé ainsi. »
- ✓ **Argent** : « Parce que cela me rapporte. »
- ✓ **Engagement** : « Parce que j'ai promis. »
- ✓ **Pressions des collègues** : « Parce que les autres le font. »
- ✓ **Amitié** : « Parce que vous m'êtes sympathique. »
- ✓ **Culture** : « C'est notre façon de faire. »

La mise en place d'une émulation entre groupes et individus fonctionne également. Il est essentiel d'appréhender, de quantifier et d'ajuster les comportements organisationnels.

Motivation du personnel informatique

En règle générale, les développeurs de logiciels détestent être gouvernés. Ils tirent une grande fierté, et prennent énormément de plaisir, à développer des solutions adaptées à chaque problématique métier. D'où leurs récriminations lorsque vous commencez à leur parler de conformité aux règles et de réutilisation.



Traitez vos développeurs en véritables professionnels. Exposez-leur systématiquement la logique d'un changement de comportement, en termes de coût, de chiffre d'affaires, de risques ou de tout autre facteur quantitatif. Automatisez, chaque fois que possible, les étapes de gouvernance et rendez-les aussi indolores que possible. Assurez-vous que les développeurs soient représentés par un porte-parole compétent, sachant se faire entendre, au sein de votre centre de compétence.

Motivation des décideurs

Les décideurs ont du mal à rester concentrés très longtemps. Or, il faut bien l'admettre, la réalisation du plan directeur SOA peut prendre plusieurs années. Il vous faut des décideurs motivés, capables d'assurer le financement, de faire preuve d'autorité pour les changements organisationnels, mais aussi de leadership.

Exposez-leur vos indicateurs sous forme de tableau de bord en recourant à des outils décisionnels (voir Figure 10.2). Permettez-leur de s'attribuer le mérite des résultats obtenus à partir de vos projets. Conviez votre supérieur hiérarchique à intervenir dans le cadre de conférences consacrées à la SOA.

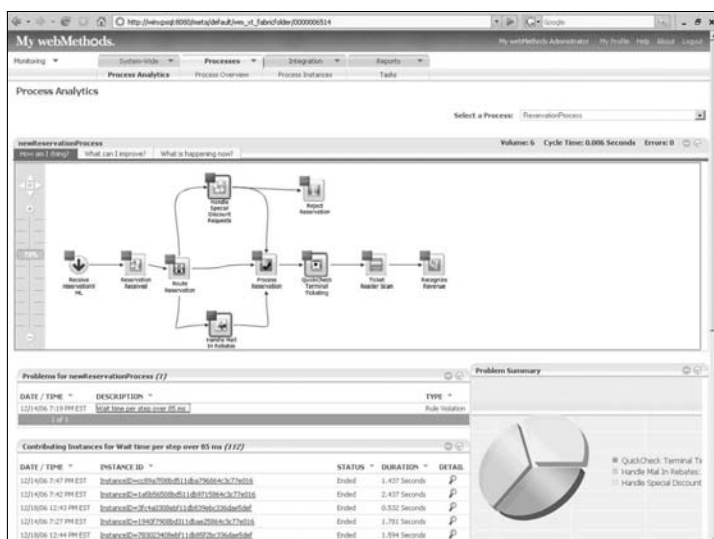


Figure 10.2 : Tableau de bord décisionnel type.

Motivation des divisions de l'entreprise

Il nous faut l'adhésion de l'entreprise pour financer et partager le coût de l'adoption de la SOA et justifier cette dernière.

Chaque division de l'entreprise a la réutilisation en horreur car, à ses yeux, ses besoins sont particuliers. Aidez les différentes divisions à comprendre qu'une solution générique est à même de leur procurer une certaine souplesse, aujourd'hui comme demain. Sachez que celles-ci détestent également partager.



Assurez-vous de l'existence de mécanismes permettant aux divisions de partager à la fois les coûts et les avantages financiers de votre infrastructure mutualisée et présentez-leur des techniques permettant de visualiser la plus-value du système, par exemple des logiciels BAM.

Motivation de tous les autres intervenants

L'entreprise peut parfaitement demeurer motivée grâce à certains moyens organisationnels et structurels (réorganisations, refacturations ou primes, entre autres), mais le leadership est un impondérable à même de faire la différence.



Soyez un porte-parole actif de votre SOA sur le marché. Intervenez dans des conférences. Publiez un ou deux articles dans une revue sectorielle. Œuvrez, de concert avec votre fournisseur, à faire connaître votre SOA. Conviez un maximum de collaborateurs à des conférences sur l'architecture SOA. Invitez des intervenants extérieurs pour dynamiser les débats. Incitez vos supérieurs hiérarchiques à promouvoir vos initiatives SOA à l'intérieur comme à l'extérieur de l'entreprise. Les collaborateurs ont besoin de constater que leur direction s'engage en faveur de la SOA.

Mettre la fusée SOA en orbite

Un *comportement orienté services* se met en place lorsque les décisions prises par les tribus en matière de réutilisation, de conception, d'activation, de modification et de tests sont dictées par l'intérêt de l'entreprise tout entière, et pas uniquement le confort d'un groupe.

Au départ, l'effort à fournir est énorme car vous changez des comportements humains qui sont profondément ancrés – et vous vous efforcez de transformer un système créé pour profiter à une tribu au détriment d'une autre.

À partir du moment où ce degré d'automatisation est atteint, votre structure informatique s'apparente à une équipe de football où chaque joueur connaît la stratégie de jeu, non seulement mentalement mais aussi physiquement. Pour chaque situation, chacun des membres de l'équipe a une riposte orientée services.

Votre destination

Vous pouvez évaluer le champ d'application de votre architecture SOA au vu du nombre de tribus qui partagent vos objectifs en la matière :

- ✓ Division unique (multi-plateforme).
- ✓ Pôle informatique centralisé (sur l'ensemble du cycle de vie).
- ✓ Pôle informatique centralisé plus division unique.
- ✓ Divisions multiples plus pôle informatique centralisé (SOA d'entreprise).
- ✓ Divisions multiples plus client ou fournisseur (SOA B2B).
- ✓ Entreprises multiples.

Ce ne sont là que quelques exemples, mais l'idée est la suivante : le champ d'application de votre SOA est proportionnel au nombre de tribus. Si votre « SOA » ne couvre qu'une seule tribu (« SOA monofournisseur » ou « SOA réservée aux développeurs »), elle ne mérite pas cette appellation.

Autre destination possible

Nous espérons vous avoir donné l'envie de collaborer avec d'autres « astronautes de la SOA ». Si tel est le cas, nous vous donnons rendez-vous sur le blog consacré à l'adoption de la SOA à l'adresse <http://blog.softwareag.com>.

Chapitre 11

En route vers les étoiles

Dans ce chapitre

- Analyse de la phase critique.
- Mise en orbite.
- En route vers l'infini.

Dans ce chapitre, nous examinons de plus près la phase critique SOA et ses implications. Une meilleure appréhension des risques de l'adoption de la SOA permet de mieux en cerner les principes inspirés de l'astronautique. En conclusion, nous réitérons les avantages induits par la mise en orbite de la fusée SOA à court comme à long terme.

Analyser la phase critique

Les trois principes d'une adoption de la SOA inspirée de l'astronautique vous permettront d'en surmonter la phase critique. Mais en appréhendant mieux les menaces qui pèsent sur votre programme SOA, vous réussirez à traduire ces principes en programmes qui réaliseront vos objectifs SOA.

Erreurs SOA

Parmi les dangers potentiels figurent les « bévues » de votre centre de compétence SOA, à savoir les erreurs commises en matière d'implémentation ou dans le plan directeur.

- ✓ **Erreurs d'implémentation** : La fusée SOA anticipe des erreurs de ce type en procédant constamment à des relevés et corrections de trajectoire. Les erreurs d'implémentation sont généralement révélées par des indicateurs clés, comme l'absence de réutilisation, le manque d'interopérabilité ou le défaut de performance. Fort heureusement, elles peuvent être corrigées avant de faire capoter le projet.

- ✓ **Erreurs d'architecture** : Ces erreurs peuvent anéantir un projet pilote SOA. Mais, avec les multiples corrections de trajectoire entre projets successifs, votre entreprise tire forcément les leçons de ses erreurs et finit par se ressaisir. À ceci près qu'un projet avorté risque de sérieusement mettre à mal l'élan apporté à votre organisation par votre programme SOA.

Anticiper l'échec possible d'un projet pilote et fixer des attentes en conséquence, c'est aussi agir pour faire en sorte que tout se déroule comme prévu. Voilà qui nous amène aux turbulences externes susceptibles de perturber la longue trajectoire de mise en orbite de votre fusée SOA.

La longue trajectoire de mise en orbite

S'il faut environ huit minutes pour mettre une fusée en orbite, la mise en orbite de la fusée SOA, elle, peut malheureusement prendre des mois ou des années. Tout simplement parce que votre programme SOA requiert une masse critique de services pour que l'agilité de ses applications composites et processus se manifeste, et qu'il faut aussi un certain temps à une structure informatique pour qu'elle s'adapte aux nouveaux impératifs du cycle de vie SOA. Ces délais ne sont pas exempts de risques :

- ✓ **Usure SOA** : La durée et la complexité d'élaboration d'une architecture SOA risquent de créer une lassitude au sein de l'organisation. Pour l'atténuer, il convient de veiller à ce que chaque projet continue d'offrir un retour sur investissement, en plus de contribuer à la réussite SOA de l'entreprise.
- ✓ **Changements au niveau de la direction** : Au cours de cette longue trajectoire de mise en orbite, des collaborateurs essentiels risquent d'être licenciés, de trouver un meilleur poste ailleurs ou de prendre leur retraite. Tout peut arriver ! Ces départs peuvent se révéler extrêmement préjudiciables mais, avec une vision commune et une équipe engagée, vous êtes en mesure de les surmonter.
- ✓ **Opposition des éditeurs** : Certains éditeurs de logiciels risquent de vous présenter leur offre comme une solution SOA monofournisseur. Rien ne les empêche d'instaurer un couplage fort ou de faire obstacle à l'interopérabilité par souci de maximiser les recettes tirées de leurs services ou licences.

- ✓ **Opposition des consultants :** Les consultants entendent garder la main et multiplier le nombre d'heures facturables. La SOA peut confirmer ou infirmer ces perspectives lucratives. Sachez que les consultants feront tout pour défendre ou accroître leurs revenus la plupart du temps.
- ✓ **Opposition des personnes chargées de la mise en œuvre :** À moins que la mise en œuvre progressive des règles ne s'accompagne de mesures incitatives, les développeurs et autres parties prenantes peuvent toujours opposer une résistance active ou passive.
- ✓ **Coupes dans les financements :** Si les projets SOA cessent de produire des résultats quantifiables par les dirigeants, des coupes claires risquent d'être effectuées dans les fonds affectés à votre programme SOA. Ne perdez pas de vue les indicateurs important le plus pour vos supérieurs hiérarchiques et veillez à mettre en adéquation votre programme SOA et ces objectifs.

Pour gérer l'ensemble de ces menaces, votre meilleur atout réside dans le relevé permanent de mesures – et pas uniquement d'indicateurs informatiques, mais aussi d'indicateurs métier. En sachant répondre aux attentes et mettre en exergue les résultats positifs induits par votre architecture SOA, vous saurez entretenir l'enthousiasme des responsables d'implémentation, financiers et autres intervenants.

Tirer parti de la mise en orbite

L'énumération de tous ces risques suscite peut-être une certaine appréhension au sujet de l'adoption de la SOA. Peut-être vous interrogez-vous sur la nécessité de cette dernière.

Sachez, tout d'abord, qu'une simple série de composants réutilisables peut donner lieu à une quasi-infinité de recombinaisons possibles. Tout comme les 26 lettres de l'alphabet ont servi à produire cet ouvrage. Et, après ses premiers balbutiements, la SOA a vite fait d'acquérir une certaine « fluidité » en termes de processus. La modification de ces derniers s'effectue au rythme de l'entreprise, dans le cadre d'une quête de longue haleine dite d'amélioration continue des processus.

L'adoption de la SOA est difficile. Faire marche arrière sur la prolifération anarchique du système informatique et de sa structure, c'est s'exposer à de nouvelles difficultés. Mais les entreprises qui gagnent ce pari écourteront leurs délais de mise sur le marché à grand renfort de produits et services différenciés. Elles sauront améliorer le service à la clientèle et aussi fidéliser cette dernière. Elles intégreront plus rapidement les entités acquises et réduiront les coûts et risques induits par leurs opérations. En résumé, elles surclasseront leurs concurrents.

Vers l'infini et au-delà...

Et après la SOA, qu'advient-il ? Sur le plan architectural, les entreprises font de plus en plus la part belle à une architecture orientée événements (EDA, *Event-Driven Architecture*) assortie d'applications dédiées au traitement des événements complexes (CEP, *Complex Event Processing*) dans leurs plans directeurs. À nos yeux, l'EDA est un ensemble de principes de conception exploitables conjointement avec la SOA, essentiellement axés sur les messages et sur la manière dont les systèmes leur répondent. Nous engageons les architectes à tenir compte de cette tendance, et à admettre qu'une SOA bien conçue n'exclut pas l'intégration d'applications orientées événements à l'avenir.

Autre orientation technologique significative : l'émergence de modèles hors site, tels que l'hébergement, le SaaS (logiciel sous forme de service), le PaaS (plate-forme sous forme de service) et le « cloud computing ». Grâce à l'adoption de la SOA, votre entreprise pourra consommer, composer et concevoir des processus pour une combinaison de services sur site et hors site. Même si certains alarmistes ont prédit une délocalisation hors site de l'ensemble des outils informatiques, il paraît évident que les entreprises qui réussissent ne se contenteront pas de consommer des services ; elles dégageront des revenus substantiels en les proposant à leurs utilisateurs, clients et partenaires commerciaux. Certaines proposent déjà une plate-forme PaaS permettant à leurs clients de combiner diverses fonctionnalités dans des « mash-up » et d'inventer plusieurs centaines de techniques inédites pour commercer avec elles.

L'adoption de la SOA prépare vos systèmes stratégiques à ces technologies dynamiques de demain.

Devenez incollable sur la SOA avec nos ressources gratuites



Maintenant que vous avez lu cet ouvrage, découvrez la SOA en pratique :



Dialoguez avec les auteurs par blogs interposés. Sur le blog « SOA Adoption for Dummies », échangez vos points de vue, vos réflexions, vos stratégies et posez vos questions aux auteurs, mais aussi à vos homologues.
Rendez-vous sur <http://blog.softwareag.com>



Êtes-vous prêt pour la SOA ? Votre évaluation en moins de 10 minutes. Identifiez les atouts qui assureront le succès de votre adoption SOA, et évitez les pièges !
Rendez-vous sur www.soatechnologyassessment.com



Calculez les avantages de la SOA. À partir d'une analyse de valeur approfondie, identifiez les domaines sur lesquels la SOA aura le plus d'impact au sein de votre entreprise. Un précieux outil qui vous aidera à rentabiliser vos investissements SOA pour accélérer votre projet.
Inscrivez-vous sur www.soavalueassessment.com



Des fournisseurs SOA notés par des analystes de renom. Réalisez de substantielles économies avec cette étude gratuite. Des fournisseurs sont évalués sur leur capacité à mettre en œuvre une architecture SOA.
Rendez-vous sur http://info.softwareag.com/lp/softwareag/SOA_Analyst.html

Accélérez votre adoption SOA en disposant des bonnes informations, dès maintenant. Pour en savoir plus : www.softwareag.com



ACCÉLÉREZ L'OPTIMISATION DE VOS PROCESSUS. EN SUIVANT NOS CONSEILS.

Mettez à profit les ressources existantes. Intégrez les silos d'informations. Optimisez les processus. Plus vite.

Accélérez vos processus et enclenchez la cinquième vitesse de votre SOA avec les logiciels d'infrastructure métier Software AG. À nos côtés, vous réduirez vos coûts d'intégration, décuplerez la valeur des applications existantes et dispenserez de nouvelles applications et de nouveaux services, plus rapidement. En vous donnant les moyens de gérer et d'accéder instantanément aux données stratégiques et aussi de superviser les opérations métier en temps réel, nous vous remettons les clés du succès. Avec Software AG, vous réalisez vos objectifs en un rien de temps. www.softwareag.com/fr.



Apprenez à appliquer les
principes SOA
à vos problématiques métier

Facilitez-vous au maximum l'adoption SOA grâce à cet ouvrage !

Ce livre n'est pas un ouvrage sur l'architecture. De nombreux écrits consacrés à l'architecture SOA existent déjà sur le marché. Cet ouvrage traite de l'adoption SOA, autrement dit de méthodes concrètes et pratiques pour donner corps à un projet SOA.

Il expose notre approche, inspirée de l'astronautique, qui pilote les projets un par un au cours de leur phase critique SOA, afin de concrétiser la vision pour le programme SOA dans son ensemble.

L'ESPRIT
DES NULS

*Le savoir et le savoir-faire des
meilleurs spécialistes*

*Une approche sans complexe
des sujets traités*

Les informations essentielles

*Un accès rapide à l'information
grâce à un système d'icônes*

Une bonne dose d'humour

Découvrez Comment :

*Résoudre les
problématiques
métier grâce à
l'architecture SOA*

*Passer du plan directeur
à l'adoption SOA*

*Définir des règles
qui guideront le
développement et
l'utilisation de votre
portefeuille de services*

*Gérer les « guerres
tribales » informatiques
qui nuisent au
déploiement SOA*

Allez sur

@ www.dummies.com

✓ Notre catalogue complet

✓ Retrouvez le meilleur des
Nuls par thème

✓ Les bons tuyaux des Nuls

Pour les Nuls®
est une marque déposée de



ISBN: 978-0-470-59358-5

Ne peut être vendu